1. **选题目的**

在肺炎防治需求紧迫的当下，肺炎决策支持系统依托大语言模型与图像识别技术，构建智能医疗辅助体系，助力医疗决策。​

大语言模型作为 “智慧大脑”，深度剖析医疗文献、病历等文本数据，梳理肺炎流行病学特征与症状，跟踪前沿诊疗动态，为个性化治疗提供数据支撑。图像识别技术化身 “火眼金睛”，通过深度学习算法精准识别胸部影像中的病变特征，量化分析病灶变化，提升诊断效率与准确性。​

系统融合两大技术，整合权威医疗文献，经智能算法综合分析。诊断时，依据症状与影像定位病原体；治疗中，评估方案风险与疗效；防控上，监测病例预测传播趋势，为肺炎防治提供科学解决方案，推动诊疗水平升级。

1. **决策目标**
2. 用户描述自身的身体状况，包括但不限于以下内容：主要症状（如发热、咳嗽、乏力等）,症状开始的时间,体温（如果发热）,是否有基础疾病（如高血压、糖尿病等）,近期旅行史或接触史等。大模型通过结合WHO指南、最新科研文献和患者提供的描述来给出合理的决策如：初步判断，居家观察建议,用药建议,就医指导。
3. 用户选择肺炎图像，系统通过神经网络和YOLO图像识别来给出根据肺炎图像患有肺炎的概率以及发生肺炎的位置。
4. 用户填写问卷表单，大模型根据表单填写结果，通过查询知识库内容来判断用户患有肺炎的风险，并给出原因以及相关建议。
5. **组内成员及工作分工**
6. **运行环境**

操作系统版本 Windows 11 家庭中文版 22H2

处理器 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz

机带 RAM 32.0 GB (31.6 GB 可用)

系统类型 64 位操作系统, 基于 x64 的处理器

开发平台 pycharm-professional-2023.1

Python版本 3.9

CUDA版本 12.8

Pytorch版本 2.7.0+cu128

1. **系统介绍**
   1. 前端介绍

本项目的前端采用Uniapp跨平台开发框架，开发基于Vue3+ts+scss的肺炎决策支持系统。本项目前已开发了web端，支持用户登录注册，获取肺炎医学决策支持，肺炎图像识别等功能。该系统使用pinia插件进行持久化存储，保障用户数据持久化存储。

