ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Yolo Object Recognition using python

K.Teja[1], B. Sai Natraj[1], K.Lavanya[1], Ch.Deepkama[l] B. Mohan Sai Krishna[1]
[#1] B.Tech UG Students, Dept.Of CSE, AITAM, Tekkali, AP, India

Abstract

The implementation of object detection using YOLO encompasses a variety of algorithms falling within the DNN (Deep Neural Network) family, with a focus on advanced models such as YOLO and SSD. The primary goal of this project is to swiftly and accurately detect objects of any type and subsequently convert these detected objects into speech. The utilization of YOLO, a DNN-based algorithm, proves to be particularly advantageous in achieving rapid and efficient object detection.

The central purpose of the project revolves around leveraging YOLO, among other advanced algorithms, for near-instantaneous object detection. YOLO stands out in terms of both speed and accuracy when compared to alternative approaches such as SSD. The project's primary objective is to detect objects using the YOLO approach, a methodology with distinct advantages over other object detection algorithms.

Unlike some alternative algorithms like Convolutional Neural Network (CNN) and Fast Convolutional Neural Network, where the model may not comprehensively analyze the entire image, YOLO takes a holistic approach. Through predictive bounding boxes generated by convolutional networks and corresponding class probabilities, YOLO ensures a thorough examination of the entire image. This comprehensive image analysis results in faster and more accurate object detection compared to its counterparts, making YOLO a preferred choice for the project's objectives.

Keywords: Neural Network, Bounding Boxes, YOLO.

Introduction

The YOLO (You Only Look Once) algorithm employs a neural network to analyze an entire image by dividing it into an S x S grid. Within each grid cell, the algorithm predicts bounding boxes, outlining the regions where objects are present, along with associated probabilities calculated using logistic regression. The bounding boxes are then weighted based on these probabilities. Class predictions are performed independently using logistic classifiers. This article aims to guide readers through the implementation of the YOLO algorithm with a pre-trained model. To start, the installation of DarkNet is necessary, as it serves as an open-source neural network framework.

Object detection stands out as a vital technique for identifying various objects, and numerous algorithms are employed for this purpose.

Cite this article as: K.Teja, B.Sai Natraj, K.Lavanya, Ch.Deepkamal & B.Mohan Sai Krishna, "Yolo Object Recognition using python", International Journal of Research in Advanced Computer Science Engineering, (IJRACSE), Volume 9 Issue 10, March 2024, Page 24-31.

OpenCV, a powerful computer vision library, proves instrumental in implementing object detection for real-time applications. Within this context, both YOLO and RCNN (Region-based Convolutional Neural Network) are utilized for detection tasks. Notably, YOLO has the capability to recognize 49 images simultaneously, demonstrating its efficiency in handling object detection tasks with speed and accuracy.

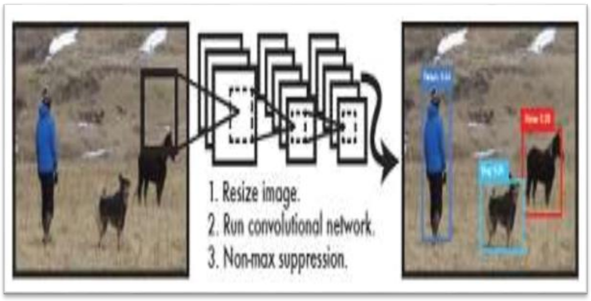


Fig. 1. The YOLO Detection System . Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model’s confidence. [1]

1.Basics of Image Processing:

