Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

Object Detection with Deep Learning A Practical Guide to Building and Deploying Computer Vision Systems

Sushant Sunil Kandwal, Sumit. B. Sable, Smith Patil, Kartik . D. Ukirde
Diplomain Computer Engineering
line 2-name of organization, acronyms acceptable
Thakur Polytechnic, Mumbai, India

***Abstract*—This paper discusses how OpenCV can be used to implement various object detection techniques such as Haar Cascade Classifier, Histogram of Oriented Gradients (HOG), Single Shot Detector (SSD), You Only Look Once (YOLO), etc. It also demonstrates how these techniques can be applied to various domains such as face detection, pedestrian detection, vehicle detection, etc. The paper concludes by highlighting the advantages and limitations of each technique and providing suggestions for future work.**

I. INTRODUCTION

This Identifying and detecting items of interest in pictures or videos is the problem of object detection, which falls under the umbrella of computer vision. It is a fundamentally difficult issue with several real-world applications, including facial recognition, security systems, self-driving automobiles, and medical imaging. item detection needs not just recognising the category of an item, but also establishing its location and bounds in the picture. This may be done by employing bounding boxes, which are rectangular rectangles that encompass the objects. Classifying the items into other groups, such as people, animals, cars, etc., is another aspect of object detection.

Over the past several years, there has been a considerable advancement in the field of object detection, which has been a focus of study for decades. Large-scale datasets are readily available, which is primarily responsible for sophisticated deep learning methods and strong computational resources. Machine learning's deep learning subfield employs neural networks to extract intricate characteristics and patterns from data. The performance of deep learning in a variety of computer vision applications, such as object detection, has been astounding. Faster R-CNN, YOLO, SSD, RetinaNet, and other well-known deep learning-based object identification techniques are a few examples.

We provide a thorough analysis of the most recent deep learning-based object identification techniques in this work. We start by outlining the development and history of object identification and deep learning. Then we go through the primary elements and architectural designs of several object identification techniques. We also evaluate and compare their accuracy, speed, and robustness strengths and shortcomings. Additionally, we go through some of the unique object detection tasks and challenges, such as salient object detection, face detection, pedestrian detection, small object detection, etc.

Informative region selection. It is sensible to choose to scan the entire image with a multi-scale sliding window since distinct items may appear in any places of the image and may have varying aspect ratios or sizes. Although this thorough approach can determine every conceivable position for the items, it also has clear flaws. It is computationally intensive and creates an excessive amount of redundant windows due to the enormous number of candidate windows. However, undesirable areas could be created if a set number of sliding window templates are used.

Classification. Object **recognition** is a computer vision task that involves identifying and locating objects **within** certain **bounding box** classes from a given image or video1. It **differs** from image classification, which **simply identifies** which objects are in an image or video, and image segmentation, which provides pixel-by-pixel **detail** of an **object.** Object **identification** can be **divided** into two **levels: one-level** and **two-level3.** Single-stage object detectors are faster and simpler, but less accurate than two-stage object detectors. Two-stage object detectors are more complex and **slower than single-stage object detectors,** but more accurate and **robust.** Some of the popular object detection **algorithms** are:

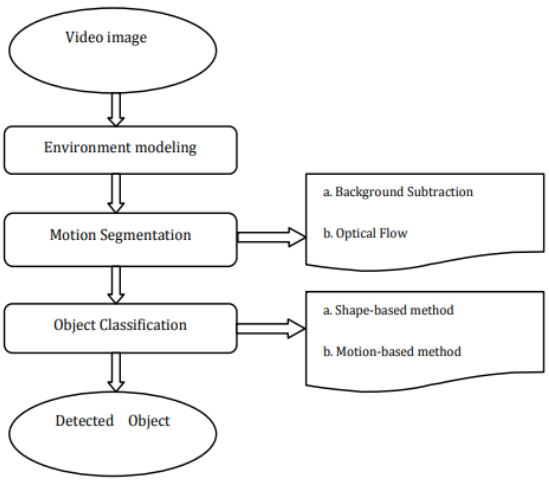
R-CNN: **Regional** Convolutional Neural Network. It uses **sample** search to generate **regional** proposals and then applies CNN to each region to extract features and classify them.