接下来展示本系统的前端页面和功能介绍。

* + 1. 登录注册页面

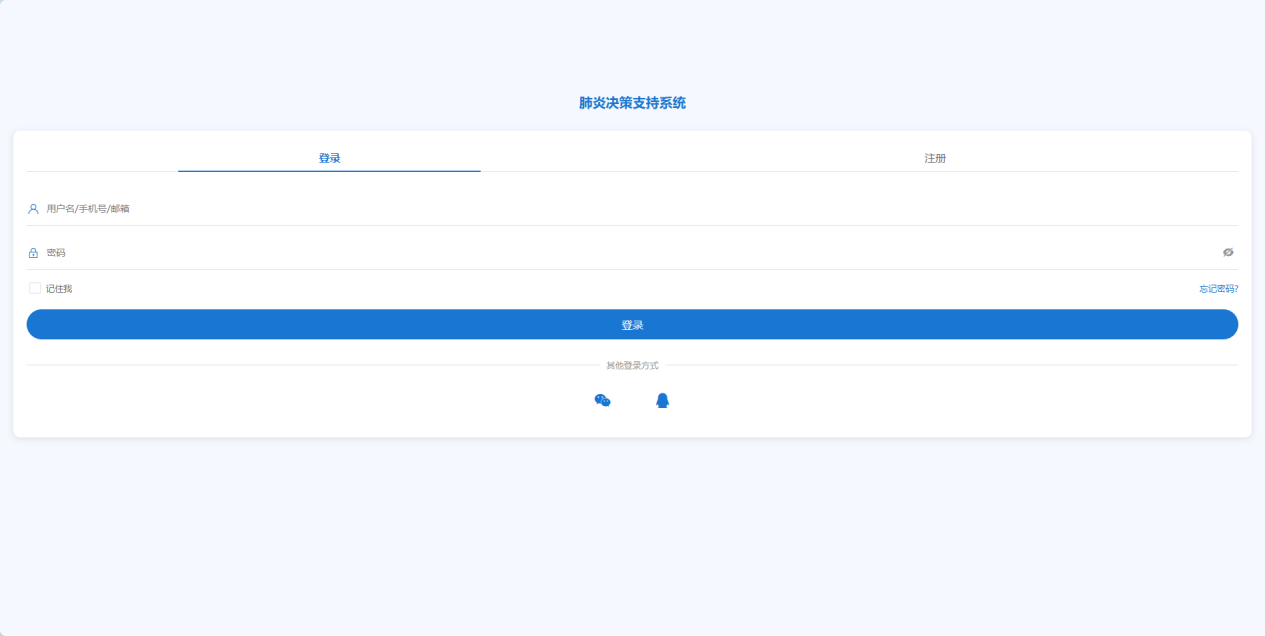


图 1登录界面

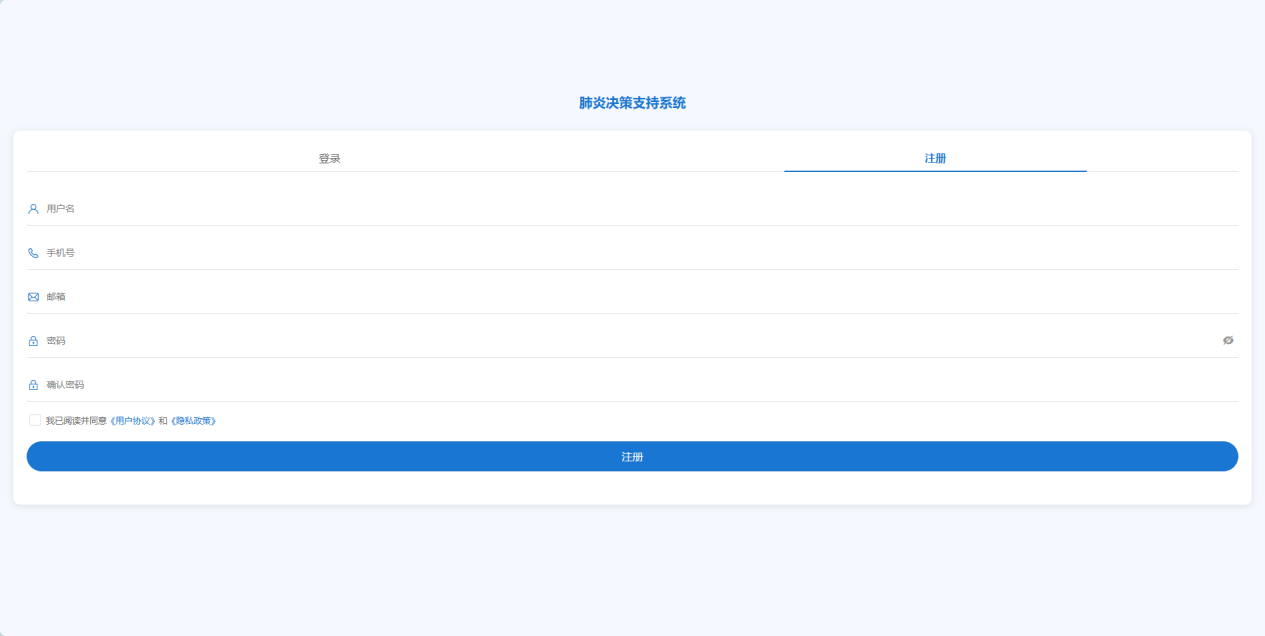


图 2注册页面

在登录注册页面，如图 1和图 2，用户可以通过用户名/手机号/邮箱进行注册，然后使用账号+密码即可登入系统。在登录页勾选了“记住我”时，浏览器会自动存储用户的登录信息，保障下次用户再次进入系统时无需再次出入账号密码。

