**YOLO NAS Model Architecture**

This document provides an overview of the YOLO NAS (You Only Look Once Neural Architecture Search) model architecture used in our project for object detection and counting in retail environments.

The YOLO-NAS (You Only Look Once - Neural Architecture Search) model architecture represents a significant advancement in object detection technology. This document outlines its components, innovations, and the implementation steps required to utilize the YOLO-NAS model for efficient and accurate object detection tasks.

**Introduction**

Object detection has revolutionized how machines perceive and interpret the world. It enables machines to recognize and locate objects within images or videos, playing a crucial role in applications like autonomous vehicles, facial recognition systems, and many more. Over the years, the development of powerful neural network architectures has significantly enhanced the capabilities of object detection, making it indispensable in various industries.

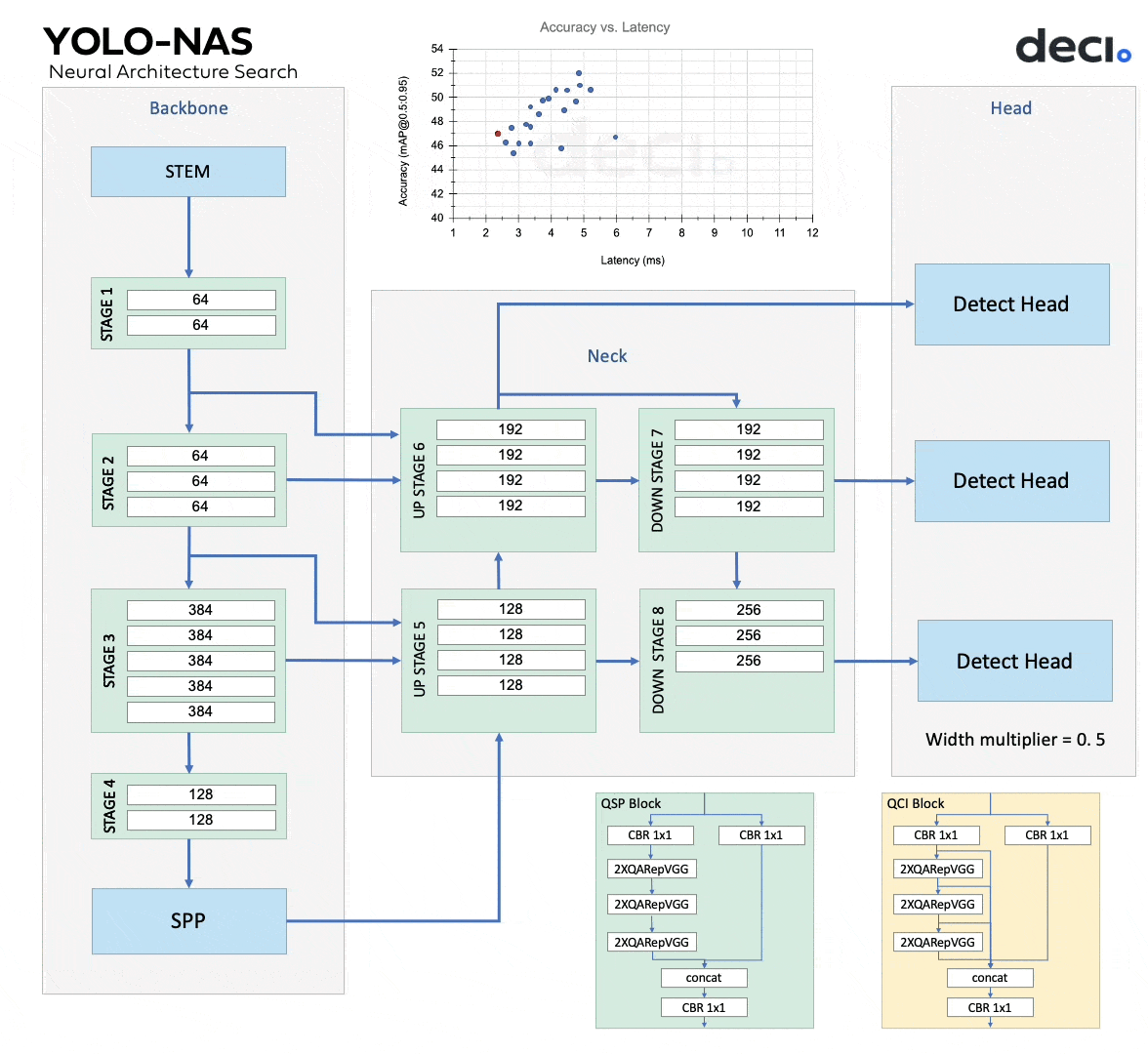
**Evolution of YOLO Architecture**

Since its introduction in 2016, YOLO (You Only Look Once) has been one of the most influential architectures in the field of object detection. YOLO redefined object detection by treating it as a single regression problem, predicting bounding boxes and class probabilities simultaneously. Despite its initial limitations in detecting small objects and localization accuracy, subsequent versions like YOLOv6, YOLOv7, and YOLOv8 have progressively improved in terms of accuracy, real-time performance, and compatibility with edge devices and cloud environments.

**YOLO-NAS: The Next Frontier**

At Deci, we have developed a new YOLO-based architecture, YOLO-NAS, to further push the boundaries of accuracy and efficiency in object detection. Leveraging Deci’s proprietary Neural Architecture Search (NAS) technology, AutoNAC, YOLO-NAS addresses existing limitations and incorporates recent advancements in deep learning.

The YOLO-NAS (You Only Look Once - Neural Architecture Search) system is a cutting-edge object detection model that excels in accuracy and efficiency. Below is a detailed block diagram of the YOLO-NAS system, explained in an easy-to-understand manner.

****

Block Diagram of a YOLO-NAS System

## Components of the YOLO-NAS System

### 1. Input

* **Description**: This is the starting point where the system receives the image or video to be processed.
* **Function**: Provides the raw data that the system will analyze to detect objects.

### 2. Backbone

* **Description**: A convolutional neural network (CNN) that extracts features from the input image.
