Zhang Yong

[hopeokay99@gmail.com]

Objective:

Dynamic and innovative professional with a proven track record in image processing, data processing, deep learning, and machine learning. Passionate about leveraging cutting-edge technologies to tackle complex challenges and drive impactful solutions.

Education:

- Master of Science in Computer Science, University of Electronic Science & Technology of China, 2015-2018
- Bachelor of Engineering in Computer Science, University of Chinese Academy of Sciences, CAS, 2012-2015

Skills:

- Proficient in image processing techniques such as feature extraction, segmentation, and object detection using Python libraries including OpenCV and scikit-image.
- Extensive experience in deep learning frameworks including TensorFlow and PyTorch for tasks such as image classification, object detection, and natural language processing.
- Skilled in data processing and analysis using Python libraries such as Pandas, NumPy, and SciPy.
- Strong understanding of computer vision algorithms and models including YOLO (You Only Look Once), CNNs (Convolutional Neural Networks), and RNNs (Recurrent Neural Networks).
- Familiarity with cloud platforms such as AWS and GCP for scalable data processing and model deployment.

Professional Experience:

<u>Senior Computer Vision Engineer</u>

[WATRIX], [Beijing, China], [2021 - present]

- Led the development of a real-time object detection system for video surveillance using the YOLO (You Only Look Once) model. Achieved high accuracy and real-time performance, reducing false positives by 30%.
- Implemented custom deep learning models for medical image analysis, including tumor detection and segmentation. Improved accuracy by 25% compared to traditional methods, leading to more reliable diagnoses.
- Designed and optimized data processing pipelines for handling large-scale image datasets, utilizing distributed computing techniques with Apache Spark and Dask. Reduced processing time by 50% while maintaining scalability.
- developed an automatic target tracking and launch system tailored for the detection, tracking, and neutralization of hazardous objects within a specific area by leveraging Python, TensorFlow, PyTorch, and YOLO (You Only Look Once) model
- Collaborated with cross-functional teams to deploy computer vision solutions in production environments, ensuring seamless integration with existing systems and workflows.

Machine Learning Researcher
[Alibaba Group Holding], [969 West Wen Yi Road, Yuhang District, Hangzhou, Zhejiang Province, China, 311121],[2017-2021]

- Conducted research on novel deep learning architectures for image super-resolution, achieving state-of-the-art results on benchmark datasets. Published findings in top-tier conferences and journals.
- Developed and trained deep learning models for anomaly detection in manufacturing processes, leveraging TensorFlow and PyTorch. Reduced false alarms by 40%, leading to significant cost savings for the company.
- Implemented reinforcement learning algorithms for robotic control in dynamic environments, optimizing task performance through self-learning and adaptation. Demonstrated robustness in real-world scenarios through extensive testing and validation.

Projects:

Real-Time Gesture Recognition System

- Developed a real-time gesture recognition system using CNNs and TensorFlow for human-computer interaction applications. Achieved high accuracy in detecting hand gestures with minimal latency.

Summary:

Developed a real-time gesture recognition system using Convolutional Neural Networks (CNNs) and TensorFlow to enable seamless human-computer interaction. Achieved high accuracy in detecting hand gestures with minimal latency.

Difficult Point:

The main challenges included:
- Ensuring high accuracy in detecting diverse hand gestures.
- Achieving minimal latency for real-time interaction.

Solving Method:

To address these challenges:
- Utilized CNNs for their ability to capture spatial features from images effectively.
- Employed data augmentation techniques to improve model robustness.
- Leveraged transfer learning to accelerate training and enhance performance.
- Optimized the inference pipeline for minimal latency, including model quantization and GPU acceleration.

This approach resulted in a real-time gesture recognition system that met high accuracy requirements while providing seamless interaction experiences.

Semantic Segmentation for Autonomous Driving

- Implemented a semantic segmentation model using PyTorch for segmenting road scenes in autonomous driving scenarios. Integrated the model into an end-to-end pipeline for real-time inference on embedded systems.