* + 1. 大模型对话页面

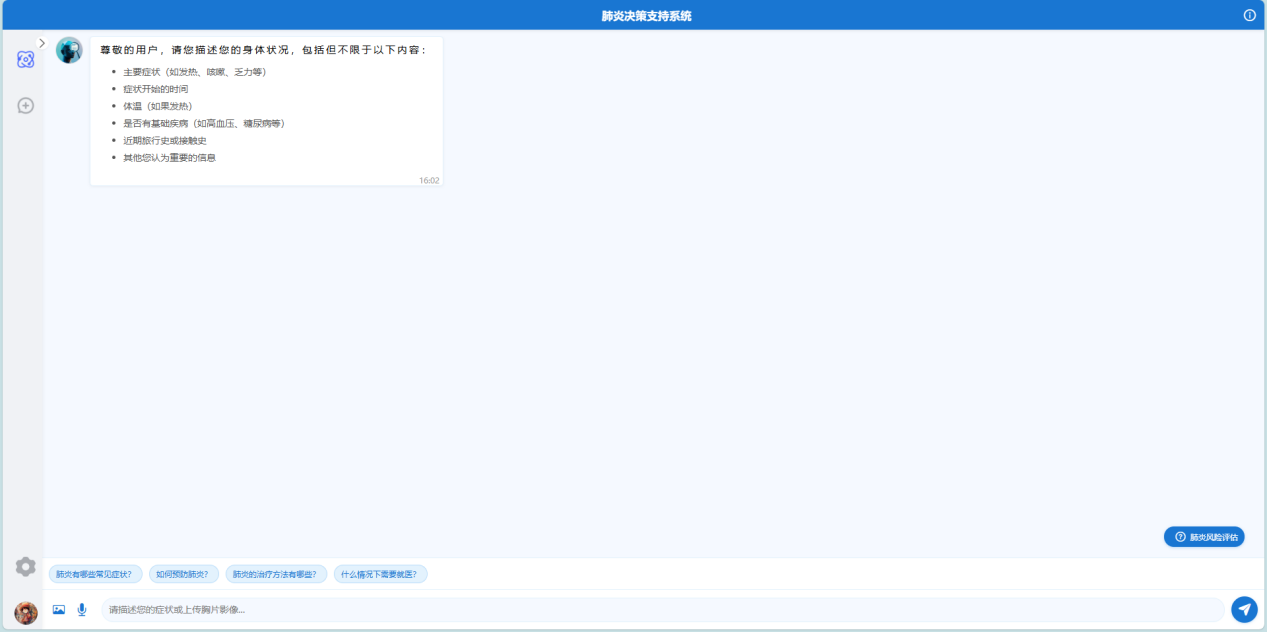


图 3大模型对话页面

该页面主要由两部分组成，侧边栏和对话栏，侧边栏默认是收缩状态，点击侧边栏的logo图标或箭头图标，或者点击用户头像，都可以唤出侧边栏。

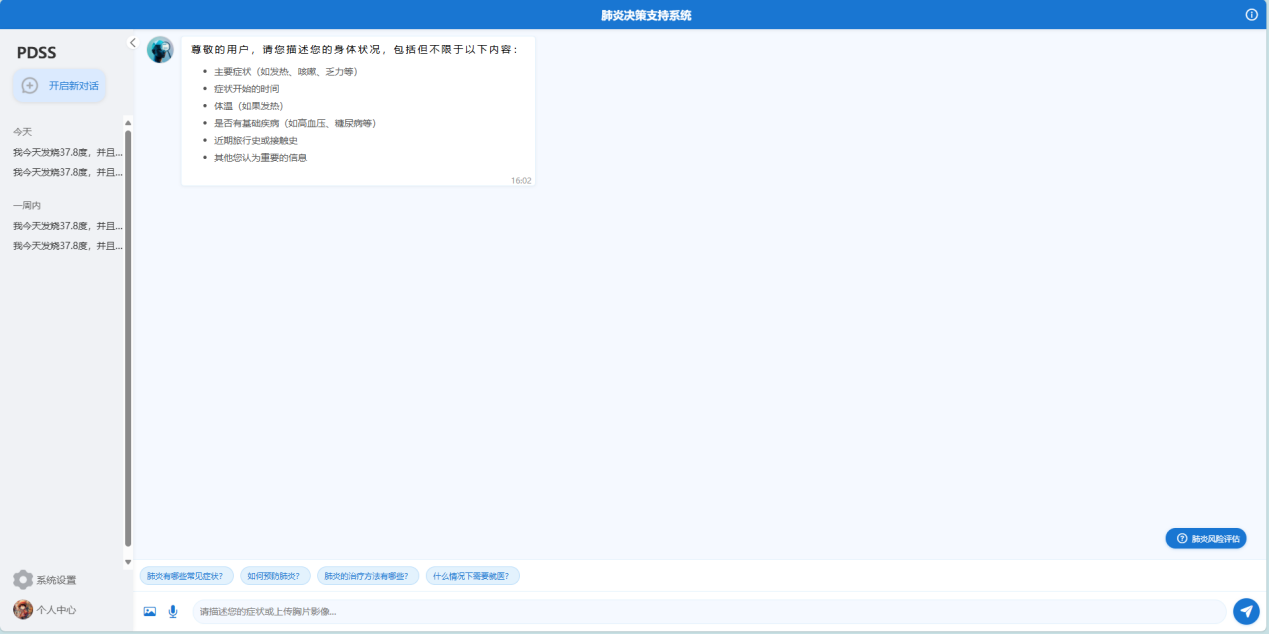


图 4侧边栏唤出

如图 4所示，侧边栏唤出后，点击开启新对话按钮，系统会自动保存上一次的对话到历史对话记录中，同时清空右边的对话记录，开启新对话。点击下方的用户头像可以唤出菜单栏。如图 5所示，点击个人信息可以查看个人信息，点击退出登录可以清空用户信息并返回至登录页面。



图 5个人信息菜单栏

用户可以根据大模型给出的提示信息在输入框中输入自己的身体状况信息，大模型会根据prompt模板和内置的知识库来给出合理的决策支持。如图 6所示，大模型输出为流式输出，显著提高响应速度和用户体验，同时支持markdown语法，使回答结构更加清晰。用户还可以点击输入框上方的快速问题，快速发送问题给大模型获取回答。



