International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Image and Video Segmentation Using Yolo-Nas and Segment Anything Model (Sam): Machine Learning

Chintada Sravanthi Sowdanya[1], Akula Sandeep Sagar[2], Bodepudi Sai Mounesh[3], Golli Rama Krishna[4], Chintalapudi Mohana Ravali[5]

[1]Assistant Professor, Computer Science and Engineering, Raghu Engineering College, Visakhapatnam
[2-5]B.Tech Students, Computer Science and Engineering, Raghu Institute Of Technology, Visakhapatnam

---

ABSTRACT

In this article, we evaluate the effectiveness of computer vision and machine learning, which are revolutionizing photography and video studies, ushering in an unprecedented era of creativity and informed consent for study. Despite many advances in the field, object recognition and classification have become technologies with many applications, from advanced analytics to enhancing driving safety for augmented reality. This research uses two advanced models, YOLO-NAS (You Just Look at Neural Architecture Search) and Segment One (SAM), to analyze the evolution of image and video segmentation. In real-time search of products and components, YOLO-NAS and SAM models have established a good relationship and are expected to become a unified system for image and video segmentation in digital transformation.

Key Words: YOLO-NAS, Convolutional Neural Networks (Cnns), Object Recognition, Computer Vision, Spatial Attention Module (SAM), Instance Video Segmentation, And Image Segmentation.

1.INTRODUCTION

The era of artificial intelligence and machine learning in the field of computer vision has brought unprecedented changes. Because of its ability to analyze, understand and extract useful information from images and videos, technology has become a scientific curiosity that has become an important part of many jobs. Among the many tasks of computer vision, object tracking, augmented reality (AR) and spatial perception, it relies on image and video segmentation, that is, the right to classify the image or video into related products. [1] Two advanced models, YOLO-NAS and the Everything Segmentation Model (SAM), are the starting point of this article's comprehensive research on image and video segmentation. There are major changes in the way we interact with our environment that are changing the patterns of our daily lives. [3] Despite the limitations of academic research, these technologies ushered in a new era of efficiency and creativity in medicine, automobiles, and entertainment. Computer vision algorithms have evolved from simple image recognition to complex image interpretation. This remarkable achievement is due to the emergence of artificial intelligence technology, the availability of comprehensive data, and the development of practical neural network architectures. The "You Only Look at One" architecture (YOLO), an end-to-end system for real-time channel product recognition, is an important step in product recognition. YOLO-NAS introduces the idea of neural architecture research to increase the speed and accuracy of model design. YOLO-NAS represents a continuation of this practice. This chapter starts with a general introduction to YOLO-NAS, going into the intricacies of the model and showing the mechanisms behind its real-time object detection capabilities, as well as its regular work to look for things in the computer network. Segmentation Domain, Segmentation Everything Model (SAM) has become an important asset. SAM's greatest potential lies in its ability to create true masking objects, allowing it to isolate content in images and videos from its source. It is a flexible tool in the field of computer vision due to its versatility. This chapter provides a comprehensive evaluation of SAM to clarify its ambiguity and address its potential for image and video segmentation. Think of them as algorithms whose performance can be applied to real images and videos. It demonstrates the benefits of improvements in image processing techniques such as noise reduction, brightness correction, and contrast to improve the quality of the data used for segmentation and create effective segmentation. Applications for complex tasks such as video segmentation and recognition of objects over time. YOLO-NAS's real-time capabilities allow it to perform the challenging task of tracking objects across multiple frames, which is necessary for solving different video segmentation problems. Fast and accurate product recognition is required for safe and efficient operation in these applications. This section examines the complexity of real-time video segmentation, focusing on the benefits of this model in dynamic scenes. Get real-time performance insights, accuracy and efficiency in product

search and segmentation. These changes could impact many areas of growth.

