



# CNN食物图像分类的优化

汇报人：2226114139 杨思涵

# 任务概述

---

《人工智能导论》创新设计

## 任务概述

初期方案

方案优化 – 数据增强与杂项

方案优化 – 半监督学习

方案优化 – ResNet50

总结

# Task:

食物图像识别分类问题:

数据集:

<https://www.kaggle.com/datasets/trolukovich/food11-image-dataset>

- Training set:
  - 280 \* 11 labeled images
  - 6786 unlabeled images
- Validation set: 60 \* 11 labeled images
- Testing set: 3347 images

0. Bread	
1. Dairy product	6. Noodles-Pasta
2. Dessert	7. Rice
3. Egg	8. Seafood
4. Fried food	9. Soup
5. Meat	10. Vegetable-Fruit

# Task:

食物图像识别分类问题:

**INPUT**



我是谁?

# Task:

食物图像识别分类问题:

**INPUT**



模型

**OUTPUT**

8.  
Seafood



# 初期方案

---

《人工智能导论》创新设计

任务概述

初期方案

方案优化 – 数据增强与杂项

方案优化 – 半监督学习

方案优化 – ResNet50

总结

# 初期方案：数据集导入

直接导入“图片-标签”数据集



```
train_tfm = transforms.Compose([
    transforms.CenterCrop(224),
    transforms.ToTensor(),
])

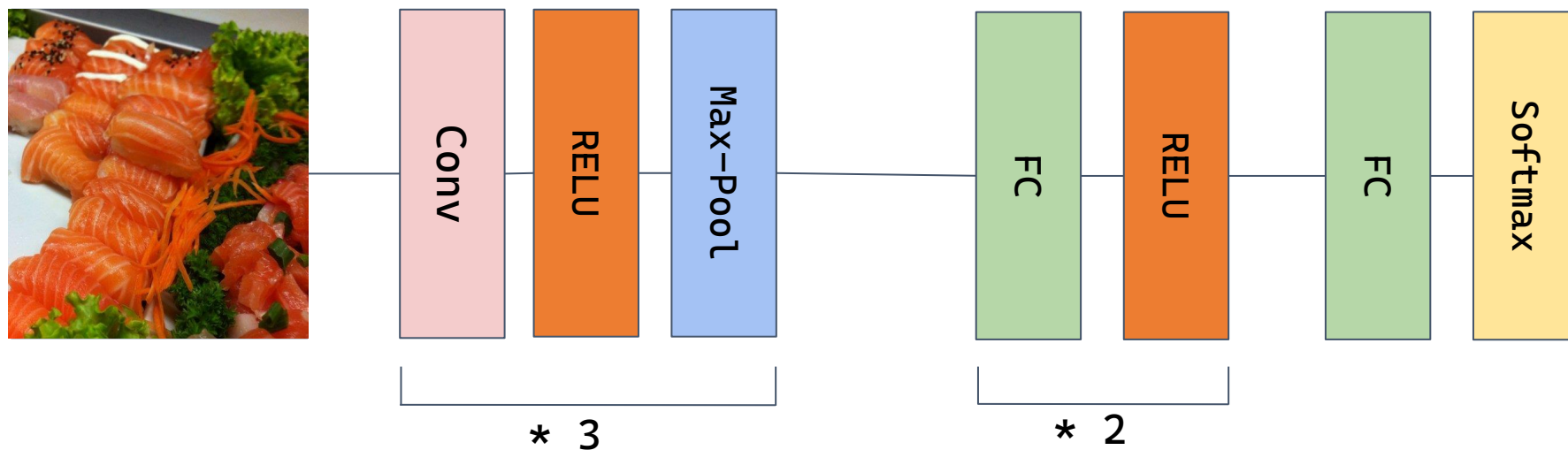
test_tfm = transforms.Compose([
    transforms.CenterCrop(224),
    transforms.ToTensor(),
])

train_set = DatasetFolder("food-11/training/labeled", loader=lambda x: Image.open(x), extensions="jpg",
transform=train_tfm)
valid_set = DatasetFolder("food-11/validation", loader=lambda x: Image.open(x), extensions="jpg", transform=test_tfm)
test_set = DatasetFolder("food-11/testing", loader=lambda x: Image.open(x), extensions="jpg", transform=test_tfm)

train_loader = DataLoader(train_set, batch_size=batch_size, shuffle=True, pin_memory=True)
valid_loader = DataLoader(valid_set, batch_size=batch_size, shuffle=True, pin_memory=True)
test_loader = DataLoader(test_set, batch_size=batch_size, shuffle=False)
```

