iJRASET

International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.70738

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Effective SSD Based Multiple Object Detection in Videos

Jehu Sheran[1], Sahana NS[2], ChanduShree M[3], Harshavardhan V[4], K.B Bini[5]
[1, 2, 3, 4]Student, [5]Assistant Professor, AMC Engineering College

Abstract: This paper aims to analyze the objects in any format of video or filmography. The web application is just simple like uploading any format of video, it processes the video and gives the finest progressive output with a bounding box. For processing the video for object detection, the SSD [algorithm] is used because the SSD [algorithm] has additional accuracy than the YOLO [algorithm]. And with SSD no other video format or other object detection processes are done, only for image classification SSD [algorithm] is used. In this project, introduced the framework of the video object detection process using the SSD [algorithm]. This technique works for surveillance video and any format of the video.

I. INTRODUCTION

Regarding the problem statement, the essential for video object-detection with good output is with bounding boxes and accuracy. The YOLO and RCNN [algorithms] are suitable with all object-detection applications, but the SSD algorithm is the best option for accuracy and speed when it comes to object-detection with multiple bounding boxes. This project looks to provide accurate video object-detection. For this subject statement, proposed a solution that involves constructing the SSD algorithm for video object-detection. The SSD [algorithm] technique is only used for image classification; video classification is difficult to attain. For this video process, separate each frame [image] of the video, apply the SSD [algorithm], and then recombine the separated frames. And now the result includes a bounding box which includes the object's class name with accuracy.

II. RELATED WORK: LITERATURE SURVEY

[1]. The Single Shot MultiBox Detector is a fast and efficient object-detection technique that uses a single deep neural-network to predict object categories and bounding boxes directly from images. Different from Faster RCNN, SSD do away with the need for region proposals, making it importantly faster while maintaining high accuracy and it uses multiple feature maps at different scales to detect objects of various sizes to make predictions. SSD achieves high detection accuracy, outermost forming YOLO in precision while Being much faster than Faster RCNN. It is acceptable for real-time applications, running at 59FPS with competitive mean average clarity on benchmark datasets like PASCAL VOC and COCO.

[2]. This paper examines why large convolutional-networks perform well in image classification and how they can be enhanced. The author introduces a visualization technique using De-convolutional networks to understand in the middle feature layers and classifies operations. They analyze network architecture and identify improvements that outperform foregoing models on ImageNet. A decrement study highlights the contributions of different layers and the model generalizes well to other datasets like Caltech-101 and Caltech-256. The study also proposes modifications to the convolutional layer to build feature retention and classification performance.

[3]. TensorFlow is a extensible machine learning framework designed for distributed computing across multiple devices, including CPUs, GPUs, and TPUs. It utilizes data flow graphs to represent computations and manage shared states expertly, enabling developers to experiment with new training algorithms and optimizations. It has been widely adopted in production and research, powering applications like image classification and language modeling. The framework is highly adaptable, allowing experimentation with various architectures and strategies. Additionally TensorFlow's open-source liberate has contributed significantly to AI advancements, and ongoing research aims to enhance automatic optimization, memory management and dynamic resources allocation for hereafter applications.

[4]. FFmpeg is a solid open-source multimedia framework for providing video and audio processing such as encoding, decoding, transcoding, and scaling. It supports a many sidedness of codecs and file formats, making it highly adaptable for video editing. The framework is command-line driven, allowing users to put in filters, alter resolution, and optimize video quality effectively. It also supports hardware acceleration, which boosts speed in real-time applications. FFmpeg is popular in media programs such as VLC, Google Chrome, and streaming services because of its speed and memory regulation. It secures high-quality video processing while keeping file size.