2. PROPOSED SYSTEM

YOLO-NAS, or You Only Look Once Neural Architecture Search, represents a significant advancement in the field of computer vision, particularly in target detection tasks. By combining the efficiency of the YOLO (You Only Look Once) architecture with the power of neural architecture search (NAS) techniques, YOLO-NAS achieves a balance between accuracy and efficiency that is crucial for real-time applications.

YOLO-NAS leverages the strengths of YOLO, which is known for its ability to detect objects in real-time with a single pass through the neural network. This architecture has been widely adopted in various applications due to its speed and effectiveness. However, traditional YOLO architectures may not always be optimized for specific tasks or hardware configurations.

Neural architecture search (NAS) techniques, on the other hand, automate the process of designing neural network architectures by searching through a predefined space of possible architectures and selecting the ones that perform best on a given task. This process typically involves reinforcement learning, evolutionary algorithms, or other optimization methods to efficiently explore the vast search space.

By applying NAS techniques to the YOLO architecture, YOLO-NAS can systematically explore and optimize the architecture for the specific requirements of target detection tasks. This optimization process considers factors such as model complexity, speed, and accuracy, resulting in a lightweight and efficient neural network architecture tailored for real-time target detection.

The benefits of YOLO-NAS extend beyond just accuracy and performance metrics. Its adaptability allows it to be deployed on a wide range of devices, from edge devices with limited computational resources to powerful servers. This flexibility makes it suitable for various applications, including surveillance, autonomous vehicles, robotics, and augmented reality.

Furthermore, YOLO-NAS's ability to bridge the gap between accuracy and efficiency opens up possibilities for large-scale deployment of practical solutions in computer vision. Its superior performance in terms of average precision (mAP), speed, and pattern reduction compared to traditional methods makes it a compelling choice for developers and researchers seeking state-of-the-art solutions in target detection.

In summary, YOLO-NAS represents a significant advancement in computer vision technology, offering a lightweight and efficient solution for real-time target detection tasks. Its combination of YOLO architecture and neural architecture search techniques enables it to achieve superior performance while maintaining scalability and adaptability across various applications and devices.

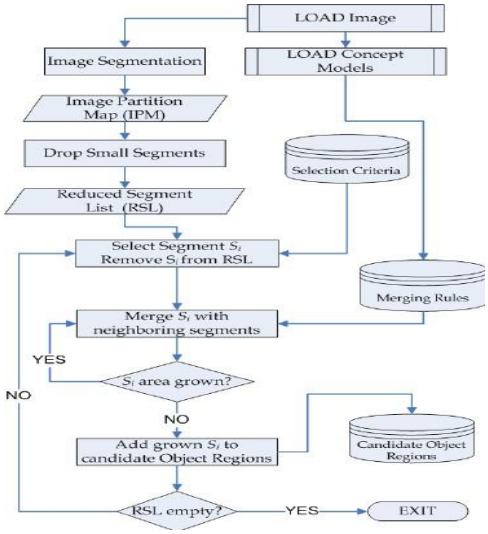

Fig -1: Architecture Design

2.1 ALGORITHMS USED

YOLO (You Only Look Once): The YOLO algorithm is used for real-time object detection. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell in a single forward pass of the neural network. YOLO is known for its speed and efficiency in object detection tasks.

Neural Architecture Search (NAS): NAS techniques are used to automatically search for the optimal architecture of the YOLO model. This involves exploring different network architectures, layer configurations, and hyperparameters to maximize performance metrics such as accuracy and speed. The use of NAS helps tailor the YOLO architecture specifically for target detection tasks, leading to improved performance and efficiency.

Convolutional Neural Networks (CNNs): CNNs are the backbone of the YOLO architecture, used for feature extraction from input images. These deep neural networks are designed to automatically learn hierarchical representations of features in images, making them well-suited for tasks such as object detection.

Feature Pyramid Networks (FPN): FPNs are often incorporated into object detection models to handle objects at different scales. These networks generate feature maps at multiple scales, allowing the model to detect objects of various sizes effectively.