# 初期方案：模型构建

简单的 CNN 架构





# 初期方案： 训练参数

用基础的训练参数



```
device = "cuda" if torch.cuda.is_available() else "cpu"
```

```
# Initialize a model, and put it on the device specified.
```

```
model = Classifier().to(device)
```

```
model.device = device
```

```
criterion = nn.CrossEntropyLoss()
```

```
optimizer = torch.optim.Adam(model.parameters(), lr=0.0003, weight_decay=1e-5)
```

```
# The number of training epochs.
```

```
n_epochs = 400
```

# 初期方案： OUTPUT

输出测试集上得到的最好的准确率  
并预测 **test** 集



```
with open("prediction.csv", "w") as f:
    f.write("Method,Acc\n")
    f.write(f"CNN_ver_0,{best_acc:.3f}\n")

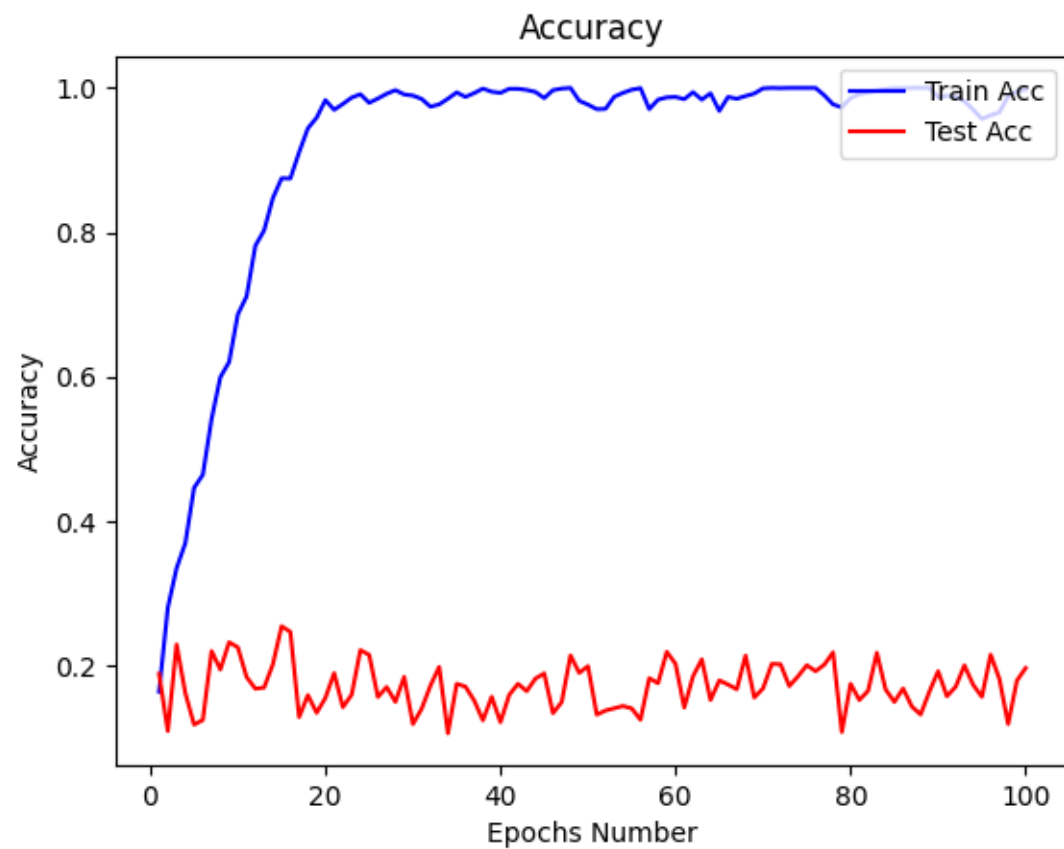
model.eval()
predictions = []
for batch in tqdm(test_loader):
    imgs, labels = batch
    with torch.no_grad():
        logits = model(imgs.to(device))

    predictions.extend(logits.argmax(dim=-1).cpu().numpy().tolist())
with open("predict.csv", "w") as f:
    f.write("Id,Category\n")
    for i, pred in enumerate(predictions):
        f.write(f"{i},{pred}\n")
```