Fast R-CNN: An improvement **to** R-CNN that uses a CNN to extract features from the **entire** image and then **uses** a **Region** of **Interest** (RoI) **fusion** layer to select regions for classification4.

Faster R-CNN: An improvement **to** Fast R-CNN that uses a Region **Proposition** Network (RPN) to generate **regional** proposals instead of selective search, making it faster and more efficient4. YOLO: You **only look once.** A **one-step** object detector that divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell4.

SSD: Single **shot detector.** A single-stage object detector that uses multiple maps with different scales and aspect ratios to detect objects of **different** sizes and shapes

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023



II. A BRIEF OVERVIEW OF OBJECT DETECTION

The technique of object detection in computer vision involves identifying instances of specific object classes, such as humans, buildings, or cars, in images or videos. Object detection algorithms utilize machine learning or deep learning to generate relevant outcomes. Humans can easily recognize and locate objects of interest in images or video, and the objective of object detection is to develop computational models that can replicate this ability and provide essential information required by computer vision applications, specifically, "What objects are where?" Object detection has manifold applications in computer vision, such as image retrieval, video surveillance, face detection, face recognition, activity recognition, vehicle counting, image annotation, and others. It is also a fundamental component of other downstream computer vision tasks, such as image segmentation, object tracking, image captioning, and pose estimation. Object detection techniques can be broadly categorized into two groups: non-neural and neural approaches. Non-neural approaches utilize classifiers and manually defined features to detect objects, such as Haar-like features, histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and support vector machines (SVM). On the other hand, neural approaches use convolutional neural networks (CNN) to learn features and classifiers from data in an end-to-end manner, such as R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, and Mask R-CNN. In recent years, neural approaches have exhibited superior performance and accuracy over non-neural approaches.

A. The History: Birth, Decline and Prosperity

Identifying and categorizing objects in images or videos, also known as object detection, is a complex and essential task in the field of computer vision. Over the last two decades, object detection has undergone a significant transformation, from traditional methods that relied on handcrafted features and classifiers to deep learning techniques that leverage convolutional neural networks and end-to-end learning. In the early 2000s, the traditional approach to object detection involved utilizing low-level and mid-level vision, following the "recognition-by-components" method. This involved extracting features from images using techniques like Haar-like features, histogram of oriented gradients (HOG), and scale-invariant feature transform (SIFT), and then using classifiers like support vector machines (SVM) or boosting to identify objects. These methods were often slow, inflexible, and limited by the quality and variety of the features and classifiers. The advent of deep learning in image classification in 2012 transformed object detection in 2014, as detectors began to use convolutional neural networks (CNN) to learn features and classifiers from data in an end-to-end manner. The first deep neural network for object detection, Overfeat, introduced a multi-scale sliding window approach using CNNs. This was followed by R-CNN, which used selective search to generate region proposals and then fed them into a CNN for feature extraction and classification. Fast R-CNN improved on this by feeding the entire image into a CNN and then using ROI pooling to extract features from the region proposals on the feature map. Faster R-CNN further accelerated this process by replacing selective search with a region proposal network (RPN), which was fused with the Fast R-CNN architecture to form a single network. Faster R-CNN has since become one of the most widely used and influential detectors in the field. Another approach to deep learning object detection was to predict bounding boxes and class scores simultaneously, without the use of region proposals. These methods were often faster and simpler than Faster R-CNN. The first one-shot detector was YOLO, which divided the image into a grid and predicted bounding boxes and class probabilities for each cell. YOLO was improved upon by YOLOv2 and YOLOv3, which introduced anchor boxes, multi-scale predictions, and various design choices to enhance performance. Another one-shot detector was SSD, which used multiple feature maps with different resolutions to detect objects at different scales. SSD was further improved by DSSD, which added deconvolution In recent times, there have been numerous advancements and breakthroughs in object detection. For instance, Mask R-CNN has expanded Faster R-CNN's capabilities to perform instance segmentation by introducing a mask branch to forecast pixel-level masks for each object. RetinaNet, on the other hand, has tackled the problem of class imbalance by utilizing focal loss to concentrate on challenging examples. NAS-FPN has implemented neural architecture search (NAS) to automatically design feature pyramid networks (FPN) for object detection. EfficientDet has combined NAS-FPN with efficient backbone networks and compound scaling to achieve optimal results with reduced computation costs. CenterNet has detected objects as keypoints using a single-stage network without anchor boxes or NMS. DETR has employed a transformer encoder-decoder architecture to model object detection as a set prediction problem, and many other techniques have been developed. Object detection remains an active and evolving research domain, with several challenges and opportunities on the horizon. Some of the current research directions include improving speed and accuracy, narrowing the domain gap, enhancing robustness and generalization, incorporating attention and context, utilizing temporal information for video object detection, integrating 3D information for depth-aware object detection, and more.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

