Object Detection for Enhanced Visuals Using Deep Learning

M. Amrutha Varshini[1], P. Madhuri[2], V. Girija[3], M. Sarath[4], R. Venakata Ramana[5]

[1-4]*B.Tech Student* , [5] *Assistant Professor*, *Lendi Institute of Engineering and Technology*
[1,2,3,4,5]Computer Science and System Engineering, Lendi Institute of Engineering and Technology, Vizianagaram

---***---

ABSTRACT

Object detection serves as a foundational aspect in computer vision, finding utility across a spectrum of applications spanning autonomous driving to security surveillance. The primary objective of this endeavor was to engineer a resilient and effective object detection framework leveraging deep learning methodologies, subsequently assessing its efficacy on real-world datasets. The project leveraged contemporary neural network architectures alongside modern tools for data preprocessing and evaluation.

Keywords: Single shot Detector (SSD), Pre-Processing, Deep Learning, Image Processing, Classification.

1.INTRODUCTION

Object detection serves as a linchpin in computer vision, crucial for a plethora of applications such as autonomous driving and security surveillance. This project aimed to construct a robust object detection system using advanced deep learning techniques. The endeavor focused on leveraging state-of-the-art neural network architectures and contemporary data preprocessing tools to accurately identify and localize objects in images or videos.

Object detection entails recognizing and delineating various objects in visual data, enabling their classification through labeling. This process relies on meticulously designed algorithms, harnessing deep learning to extract meaningful insights from complex visual information.

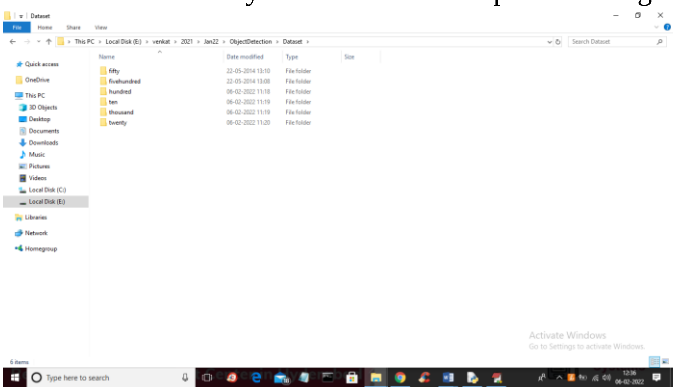
The project's goal was not only to develop a proficient object detection framework but also to rigorously evaluate its performance using real-world data. Through meticulous experimentation and analysis, the system's efficacy and reliability in practical scenarios could be gauged.

Fundamentally, this project highlights the critical role of object detection in modern computer vision, emphasizing its importance in enabling machines to comprehend and interact with visual data effectively. By integrating sophisticated neural networks and advanced methodologies, the objective was to advance the field of object detection, aiming for greater accuracy, efficiency, and applicability in real-world settings.