# 初期方案 – 数据增强与学习率

方法	测试集最大准确率
初版方案 100 epochs	0.255

# 初版方案的过拟合



# 数据增强与学习率

---

《人工智能导论》创新设计

任务概述

初期方案

方案优化 – 数据增强与学习率

方案优化 – 半监督学习

方案优化 – ResNet50

总结

# 数据增强 Image Augmentation



翻转





# 数据增强 Image Augmentation



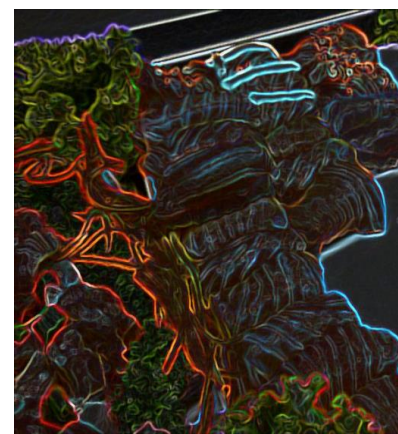
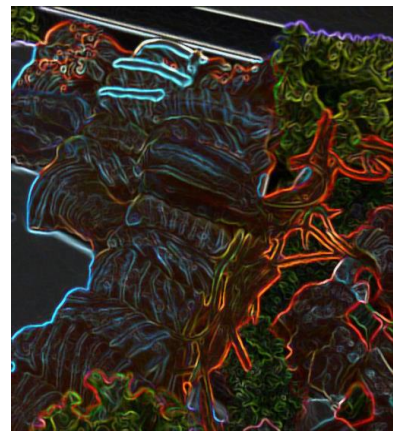
Augmentation



# 数据增强 Image Augmentation

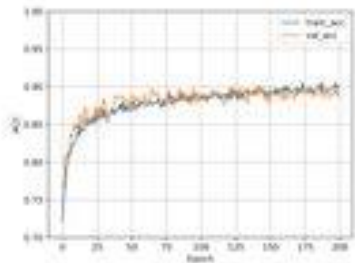
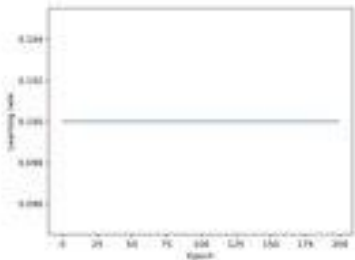


Augmentation?



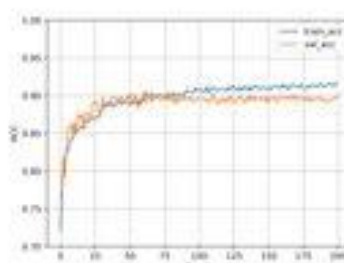
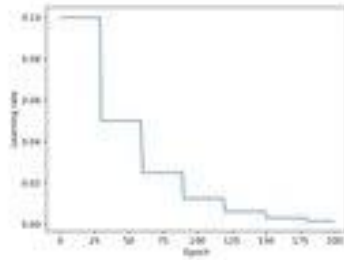
# Learning Rate Schedulers

