Utilizing Deep Learning for Object Detection in Real Time

Mr. Sourav Gayen[1], Indranil Banerjee[1] , Bikash Kumar Chand[1], Sandip Das[1], Supriya Maji[1], Md. Umar Ali[1]

[1]NSHM Knowledge Campus, Durgapur

Abstract: Real-time object detection represents a dynamic and intricate field within the realm of computer vision. When tasked with identifying a solitary object within an image, it is referred to as image localization, while the process of detecting multiple objects within an image is termed object detection. This encompasses the identification of semantic objects in digital images and videos, categorizing them into specific groups. Applications of real-time object detection span a wide range of fields, encompassing object tracking, video surveillance, pedestrian identification, crowd counting, autonomous vehicles, facial recognition, sports ball tracking, and more. Convolutional Neural Networks (CNNs) play a pivotal role in deep learning for object detection when combined with OpenCV (Open Source Computer Vision), a programming library designed primarily for real-time computer vision tasks.

Keywords: Computer vision, Deep Learning, Convolution Neural Networks.

1. Introduction:

Object detection refers to the task of detecting and identifying real-life entities, like automobiles, bicycles, televisions, flowers, and humans, within images or videos. This technique enables a more comprehensive understanding of visual content, as it enables the identification, accurate localization, and recognition of multiple objects within an image. Object detection is widely utilized in various applications, including image retrieval, security, surveillance, and advanced driver assistance systems (ADAS).

Various methods are utilized for object detection, including:

1. Object detection based on features
2. Object detection using the Viola-Jones method
3. SVM Categorizations with HOG Features
4. Object Detection through Deep Learning

In contemporary video surveillance applications, one of the primary tasks is the detection of objects within video streams. The object detection technique serves the purpose of identifying specific objects within these video sequences and grouping the pixels that constitute these objects. Detecting objects within video sequences holds significant importance, particularly in the context of video surveillance applications. The procedure for recognizing objects in a video stream typically includes multiple stages, such as pre-processing, segmentation, foreground and background extraction, and feature extraction. Humans exhibit an extraordinary capability to effortlessly identify and recognize objects in images. The human visual system is known for its speed and precision, enabling the execution of complex tasks, such as identifying multiple objects, with minimal conscious effort. Due to the abundance of data, enhanced GPU performance, and improved algorithms, it is now straightforward to train computers for the precise detection and classification of multiple objects within images.

2. Digital Image Processing

In the realm of computerized image processing, the development of viable solutions for specific problems often demands extensive testing and experimentation. A crucial element ingrained in the development of image processing systems is the significant amount of testing and trial-and-error usually required to reach a satisfactory solution. This highlights the importance of being able to design methodologies and quickly prototype potential solutions, as it often plays a substantial role in minimizing the time and cost required to achieve satisfactory system performance.

2.1. What is DIP?

DIP stands for "Digital Image Processing." It constitutes a branch of computer science and electrical engineering dedicated to employing diverse algorithms and techniques for the manipulation, analysis, and interpretation of digital images or videos and tools. Digital image processing is used to enhance, transform, compress, and extract information from visual data, making it valuable in a diverse applications, encompassing medical imaging, computer vision, and remote sensing, image and video editing, and many other fields. It involves the use of software and hardware to process and manipulate images in a digital format, as opposed to traditional analog methods. An image can be defined as a two-dimensional function denoted by f(x, y), where x and y signify spatial coordinates, and the intensity at any coordinate pair (x, y) is referred to as the brightness or grayscale level. When x, y, and intensity values are discrete and finite, the image is categorized as a digital image. Digital Image Processing (DIP) entails manipulating digital images using a computer, where a digital image comprises finite elements known as pixels, each possessing a specific position and value. Vision, being the most advanced human sense, emphasizes the crucial role images play in perception. Unlike humans, imaging devices cover the entire electromagnetic spectrum, from gamma rays to radio waves, processing images beyond human visual familiarity.

The differentiation among image processing, image analysis, and computer vision is a topic of ongoing debate. While some define image processing as a discipline in which both input and output are images, this definition is considered restrictive. Image analysis lies between image processing and computer vision, forming a continuum with no clear boundaries. Processes in this continuum are categorized into three types: low, mid, and high-level processes. Low-level processes encompass fundamental operations such as noise reduction and contrast enhancement, where both inputs and outputs are images. Mid-level processes include tasks such as segmentation and description, with inputs being images and outputs consisting of extracted properties. High-level processes involve more complex data handling. At one extreme, there is the interpretation of identified objects, as observed in image analysis, whereas at the furthest extreme, there is the performance of cognitive functions typically linked with human vision. The application of digital image processing, as defined earlier, extends across a wide range of fields, holding considerable social and economic significance.