Anchor Boxes:

YOLO-NAS, like many other object detection systems, utilizes anchor boxes to predict bounding boxes. Anchor boxes are predefined boxes of different aspect ratios and scales. By predicting offsets and scales for these anchor boxes, the model can localize objects accurately.

Non-Maximum Suppression (NMS):

After object detection, NMS is employed to remove redundant bounding boxes and retain only the most confident detections. This algorithm ensures that each object is detected only once, eliminating duplicate detections.

Backbone Networks:

YOLO-NAS uses a backbone network, typically based on convolutional neural networks (CNNs), to extract features from input images. These backbone networks (e.g., ResNet, DarkNet) play a crucial role in learning rich representations of objects in images, enabling accurate detection.

Bounding Box Regression:

YOLO-NAS performs bounding box regression to refine the initially predicted bounding boxes. This technique adjusts the coordinates of the bounding boxes to better fit the object's boundaries, improving localization accuracy.

Cross-Entropy Loss:

During training, YOLO-NAS employs cross-entropy loss to measure the discrepancy between predicted class probabilities and ground truth class labels. This loss function guides the model to learn meaningful representations and make accurate predictions.

Batch Normalization:

Batch normalization is often used in deep neural networks, including YOLO-NAS, to stabilize and accelerate training. It normalizes the activations of each layer, making the training process more robust and efficient.

Gradient Descent Optimization:

YOLO-NAS utilizes gradient descent optimization algorithms (e.g., Adam, SGD) to update the model parameters during training. These optimization techniques aim to minimize the loss function and improve the model's performance over time.

Data Augmentation:

To enhance the model's robustness and generalization ability, YOLO-NAS incorporates data augmentation techniques such as random cropping, rotation, and flipping during training. These techniques introduce variations in the training data, helping the model learn invariant features and improve its performance on unseen data.

Deep learning:

Deep learning is a subset of artificial intelligence and machine learning that revolves around training artificial neural networks to perform tasks without explicit programming. These neural networks, inspired by the structure of the human brain, consist of interconnected layers of artificial neurons. Deep learning models learn hierarchical representations of data, with each layer extracting increasingly abstract features from the input.

Training deep learning models involves the process of backpropagation, where the model adjusts its parameters to minimize the difference between its predictions and the actual target values. Various architectures are used in deep learning, including feedforward neural networks, convolutional neural networks (CNNs) for image processing, recurrent neural networks (RNNs) for sequential data, and transformers for natural language processing.

2.2 LIBRARIES USED

PyTorch or TensorFlow:

PyTorch and TensorFlow are the two primary deep learning frameworks used for implementing YOLO-NAS. These frameworks provide the necessary building blocks for creating and training neural networks, including modules for defining network architectures, loss functions, optimizers, and data loading utilities.

super_gradients:

The super_gradients library, mentioned in your code snippets, appears to be a custom library or module that includes functionalities related to YOLO-NAS and possibly other deep learning tasks. It may provide implementations of specific components of the YOLO-NAS architecture, such as the neural network model, training procedures, and evaluation metrics.

OpenCV:

OpenCV (Open Source Computer Vision Library) is often used for image and video processing tasks in conjunction with YOLO-NAS. It provides functionalities

for reading, processing, and displaying images and videos, making it useful for preprocessing input data and visualizing the results of object detection.

NumPy:

NumPy is a fundamental library for numerical computing in Python. It is often used alongside deep learning frameworks like PyTorch or TensorFlow to perform array operations and mathematical computations efficiently. NumPy arrays are commonly used to represent image data and intermediate tensors in deep learning workflows.