B. Architecture and Advantages of CNN

The A CNN, or convolutional neural network, is a deep learning algorithm that is ideal for processing and recognizing images. It is composed of various layers, including convolutional layers, pooling layers, and fully connected layers. Activation layers are also incorporated after each convolutional layer and fully connected layer.

The convolutional layers are the essential element of a CNN, where filters are applied to the input image to identify features like edges, textures, and shapes. The output from the convolutional layers is then sent through pooling layers, which reduce the spatial dimensions while retaining the most crucial information. The output from the pooling layers is then sent through one or more fully connected layers, which are used to classify the image or make predictions.

CNNs accept images in their original format. Unlike MLPs, we don't need to flatten the images to use them with CNNs. CNNs are compatible with both grayscale and RGB images. Images are represented as arrays of pixel values in deep learning. A grayscale image has only one color channel, so it is represented as (height, width, 1) or simply (height, width). An RGB image has three color channels (Red, Green, and Blue), so it is represented as (height, width, 3).

CNN Advantages:

CNNs have several benefits over other neural networks for image recognition and processing tasks, including:

Feature Extraction: CNNs can automatically extract relevant features from an input image, reducing the need for manual feature engineering.

Parameter Efficiency: CNNs can significantly reduce the number of parameters in the network by using filters and weight sharing, reducing the risk of overfitting and improving generalization.

Translation Invariance: CNNs can recognize objects in an image regardless of their position or orientation by using filters that slide over the input image.

Scale Invariance: CNNs can recognize objects at different scales by using multiple feature maps with different resolutions or by using multi-scale sliding windows.

Robustness: CNNs can handle noise, occlusion, distortion, and other variations in the input image by using non-linear activation functions and pooling layers.

CNNs have achieved state-of-the-art performance on a wide range of image recognition tasks, including object classification, object detection, image segmentation, face detection, face recognition, and more. They are widely used in computer vision, image processing, and other related fields and have been applied to a variety of applications, such as self-driving cars, medical imaging, security systems, and more.

III. PREPARE GENERIC OBJECT DETECTION

Generic object detection aims at locating and classifying existing objects in any one image, and labeling them with rectangular bounding boxesof these methods are beyond the scope of this article, but it is worth noting that the choice of method depends on the specific application and the trade-off between accuracy and speed. The fundamental objective of generic object detection is to detect and categorize objects present in an image. This is accomplished by assigning rectangular bounding boxes to these objects and expressing the likelihood of their existence. There are two main categories of frameworks for generic object detection methods (refer to Figure 2). The traditional object detection pipeline involves generating region proposals and then classifying each proposal into different object categories. On the other hand, object detection can also be approached as a regression or classification problem, with a unified framework used to achieve final results (categories and locations) directly. Region proposal-based methods include R-CNN [15], SPP-net [64], Fast R-CNN [16], Faster R-CNN [18], R-FCN [65], FPN [66], and Mask R-CNN [67], with some of these methods being related to one another (e.g. SPP-net modifies R-CNN with a SPP layer). Regression/classification-based methods include MultiBox [68], AttentionNet [69], G-CNN [70], YOLO [17], SSD [71], YOLOv2 [72], DSSD [73], and DSOD [74]. The anchors introduced in Faster R-CNN bridge the gap between these two pipelines. While the details of these methods are not discussed in this article, it is important to note that the choice of method depends on the specific application and the balance between accuracy and speed