图 6大模型决策支持

* + 1. 肺炎图像识别功能

本系统支持用户上传肺部x光图像，系统会自动识别图像，并给出患有肺炎的概率以及使用yolov11算法绘制出可能患有肺炎的位置。如图 7所示。

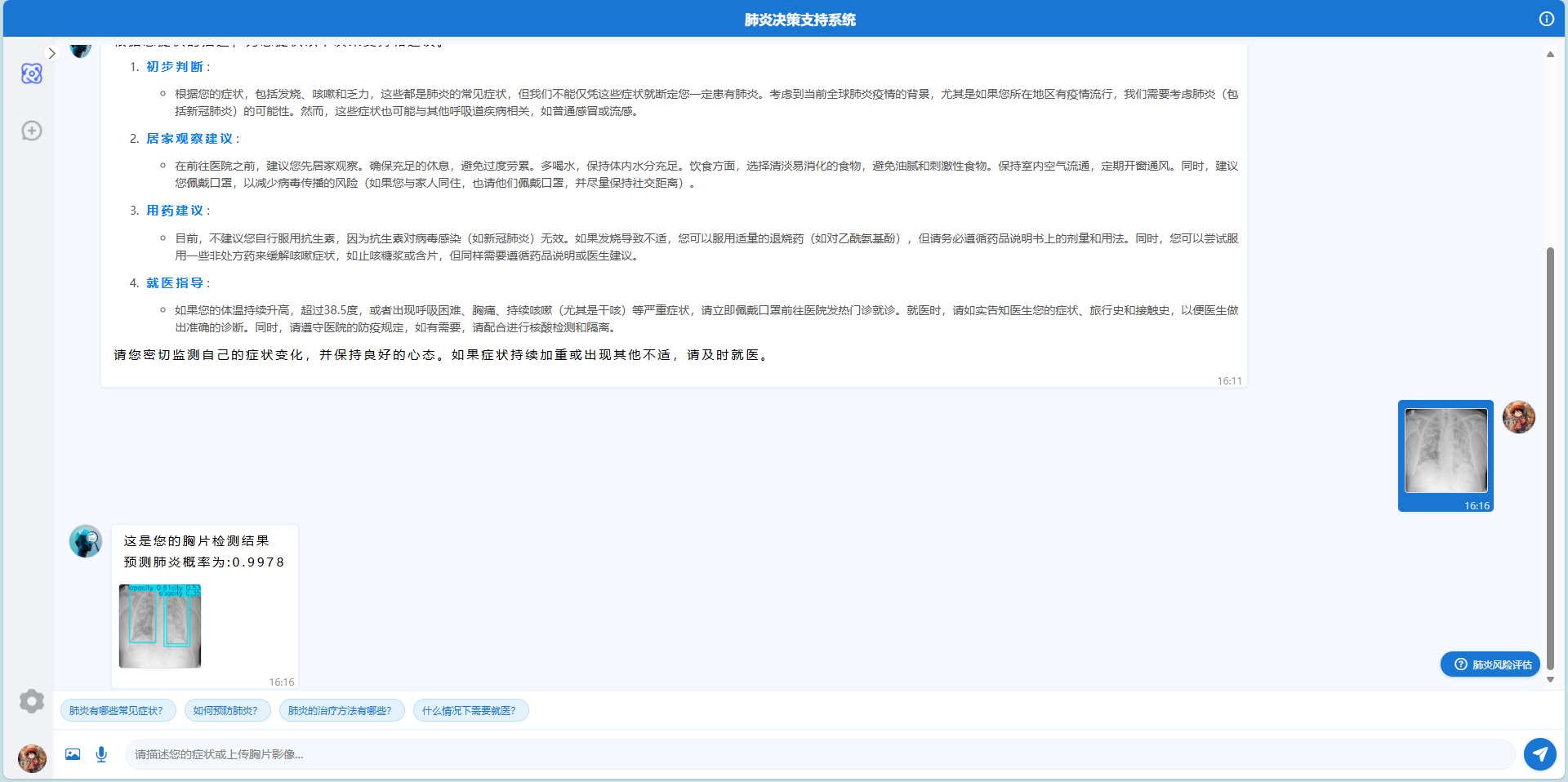


图 7肺炎图像识别

* + 1. 肺炎风险评估功能

点击右下角的肺炎风险评估，会弹出一个问卷调查，用户根据自身的实际情况填写问卷，点击提交后，大模型会根据知识库内容输出结构化数据：肺炎风险等级、判断信息、建议。如图 8所示。