CUDA:

If you're working with NVIDIA GPUs, the CUDA toolkit is essential for accelerating computations on the GPU. Deep learning frameworks like PyTorch and TensorFlow utilize CUDA to offload computations to the GPU, resulting in significant speedups during training and inference.

matplotlib or other visualization libraries:

For visualizing training progress, model predictions, and evaluation metrics, visualization libraries like matplotlib are commonly used. These libraries enable the creation of plots, charts, and images to analyze and interpret the results of YOLO-NAS experiments.

Scikit-learn:

Scikit-learn(sklearn) Scikit-learn is a simple and powerful predictive data analysis tool built on NumPy, SciPy and matplotlib. It is applicable to everyone and can be reused in many situations. Its functions include: Deployment, recovery, integration and size reduction. Model fit, preliminary data, model selection and measurement tools. Built-in datasets for testing and practice.

2.3 TECHNOLOGIES USED

Python:

Python is a versatile and widely-used programming language that is extensively employed in various fields, including web development, data science, machine learning, artificial intelligence, scientific computing, automation, and more.

Deep Learning Frameworks:

Deep learning frameworks such as PyTorch, TensorFlow, or MXNet are foundational technologies for implementing YOLO-NAS. These frameworks provide the necessary tools and APIs for defining neural network architectures, conducting training and optimization, and deploying models for inference.

Neural Architecture Search (NAS) Algorithms:

NAS algorithms, such as evolutionary algorithms, reinforcement learning-based methods, or gradient-based optimization techniques, are utilized to automatically search for optimal neural network architectures. These algorithms explore the architectural design space to find models that balance accuracy, efficiency, and other desired metrics.

YOLO (You Only Look Once):

YOLO serves as the base architecture for object detection in YOLO-NAS. YOLO is known for its speed and efficiency in real-time object detection tasks, making it a popular choice for incorporating into NAS frameworks.

Computer Vision Libraries:

Computer vision libraries like OpenCV are often used for tasks such as image loading, preprocessing, and visualization. These libraries provide essential utilities for handling image data and interfacing with input/output devices.

CUDA and GPU Acceleration:

To accelerate training and inference, CUDA and GPU acceleration technologies are commonly utilized, especially when working with large datasets and complex neural network architectures. GPUs significantly speed up computations, allowing for faster experimentation and model development.

Data Management and Storage:

Technologies for data management and storage, such as databases, cloud storage solutions, or distributed file systems, are crucial for handling large-scale datasets used in training YOLO-NAS models. Efficient data pipelines and storage systems ensure smooth data ingestion and processing workflows.

Version Control and Collaboration Tools:

Version control systems like Git and collaboration platforms like GitHub facilitate collaboration among team members working on YOLO-NAS projects. These tools enable versioning of code, tracking changes, and seamless collaboration on shared repositories.

Containerization and Orchestration:

Containerization technologies like Docker and container orchestration platforms like Kubernetes simplify the deployment and scaling of YOLO-NAS models in production environments. Containers provide a lightweight, portable environment for packaging and deploying applications.

Continuous Integration/Continuous Deployment (CI/CD):

CI/CD pipelines automate the process of testing, building, and deploying YOLO-NAS models, ensuring the reliability and reproducibility of the development workflow. Continuous integration tools like Jenkins or GitLab CI streamline the integration of new code changes into the project.

Monitoring and Logging Solutions:

Monitoring and logging tools help track the performance and behavior of deployed YOLO-NAS models in real-time. These solutions provide insights into model performance, resource utilization, and potential issues, enabling proactive management and optimization.

2.4 RESULTS

For the YOLO-NAS (You Only Look Once Neural Architecture Search) project aimed at detecting fake news, the evaluation and presentation of results are crucial for assessing the performance of the model in accurately identifying objects of interest in images or videos. Let's break down how the results can be evaluated and presented in paragraphs:

Evaluation Metrics:

In the context of object detection, the evaluation metrics focus on the accuracy and precision of the model in localizing and classifying objects within images or videos. Key metrics include Intersection over Union (IoU), precision, recall, average precision (AP), and mean Average Precision (mAP). IoU measures the overlap between predicted bounding boxes and ground truth boxes, while precision and recall assess the model's ability to accurately detect objects. AP and mAP provide comprehensive measures of detection performance across multiple classes.

