

# Academic Presentation

## Introduction and Research Achievements

10.14.2024

plote1

# Table of Contents



1 Educational Background .....	3
1.a Personal Introduction .....	4
2 Projects and Competitions .....	5
2.a Design of a guide aid for the blind based on multi-modal perception and tactile feedback .....	6
2.b Liaoning Computer Chess Competition .....	7
2.c Rust-Based OS Development (OS-Comp 2023) .....	8
2.d C Language Compiler .....	9
2.e Web Scraping and Data Analysis - Patent Publishing .....	10
2.f Book reading and paper reproduction .....	11
2.g National Intelligent Vehicle Competition - Full Model Group .....	12
2.h Room Layout Generation Model - Pix2Pix Based .....	13
3 Internship Expectations .....	14
3.a Personal understanding of the project .....	15
3.b Personal idea .....	18

# **1 Educational Background**

## Personal Information

**Name:** plote  
**Institution:** bilibili University  
**Major:** math/sofeware/ai/cs  
**GPA:** 0

## Key Courses

Advanced Mathematics	Introduction to Programming
Linear Algebra	Probability and Statistics
Computer Organization Principles	Object-Oriented Programming

## Technical Skills

**Programming Languages:** Python, Rust, C/C++

## **2 Projects and Competitions**

# Design of a guide aid for the blind based on multi-modal perception and tactile feedback

## 比赛奖项

研究背景:针对现有导盲辅具视野狭窄、手部占用等问题。研发一种基于多模态感知与触觉反馈结合的便捷式盲人辅助设备,为视障用户提供导航、避障和物体识别等功能。

研究内容:使用深度相机和麦克风阵列等传感器,利用卡尔曼滤波融合数据,减小误差

使用 Paddle 训练 PP-YOLOE 模型进行目标检测,线上评测识别率 0.997, FPS 大于 60。

ROS 环境下使用 ORB-SLAM 通过点云建图室内导航 FOC 控制无刷电机驱动动量轮提供触觉反馈

## 项目成果

2023 年已评优结题并推荐参加年会

无

无



# Liaoning Computer Chess Competition



水赛打着玩

开发并优化六子棋游戏算法，引入 $\alpha - \beta$ 剪枝技术改进传统极大极小值算法，显著提高计算机对弈效率和决策质量。

**Minimax Algorithm Optimization:** 设计并实现了基于极大极小过程的决策树，通过模拟棋手思考过程，为每种可能的走法进行评分和选择，确保在对手最佳反应下依然保持优势。通过构建深度搜索树，自底向上分析并确定最佳走法，确保稳定的棋力表现。

**Alpha-Beta Pruning Implementation:** 引入 $\alpha - \beta$ 剪枝算法减少不必要的搜索路径，处理深度搜索过程中的组合爆炸问题，显著提高搜索效率。在深度优先搜索中动态调整节点生成，利用 $\alpha$ 剪枝和 $\beta$ 剪枝规则剪除低效分支，从而在保持决策质量的同时大幅减少计算需求。

## Tsinghua University Open-Source Operating System

**System Architecture and Debugging:** 通过 `make debug` 命令使用 GDB 调试 `Qemu`，掌握代码结构和执行流程。实现了基于 `RISC-V` 特权级的简单批处理系统，使得应用程序和用户态支持库能在最低特权级 `U` 模式下运行。

**Basic Output and Control:** 通过实现 `println!` 宏和 `console_putchar` 函数，使得格式化输出在自己的库级操作系统中可用。这包括完整的 `Stdout` 结构体和 `Write trait` 的实现，提供基本的输出功能。

**Task and Privilege Management:** 设计和实现了从操作系统的 `S` 模式到应用程序的 `U` 模式的上下文切换。这包括应用程序的内存布局、系统调用的处理，以及在发生 `Trap` 时 CPU 特权级的管理和切换。

