

# Leistungsnachweis

Grade Report

Familiennamen/Family Name:

**Bai**

Vorname(n)/First Name(s):

**Yanhao**

Geburtsdatum/Date of Birth:

**13. November 2001**

13 November 2001

Geschlecht/Gender:

**männlich**

male

Geburtsort/Place of Birth:

**Shandong Province**

Matrikelnummer/Student ID Number:

**03769857**

Studiengang/Degree Program:

**Austauschprogramm Informatik**

Exchange Program Informatics

Angestrebter Abschluss/Degree in progress:

Datum/Date:

**13. Dezember 2023**

13 December 2023

Modul-ID Module ID	Bezeichnung Title	Semester Semester	Note Grade	Credits Credits
	<b>Informatik</b> Informatics			5
<b>IN2062</b>	<b>Grundlagen der Künstlichen Intelligenz</b> Techniques in Artificial Intelligence			
	Grundlagen der Künstlichen Intelligenz Techniques in Artificial Intelligence	22W	3,0	5

## Erläuterungen/Explanations:

Notenskala: 1,0-1,5 sehr gut, 1,6-2,5 gut, 2,6-3,5 befriedigend, 3,6-4,0 ausreichend, 4,1-5,0 nicht ausreichend

Grades: 1,0-1,5 very good, 1,6-2,5 good, 2,6-3,5 satisfactory, 3,6-4,0 sufficient, 4,1-5,0 fail

Bewertung von Studienleistungen: BE = bestanden NB = nicht bestanden

Performance Key: BE = pass NB = fail

Credits: Gemäß dem European Credit Transfer System (ECTS) Maßeinheit für die Arbeitsbelastung eines Studierenden; ein Credit entspricht der Arbeitszeit von 30 Stunden.

Credits: a unit of measure within the European Credit Transfer System (ECTS) representing student workload. A credit is equal to 30 hours of work.

\*) = anerkannt

\*\*) = enthält anerkannte Leistungen

\*) = accredited

\*\*) = contains accredited exams

This is to certify that

**Yanhao Bai**

completed the short academic programme

**Artificial Intelligence and Machine Learning:  
Advanced Applications of Neural Networks and Deep  
Learning**

provided by Lady Margaret Hall, a college in the University of Oxford

17<sup>th</sup> July 2023 to 4<sup>th</sup> August 2023



---

Dr Christopher Adamson

Head of Short Academic Programmes



# 国家知识产权局

**116000**

辽宁省大连高新技术产业园区高能街 35A 号 28 层 3 号 辽宁鸿文知  
识产权代理有限公司  
王海波(18504286779)

发文日:

2023 年 06 月 12 日



申请号: 202310689260.3

发文序号: 2023061201077440

## 专利申请受理通知书

根据专利法第 28 条及其实施细则第 38 条、第 39 条的规定, 申请人提出的专利申请已由国家知识产权局受理。现将确定的申请号、申请日等信息通知如下:

申请号: 2023106892603

申请日: 2023 年 06 月 12 日

申请人: 大连理工大学

发明人: 李轩衡, 葛东旭, 白岩浩, 程思科, 金灿

发明创造名称: 基于强化学习的无人机智能轨迹规划和通感资源分配方法

经核实, 国家知识产权局确认收到文件如下:

权利要求书 1 份 3 页, 权利要求项数: 2 项

说明书 1 份 7 页

说明书摘要 1 份 1 页

专利代理委托书 1 份 2 页

发明专利请求书 1 份 5 页

实质审查请求书 文件份数: 1 份

申请方案卷号: 20232566 大连理工李轩衡

提示:

1. 申请人收到专利申请受理通知书之后, 认为其记载的内容与申请人所提交的相应内容不一致时, 可以向国家知识产权局请求更正。

2. 申请人收到专利申请受理通知书之后, 再向国家知识产权局办理各种手续时, 均应当准确、清晰地写明申请号。

审查员: 自动受理

联系电话: 010-62356655

审查部门: 初审及流程管理部



200101  
2022.10

纸件申请, 回函请寄: 100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收  
电子申请, 应当通过专利业务办理系统以电子文件形式提交相关文件。除另有规定外, 以纸件等其他形式提交的文件视为未提交。