2.2. What is Image?

An image, in the context of processing of digital images, is a two-dimensional representation of visual data, typically in the form of a matrix or grid of picture elements (pixels). Each pixel in the image contains information about the color, brightness, or grayscale value at a specific location in the image. Images can

be photographs, illustrations, or any other visual content that can be captured or represented digitally. In the wider context of human perception, an image denotes a visual representation or portrayal of an object, scene, or concept. This representation can be either directly captured through the eyes and processed by the brain or exist in various forms, such as paintings, drawings, or photographs. Images are essential for conveying information, artistic expression, and communication.

An image is expressed as a two-dimensional function denoted by f(x, y), with x and y representing spatial coordinates, and the intensity of "T" at any given coordinate pair (x, y) signifies the image intensity at that specific point.

2.3. Processing on image:

Image processing can be categorized into three types: low, mid, and high level.

2.3.1 Low-level Processing:

1. Noise removal through preprocessing.
2. Enhancement of contrast.
3. Sharpening of the image.

2.3.2. Medium Level Processing:

1. Segmentation.
2. Detection of edges.
3. Extraction of objects.

2.3.3. High Level Processing:

1. Image analysis
2. Scene interpretation

2.4. What is the purpose of image processing?

Since digital images are inherently invisible, they must be formatted for presentation on one or multiple output devices, such as a laser printer or a monitor.

Optimization of the digital image can be achieved by improving the visibility and clarity of the elements within it. Three image processing techniques are commonly employed for this purpose. They are

1. Transforming Image to Image
2. Converting Image to Information
3. Converting Information to Image

2.4.1. Pixel :

A pixel functions as the smallest unit in an image, with each pixel assigned a single value. In an 8-bit grayscale image, the pixel value spans from 0 to 255. Each pixel retains a value directly linked to the light intensity at its particular location. Image resolution is typically measured in either Pixels per Inch (PPI) or Dots per Inch (DPI).

2.4.2. Resolution :

Resolution can be expressed in various ways, including pixel resolution, spatial resolution, temporal resolution, and spectral resolution. Pixel resolution pertains to the overall count of pixels within a digital image, with resolution, in this context, defined as M x N when an image comprises M rows and N columns. The quality of the image is positively correlated with pixel resolution, meaning that a higher pixel resolution results in a higher image quality.

Image resolution typically falls into two categories:

1. Low-resolution image
2. High-resolution image

Obtaining high-resolution images at a low cost is often impractical due to the expense involved. Therefore, there is a strong demand for imaging solutions that can enhance the quality of images without significant cost. Super-resolution imaging techniques employ specific methods and algorithms to generate high-resolution images from lower-resolution counterparts, addressing this need.

2.4.3. Gray scale Image:

A grayscale image is a function denoted as I(x, y), portraying the two spatial dimensions of the image plane. The function I(x, y) signifies the intensity or brightness level of the image at the position (x, y) within the image plane. The values of I(x, y) are non-negative, except when the image is constrained within a rectangular boundary.

2.4.4. Color Image:

This can be expressed through three functions: R(x, y) for red, G(x, y) for green, and B(x, y) for blue. An image may exhibit continuity concerning the x and y coordinates as well as intensity. To convert such an image into a digital format, both the coordinates and intensity must undergo digitization. The digitization of coordinate values is termed sampling, while the digitization of intensity values is referred to as quantization.

2.4.5. Related Technology:

R-CNN

R-CNN is an advanced system for detecting visual objects, which integrates region proposals from the lower layers with feature representations obtained through convolutional neural networks. This approach utilizes region proposal techniques to initially generate potential bounding boxes within an image, followed by the application of a classifier to assess these proposed regions.

3. Application of Object Detection

3.1. Facial Recognition

"Deep Face" is an advanced deep learning-based system for facial recognition that has been created to detect and identify human faces within digital images. A group of researchers at Facebook conceived and developed this technology. Google Photos also provides its facial recognition system, autonomously organizing photos based on the individuals depicted in the images. Facial recognition encompasses multiple components and focuses on various facial features such as the eyes, nose, mouth, and eyebrows to identify faces.