**SBI-Based Output and Shutdown:** 实现了基于 `SBI (Supervisor Binary Interface)` 的输出和关机功能，增强了系统的基本输入输出和控制能力。

**Multi-Tasking:** 实现了包括 `yield` 系统调用和时间片轮转调度的多任务处理能力，支持多个任务并发执行并合理分配 CPU 资源。



## Compiler and Compiler Principles

为了做这个项目,学习了编译原理并且阅读了以下书籍

**Compilers: Principles, Techniques, and Tools**

**Advanced Compiler Design and Implementation**

**Modern Compiler Implementation in C.**

**Lexical Analysis:** 实现了基于有限自动机的词法分析器。为了高效处理输入流,我采用了二重指针和缓冲区管理策略来代替 `lex` 工具使用,从而实现了快速的文本扫描和词法单元识别。

**Syntax Analysis and AST:** 使用 `Yacc` 工具进行语法分析,自动构建抽象语法树 (AST)。这一步是将词法单元按照语言的语法规则结构化,为后续的语义分析和代码生成打下基础。

**AST Visualization:** 利用 Python 开发了一个遍历并可视化 AST 的工具,通过生成图形化的树结构,使代码逻辑结构一目了然,便于调试和分析。

# Web Scraping and Data Analysis - Patent Publishing



## 实习生

**Data Acquisition:** 利用 Python 的 `Requests` 和 `Scrapy` 库结合 `XPath` 定位技术，合理合法爬取国家专利统计局开放数据。通过爬虫任务，处理和抓取网页数据。

**Skill Enhancement:** 在实习期间，学习数据预处理、清洗与转换技术，应用 `Pandas` 库进行数据处理，使用 `NumPy` 进行数值分析。

**Patent Analysis:** 针对数据，运用 `Beautiful Soup` 解析 HTML 文件，抽取关键信息，并利用 `Matplotlib` 和 `Seaborn` 等可视化工具制作图表，展示专利数据的趋势和模式，为管理决策提供了数据支持。

学习过的领域: 深度学习模型、**cv** 目标检测、生成式模型、扩散模型

## Book reading

动手学深度学习 - 李沐

统计学习方法 - 李航

机器学习花书 (*The Elements of Statistical Learning*)

*PRML* (Pattern Recognition and Machine Learning)  
*Pattern Classification*

## paper reproduction

YOLO 系列

VAE 与 GAN

Diffusion 模型

Stable Diffusion

DDPM (Denoising Diffusion Probabilistic Models)

RDDM (Refinement-based Diffusion Models)

# National Intelligent Vehicle Competition - Full Model Group



## 深度学习在线赛全国第 28 名

### 1. Data preprocessing and enhancement

- 数据格式转换：VOC 至 COCO 格式的系统性转换
- 自主开发图像增强算法：实现图像翻转、随机裁剪等技术，提高模型输入多样性
- 数据集均衡化：优化 JSON 文件中数据分布，提升数据质量

### 2. Data quality control

- 实施严格的数据清洗流程，确保标签准确性和图像质量
- 利用 PaddleLabel 进行实地数据采集和自动预标注，扩充数据集

### 3. Model optimization and selection

- 应用 Grid 搜索算法，优化模型结构和参数组合
- 实施数据蒸馏技术，提炼核心特征，加速学习过程
- 综合评估多种框架，选定 PP-YOLOE 结合 ResNet 作为主要检测框架

### 4. Model performance improvement strategies

- 实施模型剪枝技术，优化网络结构，提高计算效率
- 采用 Adam 优化器，精细调整学习率等超参数，确保模型快速收敛
- 应用先进剪枝策略，有效降低计算资源消耗，提升预测速度

### 5. Research results

- 模型性能：F1 得分达到 0.99704
- 实时性能：帧率提升至 42.85 FPS
- 显著超越前期版本，实现性能质的飞跃

# Room Layout Generation Model - Pix2Pix Based



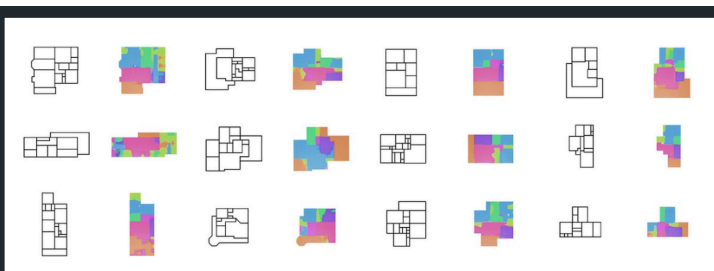
## Internship at Chinese Academy of Sciences, Institute of Automation

### Conditional Generative Adversarial

**Network (CGAN):** Based on Pix2Pix framework with U-Net generator structure, leveraging down-sampling and up-sampling with skip connections to retain image details.

**Generator:** Built a U-Net model, consisting of convolution, batch normalization, Leaky ReLU, and dropout layers to strengthen feature extraction and image generation.

**Discriminator:** Used PatchGAN to evaluate image authenticity at patch level, assisting the generator in optimizing local features.