## 中国科学院软件研究所集成创新中心实习鉴定书

实习生姓名	白岩浩	性别	男	专 业	计算机科学与技术
所在学校	大连理工大学			学 号	20201072017
实习单位	中国科学院软件研究所 集成创新中心		实习岗位		
单位联系电话	010-62662958				
实习时间	2023 年 6 月 19 日 至 2023 年 7 月 14 日				
实习生个人 实习总结	<p>本次实习围绕百度 Apollo 自动驾驶技术方案展开，该技术串联整个无人驾驶的工作流程，提出了包括系统底层、软件使用、数据学习方面的综合解决方案。我的主要工作聚焦于该技术的 HDMap 和 Planning(Routing)两个模块，前者使用 C++读取 Opendrive 格式的高精度地图并基于 kdTree 管理地图数据，后者基于 HDMap 功能实现路径规划算法并采用可视化方法呈现出 Routing 路线。</p> <p>第一周我完成了使用 Protobuf 读取地图文件，构建 kdTree 存放地图数据，理解分析 Apollo 高清地图各个模块的源码与功能；</p> <p>第二周我完成了地图的可视化实现（PCL, Qt, Vtk, OpenCV）和图形库使用，并成功实现了输入起点坐标查找所在道路和后续结点的功能，即成功实现 HDMap；</p> <p>第三周我尝试沿用 kdTree 管理地图数据的思路，以实现 Routing 模块的功能，但很遗憾 A*, GBFS, Dijkstra 等路径规划算法的输出队列均为空，因此转换思路，使用 graph 图管理地图数据，但很遗憾输出队列仍然为空；虽然没能通过使用 kdTree 或 graph 图管理地图数据的方式实现 Routing 模块的功能，但是在编写和调试程序的过程中我对 Apollo 无人驾驶系统的 Routing 模块有了很扎实的掌握；</p> <p>第四周我成功调用系统自带的 Dreamview (Apollo6.0)和 OSM (Apollo3.0)，完成了两种可视化呈现 Routing 模块的方法。</p> <p>至此，Apollo 自动驾驶技术的两个模块 HDMap 和 Routing 都已经成功实现！</p> <p style="text-align: right;">实习生签名：白岩浩</p> <p style="text-align: right;">2023 年 8 月 2 日</p>				
实习 单位 鉴定 意见	<p>综合      同学自      年   月   日至      年   月   日在我单位的实习表现， 认定其实习综合表现为：优秀 (✓) 良好 ( ) 一般 ( ) 较差 ( )</p> <p style="text-align: right;">指导教师签名：Lingzhong Meng</p> <p style="text-align: right;">2023 年 8 月 2 日</p> <p style="text-align: right;">(盖章)</p>				
备注					

## 实习证明

兹证明，大连理工大学电子信息与电气工程学部计算机科学与技术专业的本科生白岩浩，于2023年4月27日至2023年6月1日在辽宁省大连市甘井子区人民检察院实习。实习岗位为“人工智能驱动的未成年人保护法律监督研究”项目组成员。

白岩浩同学对工作充满热情，工作成绩优秀，他成功编写的代码模块包括：

1. 多模态线索的主动抓取与搜集。这部分代码通过收集文本、图片、视频和语音等多类型数据源，搭建了大数据法律监督模型的前端入口。

2. 多模态线索的初筛模型。这部分代码首先对多模态线索数据进行搜集并转化为文本，实现的技术有针对图像的 im2txt 算法、针对视频的视频抽帧技术、针对音频的 LAS 模型；接着对无效数据下定义，筛除无效线索。

3. 线索事件的智能综合判别模型。这部分代码首先根据线索事件是否达到司法保护的度划分为司法保护和非司法保护，使用的技术是 BERT 二分类；接着将司法保护进一步判别为四个子类，使用的技术是朴素贝叶斯四分类。

4. 线索事件的智能标记治理模型。这部分代码引入五大保护，对上一步的司法/非司法标签数据打分，判断起所属保护类型，使用的技术是 BERT 五分类。

5. 可靠线索数据的多维度可视化呈现。代表呈现方式是热力图。

白岩浩同学不但个人能力突出，还具备强烈的团队合作精神，在项目全程成功融入团队，并在工作中展现出出色的解决问题能力。

白岩浩同学的表现令人印象深刻，对其未来成就具有很高的期望！

特此证明。



2022  
Interdisciplinary Contest In Modeling®  
Certificate of Achievement

Be It Known That The Team Of

Yanhao Bai  
Zhaohuan Teng  
Zhiang Xu

With Faculty Advisor

Qiuhui Pan

Of

Dalian University of Technology

Was Designated As  
Finalist

  
Solomon Garfunkel, Executive Director

Administered by  


  
Amanda Beecher, Contest Director

With support from  
    

2021  
Mathematical Contest In Modeling®  
Certificate of Achievement

Be It Known That The Team Of

Xinyi Yuan  
Yanhao Bai  
Jiaxin Zhong

With Faculty Advisor

Qihui Pan

Of

Dalian University of Technology

Was Designated As  
Honorable Mention

  
Steven B. Horton, Contest Director

Administered by  
  
With support from

  
David H. Howell, Head Judge

 AMS  ASA  informs  MAA  siam.





中国工业与应用数学学会  
China Society for Industrial and Applied Mathematics



大连理工大学

学 生：白岩浩  
徐志昂  
王子鸣

指导老师：教师组

荣获二零二二年高教

社杯全国大学生数学建模

竞赛本科组二等奖。







# 全国大学生数学建模竞赛

大连理工大学

学生 白岩浩 徐志昂 王子鸣 指导教师 教师组 荣  
获 2022 年高教社杯全国大学生数学建模竞赛辽宁赛区  
本科组一等奖。特发此证，以资鼓励。

辽宁赛区组委会



中国工业与应用数学学会





2021年第十一届MathorCup高校数学建模挑战赛

## 荣誉证书

大连理工大学盘锦校区

参赛队员：白岩浩 袁欣怡 钟佳鑫

荣获 2021年第十一届MathorCup高校数学建模挑战赛 本科组

三等奖

证书编号: MC21060020040

中国优选法统筹法与经济数学研究会

2021年6月2日





homepage: <https://vanhao5103233729.github.io/edu/>

## Beam Prediction Based on Multi-Modal Fusion (in Python)

!!! In-Progress for a journal paper heading to IEEE Transactions on Vehicular and Technology.

