Grenze International Journal of Engineering and Technology, June Issue

Design and Implementation of Autopilot using Jetson Nano

Samyuktha Inala[1], N. Aleshwari[2], N. Bhanu prasad[3] and Dr. G.N. Swamy[4]
[1-4]VR Siddhartha Engineering College /EIE, Vijayawada, India
Email: samyukthainala@gmail.com, {nedadhavollualeshwari , nakkabhanuprasad7, gavini_s }@gmail.com

***Abstract*— The "Design and Implementation of Autopilot using Jetson Nano" project aims to develop an advanced autonomous control system for unmanned aerial vehicles (UAVs) and other robotic platforms. Leveraging the powerful capabilities of the NVIDIA Jetson Nano, a small yet high-performance single-board computer, this project focuses on creating a robust autopilot system that enables precise navigation, obstacle detection and avoidance, and real-time decision-making.This vehicle is designed to comply with the traffic regulations and incorporates several NVIDIA tools, including DeepStream SDK, NVIDIA GPU Cloud, TensorRT, CUDA Toolkit, Triton, and Jetbot, which serves as the central component of the project.The project's primary objectives include the integration of various sensors such as cameras and GPS for environmental perception and positioning, as well as the development of intelligent algorithms for path planning. The Jetson Nano's GPU-accelerated computing capabilities will be harnessed to handle complex data processing tasks efficiently, allowing the autopilot to react swiftly to changing conditions.**

***Index Terms*— Autopilot, Jetson Nano ,Unmanned Aerial Vehicles, Autonomous control system, Precise navigation , Obstacle detection, Real-time decision-making , NVIDIA tools, DeepStream SDK,NVIDIA GPU Cloud, TensorRT , CUDA Toolkit, Triton , Jetbot, Environmental perception, Positioning , Data processing tasks**

I. INTRODUCTION

An autopilot system is a technology that automates the control and operation of a vehicle, typically aircraft, automobiles, or drones, without direct human intervention. Autopilots are designed to assist, enhance, or even fully take over specific tasks related to navigation, flight, or driving, allowing human operators to focus on higher-level decision-making or simply reducing the workload. For instance, technologies like dynamic cruise control, automated braking in emergencies, real-time environmental mapping, digital side and rear-view mirrors, parking assistance, and lane-keeping assistance[1]-[3].These systems have a wide range of applications and are continually evolving with advancements in technology.

A. Jetson Nano

The NVIDIA Jetson Nano is a single - board computer that is reasonably priced and small and is intended for edge computing and AI applications. It can execute AI and deep learning applications effectively because to its quad-core ARM Cortex-A57 CPU and 128 CUDA core Maxwell GPU. The Jetson Nano is an affordable platform for AI research and development, featuring 4GB of LPDDR4 RAM and support for well-known AI frameworks like TensorFlow and PyTorch. It offers a thriving community and ecosystem for assistance and learning and is frequently used in robotics, computer vision, and Internet of Things projects. In addition, the Jeston Nano has

Grenze ID: 01.GIJET.10.2.841
© Grenze Scientific Society, 2024

placing in a robot or smart speaker.The Jetson Nano kit can run Linux systems directly and supports a large number of AI frameworks.

Fig 1: Jetson Nano

TABLE 1: SPECIFICATIONS OF JETSON NANO

GPU	NVIDIA Maxwell™ architecture with 128 NYDIA CUDA cores
CPU	Quad-core ARM Cortex-A57 MPCore processor
RAM	4GB 64-bit LPDDR4
STORAGE	16GB eMMC 5.1 flash memory
VIDEO CODING	4K@30(H.264/H.265)
VIDEO DECODING	4K@60(H.264/H.265)
CAMERA	12 channels(3x4 or 4x2)MIPI CSI-2 DPHY 1.1(1.5Gbps)
CONNECTION	Gigabit Ethernet
MONITOR	HDMI2.0 or DP1.2|eDP1.4 | DSI (1x2)2 sync
UPHY	1x1//2/4 PCIE , 1xUSB3.0 , 3xUSB2.0
1/0	1xSDIO/2xSPI/4x i2c/2x 12S/GPIO
SIZE	69.6mmx45mm
STANDARD SIZE	260-pin edge connector