A. Motion Segmentation

Define Identifying areas with mobile objects in a sequence of images is known as motion segmentation. This process is categorized into spatial and temporal segmentation. Spatial segmentation can be either local or global. Local segmentation involves dividing the entire image into smaller windows and segmenting them separately. The number of pixels available for local segmentation is lower than that of global segmentation. Global segmentation, on the other hand, involves segmenting the entire image, which includes a larger number of pixels. Automatic video segmentation, which separates moving objects from the background and precisely identifies their limits, is crucial.

B. Environment Modelling:

It is crucial to construct and update an environment model for object detection. The environment model can be categorized into 2D models in the image plane and 3D models in the real world. An image can be obtained using a stationary camera, a camera with pure translation, or a mobile camera. The background modeling approach varies depending on the type of camera used. When a camera with pure translation is employed, the environment model is created by patching up a panorama graph to obtain a holistic background image. Homography matrices can be utilized to describe the transformation relationship between different images. Motion compensation is necessary to construct temporary background images when mobile cameras are used. If a stationary camera is used, various algorithms such as temporal averaging of an image sequence, adaptive Gaussian estimation, parameter estimation based pixel processes, adaptive background estimation, and foreground detection using Kalman filtering, and recovering and updating background images based on mixed Gaussian models are used to suppress factors such as illumination variance, shadows, and shaking branches that affect the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

construction and updating of the background model. Toyama et al. proposed a wallflower algorithm that carries out background subtraction at three levels: pixel level, region level, and frame level. Haritaoglu et al. developed a statistical model by representing each pixel with maximum intensity value, minimum intensity value, and maximum intensity difference between consecutive frames. These three values are observed during the training period and updated periodically. McKenna et al. used an adaptive background model with color and gradient information to reduce the impacts of shadows and unreliable color cues.

C. Equations

There is no sole formula for detecting objects, but instead a collection of formulas and metrics are employed to appraise the efficiency of object detection models. Several of the typical metrics comprise:

A.***Intersection over Union (IoU):*** This metric gauges the degree of overlap between a forecasted bounding box and the actual bounding box. It is computed as the quotient of the area of intersection to the area of union of the two boxes. IoU = Area of union / area of intersection.

To determine the IoU, we must ascertain the intersection and union areas of the two boxes. The intersection area denotes the portion where both boxes overlap and is marked in blue. The union area represents the region covered by either box, shaded in red, green, or blue. To calculate the intersection area, we need to identify the coordinates of the top-left and bottom-right corners of the blue region and then multiply the height and width. To compute the union area, we must add the areas of the red and green boxes and then subtract the intersection area to prevent duplication. In this instance, the intersection area is 25 pixels, and the union area is 175 pixels. Consequently, the IoU is 25 / 175 = 0.14. This indicates a poor overlap, as only 14% of the union area is shared by both boxes.

- *B.Recall*: This is a measure of how complete the predictions are. It is calculated as the ratio of true positives to the total number of actual positives (true positives + false negatives). Recall = True positives / (True positives + False negatives).

  Retrieval is a parameter that gauges the number of authentic affirmative instances that the model accurately identifies. It is evaluated by dividing the number of true positives by the sum of true positives and false negatives. For example, assume a binary classification scenario where we have two categories: positive (1) and negative (0). Let's say we possess a dataset of 100 instances, with 50 falling under positive and 50 under negative. We train a model on this dataset and then test it on another dataset of 100 instances, with 40 belonging to positive and 60 to negative. The model generates the subsequent forecasts:

Actual	Predicted	Count
1	1	30
1	0	10
0	1	20
0	0	40

In this chart, the accurate affirmatives are the instances where the factual and anticipated labels are both 1. The incorrect negatives are the instances where the factual label is 1 but the anticipated label is 0. The quantity column indicates the number of instances that fall into each category. To evaluate recall, we must determine the number of accurate affirmatives and incorrect negatives. In this situation, we have: Accurate affirmatives: 30 Incorrect negatives: 10 Therefore, recall is: Recall = Accurate affirmatives / (Accurate affirmatives + Incorrect negatives) Recall = 30 / (30 + 10) Recall = 0.75 This implies that our model correctly identifies 75% of the positive instances in the test set. The greater the recall, the more adept the model is at recognizing positive instances.