[\[Paper\]](#) | [\[Code\]](#) | [\[Video\]](#)

### Overview

This project starts Oct, 2023, featured as a Python-coded deep learning project, developed as my bachelor thesis and a journal paper heading to IEEE Transactions mentored by [Prof. Xuanheng Li](#). Over 2 weeks, I focused on integrating radar and image data using PyTorch, designing neural network architectures, and implementing efficient data preprocessing routines. My role extended to tuning hyperparameters aiming to receive better accuracy of our proposed scheme for multi-modal data. For more insights into the project, visit the repository [here](#).

@timeFrame: Nov 13 - 27, 2023 (2 wks)

@role: Developer

@skills: Python, PyTorch, PyCharm, Data Preprocessing, DeepSense6G Dataset

@team: Prof. Xuanheng Li, **Yanhao Bai**, Dongxu Ge, Sike Cheng

### The Problem

The future of vehicular communication networks relies on mmWave massive multi-input-multi-output antenna arrays for intensive data transfer and massive vehicle access; however, reliable V2I links require narrow beam alignment, which traditionally involves excessive signaling overhead. To address this issue, existing methods rely solely on communication processing; my team noticed a need to improve beam alignment accuracy, by obtaining comprehensive environmental features.

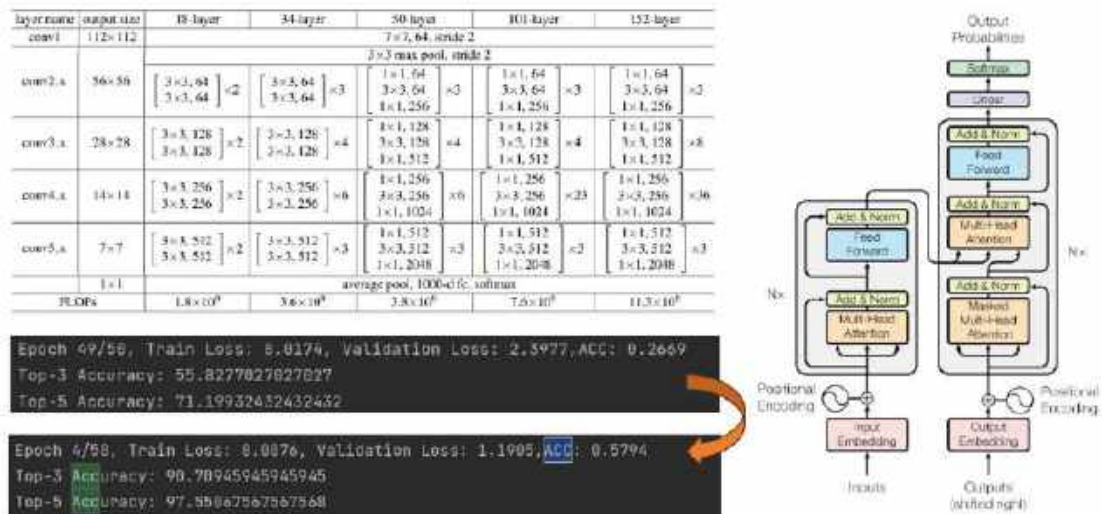
### The Goal

For this 2-week development sprint, my team's solution was a novel proactive beam prediction scheme that integrates multi-modal sensing and communications via Multi-Modal Fusion utilizing deep learning approaches, which is composed of multiple neural network components with distinct functions. Our proposed Multi-Modal Fusion scheme achieves more accurate and stable beam prediction; even in complex dynamic scenarios, robust prediction results can be guaranteed (for this, we enlarged the existing dataset by synthesizing adverse weather conditions to RGB data, for more details, see my project: Adverse Weather Synthetization on Images Based on GPT-4), demonstrating the feasibility and practicality of our proposed proactive beam prediction approach.

### Features

1. Preprocessing: Utilizes pandas for data manipulation, torchvision transforms for img preprocessing.
2. Architecture: Implements custom neural network models of encoder and attention classes, integrating multi-head attention and positional encoding to process sequential data effectively.
3. Data Handling: Employs PyTorch's DataLoader and custom dataset class for efficient data loading and batching, enabling streamlined handling of large datasets.
4. Extraction: Utilizes pre-trained ResNet model for image feature extraction and Fourier transforms for radar data processing.
5. Training: Orchestrates the training process with backpropagation and validation via cross-entropy loss and accuracy metrics, while efficiently utilizing GPU resources for computation.

## How it works



## Sample Code

# Code snippet, featured as core of positional encoding module in a transformer model, crucial for understanding the sequential aspect of input data

**class PositionalEncoding(nn.Module):**

```

def __init__(self, d_model, dropout, max_len=5000):
    super(PositionalEncoding, self).__init__()
    self.dropout = nn.Dropout(p=dropout)
    # Initialize positional encoding with shape (max_len, d_model)
    pe = torch.zeros(max_len, d_model)
    position = torch.arange(0, max_len).unsqueeze(1)
    div_term = torch.exp(torch.arange(0, d_model, 2) * -(math.log(10000.0) / d_model))
    pe[:, 0::2] = torch.sin(position * div_term)
    pe[:, 1::2] = torch.cos(position * div_term)
    pe = pe.unsqueeze(0)
    self.register_buffer("pe", pe)
  
```

```

def forward(self, x):
    x = x + self.pe[:, : x.size(1)].requires_grad_(False)
    return self.dropout(x)
  
```

homepage: <https://vanhao5103233729.github.io/edu/>

## **Adverse Weather Synthetization on Images Based on GPT-4 (in Prompts)**

**!!! Successfully deployed GPT-4 API for image processing, totally Cycle/WeatherGAN-free.**

**[Paper] | [Code] | [Video]**

### **Overview**

This project starts Nov, 2023, featured as blended-language-coded image augmentation project, deploying most advanced GPT-4 API and prompts. Over 2 weeks, I focused on encompassing existing scripts and models that manipulate images to simulate various adverse weather conditions on images serving as enlarged dataset for our proposed beam prediction scheme mentored by [Prof. Xuanheng Li](#). My role extended to writing prompts for GPT-4 to learn and deploy the above image augmentation. For more insights into the project, visit the repository [here](#).