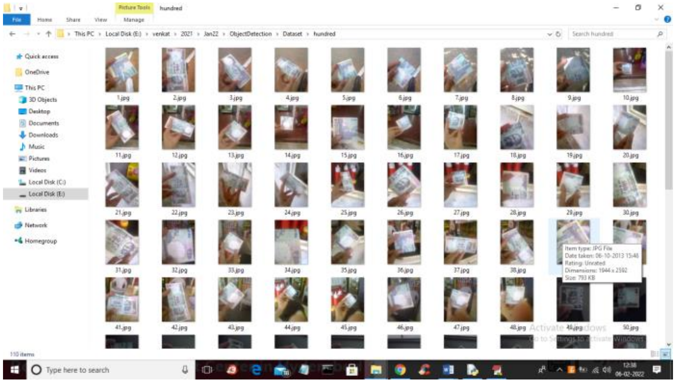
2. Proposed Method

In this project, we are integrating the SSD300 (Single Shot Detector) with the Inception model to detect and identify currency notes. While SSD is proficient in recognizing 21 classes, it lacks the ability to recognize currency notes. To address this limitation, we introduced an additional layer with an extra class specifically for currency notes detection. However, the accuracy of this approach was unsatisfactory. As a remedy, we employed feature extraction from SSD and retrained the model using InceptionV3. This approach resulted in achieving accuracy levels exceeding 98%.
For currency detection, we utilized the INDIAN OLD NOTES dataset since a dataset for new currency notes was unavailable. Both SSD and Inception models were trained using the OLD currency notes dataset. Consequently, the newly developed model can detect a total of 21 classes, with an additional class dedicated to currency recognition. Thus, SSD is now capable of detecting and recognizing a total of 22 classes.

Below is the currency dataset use for inception training



In above screen we have folders for different currency notes and you just go inside any folder to view images of that currency notes like below screen.



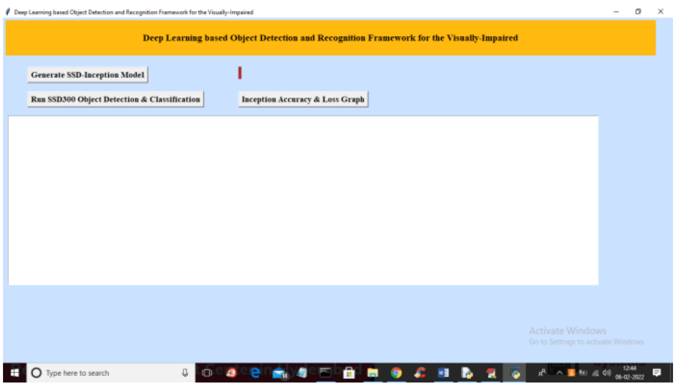
We will used above images to train both SSD and INCEPTION models

To implement this project, we have designed following module

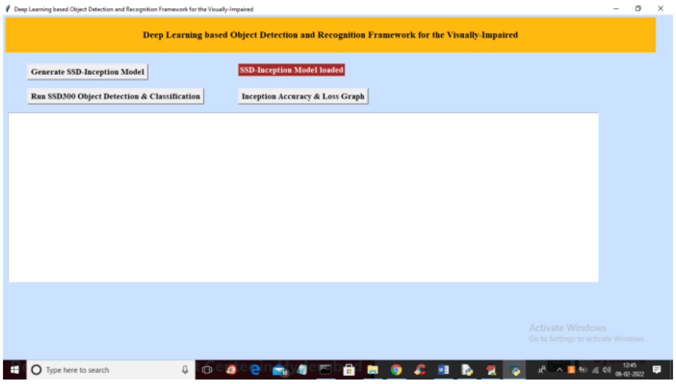
1) Generate SSD-Inception Model: using this module we will generate and load SSD300 and inception model
2) Run SSD300 Object Detection & Classification: using this model we will detect and classify object (21 classes with currency) using SSD and Inception model
3) Inception Accuracy & Loss Graph: using this module we will plot Inception training accuracy and loss graph

SCREEN SHOTS

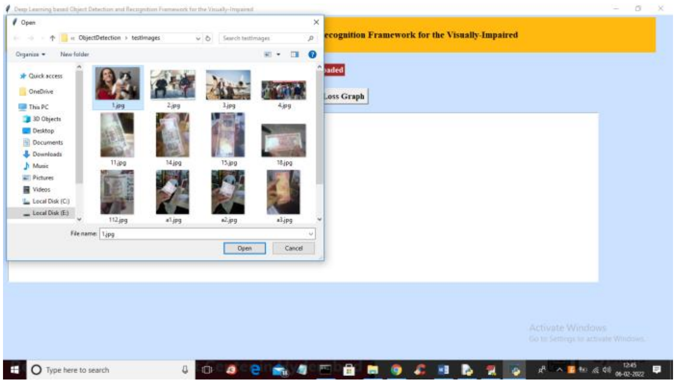
To run project double click on ‘run.bat’ file to get below screen