1) No scheduler



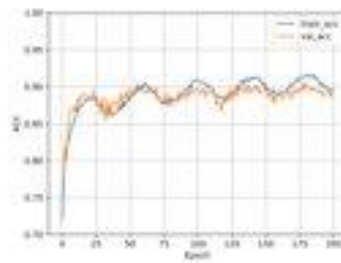
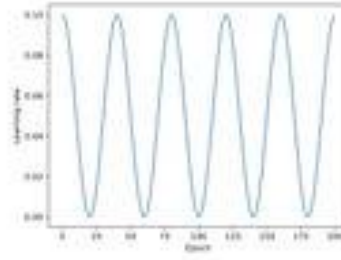
Validation acc: 88.6%  
Test acc: 88.9%

2) Step scheduler



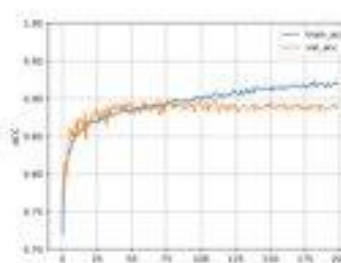
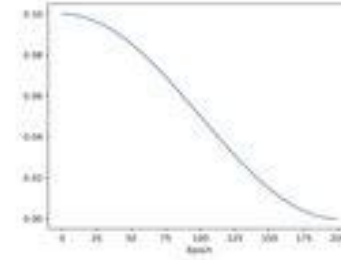
Validation acc: 89.7%  
Test acc: 89.9%

3) Cosine with restarts



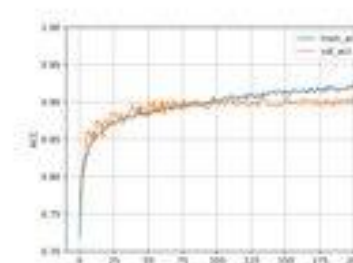
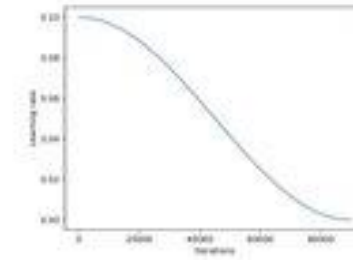
Validation acc: 89.6%  
Test acc: 89.3%

4) Classic cosine annealing (per epoch)



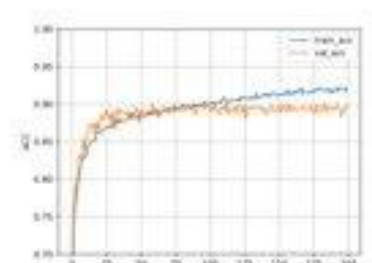
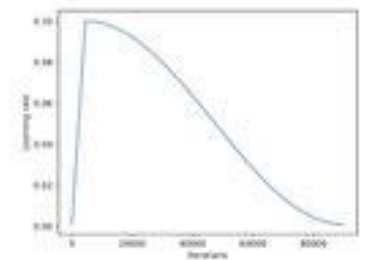
Validation acc: 89.4%  
Test acc: 89.2%

5) Classic cosine annealing (per batch)