@timeFrame: Oct 16 - 30, 2023 (2 wks)

@role: Developer

@skills: Python, MATLAB, OpenCV, TensorFlow, GPT-4, Image Processing and Augmentation

@team: Prof. Xuanheng Li, **Yanhao Bai**

### **The Problem**

The task of realistically simulating different adverse weather conditions (rain, snow, fog, dusk, etc.) on images can be quite complex, by utilizing image processing methods, such as CGAN, CycleGAN, WeatherGAN. To address this issue, my team noticed a need to adopt a generative scheme with the innovative use of GPT-4 prompts and APIs for intelligent automation in image processing.

### **The Goal**

For this 2-week development sprint, my team's solution was a comprehensive multi-tool framework then prompts for worsening images with an array of adverse weather conditions and color alterations. We ended up with successful experiments that demonstrated the effectiveness of our emphasis on deploying GPT-4 to scale up capabilities of image augmentation and processing while minimizing manual intervention, thus offering a novel, automated solution to creating diverse and enhanced datasets for AI-driven, ML-aided applications.

### **Features**

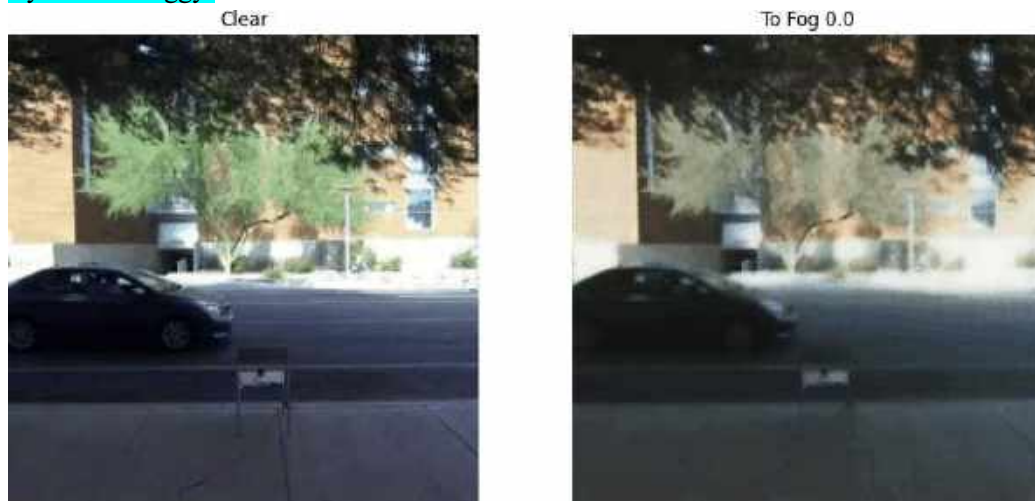
1. Image Augmentation: Advanced python scripts that apply various effects like fog, rain, shadows, sun flare, motion blur, and autumnal changes to images.
2. CycleGAN: TensorFlow-based fog simulation.
3. WeatherGAN: MATLAB-based color transfer for diverse visual effects.
4. GPT-4: Deployment of GPT-4 for interpreting and executing the above image processing tasks based on textual prompts and APIs, showcasing an LLM-assisted approach to cv.
5. Tool Versatility: Use of Python, MATLAB, OpenCV, TensorFlow, and most significantly, GPT-4, demonstrating multifaceted ways and a novel approach to image processing.



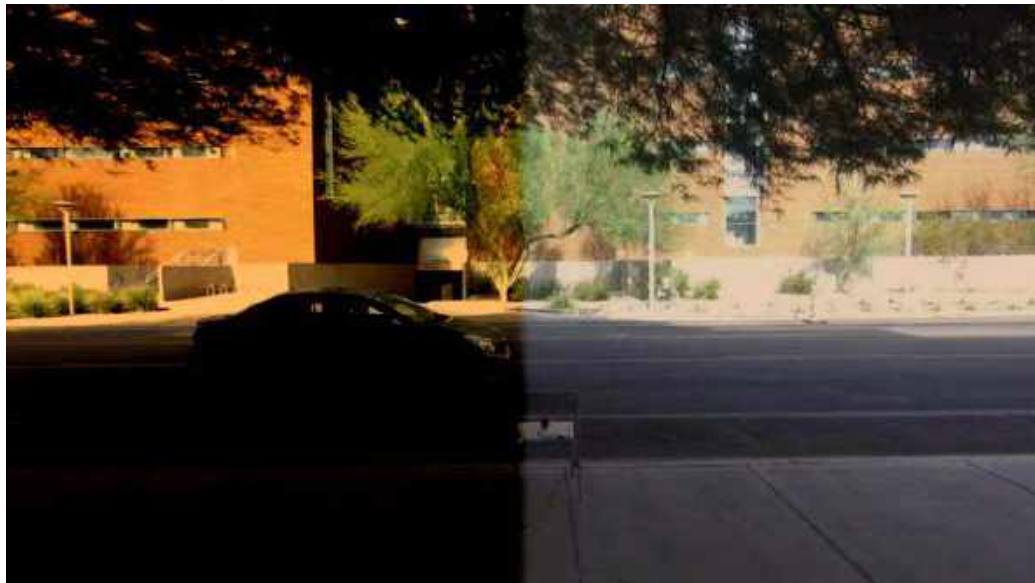
homepage: <https://vanhao5103233729.github.io/edu/>