B. Jetbot

A well-known open-source AI robot kit called the NVIDIA JetBot was created by NVIDIA in cooperation with a number of partners. It is intended to offer a platform for studying and experimenting with robots and AI. The NVIDIA JetBot has the following salient characteristics and facets:

➢ *Hardware components include*

- *Camera:* For computer vision and AI applications, it often includes a camera module.
- *Motor Controller*: The robot can move and navigate thanks to the kit's motor control hardware and software.
- *Software Stack*: The JetBot utilizes the JetPack software stack, which comprises a number of AI frameworks and libraries like TensorFlow, PyTorch, and ROS (Robot Operating System), and runs on the Jetson Nano.
- *Jupyter Notebooks*: Jupyter Notebooks, which enable interactive and instructive AI experiments and coding, are frequently used to program and manage the JetBot.

➢ *Applications*

- *AI Education*: The JetBot is frequently used in educational settings to teach students and hobbyists the basics of AI and robotics. It is accessible to beginners thanks to its user-friendly interface and Jupyter Notebook-based programming.
- *Computer Vision*: Since the robot has a camera, it is frequently utilized for computer vision tasks including object detection, image categorization, and facial recognition.
- *Navigation*: Robotics and autonomous navigation experiments can be used the JetBot since it can be taught to navigate on its own thanks to its motorized base.
- A lot of programmers and academics utilize the JetBot as a prototyping platform for testing AI ideas and concepts before putting them into practice in practical applications.
- *Community and Customization*: Users of The JetBot interact frequently and share projects, instructions, and modifications. This community-driven strategy promotes individualization and creativity.
- In the domains of artificial intelligence, computer vision, and robotics, the NVIDIA JetBot is a fantastic tool for both learning and experimenting. For individuals who wish to begin developing AI and robots, it offers a beneficial and hands-on way to learn about these technologies.

TABLE II.: NVIDIA JETBOT SPECIFICATIONS

Brand	Waveshare
CPU model	Quad Core 2.26 GHz
RAM memory	2GB
CPU speed	1.43GHz

C. Neural Network

A key idea in machine learning and artificial intelligence is the concept of neural networks. They are computational models that draw inspiration from the design and operation of the human brain. The capacity of neural networks

to learn complicated patterns and make predictions or judgments based on data has led to their enormous appeal in recent years. Here are some crucial ideas regarding neural networks:

- *Neurons and Layers*: Neurons are interconnected nodes that make up neural networks. An input layer, one or more hidden layers, and an output layer are the layers that make up this network of neurons. The input layer receives input data, and the output layer produces the output.
- *Activation Function*: An activation function is applied by neurons to the weighted sum of their inputs plus bias. As a result, the model gains non-linearity, enabling neural networks to approximation complicated functions. The sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU) are often used activation functions.
- *Feedforword Network*: Without loops or feedback, information moves in one way from input to output in feedforward neural networks. For things like image classification and regression, they are employed.
- *Recurrent Networks:* Recurrent neural networks (RNNs) include feedback loops, allowing them to simulate time-dependent input and sequences. They are frequently employed in tasks involving sequential data, including tasks involving speech recognition and natural language processing.
- *Convolutional Network*: Convolutional Neural Networks (CNNs) are designed specifically to analyze input that resembles a grid, including photos and movies. To automatically learn spatial hierarchies of features, they employ convolutional layers.
- *BackPropagation*: The main algorithm for computing gradients and adjusting weights during training is backpropagation. It entails calculating the gradient of the loss function relative to the network's parameters and making the necessary adjustments to those parameters.
- *Applications*: Image and video analysis, natural language processing, recommendation systems, autonomous vehicles, healthcare, finance, and many other fields use neural networks. For jobs involving pattern detection and intricate data linkages, they are especially effective.

Artificial intelligence and machine learning have advanced significantly in recent years thanks to neural networks, which also remain an active area of research and application.