### Dataset

I used the [CVC-FP dataset](#) for training. After downloading the dataset, the files should be placed in the `ImagesGT` folder. During the preprocessing and training phases, several issues were identified and resolved:

- ❑ **SVG Rendering Issues:** A large portion of the SVG annotations could not be rendered in browsers or other viewing tools due to corrupted code.
- ❑ **Room Classification:** The original SVG files did not explicitly classify room types, using a generic "room" label, which complicated the differentiation of room types.
- ❑ **File Naming Mismatch:** Some images and their corresponding SVG files had mismatched names or extensions.
- ❑ **Unmatched SVGs:** There were extraneous images without corresponding SVG files.
- ❑ **Duplicate Images:** Certain images were duplicated and unusable.
- ❑ **Inconsistent Color Codes:** Color codes in the SVG annotations varied across different files.

## **3 Internship Expectations**

# Personal understanding of the project



我自己对生成学习的了解

**Diffusion** 中的核心公式是:

正向扩散 **Forward Diffusion Process**

$$q(x_1, \dots, x_T \mid x_0) = \prod_{t=1}^T q(x_t \mid x_{t-1})$$

反向生成 **Reverse Generation Formula**

$$p_{\theta}(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \Sigma_{\theta}(x_t, t))$$

正向和反向扩散的平衡控制

$$\mathbf{x}_{t-1} = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon$$

## 论文的核心知识点

### Latent Diffusion Model

核心思想：在潜在空间中进行扩散和去噪

优势：降低计算成本，加速生成过程

**LDM 损失函数：**

$$L_{LDM} = E_{z, \epsilon, t} \left[ \left\| \epsilon - \epsilon_{\theta}(z_t, t, y) \right\|^2 \right]$$

因为潜在空间的维度通常比数据空间低。  
LDM 的这个损失函数与传统扩散模型的损失函数类似，但它在潜在空间中操作，使用的是压缩后的潜在表示  $z_t$  进行计算，从而加速了生成过程。

### 残差引导扩散模型(RDDM)

引入残差学习 (ResNet) 改善模型性能

**RDDM 更新公式：**

$$\begin{aligned} I_t &= I_{t-1} + \alpha_t I_{res} + \beta_t \epsilon_{t-1} \\ &= I_{t-2} + (\alpha_{t-1} + \alpha_t) I_{res} + \left( \sqrt{\beta_{t-1}^2 + \beta_t^2} \right) \epsilon_{t-2} \\ &= \dots \\ &= I_0 + \bar{\alpha}_t I_{res} + \bar{\beta}_t \epsilon, \\ \frac{I_{t-1}}{\sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}} &= \frac{I_t}{\sqrt{\bar{\beta}_t^2 - \sigma_t^2}} - (\bar{\alpha}_t - \bar{\alpha}_{t-1}) \frac{I_{res}}{\sqrt{\bar{\beta}_t^2 - \sigma_t^2}} - \frac{\bar{\beta}_t}{\sqrt{\bar{\beta}_t^2 - \sigma_t^2}} \epsilon_t, \text{ where } \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{aligned}$$



# Personal understanding of the project (iii)



## 最近新学习的论文和自己的想法

### 风格迁移中的自注意力机制

自注意力和交叉注意力机制在风格迁移中起关键作用

自注意力计算公式:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

相关研究:

Style Injection in Diffusion: “A Training-free Approach for Adapting Large-scale Diffusion Models for Style Transfer” (CVPR 2024)

### 感知损失函数(Perceptual Loss)

传统均方误差损失 (MSE) 的改进

结合感知损失和风格损失

通常基于预训练的卷积神经网络 (如 VGG)

感知损失函数:

$$L_{\text{perc}}(x, \hat{x}) = \sum_l \|\varphi_{l(x)} - \varphi_{l(\hat{x})}\|_2$$

$\varphi_l$  表示预训练网络的第  $l$  层特征提取器

相关研究: Image Style Transfer Using Convolutional Neural Networks 2016  
Diffusion Model with Perceptual Loss 2023.12

## 我的个人 idea

改进损失函数和模型结构来优化噪声处理。

- ## 利用 Attention 机制增强控制性减少风格迁移中的不协调现象

优化噪声分布与处理：通过改进噪声分布(自适应噪声处理)减少低频信息残留对生成图像的影响。

- The diagram illustrates the Diffusion in Style process. It shows a sequence of images from  $x_0$  to  $x_{1000}$ , where  $x_0$  is the original image and  $x_{1000}$  is the generated image. The process involves adding noise ( $\epsilon_t$ ) and removing noise ( $\epsilon_t$ ) at each step. The diagram also shows the style adaptation process, where the original image is adapted to a specific style (e.g., anime sketches) using a style-specific noise distribution  $N(\mu_{(t)_{style}}, \sigma_{(t)_{style}}^2)$  instead of the default  $N(\mu_{(t)}, \sigma_{(t)}^2)$ .