## How it works

### CycleGAN foggy:



### WeatherGAN dusky:



### GPT-4 all effects:



## Sample framework

### 3. Raindrop Image Formation

We model a raindrop degraded image as the combination of a background image and effect of the raindrops:

$$\mathbf{I} = (1 - \mathbf{M}) \odot \mathbf{B} + \mathbf{R} \quad (1)$$

where  $\mathbf{I}$  is the colored input image and  $\mathbf{M}$  is the binary mask. In the mask,  $M(x) = 1$  means the pixel  $x$  is part of a raindrop region, and otherwise means it is part of background regions.  $\mathbf{B}$  is the background image and  $\mathbf{R}$  is the effect brought by the raindrops, representing the complex mixture of the background information and the light reflected by the environment and passing through the raindrops adhered to a lens or windscreen. Operator  $\odot$  means element-wise multiplication.

Raindrops are in fact transparent. However, due to their shapes and refractive index, a pixel in a raindrop region is not only influenced by one point in the real world but by the whole environment [25], making most part of raindrops seem to have their own imagery different from the background scene. Moreover, since our camera is assumed to focus on the background scene, this imagery inside a raindrop region is mostly blur. Some parts of the raindrops, particularly at the periphery and transparent regions, convey some information about the background. We notice that the information can be revealed and used by our network.

Based on the model (Eq. (1)), our goal is to obtain the background image  $\mathbf{B}$  from a given input  $\mathbf{I}$ . To accomplish this, we create an attention map guided by the binary mask  $\mathbf{M}$ . Note that, for our training data, as shown in Fig. 5, to obtain the mask we simply subtract the image degraded by raindrops  $\mathbf{I}$  with its corresponding clean image  $\mathbf{B}$ . We use a threshold to determine whether a pixel is part of a raindrop region. In practice, we set the threshold to 30 for all images in our training dataset. This simple thresholding is sufficient for our purpose of generating the attention map.

## Sample prompt

PROMPT: learn the following code (alleffects.py), it adds adverse weather conditions on the imgs

GPT-4: Thank you for sharing this Python code. It appears to be a comprehensive script for applying various weather and environmental effects to images using the OpenCV library. The code includes functions for simulating rain, snow, fog, gravel, sun flare, motion blur, autumn effects, and more. These functions modify image properties like brightness, color saturation, and overlay patterns to create realistic effects.

PROMPT: you're totally right! It adds rain, snow, fog, gravel, sun flare, motion blur, autumn effects on imgs. Now apply what you have learnt on my image (test.jpg), i.e., add rain, snow, fog, gravel, sun flare, motion blur, autumn effects on my original image. Note: simply add all effects on my original image, instead of creating any fake, imaginary images.

GPT-4: I have applied all effects to your original image. You can download the processed image with the adverse weather conditions from the following link: [Download the processed image.](#)

homepage: <https://vanhao5103233729.github.io/edu/>

## Multi-Modal 6G Sensing and Communication Dataset Walkthrough (in Python)

!!! A large-scale real-world multi-modal isac dataset for 6G deep learning research.

[\[Slides\]](#) | [\[Code\]](#) | [\[Video\]](#)

### Overview

This project starts Sep, 2023, featured as a Python-coded dataset walkthrough project, to perform the essential data loading and visualization tasks on all modalities - Power, RGB images, GPS positions, Radar, and Lidar, in the DeepSense6G Dataset. Besides tasks on the main modality data, the project has acquired other relevant information such as timestamp, number of connected satellites, etc., and performed annotations such as sequence index, direction of movement, obj detection bounding boxes. The project also shows how to access and leverage the metadata and annotations to visualize and understand propagation patterns. For more insights into the project, visit the repository [here](#).

### The Problem

In the burgeoning field of 6G research, a critical challenge lies in analyzing and visualizing complex, multi-modal datasets that are quintessential for understanding real-world sensing and communication dynamics. The intricacies of such datasets include diverse data types and formats, necessitating a versatile approach to data processing and interpretation. Traditional data handling methods fall short in providing the necessary depth and clarity for insights essential in 6G technology development.