* **Function**: Identifies and captures essential visual elements like edges, textures, and shapes by processing the input through multiple convolutional layers.
* **Training**: The backbone is pre-trained on large image datasets to ensure robust feature extraction.

### 3. Neck

* **Description**: A set of convolutional layers that further process the features extracted by the backbone.
* **Function**: Enhances and refines the features, making them more suitable for the detection task performed by the head.
* **Structure**: Typically consists of one or more convolutional layers that create feature pyramids, helping in detecting objects at different scales.

### 4. Head

* **Description**: The part of the system that predicts bounding boxes and class labels for objects in the image.
* **Function**: Uses the processed features to identify the location and category of each object within the image.
* **Components**: Consists of several convolutional layers followed by a fully connected layer that outputs the bounding boxes and class labels.

### 5. Quantization

* **Description**: A process that reduces the precision of the model's weights and activations.
* **Function**: Makes the model smaller and faster, suitable for deployment on devices with limited computational resources.
* **Benefit**: Helps in running the model efficiently without significant loss of accuracy.

### 6. Output

* **Description**: The final results generated by the system.
* **Function**: Provides the detected bounding boxes and class labels for the objects in the image.
* **Usage**: Can be used in various applications like autonomous driving, surveillance, and more.

## Advanced Features of YOLO-NAS

### Quantization-Aware Blocks and Selective Quantization

* **Description**: These techniques involve using specialized blocks and a hybrid quantization method.
* **Function**: Selectively quantizes parts of the model to minimize information loss while balancing latency and accuracy.
* **Benefit**: Ensures high performance even on resource-constrained devices.

### AutoNAC Technology

* **Description**: Deci’s proprietary Neural Architecture Search (NAS) technology.
* **Function**: Determines the optimal sizes and structures of stages in the YOLO-NAS architecture.
* **Benefit**: Automates the discovery of the best model configurations, enhancing overall performance.

### AutoNAC Optimization and Pre-training

* **Description**: Utilizes AutoNAC for optimization and pre-trains the model on extensive datasets.
* **Function**: Pre-training on datasets like COCO, Objects365, and Roboflow 100 ensures the model is well-suited for various object detection tasks.
* **Benefit**: Makes the model highly effective and ready for production environments.