3.2. People Counting

People counting, a subset of object detection, serves various purposes such as individual identification or locating potential suspects. It is utilized for analyzing store efficiency or gathering crowd statistics during festivals. This task is often challenging due to the swift exit of individuals from the frame.

3.3. Industrial Quality Check

Object detection is pivotal in industrial operations, aiding in product identification and recognition. The visual inspection to locate specific items is fundamental in various industrial processes, including sorting, inventory control, machining, quality assurance, and packaging. Real-time inventory management poses challenges due to the complexity of tracking items. Automatic object counting and localization contribute to enhancing inventory accuracy.

3.4. Self Driving Cars

The future holds significant potential for self-driving technology, although the underlying mechanics can be exceptionally complex. It encompasses an array of techniques to sense the vehicle's environment, such as radar, laser sensors, GPS, odometers, and computer vision. Sophisticated control systems process sensory data to enable navigation while also detecting and responding to obstacles. This represents a significant advancement in the journey toward driverless cars, and it operates at a high speed.

3.5. Security

Object detection plays a vital role in the security domain, making substantial contributions to key areas such as Apple's facial recognition technology and the futuristic retina scans depicted in science fiction films. Governments also extensively employ this application to access security footage and compare it with their databases for identifying criminals or detecting objects such as vehicle license plates associated with illicit activities. The possibilities in this field are extensive.

3.6. Workflow for Object Detection and Feature Extraction

Every Object Detection Algorithm is grounded in the same fundamental principle, with their distinctions lying in their unique implementation and methods. These algorithms concentrate on extracting distinctive features from input images and subsequently employ these features to classify the image into its respective category.

4. Deep Learning

Deep learning is an approach within machine learning that guides a computer in processing various inputs through multiple layers of learning. The aim is to empower computers to make predictions and

categorize data. The processing of inputs, be they in the form of images, text, or audio, draws inspiration from the information processing of the human brain. Deep learning strives to replicate the intricate functions of the human brain, which consists of about 100 billion neurons, each connected to approximately 100,000 neighbors. In the human brain, neurons collaborate in a network, utilizing bodies, dendrites, and axons to execute complex tasks. Deep learning algorithms operate on a similar principle, with individual nodes (similar to neurons) receiving input signals, processing them, and generating an output signal. This algorithmic process involves passing input through layers of processing, where each layer generates an output that becomes the input for the subsequent layer. This sequence iterates until the final output is achieved. In the domain of deep learning, the input layer can be likened to human senses, translating observations into numerical values and binary data for the computer to process. The standardization or normalization of these variables ensures uniformity. Deep learning algorithms employ multiple layers for tasks like feature extraction and transformation. Each layer builds on the knowledge acquired from the preceding layer, establishing a hierarchy of abstract concepts. For instance, in image processing, the initial input might be a pixel matrix, and the first layer could represent edges and pixel compositions. Successive layers progressively transform the data into more advanced representations, recognizing features such as noses, eyes, and eventually entire faces. This hierarchical learning emulates the human brain's capacity to identify patterns and features in a hierarchical fashion.

4.1. What happens inside the neuron?

The processing of information in the input node occurs in numerical format, typically designated as an activation value assigned to each node. A more substantial activation value implies a stronger signal. Taking into account the strength of the connection, represented by weights, and the transfer function, this activation value is sent to the next node in the network. Each node consolidates the received activation values by calculating their weighted sum and then adjusts this sum based on its specific transfer function. Subsequently, it undergoes an activation function, determining whether the node should transmit a signal. Synapses are assigned specific weights, a crucial element in Artificial Neural Networks (ANNs). Weights play a pivotal role in the learning process of ANNs. By modifying these weights, the ANN determines the degree to which signals are transmitted. The adjustment of weights is a critical aspect of the learning process during network training. The activation signal traverses the network until it reaches the output nodes, where it is presented in an understandable format. To assess the model's performance, a cost function is employed to compare the output with the expected outcome. Several cost functions are at hand to gauge the network's error. The objective is to minimize the loss function, as a lower value signifies a closer match to the desired output. Following this, information undergoes backpropagation, initiating the learning process

as the neural network adjusts weights to minimize the cost function. This iterative process is referred to as backpropagation. In the course of forward propagation, information enters the input layer and traverses the network to generate output values. These values are then compared to the expected results, and errors are calculated and propagated backward. This backward propagation enables network training and weight updates. The algorithm's structure allows for the simultaneous adjustment of all weights, offering insights into each weight's contribution to the network's error. Once the weights have been fine-tuned to an optimal level, the network is prepared for the testing phase.