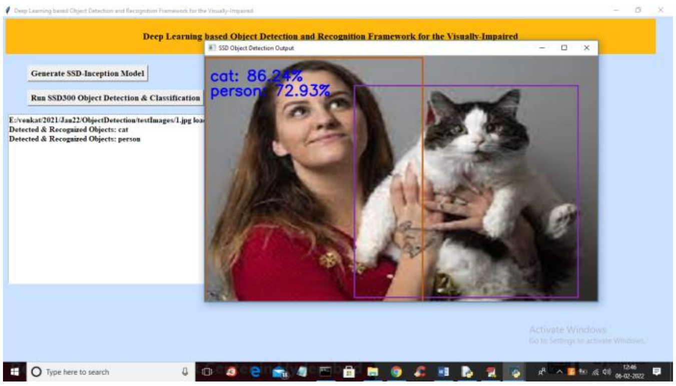
In above screen click on ‘Generate SSD-Inception Model’ button to load models and to get below screen



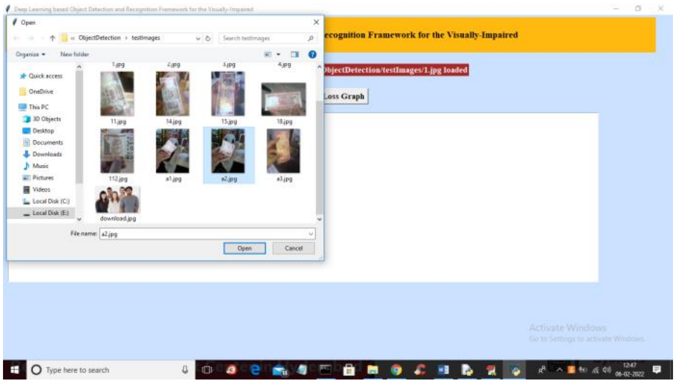
In above screen in red color text we can see models loaded and now click on ‘Run SSD300 Object Detection & Classification’ button to upload image and then classify object



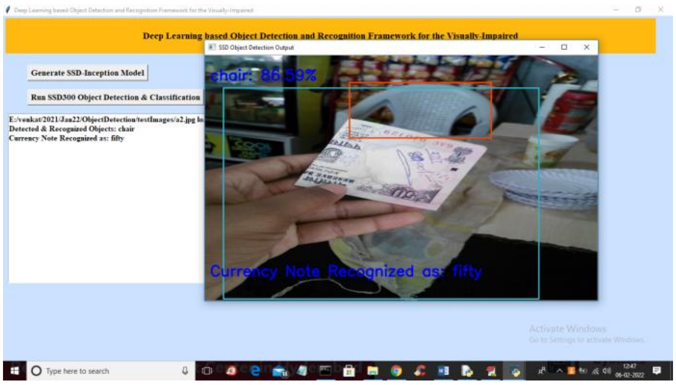
In above screen selecting and uploading 1.jpg file and then click on ‘Open’ button to get below output



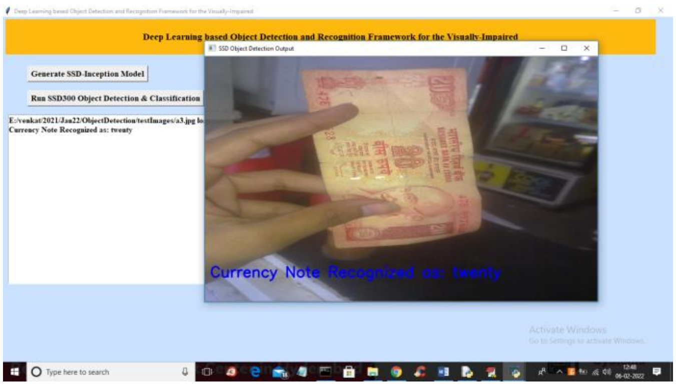
In above screen objects detected and classified as person and cat and now try other images