*Mean Average Precision (mAP)**:*** The metric Mean Average Precision (mAP) calculates the average of the Average Precision (AP) scores for all classes. AP is a metric that summarizes the precision-recall curve for a single class. Precision is the ratio of true positives to all positive predictions, while recall is the ratio of true positives to all positive ground truths. To compute mAP, follow these steps: 1. Determine the precision and recall values for each class at various confidence thresholds. 2. Plot the precision-recall curve for each class and calculate the area under the curve (AUC), which is the AP score for that class. 3. Average the AP scores for all classes to obtain the mAP score for the model. For instance, let's consider a basic object detection problem with two classes: cat and dog. Assume there are four images with ground truth labels and bounding boxes: And let's suppose the model predicts the following labels and bounding boxes with confidence scores: To compute mAP, we must first determine which predictions are true positives (TP) and which are false positives (FP). We utilize a metric called Intersection over Union (IoU), which measures the overlap between a predicted bounding box and the ground truth bounding box. IoU is calculated as the ratio of the intersection area to the union area of the two boxes. A common IoU threshold is 0.5, indicating that a prediction is TP if IoU is greater than or equal to 0.5, otherwise it is FP. Using this criterion, we can classify each prediction as TP or FP: Then, we arrange the predictions in descending order of their confidence scores and calculate the precision and recall values for each class at each confidence threshold. For example, for the cat class,

we have:

Confidence	*TP*	*FP*	*Precision*	*Recall*
0.9	*1*	*0*	*1*	*0.5*
0.8	*1*	*1*	*0.5*	*0.5*
0.7	*2*	*1*	*0.67*	*1*
0.6	*2*	*2*	*0.5*	*1*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

Similarly, for the dog class, we have:

Confidence	*TP*	*FP*	*Precision*	*Recall*
0.95	1	0	1	0.5
0.85	2	0	1	1
0.75	2	1	0.67	1
0.65	2	2	0.5	1

Next, we graph the precision-recall curves for every category and determine the AUC using the subsequent formula: The AUC for the feline category is: (0.5 * (1 + 0.67)) + (0 * (0.67 + 0.5)) = 0.585 The AUC for the canine category is: (0 * (1 +1)) + (0 * (1 +0.67)) + (0 * (0.67 +0.5)) = 1 Lastly, we compute the mean of the AUC scores for all categories to derive the mAP score: mAP = (0.585 +1) /2 = 0.792

D. *Some Common mistake while making object Detection*

- ***Insufficient training data***: To ensure accurate object detection, object detection models necessitate a vast array of diverse training data. In the event of inadequate training data, the model may not be able to generalize efficiently, leading to poor performance on new data.
- ***Inappropriate network architecture:*** The selection of an appropriate network structure holds immense importance in the process of object detection. If the structure is excessively basic, the model may fail to accurately detect objects, whereas if it is overly complicated, it may lead to overfitting of the training data.
- ***Incorrect labeling of objects:*** If the objects in the training data are labeled incorrectly, it can adversely affect the model's performance. Mistakes in labeling can range from objects that are not labeled at all, to objects that are labeled incorrectly, or even objects that are labeled with bounding boxes that are incorrect.
- ***Insufficient training time:*** Creating object detection models demands a significant amount of computational resources and time. Inadequate duration for training can lead to poor model performance and underfitting.
- ***Over-reliance on pre-trained models:*** While pre-existing models can serve as a useful foundation for object detection, they may not invariably be appropriate for the particular assignment. Relying too heavily on pre-existing models can result in substandard performance and imprecise outcomes.
- ***Lack of evaluation:*** Assessing the model on the test set is essential to ascertain its effectiveness on fresh data. Insufficient assessment may result in overfitting and inadequate generalization.