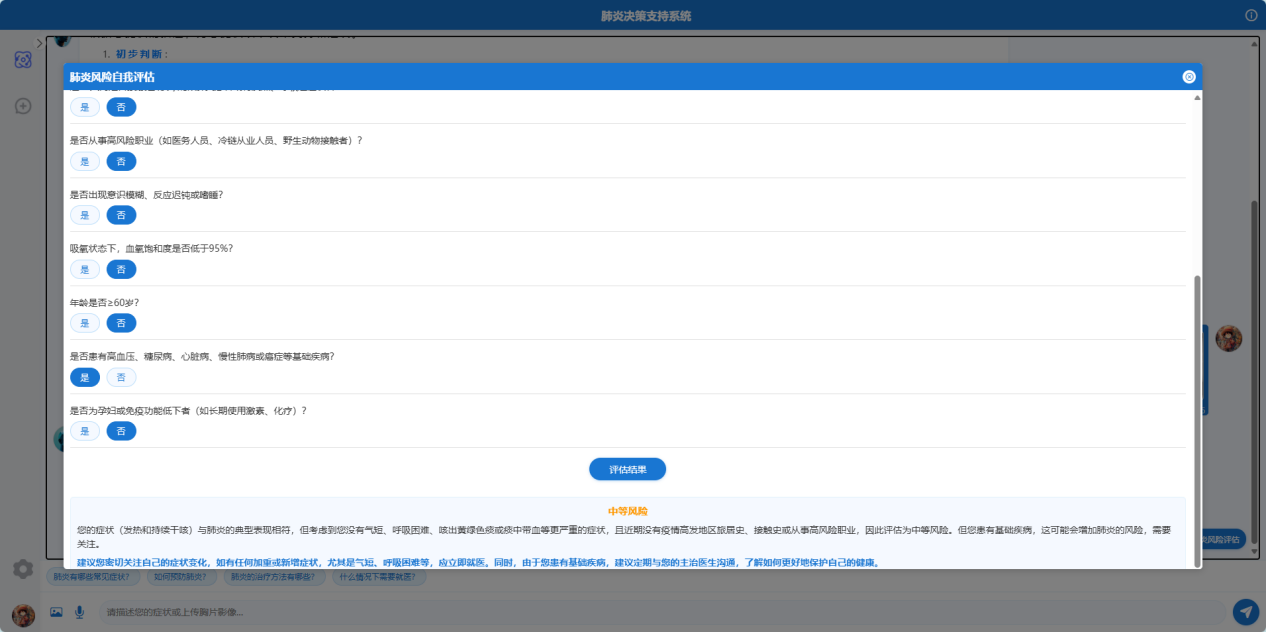


图 8肺炎风险评估

* 1. 后端介绍

后端使用springboot2开发，使用lombok插件简化开发流程。数据库使用MySql，使用jpa插件来对数据库进行curd操作。后端开发的接口如所示。



|  |  |  |  |
| --- | --- | --- | --- |
| 功能描述 | 请求方法 | URI | 备注 |
| 上传图像 | POST | /api/file/upload | 上传图像并保存到阿里云oss |
| 请求知识库 | POST | /knowledge/query | 获取知识库内容 |
| 流式获取决策支持 | GET | /ai/decisionMakingStream | 流式调用大模型获取决策支持 |
| 获取propmpt模板 | GET | /prompt/getPrompt | 获取大模型的prompt模板 |
| 非流式获取决策支持 | POST | /ai/decisionMaking | 非流式调用大模型获取决策支持 |
| 用户注册 | POST | /auth/register | 用户注册系统账号 |
| 用户登录 | POST | /auth/login | 用户登录系统 |

* 1. 算法介绍

5.3.1 yolo

算法定位：

YOLO11是Ultralytics团队于2024年推出的新一代目标检测模型，融合了YOLOv9和YOLOv10的突破性设计，在精度、速度、多任务支持三大维度实现飞跃。有着更高精度与更低参数，跨环境适配性（支持边缘设备、云平台及NVIDIA GPU部署，兼顾灵活性与效率），全任务覆盖（目标检测、实例分割、姿态估计、图像分类、旋转目标检测OBB五大视觉任务）的优点。

算法原理：

YOLO（You Only Look Once）是一种单阶段目标检测模型，其核心原理是将输入图像划分为网格，每个网格直接通过单次前向传播同时预测边界框（中心坐标、宽高）、置信度（含目标的概率与预测框准确性）和类别概率，无需生成候选区域。它通过Anchor机制优化框的定位，使用多尺度特征融合增强检测能力，并依赖非极大值抑制（NMS）消除冗余框，最终实现端到端的高效目标检测，平衡速度与精度，适合实时应用。