III. PROPOSED SYSTEM

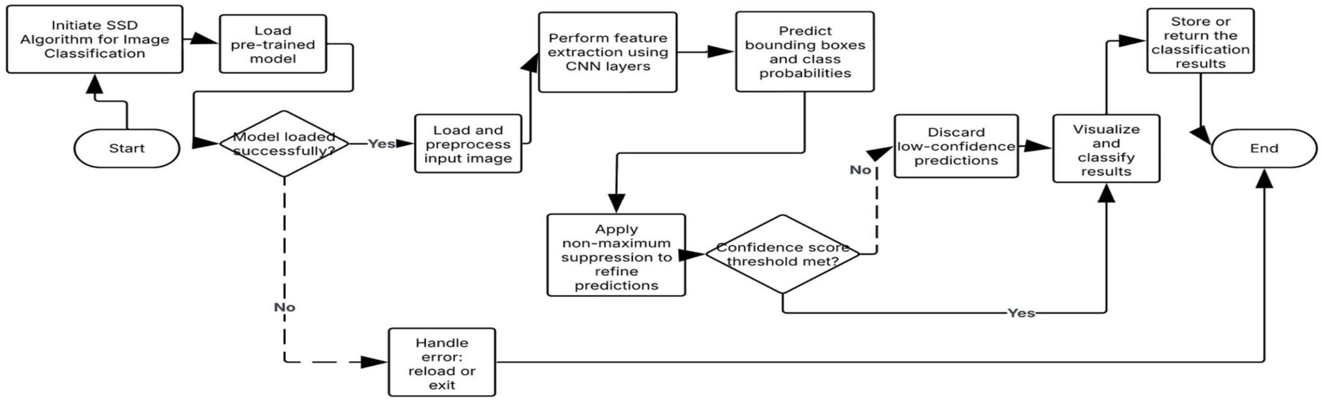
Strong object-detection solutions for video applications in all formats remain lacking in the domain. Existing advances towards primarily focus on image-based detection, leaving a gap in accurate and efficient object detection in dynamic video content. Additionally, the SSD [algorithm], while effective for images, has not been effectively utilized for video applications, limiting its potential in real-time scenarios. Giving an address to this issue is crucial for developing advanced, adaptable, and efficient video-based object detection systems that can be applied across several domains.

A. SSD [algorithm]

The SSD [algorithm] is a deep learning object-detection procedure that makes use of a convolutional neural network[CNN] to detect objects in images. This algorithm is efficient for real-time applications because of high accuracy and this algorithm aims to detect multiple object classes. The SSD algorithm is dealing with multiple bounding-boxes of the same instance of an image than other algorithms. This algorithm is faster than the compound inference speed of other algorithms. Image classification is a type of computer vision in which an image is used to anticipate an object within its bounding-box. For this procedure, the SSD [algorithm] has two components: an SSD head and a backbone; the SSD [algorithm] head is just one or two or further convolutional layers built on to the backbone. A backbone model normally is pre-trained image classification networks as an attribute extractor.

Step 1. Load Pretrained model and start: Load SSD model (Backbone: VGG16, ResNet, MobileNet)

Step 2. Preprocess the Input Image:


Initiate SSD Algorithm for Image Classification
Load pre-trained model
Start
Model loaded successfully?
Yes
No
Load and preprocess input image
Perform feature extraction using CNN layers
Predict bounding boxes and class probabilities
Apply non-maximum suppression to refine predictions
Confidence score threshold met?
No
Yes
Discard low-confidence predictions
Visualize and classify results
Store or return the classification results
End
Handle error: reload or exit