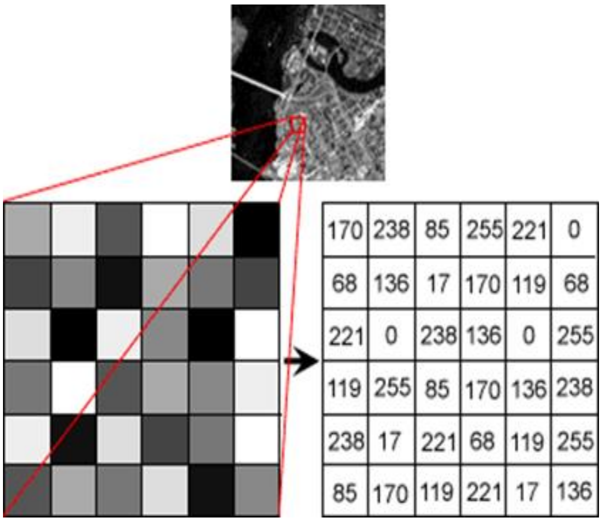
FUNDAMENTALS OF DIGITAL IMAGE

1.1 IMAGE:

An image, whether two-dimensional like a photograph or screen display, or three-dimensional like a statue, captures the appearance of a subject. It can be created through optical devices, such as cameras, or by natural means like the human eye. The term is also applied broadly to include any two-dimensional figure like maps, graphs, or abstract paintings. Images can be crafted manually through drawing or painting, produced automatically through printing or computer graphics, or a combination of methods, creating a pseudo-photograph.

Fig: Colour image to Gray scale Conversion Process

An image is a grid of pixels, each being a square with a fixed size on a display. The image has a definite height and width measured in pixels. Pixels are organized in columns and rows, with each pixel represented by numbers indicating brightness and color. Monitor variations may lead to different pixel sizes.



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Fig: Gray Scale Image Pixel Value Analysis

1.2 IMAGE FILE SIZES: Image file size depends on the number of pixels and color depth, with higher resolution and greater color depth resulting in larger files. Compression algorithms are used to reduce file size. High-resolution cameras produce large files, e.g., a 12-megapixel image occupies 36,000,000 bytes. To manage large sizes, image file formats were developed.

1.3 IMAGE FILE FORMATS:

Image file formats are standardized methods for organizing and storing digital images, comprising pixel or vector data rasterized for display. Numerous formats exist, with PNG, JPEG, and GIF commonly used for online display. Metafile formats, portable and containing both raster and vector data, serve as intermediate formats, often opened in Windows applications and saved in native formats.

2.IMAGE PROCESSING:

Digital image processing, a recent development in mankind's fascination with visual stimuli, has seen widespread application with varying success. Despite its subjective appeal, the field faces myths and misunderstandings. It encompasses optics, electronics, mathematics, photography, graphics, and computer technology. The declining cost of computer equipment and emerging technologies like parallel processing and CCDs indicate a promising future for digital image processing.

FUNDAMENTAL STEPS IN DIGITAL IMAGE

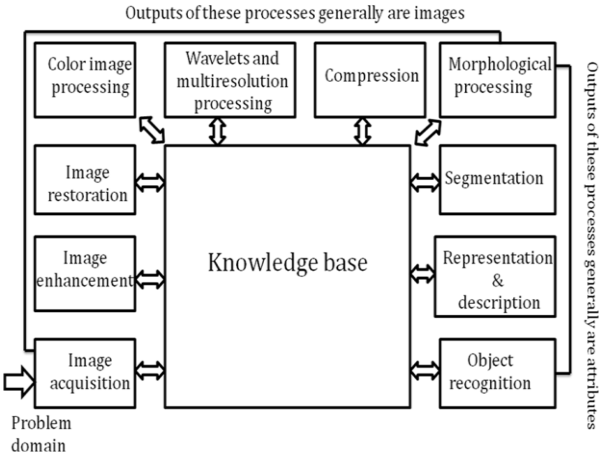


Fig: Basics steps of image Processing

2.1 Image Acquisition:

Image acquisition involves obtaining a digital image using an image sensor and digitizing the sensor's signal. The sensor, whether monochrome or color, can be a TV camera capturing a complete image every 1/30 sec or a line scan camera capturing one image line at a time, particularly suitable for objects in motion.

2.2 Image Enhancement:

Image enhancement, a straightforward aspect of digital image processing, aims to reveal obscured details or highlight specific features in an image. Increasing contrast, for instance, is a common enhancement to improve visual appeal, but it's crucial to recognize that enhancement is highly subjective in image processing.

