IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Faster R-CNN Towards Real-Time Object Detection with Region Proposal Networks

[1]Suggu Chandu, [2]Rahul Kumar Behra, [3]Gagan Agrawal, [4]M. Sai Ambika Rani, [5]Prerna Dusi

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor
[1]Computer Science and Engineering,
[1]Kalinga University, Naya Raipur, India

Abstract: object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. Object detection using neural networks has gained significant traction, with classification playing a key role in the identification of objects within images or video frames. In this approach, the neural network is tasked with not only classifying the objects present but also predicting their locations, typically by segmenting images into regions and classifying each region. Modern neural network architectures, such as Convolutional Neural Networks (CNNs), are particularly well-suited for this task due to their ability to automatically extract spatial hierarchies of features.

Classification-based object detection methods, such as Region-based CNN (R-CNN) and its derivatives (Fast R-CNN, Faster R-CNN), first classify potential object regions (proposals) and then refine their positions. Single-stage detectors like You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD) streamline this process by performing classification and localization simultaneously, offering faster detection speeds. These classification models are trained on large annotated datasets and optimized to generalize across diverse object categories.

The combination of classification and localization in neural networks enables robust and scalable solutions for real-time applications, including autonomous driving, medical diagnostics, and industrial automation. The ongoing research continues to improve the trade-offs between accuracy, speed, and computational efficiency in object detection.

Keywords— Neural Networks, Deep learning, CNN, Convolutional Neural Networks, Classification, Architecture, Pooling Layers, Object Detection, RoI (Region of Interest), Dataset, Training Model, Feature Extractor, Classifier, Selective search, Cost Function, Optimizer.

1. Introduction

A neural network is a sort of computer architecture based on human brain, thus the term “neural” and helps machine to achieve human thinking capabilities. They are an important component of AI (Artificial Intelligence). These network models are developed by collection of processing units known as “nodes”. Thes nodes transfer data with each other resembling functionality of neurons in human body.

The sole purpose of neural networks is to detect relationship between features in a dataset and this is achieved using set of learning algorithms that is inspired from human brain and mimics it. Neural networks are used in Machine Learning, a set of computer programs where machine learns without definite information. Specifically, they are used in Deep Learning (an advanced type of machine learning) where conclusions are drawn without human intervention.

Neural Networks are quite old. The Concept of neural networks first introduced in a 1943 mathematical paper modelling how human brain could work. It is then, some computer scientists began experiments by constructing

simple neural network models in 1950s and 1960s but eventually they failed and the concept fell over. In 1980s the concept of neural networks was revived and by 1990s they were everywhere in AI research.

However, only with the advent of hyper-fast processing, massive data storage capabilities, and access to computing resources neural were networks able to advance to the point they have reached today, where they can imitate or even exceed human cognitive abilities. Developments are still made in this field through new experiments and research.

Starting since 1950, researchers are working to develop systems that can understand visual information and the hard work finally paid and gave birth to Computer Vision. In 2012, a significant breakthrough occurred when researchers from the University of Toronto developed Alex Net, an AI model that significantly outperformed previous image recognition algorithms with an accuracy of 85% which was unbelievable and this success was driven by CNN Models.

Convolutional Neural Networks (CNN): Convolutional neural networks are a kind of deep neural network specifically involved in interpretating of visual information. Convolutional Neural Networks (CNNs) resemble traditional ANNs in a way that they are comprised of neurons that self-optimise through learning. Each neuron gets an input and perform operation it (such as a scalar item taken after by a non-linear work). From the input crude picture vectors to the last yield of the course score, the whole of the arrange will still express a single discerning score work (the weight) and final layer will contain misfortune capacities related with the classes. CNNs are essentially utilized in the field of recognizing designs inside pictures. This makes the design more reasonable for picture centred errands with encoded picture particular highlights.

CNNs are composed of three layers. These three layers when stacked form the CNN architecture:

- The Convolutional Layers: The convolutional layer consists of a large number of filters, basically small matrices applied to the input image.
- The Pooling Layers: This layer down samples the feature maps thus reducing
- computational cost making the model more invariant to small translations and rotations.
- Fully-Connected layers: These are the final layers of the CNN. They merge the extracted features of the above two layers and produce a final output.