Validation acc: 89.7%  
Test acc: 89.7%

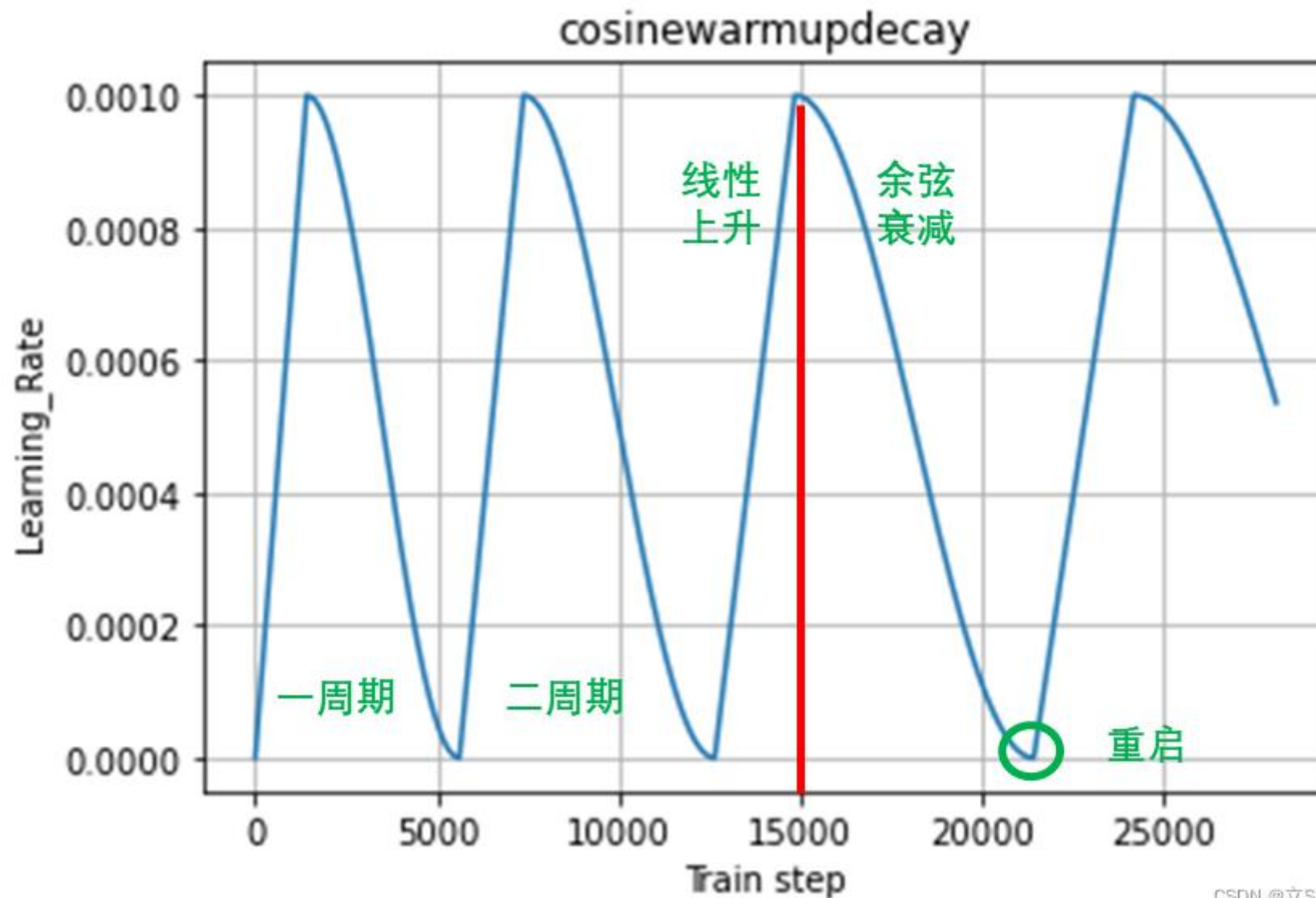
6) Classic cosine annealing (per batch) with warmup