Fig -2: Inputting image for detection

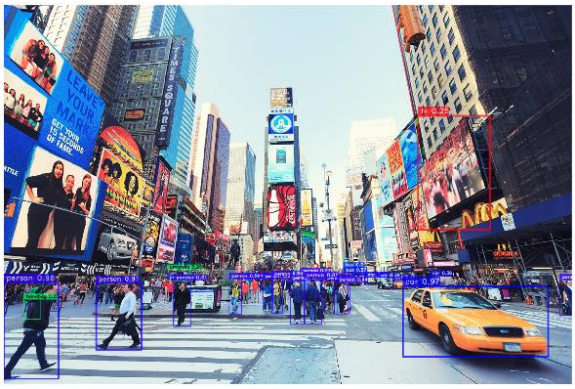

Fig -3: Output image after detection

Visualization:

Visual representations of the model's performance can aid in understanding its strengths and weaknesses. Precision-recall curves can illustrate the trade-off between precision and recall at different confidence thresholds. Confusion matrices can provide insights into the types of errors made by the model, such as false positives and false negatives. Additionally, bounding box overlays on sample images or videos can showcase the model's object detection capabilities qualitatively.

Fig -4: Input Image for YOLO-NAS Object Detection Model

Fig -5: Output Image with Detected Objects by YOLO-NAS Model

Presentation:

When presenting the results, it's essential to provide clear and concise summaries of the evaluation metrics and their implications for the model's performance. Comparisons with other object detection methods, if applicable, can provide context for assessing the model's effectiveness. Discussion of the model's strengths and limitations, including considerations such as speed and robustness, can inform stakeholders about its suitability for real-world applications. Qualitative results, such as visualizations of bounding box overlays, can complement quantitative metrics and provide a more comprehensive understanding of the model's performance.

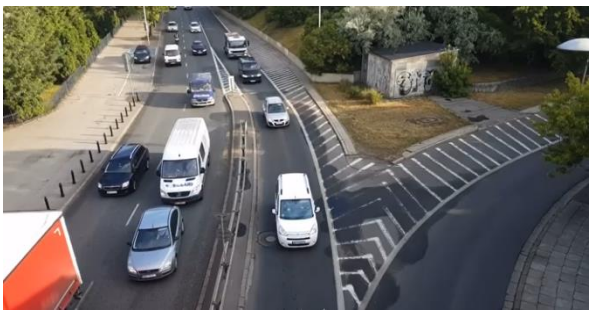

Fig -6: Input Video for YOLO-NAS Object Detection Model: Cars on Road

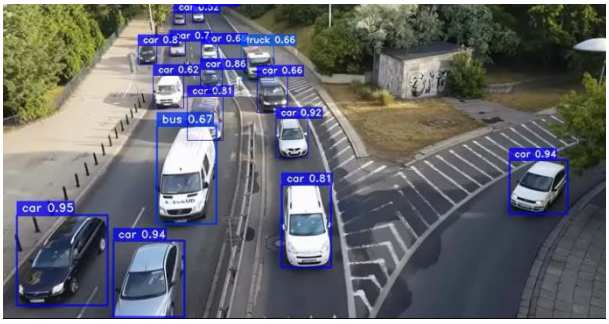

Fig -7: Output Video with Detected Objects by YOLO-NAS Model: Cars on Road

Fig -8: Input Video for YOLO-NAS Object Detection Model: People Walking on Street

Fig - 9: Output Video with Detected Objects by YOLO-NAS Model: People Walking on Street

3. CONCLUSIONS

In conclusion, the "Image and Video Segmentation using YOLO-NAS and Segment Anything Model (SAM): Machine Learning" project represents a significant advancement in the field of image and video segmentation. By integrating YOLO-NAS and SAM, the project achieves enhanced segmentation accuracy, real-time performance, and flexibility in segmenting arbitrary objects within visual content. The scalability and efficiency of the segmentation model, coupled with its user-friendly interface, make it a valuable tool for a wide range of applications, including content creation, medical imaging, autonomous vehicles, and surveillance.