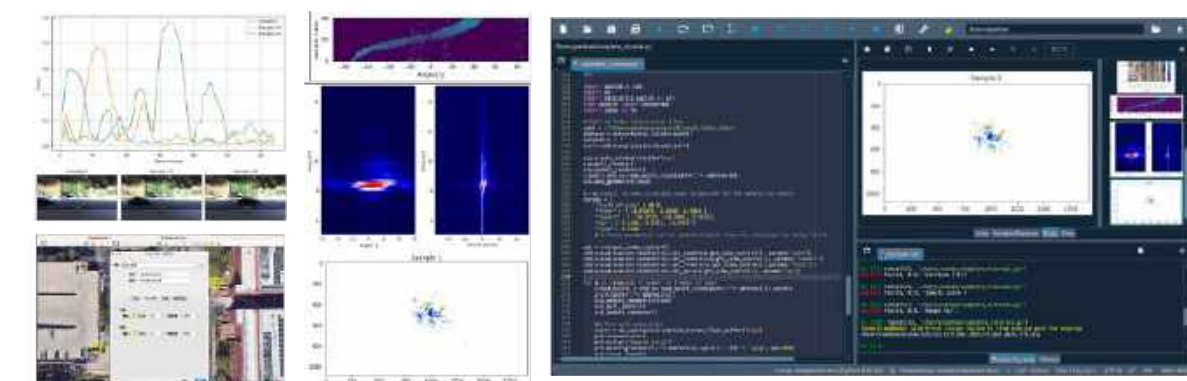
### The Goal

This project aims to develop a robust framework for the comprehensive analysis and visualization of multi-modal datasets pertinent to 6G deep learning research. By leveraging advanced py libraries and visualization tools, the project aspires to not only streamline the processing of data of Power, RGB, GPS, Radar, and Lidar but also enhance understanding of data through intuitive visualizations.

### Features

1. Data Processing: Uses Python library Pandas for efficient data manipulation and management.
2. Visualization: Employs Matplotlib and Open3D for visual representation of various data types.
3. Multi-Modal Data Handling: Integrates various data forms including Power, RGB images, GPS, Radar, and Lidar data, ensuring a holistic approach to data analysis.
4. Interactive Analysis: Provides tools for dynamic data exploration, enabling users to interact with and derive insights from complex datasets.

### How it works



homepage: <https://vanhao5103233729.github.io/edu/>

### Sample Code

# Code snippet, featured as utilizing open3d library to load and visualize point cloud data, by setting up a 3D visualization window, loading point cloud files, and adjusting the display for custom view

```
import open3d as o3d
import os
from natsort import natsorted
import numpy as np

# Path to lidar (point cloud) files
path = '/Users/yanhao/scenario31/unit1/lidar_data/'
address = natsorted(os.listdir(path))
vis = o3d.visualization.Visualizer()
vis.create_window(visible=True)
vis.poll_events()
vis.update_renderer()
cloud = o3d.io.read_point_cloud(path + "/" + address[0])
vis.add_geometry(cloud)

# Setting custom view parameters
params = {
    "field_of_view": 60.0,
    "front": [-0.01093, 0.0308, 0.9994],
    "lookat": [-18.9122, -18.4687, 7.3131],
    "up": [0.5496, 0.8351, -0.0197],
    "zoom": 0.3200
}

vis.get_view_control().set_front(params['front'])
vis.get_view_control().set_lookat(params['lookat'])
vis.get_view_control().set_up(params['up'])
vis.get_view_control().set_zoom(params['zoom'])

for file_name in address:
    if file_name.endswith('.ply'):
        pcd = o3d.io.read_point_cloud(os.path.join(path, file_name))
        vis.add_geometry(pcd)
        vis.poll_events()
        vis.update_renderer()

vis.destroy_window()
```

homepage: <https://vanhao5103233729.github.io/edu/>

## Baidu Apollo Autonomous Driving Project at ISCAS in Beijing (in C++)

!!! This is my 2023 summer intern at Institute of Software Chinese Academy of Sciences in Beijing.  
[Paper] | [Code] | [Video]

### Overview

This project starts Jul, 2023, featured as a C++-coded comprehensive solution for the Baidu Apollo Autonomous Driving Project, which focuses on linking functionalities of bottom layers, software usage, and data learning, etc., to connect the entire autonomous driving workflow. In Summer Intern, I worked closely with various teams at Integration Center, ISCAS, to propose a comprehensive solution for such project; I coded, tested, debugged, implemented and documented the two modules (hdmap, routing) using C++ and Python. For more insights into the project, visit the repository [here](#).

### The Problem and Goal

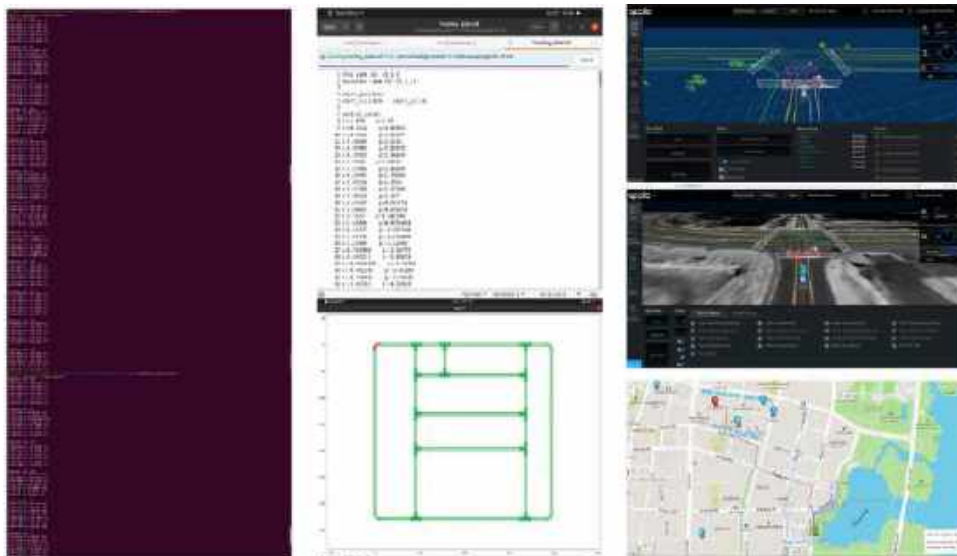
For HDMap, its primary goal is to implement an efficient algorithm by utilizing protobuf to read high-precision maps in Opendrive format, build up KDTree storing map data, implement the functionality of finding the road and subsequent nodes based on the entered starting point coordinates, lastly visualize results via PCL, Qt, Vtk, OpenCV.