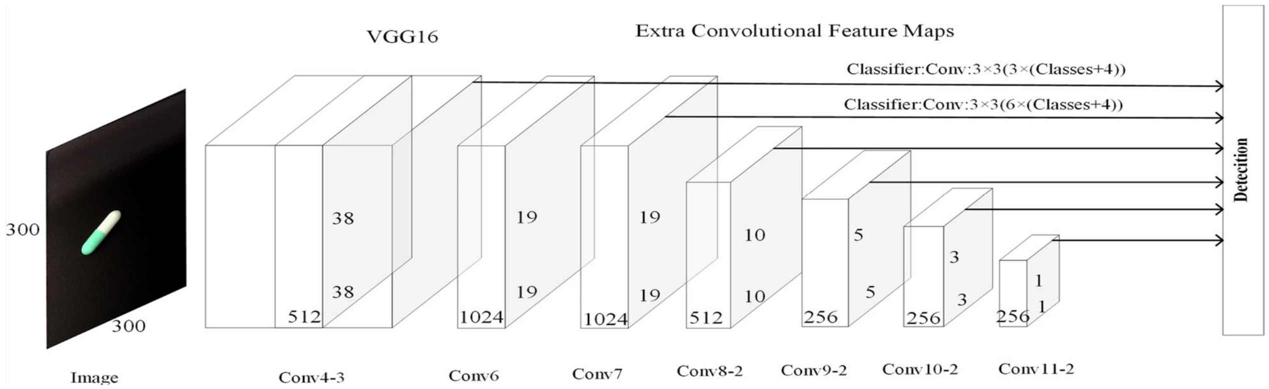
Input: Image

Resize image to (300*300) or (512*512)

Standardize pixel values

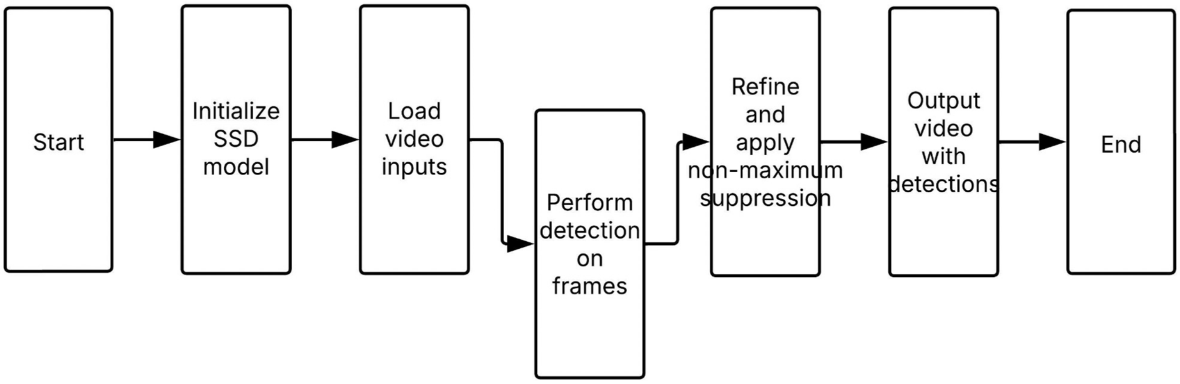
Convert image to tensor

Step 3. Extract Features using Base Network



```
Feature_Maps=Base_Network (Preprocessed_Image)
Step 4. Create Multi-Scale Feature Maps:
for each Feature_Map in Feature_Maps:
Create multiple Default Boxes (Anchor Boxes) with different aspect ratios
Predict Class Scores for each Default Box
Predict Bounding Box Offsets for each Default Box
Step 5. Decode Predictions:
for each Predicted_Box:
Apply Bounding Box Offsets to clarify box positions
Assign class label with highest confidence score
Step 6. Apply Non-maximum Suppression (NMS)
Remove overlapping boxes using IoU (Intersection over Union) threshold and hold on to boxes with highest confidence scores.
Step 7. Output Final Detected Objects
for each Final_Bounding_Box:
Draw Bounding Box on Image
Label Object with Class Name and Confidence Score
RETURN: figure with Detected Objects
```