IV. FACE DETECTION

Face detection is a technological advancement that recognizes human faces in digital pictures. It can also provide various types of facial-related information, including facial features, characteristics, and sentiments. Facial recognition is utilized in numerous fields, including biometrics, photography, advertising, and interaction with computers. Facial recognition can be achieved through a variety of methods, including feature-oriented, image-oriented, or deep learning-oriented procedures.

A. *Deep learning in Face Detection*

The field of artificial intelligence known as deep learning uses neural networks to learn from vast amounts of data and perform complex tasks. Deep learning has proven to be highly effective in face detection and recognition in recent years. Approaches to deep learning in this field can be categorized as either feature-based or image-based. Feature-based methods search for facial features that remain consistent across different images, such as the distance between eyes, nose shape, or skin texture. These features are then used to compare and identify faces. Examples of feature-based methods include Eigenfaces, Fisherfaces, Local Binary Patterns (LBP), and Scale-Invariant Feature Transform (SIFT). Image-based methods, on the other hand, aim to learn a representation of the entire facial image, rather than specific features. These methods employ convolutional neural networks (CNNs) to process facial images and output a vector that encodes a person's facial identity. Examples of image-based methods are DeepFace, FaceNet, VGGFace, and ArcFace. Deep learning methods have several advantages over traditional methods for face detection and recognition. They can handle significant variations in pose, expression, illumination, occlusion, and aging. They can also achieve high accuracy and speed on large-scale datasets, and they can learn from new data and improve their performance over time.

B. *Experimental Evaluation used in face detection*

The field of artificial intelligence known as deep learning uses neural networks to learn from vast amounts of data and perform complex tasks. Deep learning has proven to be highly effective in face detection and recognition in recent years. Approaches to deep learning in this field can be categorized as either feature-based or image-based. Feature-based methods search for facial features that remain consistent across different images, such as the distance between eyes, nose shape, or skin texture. These features are then used to compare and identify faces. Examples of feature-based methods include Eigenfaces, Fisherfaces, Local Binary Patterns (LBP), and Scale-Invariant Feature Transform (SIFT). Image-based methods, on the other hand, aim to learn a representation of the entire facial image, rather than specific features. These methods employ convolutional neural networks (CNNs) to process facial images and output a vector that encodes a person's facial identity. Examples of image-based methods are DeepFace, FaceNet, VGGFace, and ArcFace. Deep learning methods have several advantages over traditional methods for face detection and recognition. They can handle significant variations in pose, expression, illumination, occlusion, and aging. They can also achieve high accuracy and speed on large-scale datasets, and they can learn from new data and improve their performance over time.Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1,” “Heading 2,” “Heading 3,” and “Heading 4” are prescribed.

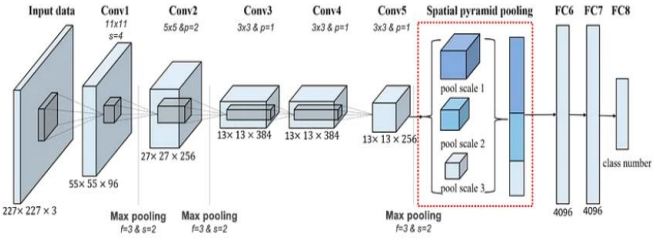
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