For Routing, its primary goal is to deploy the two methods of visualizing the system's expected routing functions, by utilizing Apollo's built-in tools of Dreamview (Apollo6.0) and OSM (Apollo3.0).

### Features

1. Map Parsing: Utilizes protobuf for reading Opendrive format maps.
2. Data Structuring: Designs data structures to store map data post-parsing, constructs a kdTree for efficient data storage and retrieval.
3. Map Visualization: Displays maps using graphical libraries, by employing PCL, Qt, Vtk, OpenCV for map data and routing paths.
4. Route Identification: Identifies roads and subsequent path points based on input coordinates.
5. Routing: Implements visualization of the system's routing capabilities using Apollo's Dreamview (Apollo6.0) and OSM (Apollo3.0) tools.

### How it works





homepage: <https://vanhao5103233729.github.io/edu/>

### Sample Code (hdMap)

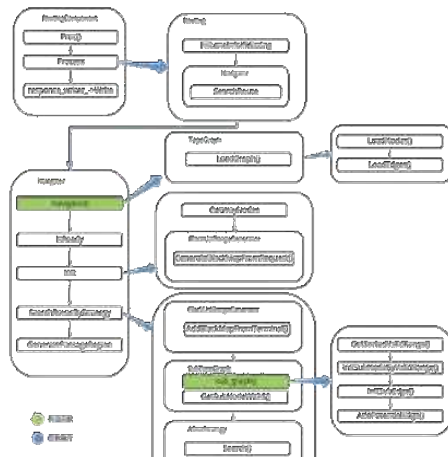
// Code snippet, featured as constructing a kdTree for efficient spatial data management, by iterating through each lane in the map, processing points from the central curve, start position, left and right boundaries of each lane; for each point, creating a kdTree Node, assigning it the corresponding lane ID, and inserting it into the kdTree, all facilitating efficient queries for spatial data related to lanes, such as finding the nearest points or lanes based on specific coordinates.

```
// Building a kdTree from lane data in the map
kdTree *mykdTree = new kdTree();
kdTreeNode *node;
for (int i = 0; i < map.lane_size(); i++) {
    const apollo::hdmap::Lane &lane = map.lane(i);
    // Process each point in the central curve of the lane
    for (int j = 0; j < lane.central_curve().segment(0).line_segment().point_size(); j++) {
        node = new kdTreeNode(lane.central_curve().segment(0).line_segment().point(j).x(),
                               lane.central_curve().segment(0).line_segment().point(j).y());
        node->setLane_id(lane.id().id()); // Set the lane ID for the point
        mykdTree->insert(node); // Insert the node into the kdTree
    }
    // Process start position of the lane
    node = new kdTreeNode(lane.central_curve().segment(0).start_position().x(),
                           lane.central_curve().segment(0).start_position().y());
    node->setLane_id(lane.id().id());
    mykdTree->insert(node);
    // Process left boundary of the lane
    for (int j = 0; j < lane.left_boundary().curve().segment(0).line_segment().point_size(); j++) {
        node = new
kdTreeNode(lane.left_boundary().curve().segment(0).line_segment().point(j).x(),

lane.left_boundary().curve().segment(0).line_segment().point(j).y());
        node->setLane_id(lane.id().id());
        mykdTree->insert(node);
    }
    // Process right boundary of the lane
    for (int j = 0; j < lane.right_boundary().curve().segment(0).line_segment().point_size(); j++) {
        node = new
kdTreeNode(lane.right_boundary().curve().segment(0).line_segment().point(j).x(),

lane.right_boundary().curve().segment(0).line_segment().point(j).y());
        node->setLane_id(lane.id().id());
        mykdTree->insert(node);
    }
}
```

### Sample Code (routing)



// Code snippet for the above flowchart, featured as first reading the routing\_map and initializing the Navigator class, then iterating through the routing\_request which comprises a series of points, and finding the nearest road for each point that is not on a road, while supplementing that such points are inaccessible, finally, calling "navigator\_ptr ->SearchRoute" to return routing response.

```
apollo::common::Status Routing::Init() {
    // Read the routing_map, which consists of nodes and edges
    const auto routing_map_file = apollo::hdmapi::RoutingMapFile();
    navigator_ptr_.reset(new Navigator(routing_map_file));
    // Read the map to find the nearest lane to the points in the routing request, and
    // return the corresponding lane id. This is akin to finding the nearest pickup point
    // in a residential area when hailing a taxi - in essence, finding the closest road.
    hdmapi_ = apollo::hdmapi::HDMapiUtil::BaseMapPtr();
}
```