### Hybrid Quantization Method

* **Description**: Selectively quantizes certain layers while leaving others untouched.
* **Function**: Balances latency and accuracy by optimizing which parts of the model to quantize.
* **Benefit**: Maintains high accuracy with improved computational efficiency.

### Pre-training Regimen

* **Description**: Involves a comprehensive training process using multiple datasets and techniques.
* **Function**: Includes pre-training on Object365, COCO Pseudo-Labeled data, Knowledge Distillation, and Distribution Focal Loss.
* **Benefit**: Enhances the model’s ability to handle complex object detection tasks with higher accuracy and robustness.

**Key Features**

**Accuracy and Latency Trade-off**

* **Superior Trade-off**: YOLO-NAS excels in achieving a superior trade-off between accuracy and latency compared to other models. It ensures real-time object detection with high precision, making it suitable for various applications.
* **Performance**: It is designed to be more accurate and faster, leveraging advanced neural architecture and optimization techniques.

**Architecture and Optimization**

* **Neural Architecture Construction (NAC)**: YOLO-NAS employs a specialized NAC tailored for classification and identification using neural networks. This technology enables the model to automatically correct and adapt its architecture to improve performance.
* **Neural Search**: The architecture incorporates a comprehensive neural search process, exploring a vast search space of 101410^{14}1014 potential architectures. This extensive search ensures the discovery of the most efficient model configurations.
* **AutoNAC**: Automated Neural Architecture Construction (AutoNAC) is a key differentiator, providing a sophisticated algorithm for optimal architecture discovery. This approach reduces reliance on trial and error, focusing on balancing accuracy, operations, and model size.

**Quantization**

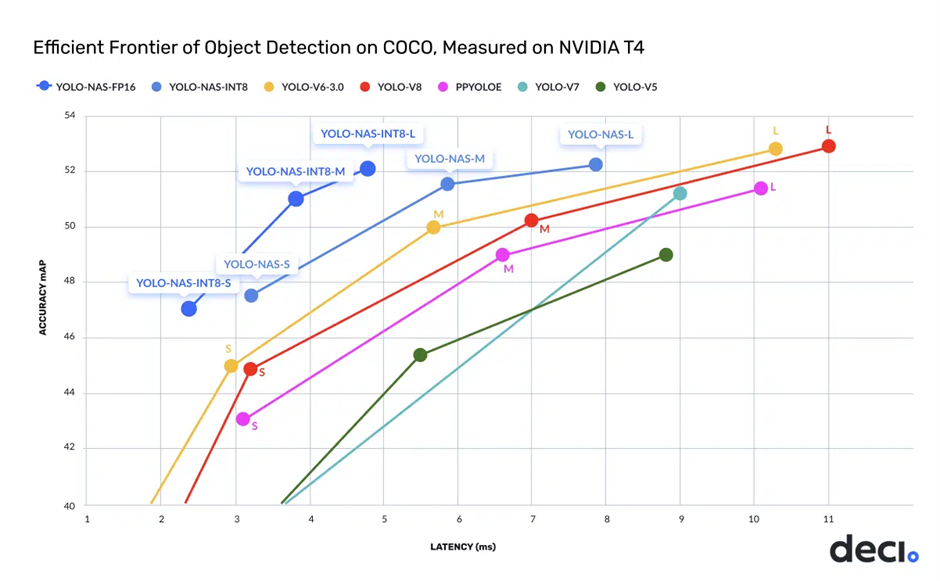
* **Quantization Support**: YOLO-NAS supports quantization, specifically the quantization-aware INT8 technique. This method converts weights from float 32 to int 8, enhancing efficiency on edge devices with limited resources.
* **Memory and Inference Optimization**: By focusing on quantization, YOLO-NAS achieves a balance between memory usage and faster inference, optimizing for both accuracy and efficiency.

**GPU Support**

* **Broad Compatibility**: YOLO-NAS addresses the problem of GPU requirements, offering support for Tesla T4 and other GPUs. This compatibility ensures that the model can be deployed across various hardware platforms, enhancing its usability in real-world scenarios.

**Industry Applications**

* **Retail and Beyond**: YOLO-NAS is designed with the specific needs of various industries in mind, including retail. Its adaptable architecture allows for optimal performance in diverse applications, making it a versatile solution for object detection tasks.
* **Dataset Fine-Tuning**: The model is fine-tuned using comprehensive datasets like RoboFlow100, demonstrating its ability to handle complex object detection tasks with high adaptability.