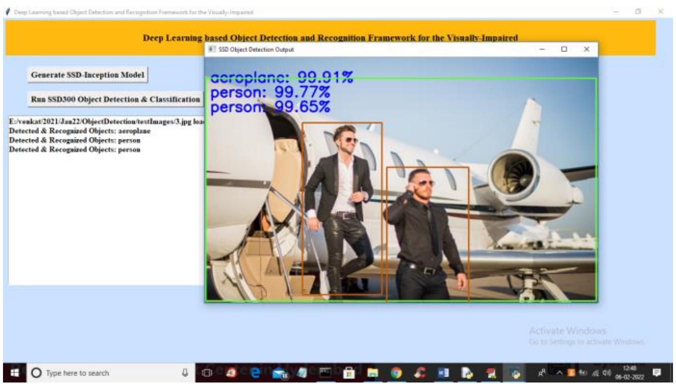


For above a2.jpg selection below is the output

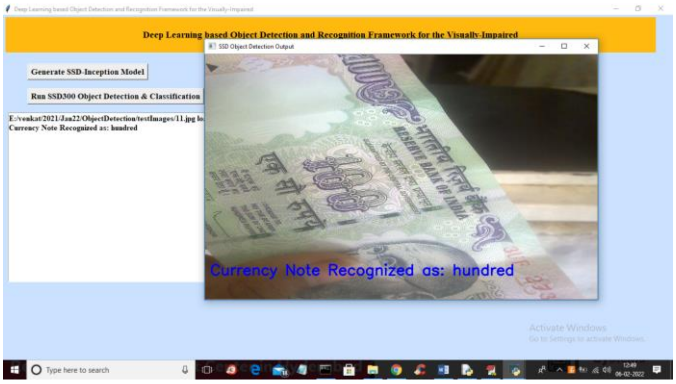


In above screen application displaying chair and currency note fifty detected and no test other images

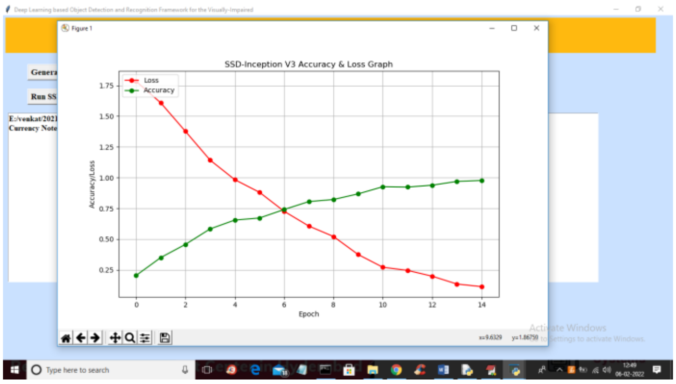




In above screen in text area as well as images you can see printing of detected objects and similarly you can upload and test other images



Now click on 'Inception Accuracy & Loss Graph' button to get below graph



To train Inception we took 15 epoch so in above graph x-axis represents epoch and y-axis represents accuracy and loss values and in above graph green line represents accuracy and red line represents loss and we can see with each increasing epoch accuracy get increase and loss get decrease and at final epoch accuracy reached to 100% and loss reached to 0. So this proves that inception is train efficiently to detect all classes

2.1. LITERATURE SURVEY:

SIFT, short for Scale-Invariant Feature Transform, is a computer vision algorithm used for detecting and describing local features in images. These features are essential for identifying and defining objects within an image. What makes SIFT particularly powerful is its ability to maintain its effectiveness across different scales, rotations, lighting conditions, and small perspective shifts. This scale invariance is achieved through a series of transformations applied to the local image characteristics.

Initially introduced by David G. Lowe in 1999, SIFT has undergone further improvements and refinements over the years. Five years after its initial presentation, Lowe revisited and summarized the algorithm's advancements.