Summary:

Implemented a semantic segmentation model using PyTorch to segment road scenes in autonomous driving scenarios. Integrated the model into an end-to-end pipeline for real-time inference on embedded systems.

Difficult Point:

The main challenges included:
- Achieving accurate and real-time segmentation of road scenes.
- Integrating the model into an embedded system with limited computational resources.

Solving Method:

To address these challenges:
- Implemented a deep learning-based semantic segmentation model using PyTorch, leveraging architectures such as U-Net or DeepLab.
- Utilized large-scale annotated datasets for training the model to accurately classify pixels in road scenes into semantic categories (e.g., road, lane markings, vehicles, pedestrians).
- Optimized the model architecture and training process to balance accuracy and computational efficiency.
- Integrated the trained model into an end-to-end pipeline for real-time inference on embedded systems, optimizing for performance and memory footprint.
- Employed techniques such as model quantization and pruning to reduce the model size and computational requirements without sacrificing accuracy.

This approach resulted in a semantic segmentation solution tailored for autonomous driving applications, capable of providing accurate scene understanding in real-time while meeting the computational constraints of embedded systems.

Automatic Target Tracking and Launch System for Hazardous Object Destruction

Summary:

This project focuses on developing an automatic target tracking and launch system tailored for the detection, tracking, and neutralization of hazardous objects within a specific area. Leveraging Python, TensorFlow, PyTorch, and YOLO (You Only Look Once) model, the system integrates advanced sensor technologies and machine learning algorithms to identify threats, track their movements, and initiate the appropriate response for effective hazard mitigation.

Key Components:

1. Detection Sensors: The system integrates various sensors, including thermal cameras, LiDAR, and acoustic sensors, to detect hazardous objects within the designated area. These sensors feed real-time data into the system for analysis and processing.

2. Object Detection with YOLO: The YOLO model is utilized for real-time object detection, enabling the system to quickly identify hazardous objects in the sensor data stream. YOLO's efficiency and accuracy make it well-suited for rapid threat detection in dynamic environments.

3. Tracking with TensorFlow and PyTorch: TensorFlow and PyTorch are employed for object tracking, utilizing deep learning algorithms to continuously monitor the position, velocity, and trajectory of detected objects. These frameworks enable the system to predict object movements and anticipate potential threats.

4. Decision-Making Module: Based on the tracked object's characteristics and trajectory, the system's decision-making module determines the appropriate response strategy. This may involve deploying countermeasures, such as intercepting projectiles or triggering controlled detonations, to neutralize the threat.

5. Launch Control: Once the decision is made to neutralize a hazardous object, the system initiates the launch sequence for the appropriate countermeasure. This involves coordinating the activation of launch platforms, such as UAVs or robotic arms, to execute the response strategy with precision.

Benefits:

- Enhanced Safety: The system offers proactive hazard mitigation, minimizing the risk of accidents and ensuring the safety of personnel and assets within the area of operation.
- Rapid Response: By automating threat detection and response, the system can react swiftly to emerging hazards, reducing response times and mitigating potential damage.
- Scalability: Leveraging Python, TensorFlow, PyTorch, and YOLO, the system's modular design allows for scalability and adaptability to different environments and threat scenarios.

Applications:

- Emergency Response: The system can be deployed in disaster-stricken areas to detect and neutralize hazardous objects, such as unexploded ordnance or chemical containers, posing risks to rescue teams and survivors.
- Critical Infrastructure Protection: It can safeguard critical infrastructure, including power plants and transportation hubs, by identifying and neutralizing threats such as unauthorized drones or suspicious packages.

Conclusion:

The Automatic Target Tracking and Launch System for Hazardous Object Destruction, utilizing Python, TensorFlow, PyTorch, and YOLO, represents an innovative solution for proactive threat mitigation and risk management. By integrating advanced sensor technologies and machine learning algorithms, the system enhances safety, responsiveness, and security in various operational scenarios.

Certifications:

- TensorFlow Developer Certificate
- PyTorch Scholarship Challenge Nanodegree

Languages:

- Proficient in Python and Java
- Familiar with C++, MATLAB