// Subsequent execution of the "Process" main flow proceeds as follows:

```
bool Routing::Process(const std::shared_ptr<RoutingRequest>& routing_request,
                    RoutingResponse* const routing_response) {
    // Find the road closest to the nodes in the routing_request
    const auto& fixed_request = FillLaneInfoIfMissing(*routing_request);
    // Check if a planning path can be found
    if (!navigator_ptr_->SearchRoute(fixed_request, routing_response)) {
        monitor_logger_buffer_.WARN("Routing failed! " + routing_response->status().msg());
        return false;
    }
    monitor_logger_buffer_.INFO("Routing success!");
    return true;
}
```

## **UAV Machine Learning Aided Integrated Sensing & Communication (in MATLAB)**

**!!! Published a China National Patent on Jun 12, 2023; awarded as China National Innov. Project.**

[\[Paper\]](#) | [\[Slides\]](#) | [\[Code\]](#) | [\[Video\]](#)

This project starts Dec, 2021, featured as a Matlab-coded integrated sensing and communication (isac) project on SDR devices, developed as my research project for China Innovation & Entrepreneurship Program for College Students in Dec, 2022, mentored by [Prof. Xuanheng Li](#). Over 1 year, I focused on developing a unified system capable of executing communication and sensing tasks simultaneously, leveraging Matlab for programming and gnuradio for modular programming. My role extended to proposing an innovative method for UAV intelligent trajectory planning and resource allocation, using reinforcement learning (deep q-learning) to further demonstrate the effectiveness of our built platform.

### **The Problem (Part 1)**

The future of 6G systems relies on performing wireless communication and environmental sensing simultaneously, bridging gaps between the separation of the two modalities; such separation between sensing and communication mainly attributes to the differing requirements of the two (sensing looks for accuracy and resolution, while communication aims for max. of spectral efficient and data rates). With existing advancements of massive antenna arrays and high-frequency bands, my team noticed a pressing need to merge the two fields of functionalities into one single, efficient isac system. For more insights into the project, visit the repository [here](#).

### **The Problem (Part 2)**

After building up an isac platform resolving the problem, how to ensure efficient resource allocation and trajectory planning for UAVs in dynamic and complex isac networks is another key issue. For this, my team noticed an urging need for adaptable strategies in response to the changing isac demands.

### **The Goal**

For this 1-year development marathon, my team's solution was a novel and comprehensive system that can send and receive ofdm signals while detecting environmental changes using doppler radar at the same time, which fully utilized the inherent properties of the two modalities, achieving high data rates and accurate sensing, demonstrating the feasibility of integrated sensing and communication on a single platform, providing a foundation for enabling UAVs to autonomously and intelligently formulate optimal strategies for trajectory planning and resource allocation under varying environs.

### **Features**

1. Communication (OFDM): Uses Matlab to develop a robust ofdm-based communication system, incorporating frame synchron, frequency offset correction, channel equalization, while using signal processing for data modulation, demodulation, error correction, to ensure reliable data transmission.
2. ISAC: Integrates the communication and sensing systems on the Pluto-SDR platform.
3. Reinforcement Learning (DQN): Implements a deep q-learning algorithm for UAVs, enabling them to autonomously adapt trajectories and resource distribution based on real-time environmental data.
4. Trajectory Planning / Dynamic Resource Allocation: Allows drones to optimize their flight paths to balance communication needs and sensing accuracy; allows drones to allocate bandwidth and power resources dynamically, enhancing the overall efficiency of the isac network.

homepage: <https://vanhao5103233729.github.io/edu/>

5. Off/Online Learning: Incorporates the capability for drones to learn from historical or simulated data offline and refine strategies online in real environments.

6. Adaptive Strategy: Contributes to drones' ability to formulate, update complex coupling strategies for power and bandwidth allocation, adapting to the specific demands of the isac environment.

### Sample framework

Initialization:

- Set size of experience pool as  $G$ , size of mini-batch sampling as  $S$ , the update period as  $J$ , the discount factor as  $\gamma$ , the learning rate as  $\alpha$ , the greedy factor as  $\varepsilon=0.9$ , the parameters of the Q-main-network as  $\theta$ , the parameters of the Q-target-network as  $\hat{\theta}$ , the initial state as  $s_0=1$

Repeat:

- Formulate a shared strategy  $a_t$  based on the  $\varepsilon$ -greedy policy in the current state  $s_t$
- Move to the next state  $s_{t+1}$  and calculate the reward value  $\gamma_{t+1}$
- Collect experience tuple info: current state  $s_t$ , action  $a_t$ , next state  $s_{t+1}$ , reward value  $\gamma_{t+1}$ , then store this experience tuple info in the memory unit
- $t \leftarrow t+1$
- $\varepsilon \leftarrow 0.9 \times 0.1^{\frac{1}{10000}}$
- If  $\varepsilon \leq 0.1$   
 $\varepsilon = 0.1$
- If  $t > G$   
Update the experience tuple  
Sample randomly  $S$  numbers of experience tuples to train the network  
Calculate loss function  $L(\theta)$ , perform a gradient descent, update Q-main-network's  $\theta$   
If  $t - G \bmod J = 0$   
Copy Q-main-network's parameters  $\theta$  to Q-target-network's parameters  $\hat{\theta}$