4.2. How does a neural network of the artificial kind acquire knowledge?

There are two primary approaches to instructing a program for task execution. The first method entails explicit guidance and hard programming, where one precisely outlines the desired actions for the program. On the other hand, there are neural networks. In neural networks, you supply the network with input data and the desired outputs, enabling it to autonomously acquire the necessary knowledge. By allowing the network to learn independently, the need for manually specifying all the rules is eliminated. You can establish the network's architecture and grant it the freedom to learn on its own. Once the training is completed, the network can be presented with new input data and is proficient in generating the desired outputs.

4.3. Feedforward and feedback networks

A feedforward network comprises input, output, and hidden layers, facilitating the transmission of signals in a single direction—forward. Input data enters a layer where computational operations occur, with each processing unit calculating its output based on the weighted sum of input values. These outputs then become inputs for the next layer, creating a sequential feed-forward process that spans all layers and culminates in the network's output. Feedforward networks are commonly used in fields such as data mining. In contrast, feedback networks, exemplified by recurrent neural networks, feature pathways that allow signals to move bidirectionally through loops. These networks enable all possible connections between neurons. The loops introduce non-linearity and dynamism, resulting in continuous changes until reaching a state of equilibrium. Feedback networks are frequently applied in optimization problems, aiming to identify the optimal configuration of interrelated factors.

5. Convolution Neural Networks

5.1. Artificial Neural Networks

The foundation of Artificial Neural Networks (ANNs) lies in the concept that the intricate neural connections driving the functioning of the human brain can be replicated using silicon and wires. The emulation endeavors to replicate the functions of living neurons and dendrites. The human brain consists of a vast network of 86 billion nerve cells, or neurons, interconnected with thousands of others through axons. Dendrites receive sensory stimuli from the external environment or inputs from sensory organs, initiating electrical impulses that swiftly traverse the neural network. Following this, a neuron determines whether to transmit the message to another neuron to address the issue. Artificial Neural Networks are structured with multiple nodes to mimic the biological neurons in the human brain. These artificial neurons are interconnected and engage in interactions. Nodes receive input data, perform basic operations on this data, and transmit the results to other

neurons. The output at each node is termed its activation or node value. Each connection between nodes is associated with a weight, and ANNs have the ability to learn, achieved through the adjustment of these weight values.

5.2. Neural network:

A neural network is essentially a network or circuit composed of neurons. In a modern context, it can refer to an artificial neural network consisting of artificial neurons or nodes. Therefore, a neural network can be a biological neural network, composed of real biological neurons, or an artificial neural network used for addressing artificial intelligence (AI) problems. In biological neurons, connections are represented as weights, where positive weights indicate excitatory connections, and negative values signify inhibitory connections. These weights adjust all input signals and are then summed, a process known as linear combination. Finally, an activation function regulates the amplitude of the network's output, typically falling within a specified range, such as 0 to 1 or -1 to 1. Artificial neural networks find applications in predictive modeling, adaptive control, and scenarios where training with datasets is feasible. These networks can also exhibit self-learning, drawing conclusions from complex and seemingly unrelated sets of information based on their experiences. A deep neural network (DNN) is an artificial neural network (ANN) characterized by multiple intermediate layers positioned between the input and output layers. The primary function of a DNN is to discern the appropriate mathematical transformations required to convert input data into output data, encompassing both linear and non-linear relationships.

6. Open computer Vision

OpenCV, an abbreviation for Open Source Computer Vision Library, is an open-source software library crafted as a companion for computer vision and machine learning. The inception of OpenCV aimed to create a standardized infrastructure for computer vision applications, simplifying the integration of machine perception into commercial products. Its BSD-licensed nature allows for easy adaptability and modification for businesses. Featuring 2500 optimized algorithms covering both traditional and cutting-edge computer vision and machine learning algorithms, OpenCV stands as a versatile and robust library. These algorithms play roles in various functions, including detecting and recognizing faces, categorizing objects, analyzing human actions, monitoring camera motions, and tracking dynamic objects in videos. OpenCV can generate 3D models of objects, create 3D point clouds using stereo cameras, merge images to produce high-resolution representations of entire scenes, identify similar images within a picture database, remove red-eye effects from flash photography, monitor eye movements, recognize landscapes, and establish markers for augmented reality overlays. Officially introduced in 1999, the OpenCV paper originated as an initiative by Intel Research aimed at enhancing CPU-intensive applications. It was part of a broader series of papers, including real-time ray tracing and 3D display walls. Key contributors to the paper included optimization experts within Intel Russia, along with participation from Intel's Performance Library Team.