2.3 Image restoration:

Image restoration focuses on improving image appearance using objective techniques, unlike subjective enhancement. Restoration relies on mathematical or probabilistic models of image degradation for its methods.

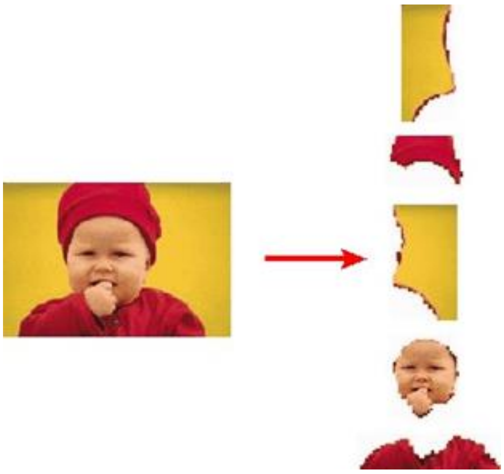
2.4 Color image processing:

Color in image processing serves two key purposes: it simplifies object identification and extraction, aiding in scene interpretation, and takes advantage of humans' ability to discern thousands of color shades, crucial for manual image analysis compared to limited shades of gray.

2.5 Segmentation:

Segmentation divides an image into constituent parts, a challenging task in digital image processing. Effective segmentation is crucial for successfully solving imaging problems requiring individual object identification. The accuracy of segmentation greatly influences the likelihood of successful recognition. In digital image processing, an image is a two-dimensional function, denoted as f(x, y), where x and y are spatial coordinates, and the field involves processing such images with a digital computer.

Image elements, often called pixels, are finite elements with specific locations and values.



2.6 Image Compression:

Digital image compression reduces data needed to represent an image by eliminating redundancy. Mathematically, it transforms a 2D pixel array into an uncorrelated dataset. Data redundancy, quantifiable through n1 and n2 information-carrying units, defines the relative redundancy (RD) of the first dataset.

RD=1-1/CR

Where CR is called as compression ration.CR=n1/n2

Image compression targets coding, interpixel, and psychovisual redundancies for reduction. Primarily used in image transmission and storage, applications include broadcast television, remote sensing, teleconferencing, computer communications, facsimile transmission, educational/business documents, medical images (CT, MRI, digital radiology), motion pictures, satellite images, weather maps, and geological surveys.

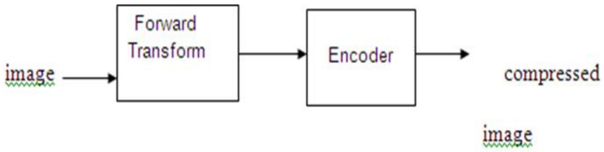


Figure 1.1.a) Block Diagram of Image compression

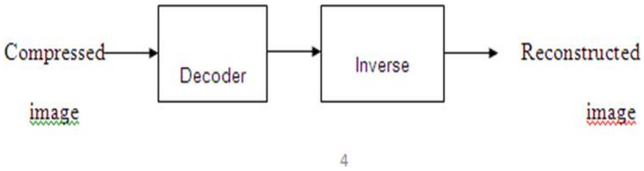


Image Compression Types

There are two types' image compression techniques.

1. Lossy Image compression
2. Lossless Image compression

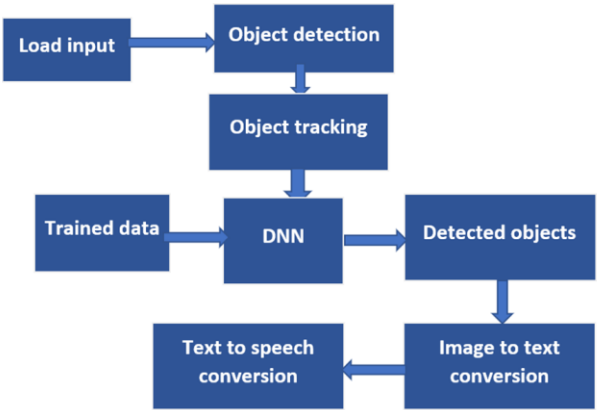
Compression ratio:

$$compression\ ratio = \frac{B_0}{B_1}$$

B_0 – number of bits before compression
B_1 – number of bits after compression

3. Methodology

This section details the proposed model, starting with enhancements based on the loss function, followed by improvements using the inception structure model. Lastly, the incorporation of the spatial pyramid pooling layer is outlined, including symbolic representations.