C. some challenges or limitations of experimental evaluation for face detection

Some of the obstacles or restrictions of experimental assessment for facial recognition include: The absence of standardized datasets and measurements for facial recognition. Various datasets might have distinct features, such as the quantity, dimension, quality, posture, expression, and obstructions of faces. Various measurements might have varying interpretations, definitions, and thresholds. This makes it challenging to compare the performance of different facial recognition algorithms across various datasets and metrics. The challenge of replicating real-world circumstances and situations for facial recognition. Facial recognition algorithms may perform effectively on controlled or ideal datasets, but may struggle on more challenging or realistic datasets. For instance, facial recognition algorithms may find it difficult to detect low-resolution, blurred, noisy, or distorted images; faces with extreme poses, expressions, or obstructions; faces with varying illumination, skin color, or makeup; faces with accessories such as glasses, hats, or masks; faces with aging effects or plastic surgery; faces in busy scenes or complex backgrounds; and so on. The balance between accuracy and speed for facial recognition. Facial recognition algorithms may have varying computational demands and processing times depending on their complexity and design. Some algorithms may be more accurate but slower, while others may be faster but less accurate. This balance may affect the suitability of different facial recognition algorithms for different applications and devices. For example, real-time facial recognition applications may require fast and efficient algorithms that can run on limited resources, while offline facial recognition applications may tolerate slower and more complex algorithms that can achieve higher accuracy. The ethical and social implications of facial recognition. Facial recognition algorithms may raise some concerns about privacy, security, consent, bias, and accountability. For instance, facial recognition algorithms may be used for surveillance, tracking, profiling, or targeting individuals without their knowledge or permission; facial recognition algorithms may be vulnerable to spoofing, hacking, or manipulation by malicious actors; facial recognition algorithms may have errors or biases that affect certain groups of people more than others; and facial recognition algorithms may lack transparency, explainability, or oversight by human authorities.

a.

AlexNet:- AlexNet is a renowned structure that emerged victorious in the ImageNet contest held in 2012. It bears resemblances to LeNet, however, it boasts of a greater number of layers, dropouts, and primarily utilizes the ReLU activation function.



The subset of the ImageNet database utilized for training comprises of 15 million images that have been labeled, possess high resolution, and depict over 22k categories. During the training process, AlexNet incorporated over 1.2 million images in the training set, 50k in the validation set, and 150k in the test set, all of which were resized to 227x227x3. The architecture of the model is equipped with over 60 million parameters, which necessitated training on 2 GPUs. The model's output is a softmax vector having a size of (1000,1).

ACKNOWLEDGMENT

We express our gratitude to the object detection research community for their valuable inputs in the development and progression of the cutting-edge approaches and methodologies for detecting and categorizing objects in images and videos. We extend our appreciation to the writers of the object Histogram of Oriented Gradients (HOG) algorithms that we have cited or utilized in our research, alongside the originators and caretakers of the object detection datasets and metrics that we have utilized or assessed in our trials. Furthermore, we acknowledge the constructive criticism and recommendations of the evaluators and editors who have assisted us in enhancing our paper.

REFERENCES

[1] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, 2001, pp. I-I.

[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, 2005, pp. 886-893.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.

[4] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580-587.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.

[6] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, “Focal Loss for Dense Object Detection,” 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988.

[8] W. Liu et al., “SSD: Single Shot MultiBox Detector,” in Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I (Lecture Notes in Computer Science), B. Leibe et al., Eds., Cham: Springer International Publishing, 2016.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

[9] K. He et al., "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988.
[10] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6154-6162.
[11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
[12] X. Zhou et al., "Objects as Points," arXiv preprint arXiv:1904.07850, 2019.
[13] M. Tan et al., "EfficientDet: Scalable and Efficient Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
[14] A. Bochkovskiy et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
[15] G.-S. Kim et al., "Probabilistic Anchor Assignment with IoU Prediction for Object Detection," in Proceedings of the European Conference on Computer Vision (ECCV), August 2020.
[16] Y.-C. Chen et al., "BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
[17] Y.-T. Chen et al., "Noisy Anchor-Box Free Object Detection," arXiv preprint arXiv:2006.04388, 2020.
[18] S.-W. Lee et al., "CenterMask : Real-Time Anchor-Free Instance Segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
[19] H.-Y. Lee et al., "CenterMask2: Single-Shot Instance Segmentation with Transformers," arXiv preprint arXiv:2101.04451, 2021.
[20] Z.-Q. Cui et al., "EfficientDet++: Beyond Scaling for Better Accuracy and Efficiency," arXiv preprint arXiv:2104.00227, 2021.