In its initial phases, the paper's objectives were outlined as follows:

1. Advance vision research by supplying open and optimized code for fundamental vision infrastructure, thereby reducing the necessity for duplicative development efforts.
2. Disseminate vision knowledge by offering a common infrastructure that developers can build on, improving code readability and transferability.
3. Advance vision-based commercial applications by furnishing freely available, portable, and performance-optimized code, utilizing a licensing model that doesn't mandate the code to be open or free itself.

The initial alpha release of OpenCV made its public debut at the IEEE Conference on Computer Vision and Pattern Recognition in 2000. Following that, five beta versions were released between 2001 and 2005. The first official 1.0 version was released in 2006, followed by a "pre-release" of version 1.1 in October 2008.

In October 2009, OpenCV experienced its second major release. OpenCV 2 introduced significant changes to the C++ interface, prioritizing simpler, more type-safe patterns, introducing new functions, and improving the performance of existing ones, especially on multi-core systems. The paper embraced a biannual release schedule, with official releases occurring every six months. Currently, development is undertaken by an independent Russian team with support from commercial corporations.

7. Result

In this developed paper, we have selected approximately 10 to 20 objects for detection during the training phase. Some of these objects include 'person,' 'car,' 'cell phone,' 'laptop,' 'cup,' 'bottle,' and others. The paper's output showcases the detected objects with a rectangular box surrounding each object, accompanied by a label which indicates its name and the accuracy of the detection displayed above it. The system can confidently identify numerous objects within a single image.

Figure -1

Figure -2

Figure -3

Figure -4

8. Conclusion

The field of deep learning-based object detection has gained significant prominence in recent years. This paper begins with the implementation of generic object detection pipelines, presenting foundational architectures for associated tasks. Through the integration of object detection with deep learning and OpenCV, coupled with the utilization of efficient, threaded video streams using

OpenCV, the authors successfully accomplished three common tasks: object detection, face detection, and pedestrian detection. Recognizing the potential impact of camera sensor noise and varying lighting

conditions on results, the authors identified these factors as potential challenges in object recognition. Typically, this entire process demands the use of GPUs due to computational intensity, but the authors effectively executed it using CPUs, achieving efficient performance in considerably less time. Object detection algorithms seamlessly combine aspects of both image classification and object localization. They take an input image, producing output with bounding boxes corresponding to the number of objects in the image, each labeled with its category. The output includes information about the position, height, and width of the bounding boxes, offering a comprehensive representation of the detected objects.

References:

1. https://ieeexplore.ieee.org/abstract/document/6731341/authors#authors
2. https://www.sciencedirect.com/science/article/pii/S187705092031276X
3. Redmon, J, Divvala, S, Girshick, R., Farhadi, A (2016) You only look once: Unified, real-time object detection. In: CVPR
4. P. Poirson, P. Ammirato, C.-Y. Fu, J. Liu, Wei Koeck, A. C. Berg (2016) Fast single shot detection and pose estimation. International Conference on 3DVision(3DV).
5. R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, J. M. Z. Maningo (2018) Object Detection Using Convolutional Neural Networks. TENCON 2018-2018 IEEE Region 10 Conference
6. D. Alamsyah and M. Fachrurrozi (2017) Faster R-CNN with Inception V2 for Fingertip Detection in homogeneous Background Image. IOP Conf. Series: Journal of Physics: Conf. Series 1196.
7. P. Viola, M. Jones (2001) Rapid object detection using a boosted cascade of simple features. vol. 1, pp. 511-518
8. R. Girshick (2015) Fast R-CNN. in IEEE International Conference on Computer Vision (ICCV).
9. S. Ren, K. He, R. Girshick, and J. Sun. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99.
10. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:1–10.
11. Abdulghafoor, Nuha H., and Hadeel N. Abdullah. “A novel real-time multiple objects detection and tracking framework for different challenges.” Alexandria Engineering Journal 61, no. 12 (2022): 9637-9647.
12. Hewawasam, Hasitha S., M. Yousef Ibrahim, Gayan Kahandawa, and Tanveer A. Choudhury. “Comparative study on object tracking algorithms for mobile robot navigation in gps-denied environment.” In 2019 IEEE International Conference on Industrial Technology (ICIT), pp. 19-26. IEEE, 2019.