常见应用：

医疗影像：病灶区域检测；

工业检测：PCB板缺陷识别、安全帽佩戴监测；

自动驾驶：实时行人/车辆多目标跟踪。

5.3.2 EfficientNet网络模型

EfficientNet模型，如图 9中的compound scaling所示，是一种创新的卷积神经网络架构，它在可扩展网络设计领域引领了一种新的范式。该模型的创新之处在于它采用了一种称为复合缩放的系统化网络缩放方法。这种方法与传统的网络缩放做法不同，传统方法通常仅增加网络的深度或宽度，而EfficientNet通过固定的缩放系数，均衡地扩展了网络的深度、宽度以及输入图像的分辨率。

在本次实验中，选择了EfficientNets系列中的基础网络模型EfficientNet-B0作为分析对象。EfficientNet-B0是一种采用复合缩放方法设计的高效卷积神经网络。EfficientNet的复合缩放方法能够产生一系列模型，从EfficintNet-B0到B7不等，这些模型提供了一系列的能力，实现了计算效率和预测准确性之间的和谐平衡。

EfficientNet-B0，是通过自动化的神经架构搜索过程在预定义的资源限制下优化得到的，旨在效率和准确性之间找到最佳平衡。EfficientNet-B0的核心结构是移动翻转瓶颈卷积（Mobile Inverted Bottleneck Convolution, MBConv）模块，该模块融合了压缩与激发网络（Squeeze-and-Excitation Network, SENet）的注意力机制。SENet在提出时在 ImageNet 数据集上达到了当时最高的准确率，突显了其有效性。移动翻转瓶颈卷积模块也是通过神经网络架构搜索得到的，其结构与深度分离卷积类似。

在 MBConv 模块中，如图 10 所示，首先进行1×1的逐点卷积，根据扩展比例调整输出通道维度。随后执行k ×k 的深度卷积。如果需要引入压缩与激发操作（SE 模块），则该操作在深度卷积之后进行。模块的最后部分是另一个1×1的逐点卷积，用以恢复到原始的通道维度。此外，MBConv 模块还融入了连接失活和输入的跳跃连接，不仅有效地缩短了训练时间，还提升了模型的整体性能。