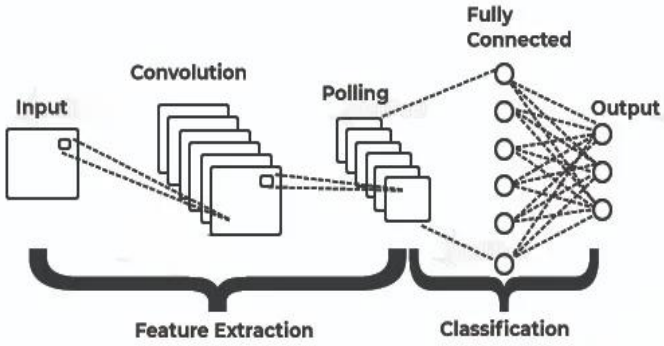

Fig 1.1 Demonstration of Layers of CNN Architecture.

Classification: Classification as the name suggests is a situation/problem where a specific type of class label is produced as output from given input data on the particular data. It involves prediction if something belongs to a particular group or another. Classification plays a vital in almost all real-life scenarios. For instance, classifying spam emails or classifying user observation. For any training model, we need a training dataset with all the possible scenarios, their inputs and outputs. There are mainly three types of classifications:

- Binary Classification: Used in scenarios where the possible outcomes are at most two.
- Multi-class Classification: Used when there are more than two possible outcomes.
- Multi-label Classification: Used in the cases where categories should be assigned.

Our Contributions:

Region proposal methods typically rely on inexpensive features and economical inference schemes. Selective Search, one of the most popular methods, greedily merges superpixels based on engineered low-level features. Yet when compared to efficient detection networks , Selective Search is an order of magnitude slower, at 2 seconds per image in a CPU implementation. EdgeBoxes currently provides the best tradeoff between proposal quality and speed, at 0.2 seconds per image. Nevertheless, the region proposal step still consumes as much running time as the detection network.

One may note that fast region-based CNNs take advantage of GPUs, while the region proposal methods used in research are implemented on the CPU, making such runtime comparisons inequitable. An obvious way to accelerate proposal computation is to reimplement it for the GPU. This may be an effective engineering solution, but re-implementation ignores the down-stream detection network and therefore misses important opportunities for sharing computation.

Our observation is that the convolutional feature maps used by region-based detectors, like Fast RCNN, can also be used for generating region proposals. On top of these convolutional features, we construct an RPN by adding a few additional convolutional layers that simultaneously regress region bounds and objectness scores at each location on a regular grid. The RPN is thus a kind of fully convolutional network (FCN) and can be trained end-toend specifically for the task for generating detection proposals

2. Literature Review

2.1 **Region- Based Convolutional Neural Network (R-CNN):** Region-Based Convolutional Neural Network (R-CNN) is a deep learning model specifically designed for object detection tasks. It was proposed to overcome limitations in traditional image detection methods, especially when working with complex images where objects need to be precisely localized. The core idea of R-CNN is to combine region proposals with convolutional neural networks, allowing it to locate and classify multiple objects within an image. Here are the steps explaining the basic steps of R-CNN:

2.1.1 Input Images from dataset are fed to the model.

2.1.2 Extract region proposals using selective search. This is done using first generating initial sub-segmentation, then using greedy search algorithm we recursively combine similar regions. These regions are then used to produce the final candidate region proposals.

2.1.3 This warped region is then used to compute CNN features.

Here's a breakdown of the R-CNN approach:

Region Proposal: R-CNN starts by generating "region proposals," which are potential bounding boxes around objects in the image. Selective Search is commonly used for this, producing a set of bounding boxes that could contain objects of interest.

Feature Extraction with CNN: Each region proposal is fed into a Convolutional Neural Network (CNN) to extract feature vectors. This step allows R-CNN to leverage CNN's strong capability for visual feature extraction.

Classification and Bounding Box Regression:

The extracted features from each region are passed through a classifier, typically a Support Vector Machine (SVM), to identify the class of the object within the region.

2.2 **Fast R-CNN: Fast R-CNN** is an improved version of R-CNN, proposed by Ross Girshick in 2015, aimed at addressing the high computational cost and slow processing time of the original R-CNN model. Fast R-CNN significantly speeds up the object detection pipeline by optimizing how region proposals are processed and features are extracted, making it both faster and more efficient.