D. Broker MQTT

A lightweight publish-subscribe messaging protocol called MQTT (Message Queuing Telemetry Transport) was created for effective communication in limited settings, particularly in the context of the Internet of Things (IoT). Here is a quick introduction to MQTT:

- *Publish-Subscribe Model*: The publish-subscribe model is how MQTT works. Other devices (subscribers) get messages from subjects they are interested in, and devices (publishers) transmit messages to specified topics.
- *Lightweight:* MQTT is intended to be lightweight, making it suited for devices with constrained resources and little computing power and bandwidth.
- *Quality of Service (QoS):* MQTT supports three levels of quality of service to ensure message delivery: 0 (At most once, no acknowledgment), 1 (At least once, acknowledgment), and 2 (Exactly once, assured delivery).
- *Retained Messages*: MQTT allows the broker to retain the last message sent on a particular topic. When a new subscriber joins, it can immediately receive the last message sent to that topic.
- *Last Will and Testament*: MQTT provides the option for clients to specify a "last will" message. If a client disconnects unexpectedly, the broker can send the last will message to a specified topic to notify others of the client's unavailability.
- *Security*: MQTT can be configured with security mechanisms such as Transport Layer Security (TLS) for encryption and authentication, allowing for secure communication.
- *Open Protocol*: MQTT is an open protocol with specifications maintained by OASIS, and it is not tied to any particular vendor or technology stack, making it versatile and interoperable.
- Overall, MQTT is an essential tool for building scalable, efficient, and reliable IoT and messaging systems, and it continues to play a significant role in the development of IoT applications and other scenarios where lightweight, publish-subscribe communication is required.

E. Cuda toolkit

NVIDIA offers a collection of libraries and software development tools called the CUDA Toolkit that are used to program NVIDIA GPUs. It makes it possible for programmers to use GPU parallel processing capabilities for a variety of computational tasks. CUDA C/C++, libraries for efficient mathematical operations, a runtime API for GPU resource management, and profiling and debugging tools are all included in the toolkit. It is extensively utilized in domains such as computer graphics, machine learning, and scientific computing. Comprehending

CUDA enables developers to design high-performing applications that take advantage of NVIDIA GPU's processing power.

F. Deep Stream SDK

An enormous global network of cameras and sensors continuously collects enormous volumes of data. This data is a great tool for improving income streams, creating company insights, and optimizing procedures. Reliable, real-time Intelligent Video Analytics (IVA) is required in every situation, whether it is for improving consumer pleasure in retail establishments, guaranteeing health and safety in hospitals, streamlining traffic at junctions, or identifying defects in manufacturing components.

Massive amounts of data are continuously collected by a massive worldwide network of cameras and sensors. This data is an excellent resource for streamlining processes, developing business insights, and enhancing revenue sources. Intelligent video analytics (IVA) that is dependable and up-to-date is necessary in all scenarios, be it enhancing customer experience in retail stores, ensuring patient safety in medical facilities, optimizing traffic at intersections, or detecting flaws in manufacturing parts.

G. NVIDIA GPU cloud

The platform known as NVIDIA GPU Cloud (NGC) provides GPU-optimized containers for a range of uses, such as high-performance computing and deep learning. It makes it simpler to implement GPU-accelerated workloads in cloud and on-premises systems by offering pre-configured Docker containers that fully utilize NVIDIA GPUs. Researchers and developers use NGC extensively to quickly access and employ GPU resources for purposes like training deep learning models and scientific simulations.

H. NVIDIA TensorRT

The deep learning inference optimizer and runtime included in NVIDIA® TensorRTTM SDK for high-performance deep learning inference provide low latency and high throughput for inference applications.Benefits of NVIDIA TensorRT :

1) Speed Up Inference by 36X
2) Optimize Inference Performance
3) Accelerate Every Workload
4) Deploy, Run, and Scale With Triton

II. AUTOMATIC DRIVING

A. Introduction to linear regression

The linear regression output is a continuous value, so it is suitable for regression problems. Regression problems are very common in practice such as the prediction of continuous values such as house prices ,temperatures and sales . Different from the regression problem, the final output of the model in the classification problem is a discrete value. Softmax regression is suitable for classification problems.

B. Basic elements of linear regression

The goal of this application is to predict the variable means of x and y, respectively. As with the correlation coefficient r, once again, everything revolves around variances.

$$y = x_1 w_1 + x_2 w_2 + b \qquad (1)$$

Where w_1 and w_2 are weights, b is bias and both are scalars.They are parameters of the linear regression. We usually allow some error between them.