EfficientNet-B0 的结构包括 16 个移动翻转瓶颈卷积（MBConv）模块、2 个卷积层、1 个全局平均池化层和 1 个分类层。在其结构示意图中，图 4-3 展示了这些组件，其中不同的颜色代表了网络中的不同阶段，以便于区分和理解各个部分的功能和组织方式。在多个标准图像识别基准测试中，EfficientNet 展现了显著的性能，尤其是在 ImageNet 数据集上实现了最先进的准确率，同时保持了较低的计算复杂度和参数数量。这种效率使得 EfficientNet 非常适合广泛的应用和设备，从计算能力有限的移动设备到高性能计算环境。此外，EfficientNet-B0 的成功也激励了后续研究者在保持模型性能的同时，进一步探索模型效率的提升空间，推动了深度学习领域向更高效、更智能的方向发展。通过对 EfficientNet-B0 及其衍生版本的持续优化和应用，我们可以期待在图像处理、模式识别、机器视觉等领域实现更加精准和高效的解决方案，从而为实际应用带来革命性的改进。

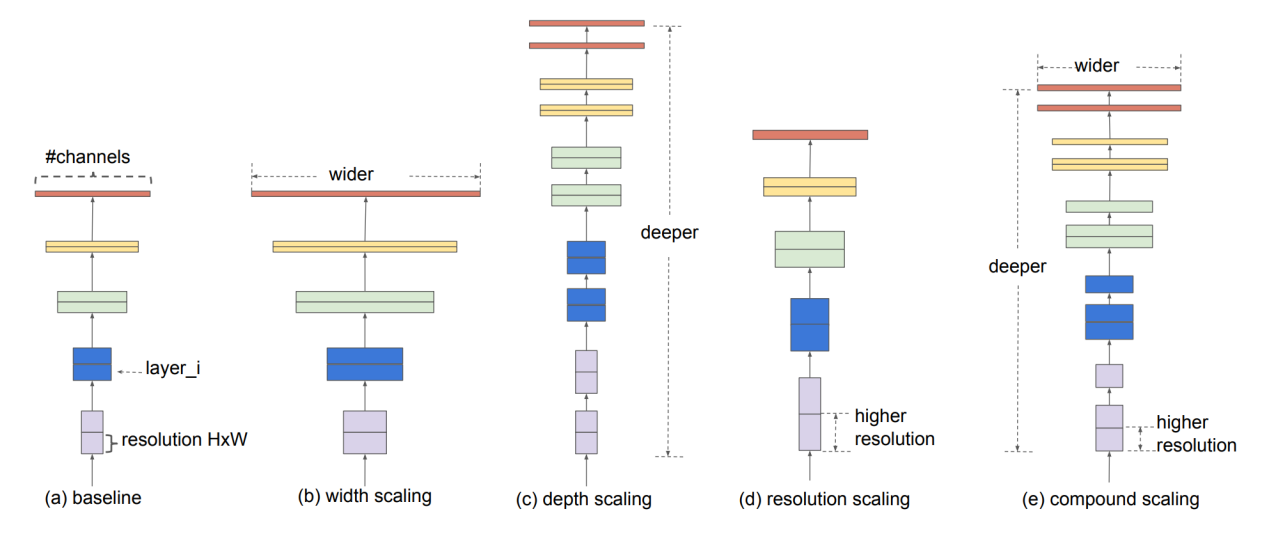


图 9 Model Scaling.