3.1 Improvement in Network Design:

The following improvements to the YOLO network model are made while maintaining the original model dominant idea.

3.2 Improvement Based on Loss Function:

The original YOLOv1 network's loss function treats large and small objects equally, causing issues in detecting neighboring or small objects. The new loss function, more flexible and optimized, replaces the original difference with proportionality. YOLOv1's original loss function combines bounding box and object classification, with parts focusing on bounding box coordinates, confidence differences, and class probability disparities. Scalars λcoord and λnoobj weight each loss function. The improved model introduces the variance function, normalizes contrast, and uses a proportion style, considering the object size in the image. The loss function guides class optimization and boundary box position for object detection.

3.3 Computer Vision Resources:

3.3.1Packages and Frameworks

OpenCV – "OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics."

SimpleCV – "SimpleCV is an open source framework for building computer vision applications. With it, you get access to several high-powered computer vision libraries such as OpenCV – without having to first learn about bit depths, file formats, color spaces, buffer management, eigenvalues, or matrix versus bitmap storage."

Mahotas – "Mahotas is a computer vision and image processing library for Python. It includes many algorithms implemented in C++ for speed while operating in numpy arrays and with a very clean Python interface. Mahotas currently has over 100 functions for image processing and computer vision and it keeps growing."

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

3.3.2 Experiment

Pascal VOC is divided into two datasets: Pascal VOC 2007 and Pascal VOC 2012 dataset. newly designed network was tested on both datasets [41]. 'e Pascal VOC dataset consist of 20 categories: person, bird, cat, cow, horse, sheep, airplane, bike, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and TV monitor. Figures 4 and 5 show the sample image.

3.4 Computer Vision on Algorithm

Algorithmia simplifies the deployment of scalable microservices for computer vision applications. The marketplace offers algorithms like SalNet for identifying crucial image parts, Nudity Detection, Emotion Recognition, DeepStyle for unique filters, Face Recognition, and Image Memorability. A typical workflow involves using these algorithms, such as Emotion Recognition for security cameras or Nudity Detection to screen inappropriate profile pictures. For in-depth guidance on implementing computer vision tasks, explore the Algorithm platform.

4.Algorithm:

4.1 RCNN: Region convolutional neural network. It is two stage of cnn. using this we can detect object easily with grid lines for the classification.

4.2 Darknet: The architecture, known as DarkNet, was created by the primary author of the YOLO paper. This algorithm divides an image into a grid, predicting bounding boxes, confidence scores, and class probabilities for each cell. It excels in image classification, offering clarity and the ability to detect numerous objects in real-time using a webcam. Implemented in C++ and Python, it leverages a caffe model dataset. The process involves four key steps for feature detection, particularly on faces.

4.2.1 Data Preparation:In this step ,clean the image and store them in a format that can be used by caffe.we will write write python script that will handle both pre-processing and storage.

4.2.2 Model Definition:in this,we choose CNN architecture and define parameters in a configuration file with extension..prototxt.

4.2.3 Solver Definition :the solver is responsible formodel optimization.we define solver parameters in a configuration file with extension..Prototxt.

4.2.4 Model Training:we train thye model by executing one caffe command from the terminal.we willl get trained model in the file extension..caffemodel.

After training the phase, we will the the .caffemodel. trained model to make prediictions of new unseen data.