C. Model training

We need to find specific model parameter values through the data, so that the error of the model in the data is as small as possible. This process is called model training.

D. Training data

We usually collect a series of real data. We hope to find model parameters from these data to minimize the eror between the model's predicted and the real data. In machine learning terminology, this data set is called training data or training set. Two factors used to predict a label are called features, which are used to characterize the sample. Suppose that the number of samples we collect is n , the characteristics of the sample with index I are $x(i)_1$ and $x(i)_2$ and the label is y(i).

$$y'^{(i)} = x_1^{(i)} w_1 + x_2^{(i)} w_2 + b. \quad (2)$$

E. Loss Function

In model training, we need to measure the error between the predicted value and the true value. Usually we will choose a non-negative number as the error , and the smaller the value is,the smaller the error is. A common choice is the square function .Its expression for evaluating the sample error at index i is :

$$l^{(i)}(w_1, w_2, b) = \frac{1}{2}(y'^{(i)} - y^{(i)})^2, \quad (3)$$

Where the constant is 1/2;the constant coefficient after derivation of the square term is 1,so that it looks simpler in form. Obviously , the smaller the error is, the closer the predicted value is to the real value and the error is 0 when the two equal.Given the training data set , this error is only related to the model parameters, so we will record it as a function that takes the model parameters as parameters.In machine learning, the function that measures error is called the loss function. The square error function used here is also called square loss.

Usually, we use the average of all sample errors in the training data set to set to measure the quality of the model prediction , that is :

$$l(w_1, w_2, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(w_1, w_2, b)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(x_1^{(i)} w_1 + x_2^{(i)} + b - y^{(i)}\right)^2. \quad (4)$$

In model training, we hope to find a set of model parameters, which are recorded as w*1,w*2,b*, to minimize the average loss of training samples:

$$w_1^*, w_2^*, b^* = argmin\, l(w_1, w_2, b). \quad (5)$$

F. Model prediction

After the model training is completed, we record the values of the model parameters w1,w2,b when the optimization algorithm is stopped as w1,w2,b^.

III. Steps For Automatic Driving

1. Data collection
2. Training
3. Deploy

In addition to classification, the linear regression uses it to enable JetBot to advance along a road.

1. Place Jetbot at different locations on the path .
2. Display real-time camera input from Jetbot robot.
3. Using the game pad controller ,place a “green dot” on the image ,which we want the robot to move.
4. Store the X,Y values of this green dot together with the image of the robot camera.

Then, in the training notebook,we will train a neural network to predict the X,Y values of our labels.In the live demonstration , we will use the predicted X and Y values to calculate an approximate steering value .Some guidelines how to determine the target location.

1. Look at the live video of the camera.
2. Imagine the path the robot should follow .
3. Place the target on the path as far as possible, so that the robot can rush directly to the target without “running away” from the road.

For example, if we are on a very straight road, we can place it on a horizontal line. If we are making a sharp turn, it may need to be placed closer to the robot so that it will not run out of the boundary.

IV. Conclusion

In conclusion, the autopilot system built around the Jetson Nano offers several key advantages like high performance, energy efficiency, versatility, Deep Learning Capabilities . Implementing cam detection with a JetBot, you'll typically create functions or scripts that control the motor speeds, monitor sensor data, and react to various situations. Depending on your goals, you can also explore more advanced locomotion techniques like path planning and obstacle avoidance to make your JetBot navigate autonomously.

REFERENCES

[1] Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Sensors 2021, 21, 2140. https://doi.org/10.3390/s21062140

[2] Baïou M., Quilliot A., Adouane L., Mombelli A., and Zhu Z., Algorithms for the Safe Management of Autonomous Vehicles, Annals of Computer Science and Information Systems. IEEE, Sep. 26, 2021. doi: 10.15439/2021f18.

[3] Podbucki K., Possibilities and limitations of environment monitoring with usage of LiDAR scanner, Przegld Elektrotechniczny, vol. 1, no. 1. Wydawnictwo SIGMA-NOT, sp. z.o.o., pp. 1863189, Jan. 04, 2022. doi: 10.15199/48.2022.01.40.

[4] XiaoR GEEK Classroom-Jetbot 1.0