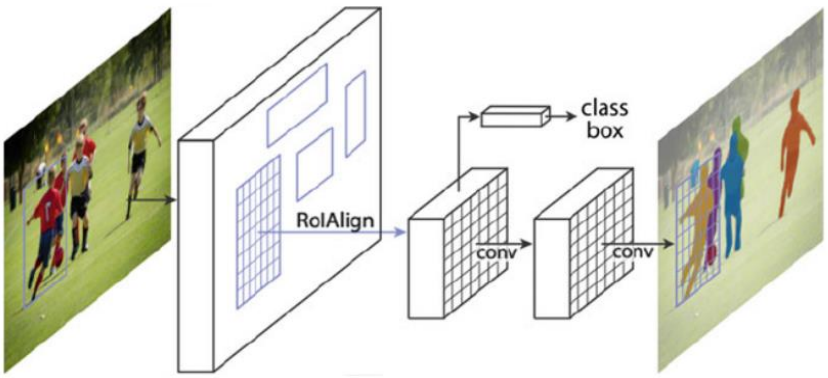

Fig2.1 Demonstration of mask R-CNN Framework

2.4 Computer Vision: Computer Vision is a field of artificial intelligence (AI) that enables computers to interpret, analyze, and understand visual information from the world, typically in the form of images and videos. The primary goal of computer vision is to enable machines to "see" and perform tasks similar to human vision, such as recognizing objects, identifying patterns, and making decisions based on visual input.

1. Image Processing:

- This involves transforming or analyzing images using techniques like filtering, edge detection, and color adjustment.
- Image processing often serves as a preliminary step in computer vision to enhance image quality or isolate relevant parts of the image for further analysis.

2. Object Detection:
   - Object detection identifies and locates objects within an image or video by drawing bounding boxes around them.
   - Models like YOLO (You Only Look Once), Faster R-CNN, and SSD (Single Shot Detector) are commonly used for real-time object detection.
3. Image Classification:
   - In image classification, the model assigns a label or category to an image based on its content.
   - Convolutional Neural Networks (CNNs) are typically used for this task due to their strong performance in extracting and learning image features.
4. Image Segmentation:
   - Image segmentation involves partitioning an image into multiple regions, with each region corresponding to a different object or part of an object.
   - Techniques include semantic segmentation (labeling every pixel in an image) and instance segmentation (differentiating between separate instances of the same object class).
   - Models like U-Net and Mask R-CNN are often used for segmentation tasks.
5. Facial Recognition:
   - Facial recognition identifies and verifies individuals by analyzing facial features.
   - Applications include security (face ID), social media tagging, and biometric identification.
6. Pose Estimation:
   - Pose estimation identifies the position and orientation of body parts or objects within an image.
   - This is useful for applications like human-computer interaction, sports analysis, and motion tracking.
7. Optical Character Recognition (OCR):
   - OCR involves recognizing and extracting text from images. It's widely used in document digitization, license plate recognition, and real-time translation.
   - Modern OCR uses deep learning to handle a variety of fonts, orientations, and backgrounds.
8. 3D Vision and Depth Estimation:
   - 3D vision seeks to understand the three-dimensional structure of a scene, often using multiple 2D images or depth-sensing cameras.
   - Depth estimation helps in tasks like autonomous driving and robotic manipulation.

Techniques and Models in Computer Vision:

- Convolutional Neural Networks (CNNs): CNNs are widely used for image-related tasks due to their ability to capture spatial hierarchies in images and learn complex patterns.
- Transfer Learning: This involves using pre-trained models (e.g., ResNet, VGG, Inception) as a base for specific tasks, which saves time and computation.
- Generative Adversarial Networks (GANs): GANs are used for generating new images, image super-resolution, and style transfer, among other tasks.

Applications of Computer Vision:

- Autonomous Vehicles: Computer vision is crucial for self-driving cars to detect objects, understand road signs, and navigate safely.
- Healthcare: In medical imaging, computer vision helps in diagnosing diseases by analyzing MRI, CT, and X-ray images.
- Retail and E-commerce: Vision-based systems can perform tasks like visual search, virtual try-ons, and inventory management.
- Agriculture: Computer vision helps monitor crop health, detect pests, and automate harvesting.
- Manufacturing and Quality Control: Vision systems inspect products on assembly lines for defects or quality issues.

Tools and Frameworks in Computer Vision:

Popular libraries and frameworks for developing computer vision applications include:

- OpenCV: An open-source library with numerous functions for image processing and computer vision.
- TensorFlow and PyTorch: Popular deep learning frameworks with support for image-based models.
- Keras: A high-level neural network API that simplifies building and training deep learning models for vision tasks.