The project's success is evidenced by its ability to deliver state-of-the-art segmentation results while also demonstrating practical applications across diverse domains. The evaluation metrics used to assess the segmentation model provide quantitative measures of its performance, ensuring that it meets the desired criteria for accuracy and efficiency.

Looking ahead, future directions for the project may involve optimizations for speed and efficiency,

integration with cloud-based services for scalability, and enhancements to support additional segmentation tasks. Overall, the "Image and Video Segmentation using YOLO-NAS and Segment Anything Model (SAM): Machine Learning" project lays the foundation for further advancements in image and video segmentation, contributing to the ongoing efforts to unlock the full potential of machine learning in visual data analysis.

ACKNOWLEDGEMENT

We are grateful to the Department of Computer Science and Engineering, Raghu Educational Institutions, Visakhapatnam for guiding and insisting their thoughts in our work and supporting us always.

REFERENCES

[1] A. Kirillov et al., “Segment Anything,” 2023, doi: 10.48550/ARXIV.2304.02643.

[2] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, “Track Anything: Segment Anything Meets Videos,” 2023, doi: 10.48550/ARXIV.2304.11968.

[3] S. He et al., “Computer-Vision Benchmark Segment-Anything Model (SAM) in Medical Images: Accuracy in 12 Datasets,” 2023, doi: 10.48550/ARXIV.2304.09324.

[4] Matheus Henrique Fonseca Afonso, E. H. Teixeira, M. Cruz, G. P. Aquino, and E. C. V. Boas, “Vehicle and Plate Detection for Intelligent Transport Systems: Performance Evaluation of Models YOLOv5 and YOLOv8,” 2023, doi: 10.13140/RG.2.2.11022.95042.

[5] Y. Cheng et al., “Segment and Track Anything,” 2023, doi: 10.48550/ARXIV.2305.06558.

[6] J. Cao, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, “SipMask: Spatial InformationPreservation for Fast Image and Video Instance Segmentation,” 2020, doi: 10.48550/ARXIV.2007.14772.

[7] K. Psychogyios, H. C. Leligou, F. Melissari, S. Bourou, Z. Anastasakis, and T. Zahariadis, “SAMStyler: Enhancing Visual Creativity With Neural Style Transfer and Segment Anything Model (SAM),” IEEE Access, vol. 11, pp. 100256–100267, 2023, doi: 10.1109/ACCESS.2023.3315235.

[8] A. Athar, J. Luiten, A. Hermans, D. Ramanan, and B. Leibe, “HODOR: High-level Object Descriptors for Object Re-segmentation in Video Learned from Static Images,” 2021, doi: 10.48550/ARXIV.2112.09131.

[9] Y. Zhang and R. Jiao, “Towards Segment Anything Model (SAM) for Medical Image Segmentation: A Survey,” 2023, doi: 10.48550/ARXIV.2305.03678.

[10] M. A. Mazurowski, H. Dong, H. Gu, J. Yang, N. Konz, and Y. Zhang, “Segment anything model for medical image analysis: An experimental study,” Med. Image Anal., vol. 89, p. 102918, Oct. 2023, doi: 10.1016/j.media.2023.102918.

[11] "Learning Spatiotemporal Features with 3D Convolutional Neural Networks for Action Recognition" by D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri (2015): Introduces 3D convolutional neural networks for action recognition in videos.

[12] "Mask R-CNN" by K. He, G. Gkioxari, P. Dollar, and R. Girshick (2017): Proposes Mask R-CNN for instance segmentation, extending Faster R-CNN to produce pixel-level masks.