### QARepVGG Blocks

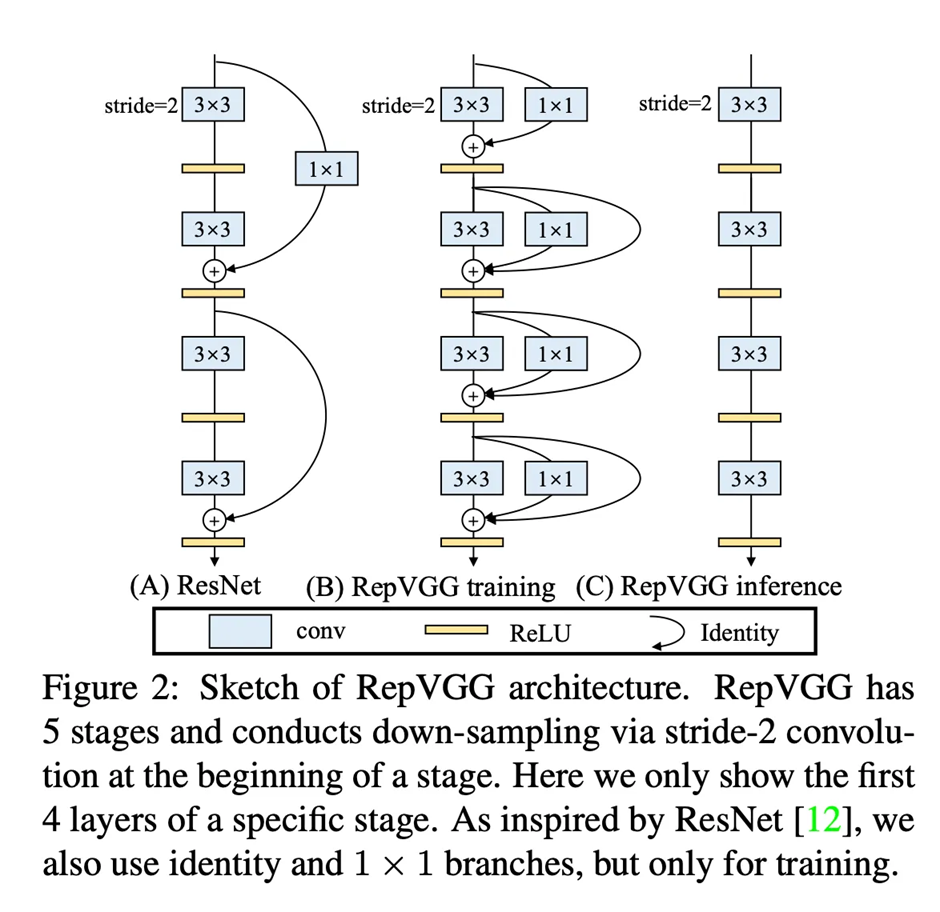
The incorporation of QARepVGG blocks significantly enhances the model's accuracy during quantization.

#### **What is QARepVGG?**

RepVGG is an architecture designed like a multi-branch model (e.g., ResNet, Inception) but can be converted through structural re-parameterization into a VGG-like model. This conversion results in successive stacks of 3 × 3 convolutions and ReLU layers, which yield the same results during inference while being highly optimized by modern computing libraries.

### Quantization for Large Models

Quantization is essential for large models as it reduces the required time and space, making the model more efficient without significantly compromising accuracy. This feature is particularly beneficial for deploying models on devices with limited resources.

****

### Components of Neural Architecture Search (NAS)

YOLO-NAS utilizes a structured approach to NAS, comprising three main components:

1. **Search Space**: The set of all possible architectures that can be considered.
2. **Search Algorithm**: The method used to explore the search space.
3. **Evaluation Strategy**: The criteria and metrics used to assess the performance of different architectures.

### Open-Source Accessibility

YOLO-NAS distinguishes itself by offering an open-source architecture. This accessibility facilitates research and development, allowing the community to innovate and improve upon the existing model.