2.4.1 Facial Recognition: Facial Recognition using OpenCV involves detecting and identifying faces in images or videos. OpenCV provides a set of tools and pre-trained models that make it straightforward to implement facial recognition, which can be applied to security, authentication, tagging, and many other applications.

2.4.2

2.5 TensorFlow: TensorFlow is an open-source machine learning (ML) and deep learning framework developed by Google. It was created to simplify the process of building, training, and deploying machine learning models for various applications, ranging from image recognition to natural language processing. TensorFlow provides a comprehensive ecosystem of tools, libraries, and community resources, making it popular among researchers, developers, and businesses for both research and production use.

2.6 Keras: Keras is Python's neural network application programming interface (API) tightly integrated with TensorFlow for developing machine learning models. Keras models provide a simple, user-friendly way to describe a neural network, which is then built for you by TensorFlow.

2.6.1 Keras Model Overview: A model is the important settings you work with when using Keras. This model is used to describe TensorFlow neural networks by specifying the required features, functions, and layers. Keras provides several APIs for describing neural networks, including a section of the API that lets you build models layer by layer for most problems. This is simple (just a simple list of processes), but it is limited to just one input, one output layer stack. It is flexible and more complex than the API section. It is suitable for research and use of complex data, but is rarely used in practice.

3 Related Work

The effectiveness of any object detection model is determined using various parameters including average precision (AP), average recall (AR), mean average precision (mAP), and intersection over union (IoU).

Object Detection models are used by several researchers till date in different application areas such as agriculture, healthcare, transport systems, satellite imaging, etc. The following table explains briefly about the work done with some of the object detection models used to train for detecting plants and animals along with their activities with their pre-trained models and datasets used.

4 Methodology

A Region Proposal Network (RPN) takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an objectness score. We model this process with a fully convolutional network , which we describe in this section. Because our ultimate goal is to share computation with a Fast R-CNN object detection network , we assume that both nets share a common set of convolutional layers. In our experiments, we investigate the Zeiler and Fergus model (ZF), which has 5 shareable convolutional layers and the Simonyan and Zisserman model (VGG-16), which has 13 shareable convolutional layers. To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes as input an n × n spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature (256-d for ZF and 512-d for VGG, with ReLU following). This feature is fed into two sibling fullyconnected layers—a box-regression layer (reg) and a box-classification layer (cls). We use n = 3 in this paper, noting that the effective receptive field on the input image is large (171 and 228 pixels for ZF and VGG, respectively). Note that because the mini-network operates in a sliding-window fashion, the fully-connected layers are shared across all spatial locations. This architecture is naturally implemented with an n×n convolutional layer followed by two sibling 1 × 1 convolutional layers (for reg and cls, respectively).

4.1 Define the Classification Criteria

Purpose: Determine why you are classifying the objects (e.g., for sorting, identification, analysis).
Attributes: Identify relevant features or attributes (size, colour, shape, material).

4.2 Data collection

Gather Samples: Collect a representative sample of the objects to be classified.
Feature Extraction: Measure and record the identified attributes for each object.

4.3 Choose a Classification Method

Rule-Based Classification: Use predefined rules to categorize objects (e.g., if the object is red and round, classify as an apple).

Statistical Methods: Utilize statistical techniques (e.g., k-means clustering, decision trees) to classify based on data patterns.
Machine Learning: Implement algorithms such as support vector machines, neural networks, or random forests to learn from labelled data.
Hierarchical Classification: Organize objects into a tree structure, where broad categories are divided into more specific subcategories.