There have been various modifications and enhancements proposed based on the original SIFT algorithm. For example, Zhuang Zhemin et al. introduced an improved SIFT target tracking algorithm that leverages compressed sensing principles. This modified approach includes updates to the classifier design within the compressive theory, enhancing real-time performance. Additionally, they modified the vector neighborhood of SIFT to reduce vector dimensions, thereby reducing computational complexity. Their algorithm has shown effectiveness in tracking objects under complex conditions while improving detection speed and real-time performance.

The Histogram of Oriented Gradients (HOG) is a feature descriptor widely used in computer vision and image analysis for object detection. It operates on the principle that local objects in an image can be characterized by the distribution of intensity gradients or edges within the image. The process involves dividing the image into smaller connected regions called cells, and for each pixel within a cell, a histogram of gradient directions is computed. These histograms are then concatenated to form the descriptor.

One of the primary advantages of the HOG descriptor is its invariance to geometric and photometric transformations, except for changes in object orientation. This makes it robust in various conditions.

Guangyuan Zhang et al. optimized the HOG features for effective human identification systems, particularly in pedestrian detection, by combining them with Support Vector Machine (SVM) classifiers. Simulation results demonstrated the effectiveness of this approach, leading to the widespread use of HOG features in pedestrian detection systems.

Furthermore, researchers have proposed modified methods based on HOG and SVM for improved performance. For example, CHENG Guang-tao et al. introduced a pedestrian detection method based on HOG and SVM, utilizing a pedestrian segmentation approach grounded on vertical edges and their symmetry properties. Experimental results showcased effective performance in complex conditions with higher detection speeds.

Similarly, YAO Xue-qin et al. addressed the low detection speed issue of traditional HOG-based pedestrian detection methods by proposing a method based on edge symmetry and HOG features. This approach rapidly detected candidate pedestrians using vertical edge symmetry and validated them using HOG features and SVM, achieving increased detection speeds while maintaining comparable detection rates.

Additionally, CHEN lifeng et al. proposed a pedestrian detection method based on HOG features and an Adaboost-BP model, which combines edge detection technology for rapid candidate region detection with multi-scale and multi-directional HOG features. The Adaboost algorithm trained multiple backpropagation neural networks to build a strong classifier for improved detection and recognition of test sample images, achieving higher detection rates and lower false-positive rates.

In summary, HOG-based methods, particularly when combined with SVM or other classification techniques, have proven effective in object detection tasks, particularly in pedestrian detection, with various modifications and enhancements continuously being proposed to improve their performance in complex conditions.

2.2. DATASET STUDY :

Applications of object detection have a broad range covering autonomous driving, detecting aerial objects, text detection, surveillance, rescue operations, robotics, facing detection, pedestrian detection, visual search engine, computation of object of interest, brand detection, and many more [1, 58]. The major challenges in the object detection includes; (i) The occupancy of Fig. 1 Classification, Localization, and Segmentation in Single and Multiple Objects image [13] Multimedia Tools and Applications (2023) 82:9243–9275 9245 an object in an image has an inherent variation such as objects in an image may occupy majority of the pixels i.e., 70% to 80%, or very few pixels i.e., 10% or even less, (ii) Processing of low-resolution visual contents, (iii) Handling varied sized multiple objects in an image, (iv) Availability of labelled data, and (v) Handling overlapping objects in visual content. Most of the object detectors based on machine learning and deep learning algorithms fail to address commonly faced challenges, are summarized as follows: & Multi-scale training: Most object detectors are trained for a specific resolution of input. These detectors generally underperform for inputs having different scales or resolutions. & Foreground-Background class imbalance: Imbalance or disproportion among the instances of different categories can majorly affects the model performance. & Detection of relatively smaller objects: All the object detection algorithms will tend to perform well on larger objects if the model is trained on larger objects. However, these models show poor performance on comparatively smaller sized objects. & Necessity of large datasets and computational power: Object detection algorithms in deep learning need larger datasets for computation, labor intensive approaches for annotations, and powerful computational resources for processing [45]. Due to the exponential increase of

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 02 | February - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

generated data from various sources, it has become a tedious and laborintensive task to annotate each and every object in the visual contents & Smaller sized datasets: Though deep learning models outperform traditional machine learning approaches by a great margin, they demonstrate poor performance while evaluating on the datasets with fewer instances. & Inaccurate localization during predictions: Bounding boxes are the approximations of the ground-truth. Generally, background pixels are also included during predictions, this affects the accuracy of the algorithm. Mostly, localization errors are either due to occupancy of background in the predictions and detecting similar objects [45].