4.3 Deep Learning Classification: Deep learning, a subset of machine learning, simulates the brain's neural network using connected layers, including input, hidden, and output layers. The term "deep" indicates networks with more than two layers, allowing for the learning of complex features. Neural networks consist of interconnected neurons in hidden layers, processing input signals and propagating them based on weights, bias, and activation functions. Deep learning finds applications in driverless cars, mobile phones, search engines, fraud detection, and more. There are two main types of neural networks: shallow, with one hidden layer, and deep, with multiple layers. Feed-forward neural networks, the simplest type, transmit information only in a forward direction, from input to hidden to output layers, without loops.

4.4 Recurrent neural networks (RNNs): RNN, or Recurrent Neural Network, is a multi-layered neural network with loop connections, allowing it to learn data sequences and predict numbers or sequences. It is effective for tasks involving time-series or sequential data. Common applications include generating analytic reports, detecting abnormalities in financial statements, identifying fraudulent credit card transactions, providing image captions, and powering chatbots.

Object detection has seen significant advancements with convolutional neural networks (CNNs). These networks have proven effective in learning features directly from image pixels, eliminating the need for manual feature extraction. Techniques like FPN, SSD, and DSSD have improved object detection and recognition. Recent developments in deep convolutional neural networks (DCNN) have enhanced image classification accuracy, with approaches like faster R-CNN, YOLO, and SSD achieving state-of-the-art performance. Despite their accuracy, these methods often require intensive computation and substantial labeled training data. Ongoing research aims to address these challenges, focusing on modifying architectures for efficient real-time object detection with high accuracy.

4.5 YOLOv3 (You Only Look Once, Version 3): YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm developed by Joseph Redmon and Ali Farhadi. It uses deep convolutional neural networks to identify specific objects in videos, live feeds, or images. YOLOv3, released in 2018, is an improved version of its predecessors. It is implemented using Keras or OpenCV libraries and is known for its speed and accuracy. YOLO processes input images as structured arrays, allowing it to look at the entire image at test time for globally informed predictions. The algorithm scores regions based on similarities to predefined classes, enabling it to detect various objects in real-time applications like traffic monitoring.

5.Conclusion

Introducing YOLO, a unified model for object detection that stands out for its simplicity and direct training on full images. Unlike classifier-based approaches, YOLO is trained on a loss function directly tied to detection performance, enabling joint training of the entire model. Fast YOLO holds the title of the fastest general-purpose object detector in current literature, pushing the boundaries of real-time object detection. YOLO's versatility extends to new domains, making it an ideal choice for applications requiring swift and reliable object detection. Following detection, the image is converted to text and undergoes text-to-speech conversion.

6.References

1.A. Tiwari, A. Kumar, and G. M. Saraswat, “Feature extraction for object recognition and image classification,” International Journal of Engineering Research & Technology (IJERT), vol. 2, pp. 2278–0181, 2013.

2. J. Yan, Z. Lei, L. Wen, and S. Z. Li, “"e fastest deformable part model for object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2497–2504, New York, NY, USA, 2014.

3. T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, “Fast, accurate detection of 100,000 object classes on a single machine,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1814–1821, New York, NY, USA, 2013.

4. P. Viola and M. J. Jones, “Robust real-time face detection,” International Journal of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004.

5. C.-J. Du, H.-J. He, and D.-W. Sun, “Object classification methods,” in Computer Vision Technology for Food Quality Evaluation, pp. 87–110, Elsevier, Berlin, Germany, 2016.

6. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

7.N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proceedigs of the International Conference on Computer Vision & Pattern Recognition (CVPR’05), pp. 886–893, Berlin, Germany, 2005.

8.J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, Las Vegan, NV, USA, 2016.

9.Y. Zheng, C. Zhu, K. Luu, C. Bhagavatula, T. H. N. Le, and M. Savvides, “Towards a deep learning framework for unconstrained face detection,” in Proceedings of the 2016 IEEE 8th International Conference on BiometricsAeory, Applications and Systems (BTAS), IEEE, New York, NY, USA, pp. 1–8, 2016.

10.R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, New York, NY, USA, 2014.

11.R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448, Berlin, Germany, 2015.