4.4 Training and Testing

Training Set: If using machine learning, split the data into training and testing sets to build and validate the model.
Feature Selection: Optimize the set of features used for classification to improve accuracy.

```
$ time python fine_tune_rcnn.py
[INFO] loading images...
[INFO] compiling model...
[INFO] training head...
Train for 94 steps, validate on 752 samples
Train for 94 steps, validate on 752 samples
Epoch 1/5
94/94 [==============================] - 77s 817ms/step - loss: 0.3072 - accuracy: 0.8647 -
val_loss: 0.1015 - val_accuracy: 0.9728
Epoch 2/5
94/94 [==============================] - 74s 789ms/step - loss: 0.1083 - accuracy: 0.9641 -
val_loss: 0.0534 - val_accuracy: 0.9837
Epoch 3/5
94/94 [==============================] - 71s 756ms/step - loss: 0.0774 - accuracy: 0.9784 -
val_loss: 0.0433 - val_accuracy: 0.9864
Epoch 4/5
94/94 [==============================] - 74s 784ms/step - loss: 0.0624 - accuracy: 0.9781 -
val_loss: 0.0367 - val_accuracy: 0.9878
Epoch 5/5
94/94 [==============================] - 74s 791ms/step - loss: 0.0590 - accuracy: 0.9801 -
val_loss: 0.0340 - val_accuracy: 0.9891
[INFO] evaluating network...
              precision    recall  f1-score   support

   no_raccoon       1.00      0.98      0.99       440
      raccoon       0.97      1.00      0.99       312

    accuracy                           0.99       752
   macro avg       0.99      0.99      0.99       752
weighted avg       0.99      0.99      0.99       752

[INFO] saving mask detector model...
[INFO] saving label encoder...

real    6m37.851s
user    31m43.701s
sys     33m53.058s
```

4.5 Implementation

Algorithm Application: Apply the chosen method to classify new objects.
Model Evaluation: Use metrics like accuracy, precision, recall, and F1-score to assess performance.

4.6 Iterative Improvement

Feedback Loop: Continuously gather data on misclassified objects and refine the classification criteria or model.
Adaptation: Modify the methodology based on new information or changing conditions.

5 Results

Results: Document the classification results, methodologies used, and any insights gained.

Visualization: Use graphs, charts, or other visual tools to present findings clearly.
Applications.

Image Classification: Used in computer vision for identifying objects in images.

Text Classification: Applied in natural language processing for categorizing documents.

Biological Classification: Used in taxonomy to categorize organisms.

By following these steps, you can develop a systematic approach to classifying objects effectively.

6 **Conclusion:** This study endeavours to explore the use of neural networks for addressing the first, and very important, natural language processing (NLP) problem – topic classification. It is quite different from ordinary machine learning approaches where one has to manually define the features. Neural networks learn complex patterns and characteristics from the data itself. Noticeable development has been observed with tagging themes of compact texts utilizing such models as transformers, recurrent neural networks, and convolutional networks. Long- and short-term memory (LSTM) networks are highly efficient in working with the order of words and grasping the 'whole picture' of the text. On the contrary, convolutional networks are efficient in the extraction of important elements or patterns in brief texts. Topics even beyond this level are posed by transformers, which have gone further than previous builds due to the incorporation of focus methods into the entire text at once in the present study, we sought into how neural networks can be used for solving the first and one of the key problems of natural language processing NLP - topic classification. This is quite different from ordinary machine learning Normal approach where you have to specify the features. Neural networks learn complex patterns and features from the data itself. Noticeable development has been observed with tagging themes of compact texts utilizing such models as transformers recurrent neural networks and convolutional networks Long and short term memory LSTM for example are very efficient working with commands of lexemes caching all lexemes. As its. Cognitive bias – taking on Weaponizing than Documenting with Few-Round Memory/Harsha Rahman s Pinaki r pairing image distortion adjustment homogeneous separation transfusion rounds are doable through uniform process via compelling struggles Yet CNNs are more efficient in where important patterns or mostly key phrases need to be looked for in the small snippets of text. Topics even beyond this level are posed by transformers, which have gone further than previous builds due to the incorporation of focus methods into the entire text at once

7 References:

7.1 R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
7.2 K. Lenc and A. Vedaldi, “R-CNN minus R,” in British Machine Vision Conference (BMVC), 2015
7.3 J. Johnson, A. Karpathy, and L. Fei-Fei, “Densecap: Fully convolutional localization networks for dense captioning,” arXiv:1511.07571, 2015.
7.4 P. O. Pinheiro, R. Collobert, and P. Dollar, “Learning to segment object candidates,” in Neural Information Processing Systems (NIPS), 2015.
7.5 Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural computation, 1989
7.6 P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in International Conference on Learning Representations (ICLR), 2014.
7.7 . Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Neural Information Processing Systems (NIPS), 2012.
7.8 C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” in Neural Information Processing Systems (NIPS), 2013.
7.9 Fig 1.1: https://intellipaat.com/blog/wp-content/uploads/2022/02/CNN-Architecture.png diagram representing layers of CNN Architecture
7.10 https://arxiv.org/pdf/1511.08458 a research paper based on introduction to convolutional neural networks (rXiv:1511.08458v2 [cs.NE] 2 Dec 2015)