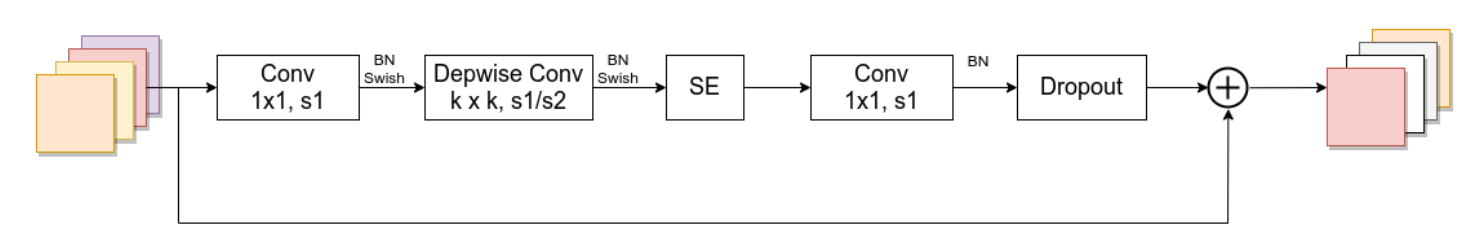


图 10 MBConv模块

1. **实验过程**
   1. 环境准备

在已有的环境上增加说明

6.1.1 硬件环境

GPU 3070Laptop

内存容量 32GB

6.1.2 深度学习框架

PyTorch 2.7.0 + Ultralytics YOLO11

通过git clone从Ultralytics官方仓库克隆获取最新代码并安装依赖

repo\_url = "https://github.com/ultralytics/ultralytics"

if not os.path.exists("ultralytics"):

!git clone {repo\_url}

* 1. 数据集准备

6.2.1 数据来源

通过以下链接 [SIIM-FISABIO-RSNA COVID-19 Detection | Kaggle](https://www.kaggle.com/competitions/siim-covid19-detection/data) 获取的SIIM COVID-19胸部X光数据集，内容为关于COVID-19胸部X光影像专业的医学影像文件及标注文件

6.2.2 数据预处理

图像处理：

统一缩放至256×256像素

路径标准化

标签处理：

解析原始标签中的边界框坐标

根据图像缩放比例调整边界框

转换为YOLO格式（中心坐标+宽高归一化）

数据划分：

按8:2比例划分训练集/验证集

分层抽样

* 1. 模型训练

模型选择：yolo11n.pt

输入尺寸：256\*256

超参数：

Batch Size-16

Epochs-100

优化器-默认SGD（动量0.937）

学习率-默认0.01（未显式设置则省略）

数据增强=默认YOLOv8增强策略（翻转、色彩抖动等）

实验监控-通过wandb记录损失曲线、mAP等指标

!yolo detect train \

data=tmp/ultralytics/data/data.yaml \

imgsz=256 \

batch=16 \

epochs=100 \

model=yolo11n.pt \

project=siim-covid-detection \

name=yolo11n-256 \

save\_period=1

6.4.模型推理与评估

6.4.1 测试集预测

TEST\_PATH = Path('C:/Users/hxj/PycharmProjects/pythonProject/data/siim-covid19-resized-to-256px-jpg/test/')

MODEL\_PATH = Path('C:\\Users\hxj\PycharmProjects\\pythonProject\\covid-work\\siim-covid-detection\\yolo11n-256\\weights\\best.pt')

IMG\_SIZE = 256

CONF\_THRES = 0.281

IOU\_THRES = 0.5

MAX\_DET = 3

model = YOLO(MODEL\_PATH)

results = model.predict(

source=TEST\_PATH,

imgsz=IMG\_SIZE,

conf=CONF\_THRES,

iou=IOU\_THRES,

max\_det=MAX\_DET,

save\_txt=True,

save\_conf=True,

project='runs/detect',

name='exp',

exist\_ok=True

)

6.4.2 模型评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | mAP50-95 | mAP50 | 精确率 (P) | 推理速度 (ms/img) |
| ​​YOLOv8s​​ | 0.342 | 0.678 | 0.844 | 1.3 |
| ​​YOLOv8m​​ | 0.333 | 0.653 | 0.824 | 2.3 |
| ​​YOLOv11n​​ | 0.356 | 0.720 | 0.847 | 2.9 |

YOLOv11n在保持较高推理效率（2.9ms/img）的同时，实现较YOLOv8s提升1.4% mAP50-95，精确率提高0.3%，更符合需求。

1. **项目总结**

本实验在配备RTX 3070laptop GPU的硬件环境下，基于PyTorch和Ultralytics YOLOv11框架，对SIIM COVID-19胸部X光数据集进行预处理（统一缩放至256×256像素并转换为YOLO格式）后，采用预训练模型yolo11n.pt进行100轮训练，最终模型在测试集上达到0.356 mAP50-95和0.847精确率，较YOLOv8s提升1.4% mAP50-95且保持2.9ms/img的推理速度，验证了其在医学影像检测任务中的高效性与准确性。