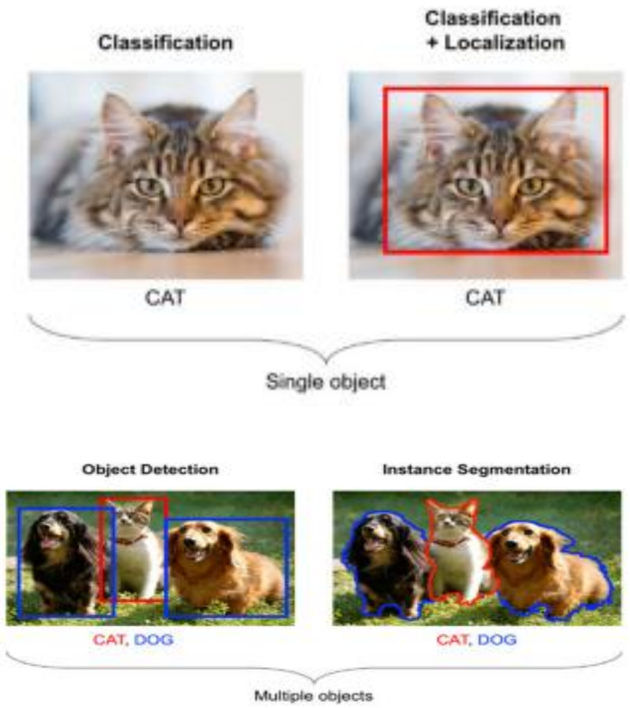


Fig. 1 Classification, Localization, and Segmentation in Single and Multiple Objects image.

2.3. STAGES OF DATA PROCESSING:

In supervised machine learning, there are two types of problems; (i) Regression and (ii) Classification. However, image classification is no different from the traditional classification problem. The next task after classifying the object in an image is to localize it, if required. A rectangular box, commonly known as a bounding box is determined around an object with the help of the deep neural networks. This object detection problem generally performs the features extraction followed by the classification and/or localization, known as two-stage object detectors if implemented in two stages. First stage generates Regions of Interest (RoI) using Region Proposal Network (RPN), however, the second stage is responsible for predicting the objects and bounding boxes for the proposed regions. First stage mainly responsible for selecting plausible region proposals by applying various techniques such as negative proposal sampling. The popular models in this category include Region based convolutional neural networks (RCNN), Fast RCNN, and Faster RCNN. Single stage object detectors enjoying the simpler architecture, specially designed for the object detection in single stage by considering all the region proposals. These detectors output the bounding boxes and class specific probabilities for the underlying objects by considering all the spatial sizes of an image in one shot. Though, two stage object detectors perform better in comparison with 9246 Multimedia Tools and Applications (2023) 82:9243–9275 single stage object detectors as it works on highly probable regions only for the object detection. However, with the advent of You Only Look Once (YOLO) and its successors, attempts are being heavily appreciated for solving this task in one shot/stage wherein localization problem is formulated as a regression problem with the help of deep neural networks. YOLO is not the first algorithm that uses Single Shot Detector (SSD) for object detection. There are numerous other algorithms that have been introduced in recent past such as Single Shot Detector (SSD) [43], Deconvolution Single Shot Detector (DSSD) [16], RetinaNet [41], M2Det [86], RefineDet++ [85], are based on single stage object detection. Two stage detectors are complex and powerful and therefore they generally outperform single stage detectors. YOLO can be seen as giving a tough fight to not only two staged detectors but previous single staged detectors also in terms of accuracy and inference time. It is considered as one of the most common choices in production only because of its simple architectural design, low complexity, and easy implementation. Figure 2 shows the generic schematic architecture of single stage object detectors wherein it generates all the bounding boxes along with the class probabilities by considering all the spatial regions in one shot.