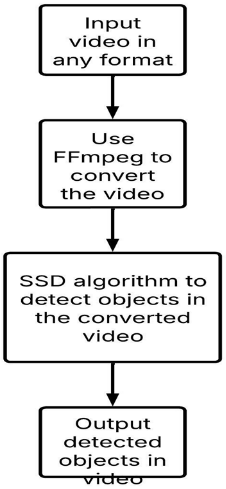
IV. METHODOLOGIES


Start
Initialize SSD model
Load video inputs
Perform detection on frames
Refine and apply non-maximum suppression
Output video with detections
End

The project’s expected solution is to upload the file to a web application, and then process the video in the backend for object-detection using CNN technique and SSD[algorithm]. However, a lack of structure was encountered when attempting to solve the video object-detection. The SSD algorithm is used for image[figure] classification. When it came to using video object detection, professional a number of structural difficulties. To fix this problem, arrange the video frame work; SSD operates with a single image and multibox detection. Video detection varies from image classification. But every video is projected at a rate of 24 frames[images] per second. It means per second, the video shows 24 complete still pictures[image] in sequences. This is the entire process of creating video and Then aimed to apply an SSD[algorithm] to each and every single frame of the per second[video]. Per second 24 images, when it comes to per minute nearly 1440 images in sequences. Applying an algorithm for each single image of the frame and proceeding the video for detection.

The TensorFlow engage in a crucial role in implementing the SSD algorithm for video object-detection. It provides pre-trained SSD models optimized for video processing and these models are available in TensorFlow object detection API and can be fine-tuned for custom applications. It processed the input video frame by frame using OpenCV and each frame is treated as an image for SSD grounded object-detection. The TensorFlow loads the SSD model and extracts features from each video frame using CNN-based feature extractors and the model generates multi-scale feature maps to detect objects of different size. TensorFlow bounds a convolutional layer to predict class labels and bounding box co-ordinates for objects in each frame and NMS is applied to remove unnecessary detections. It overlays detected bounding boxes and labels onto the video frames and the processed frames are recombined into a video visualization.

In addition, for video object-detection FFmpeg is used for converting video format files between different formats alike as HSL to mp4 or MOV to mp4. It is a free open source application. Every different video format in computer vision image applications are different there. To solve this FFmpeg is used to convert any format video to a suitable SSD algorithm to detect objects in videos.


Input video in any format
Use FFmpeg to convert the video
SSD algorithm to detect objects in the converted video
Output detected objects in video

V. RESULT EXPERIMENTAL AND PERFORMANCE ANALYSIS

A. DataSet

The aim of the result is to detect the object in the format of video using [ssd_resnet50_v1_fpn_640*640_coco17_tpu-8] model dataset. In this dataset model image classes are there to detect the objects. The"ssd_resnet50_v1_fpn_640*640_coco17_tpu-8" is a pre-trained object-detection model that blends the single shot multiBox detector [SSD] framework with a Resnet-50 backbone. Resnet-50 is a convolutional backbone neural network architecture and it is used as the feature extractor. Feature pyramid network is used to enhance the feature extraction process by creating multi-scale feature pyramids and this allows the model to detect objects at different scales. The size of 640*640 to take input of image pixels, which balances between computational efficiency and detection.

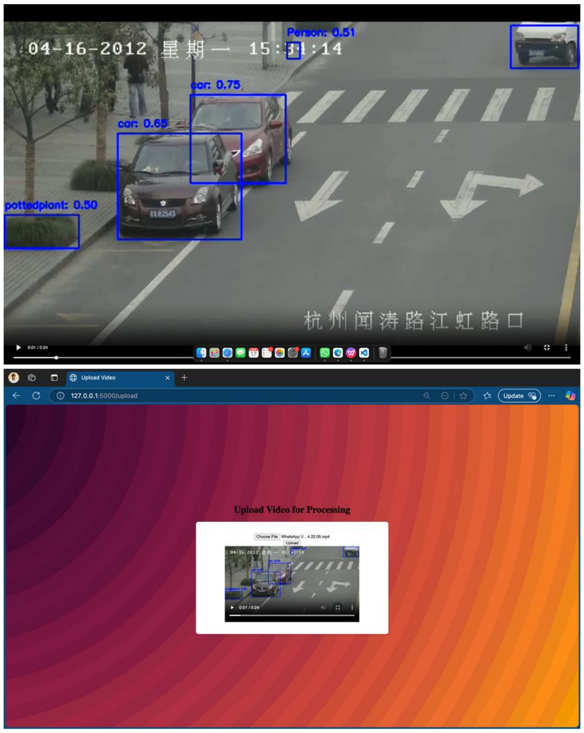
Note: You can use other dataset models also; below mentioned them, and you can use the MobileNet model also.