### Benchmark Achievements

The architecture and training regimen of YOLO-NAS contribute to its success in setting a new benchmark for accuracy and latency tradeoffs in object detection. This model demonstrates exceptional performance, making it a valuable tool for real-time applications.

**YOLO-NAS Pose: A Breakthrough in Object Detection and Pose Estimation**

YOLO-NAS Pose stands out by combining two crucial tasks: detecting people and estimating their poses simultaneously. Unlike other methods that follow a two-stage process, it operates swiftly, akin to bottom-up approaches, yet with its own unique twist.

**Architectural Innovation**

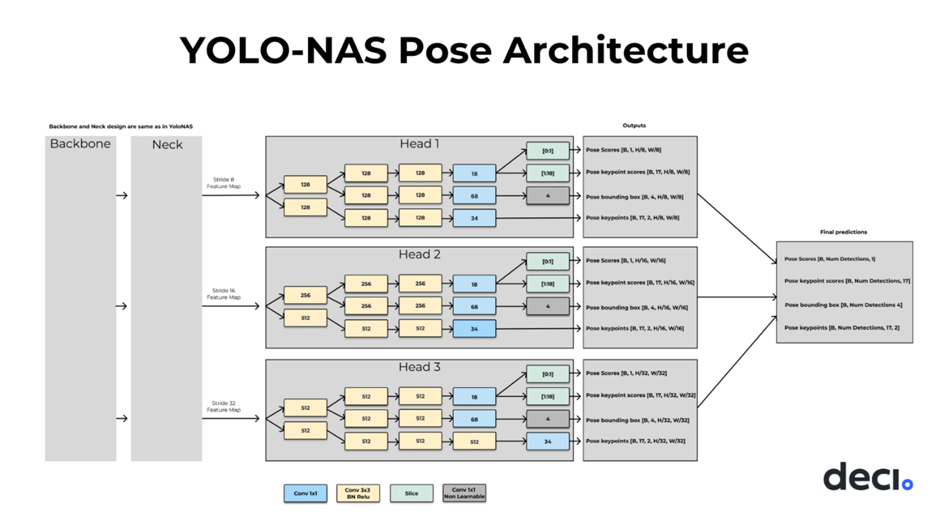
Built upon the YOLO-NAS framework used for object detection, YOLO-NAS Pose shares a similar backbone and neck design. However, what truly distinguishes it is its innovative head design tailored for two tasks: detecting people and estimating their poses accurately in one go.

**Crafting a Novel Head Design**

The head design of YOLO-NAS Pose plays a vital role in balancing runtime performance and prediction accuracy. To find the best head design, Deci employs AutoNAC, their proprietary NAS-powered engine. AutoNAC efficiently explores various architectural designs without the need for exhaustive manual search.

**Hyperparameters for Head Design Search**

AutoNAC's search focuses on key hyperparameters, including the number of Conv-BN-Relu blocks and intermediate channels for both pose and box regression paths, and the decision between shared or distinct stems for pose/box regression

****

**Advanced Training Process**

Training YOLO-NAS Pose involves several refined techniques:

* **Refining the Loss Function**: The loss function incorporates both IoU score for bounding boxes and Object Keypoint Similarity (OKS) score for pose estimation, ensuring accurate predictions for both tasks.
* **Leveraging YOLO-NAS Weights and Data Augmentation**: YOLO-NAS weights initialize the backbone and neck, while data augmentation techniques further enhance model performance.
* **Additional Training Details**: Details such as training time, learning rate, and batch size per variant are meticulously optimized for each model size.

**Simplified Post Processing**

YOLO-NAS Pose simplifies post-processing by using Non-Maximum Suppression on box detections and pose predictions. This results in a streamlined process for selecting final predictions with high confidence.

**Additional Capabilities**

* **Seamless Deployment**: Easy integration into various environments, thanks to its deploy-friendly architecture.
* **Flexibility in Crowd Density**: Accurate pose estimation for both single individuals and crowded environments.
* **Comprehensive Solution**: Integrated object detection and pose estimation for a holistic approach to computer vision tasks.

**Community Availability**

Deci is committed to promoting widespread adoption by releasing YOLO-NAS Pose under an open-source license. Researchers and developers are encouraged to leverage its capabilities, facilitated by the open-source SuperGradients training toolkit for model refinement.