2.4. METHODOLOGY:

The project can be broken down into several key methodologies:

1. Data Collection and Preprocessing: Gather a diverse dataset of images or videos containing various objects relevant to the application domain. Preprocess the data to ensure uniformity, clarity, and compatibility with the deep learning models.

2. Model Selection and Training: Choose appropriate deep learning architectures, such as Faster R-CNN, YOLO, or SSD, known for their effectiveness in object detection tasks. Train the selected model on the prepared

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 02 | February - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

dataset, adjusting hyperparameters and optimizing performance.

3. Evaluation and Fine-Tuning: Assess the trained model's performance using metrics like precision, recall, and mean average precision (mAP). Fine-tune the model if necessary to improve its accuracy and robustness, based on insights gained from evaluation.

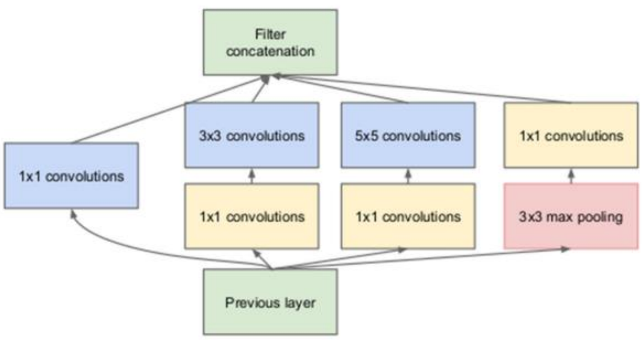
4. Inference and Object Detection: Deploy the trained model to perform object detection on real-world data, such as images or video streams. Utilize techniques like non-maximum suppression to refine the detected objects and eliminate redundant detections.

5. Performance Analysis and Optimization: Analyze the system's performance in terms of detection accuracy, speed, and resource efficiency. Optimize the model and inference pipeline for better performance and scalability, if required.

6. Integration and Deployment: Integrate the object detection system into the target application environment, ensuring seamless operation and compatibility. Deploy the system for real-world usage, monitoring its performance and making adjustments as needed.

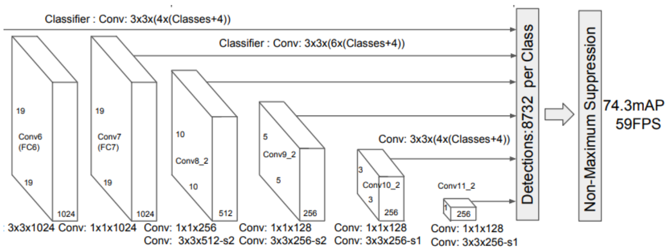
2.5. ARCHITECTURAL DESIGN OF YOLO:

The basic idea behind YOLO is to divide the input image into a grid of cells and, for each cell, predict the probability of the presence of an object and the bounding box coordinates of the object. The process of YOLO can be broken down into several steps:
1. Input image is passed through a CNN to extract features from the image.
2. The features are then passed through a series of fully connected layers, which predict class probabilities and bounding box coordinates.
3. The image is divided into a grid of cells, and each cell is responsible for predicting a set of bounding boxes and class probabilities.



2.6. ARCHITECTURAL DESIGN OF SSD:

Single Shot Detector (SSD) represents a pivotal object detection algorithm in the realms of computer vision and deep learning. Renowned for its real-time capabilities, SSD excels in detecting multiple objects within an image with remarkable precision. This algorithm hinges on a deep convolutional neural network (CNN), which ingests an image and yields a collection of bounding boxes alongside class predictions for every object contained within the image.