ssd_resnet101_v1_fpn_1024x1024_coco17_tpu-8.config
ssd_resnet101_v1_fpn_640x640_coco17_tpu-8.config
ssd_resnet152_v1_fpn_1024x1024_coco17_tpu-8.config
ssd_resnet152_v1_fpn_640x640_coco17_tpu-8.config
ssd_resnet50_v1_fpn_1024x1024_coco17_tpu-8.config
ssd_resnet50_v1_fpn_640x640_coco17_tpu-8.config
etc…..

The preprocessing of the results is the input images is resized to 640*640 pixels and normalized before being fed into the model. The frame from a video is converted to a tensor using tf.convert_to_tensor. For reasoning the model performs inference on the preprocessed input to produce detections, which include bounding boxes, classes and accuracy. In the post processing the detection results are proposed to extract bounding boxes, classes and accuracy. Visually in the screen, detected objects are visualized by drawing bounding boxes on the frames with a confidence score more than a not in doubt threshold.

B. ScreenShot

Different timeline of the video screenshot for reference. You can see the bounding box variation. In every single frame the SSD[algorithm] allows analysis with the dataset which you are given [any pre-trained dataset].



VI. CONCLUSION AND FUTURE WORK

This project successfully demonstrates the application of the SSD[algorithm] for video object detection, addressing the challenges of adapting an images based detection model to video content. By processing each frame of the video individually and applying the SSD[algorithm]. The system achieves accurate object-detection with bounding boxes, enhancing the utility of SSD in video applications. The integration of FFmpeg ensures compatibility across various video formats. This approach not only fills a gap in existing object detection methodologies but also sets a foundation for future advancements in video based object detection systems, offering a balance between speed and accuracy that is crucial for practical deployment in diverse domains such as surveillance and multimedia processing.

REFERENCES

[1] Wei Liu; Dragomir Angular; Dumitru Esha; Christian Szegedy; Scott Read; Cheng-Yang Fu: 2016, “SSD: Single shot multiBox Detector” arXiv: 1512.0232

[2] Zeiler; Matthew D; Rob Fergus: 2014, “Visualizing and understanding convolutional networks” in European conference on computer vision, pp.818-833

[3] Xlaohua Lei; Xiahhua jiang; Cashong Wang: 2013, “ Design and Implementation of a Real-time video stream analysis System Based On FFMPEG”, 10.1109/WISE.2013.38

[4] H Sumesh Singha; Dr. Bhuvana : 2021, “A Study On FFmpeg MultiMedia Framework”, ISSN-2456-6470.

[5] Jonathan Heri: 2018, “SingleShot MultiBox Detector for real-time processing”

[6] Dang HaThe Hien: A Guide to receptive field arithmetic for CNN

[7] Suramya Tomas: 2006, “Converting Video formats with FFMPEG”

[8] FFmpeg :[online] https://www.ffmpeg.org/

[9] Howard Jeremy: 2019, Lesson 09: “Deep learning part-2 multi object detection”

[10] J. Ba, V. Mnih, and K. Kavukcuoglu: 2014,”Multiple Object Recognition with visual attention”, arXiv:1412.7755, 2014.

[11] Hafiz Nur: 2022, “Research on TensorFlow with a system for large-scale machine learning”, Researchgate.