2.7. OVERVIEW OF TECHNOLOGIES:

LIBRARIES-

NUMPY:
NumPy, short for Numerical Python, offers a comprehensive range of multidimensional array objects and associated functions for array manipulation. Through the NumPy package in Python, users can efficiently handle arrays for mathematical and logical operations.

CV2:
The OpenCV module is commonly imported using the name cv2. It is a widely-used open-source library essential for various computer vision tasks such as processing video and CCTV footage, as well as analyzing images.

KERAS:
Keras stands out as a robust and user-friendly open-source Python package, renowned for its versatility in machine learning tasks. Developed atop foundational frameworks like TensorFlow, Theano, and Cognitive Toolkit (CNTK), Keras facilitates seamless neural network development. Theano, a Python program, enables swift numerical computations, while TensorFlow serves as the predominant symbolic math library for building neural networks..

TENSOR FLOW:

Leveraging the TensorFlow platform facilitates the implementation of top-tier practices for data automation, model tracking, performance monitoring, and model retraining. Employing production-grade tools to automate and monitor model training throughout the lifecycle of a product, service, or business process is indispensable for achieving success..

2.8. RESULTS

The project aimed to create an efficient and reliable object detection system using cutting-edge deep learning techniques, which is vital for various applications like autonomous driving and security surveillance. Object detection involves identifying and labeling different objects within images or videos, a task crucial for understanding visual content.
Utilizing state-of-the-art neural network architectures and modern data preprocessing tools, the project focused on developing algorithms capable of accurately recognizing and localizing objects in diverse real-world scenarios. Deep learning played a central role in this process, enabling the generation of meaningful results by leveraging large datasets and complex models.
The system's performance was rigorously evaluated using real-world data to ensure its robustness and effectiveness across various environments and conditions. By combining advanced algorithms with sophisticated evaluation techniques, the project aimed to push the boundaries of object detection capabilities and contribute to advancements in computer vision technology.

3. CONCLUSIONS

In conclusion, the literature reviewed offers valuable insights into the evolution of object detection techniques employing deep learning methodologies. Each method discussed presents distinct advantages alongside inherent limitations. Notable advantages encompass heightened accuracy, expedited processing speeds, resilience against occlusions and scale variations, and adeptness in real-time environments. Nevertheless, challenges persist, including the requisite for copious annotated training data, complexities in detecting diminutive or heavily obscured objects, susceptibility to fluctuations in lighting and backgrounds, and the perpetual balance between accuracy and processing efficiency.

Collectively, the body of research underscores significant strides in object detection through deep learning, continually propelling advancements in computer vision applications. The synthesis of findings furnishes a holistic comprehension of cutting-edge techniques, impediments, and prospective avenues in object detection, thereby catalyzing further innovation in the domain. By leveraging these insights, both researchers and practitioners can pioneer enhanced algorithms and systems across diverse domains such as autonomous driving, surveillance, healthcare, and beyond, thereby driving progress in numerous societal and industrial spheres

REFERENCES

[1]. Agarwal S, Terrail JO, Jurie F (2018) Recent advances in object detection in the age of deep convolutional neural networks. arXiv preprint arXiv:1809.03193.
https://doi.org/10.48550/arXiv.1809.03193

[2]. Albelwi S, Mahmood A (2017) A framework for designing the architectures of deep convolutional neural networks. Entropy 19(6):242

[3]. Bengio Y, Courville AC, Vincent P (2012) Unsupervised feature learning and deep learning: a review and new perspectives. CoRR, abs/1206.5538, 1(2665)

[4]. Bhattacharya S, Maddikunta PKR, Pham QV, Gadekallu TR, Chowdhary CL, Alazab M, Piran MJ (2021) Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey. Sustain Cities.So65:102589.
https://doi.org/10.1016/j.scs.2020.102589

[5]. Bochkovskiy A, Wang CY, Liao HY (2020) YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934