Validation acc: 89.9%  
Test acc: 89.1%



# Learning Rate Schedulers



# 方案优化 – 数据增强与学习率

方法	测试集最大准确率
初版方案 100 epochs	0.255
数据增强 + 学习率调度器 400 epochs	0.568

# 半监督学习

---

《人工智能导论》创新设计

任务概述

初期方案

方案优化 – 数据增强与学习率

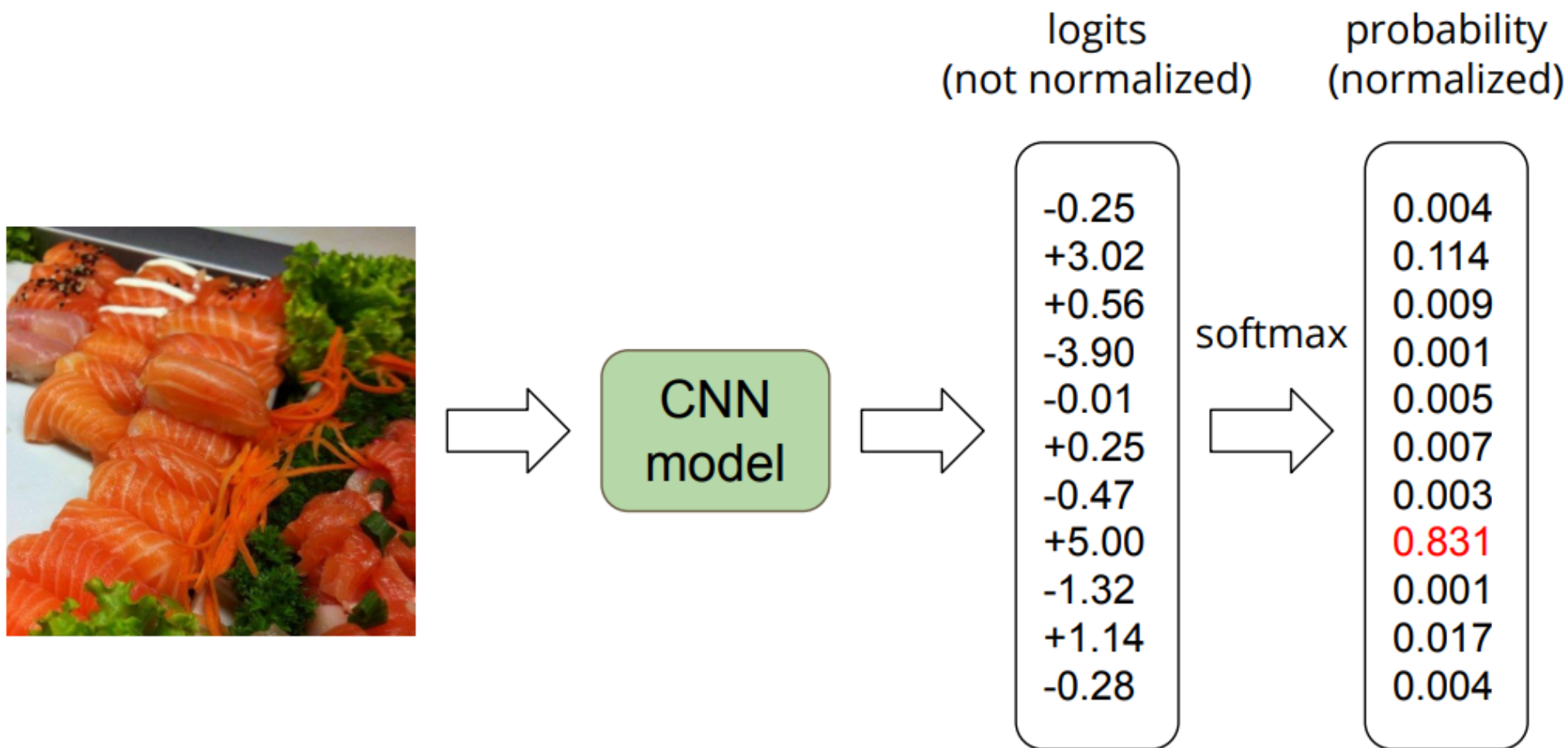
方案优化 – 半监督学习

方案优化 – ResNet50

总结

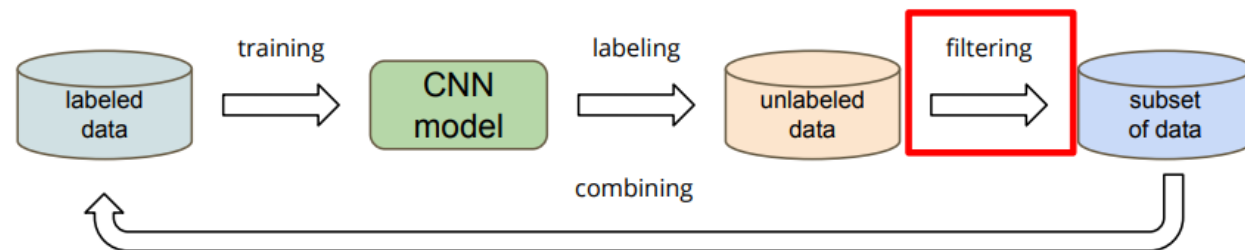


# 半监督学习

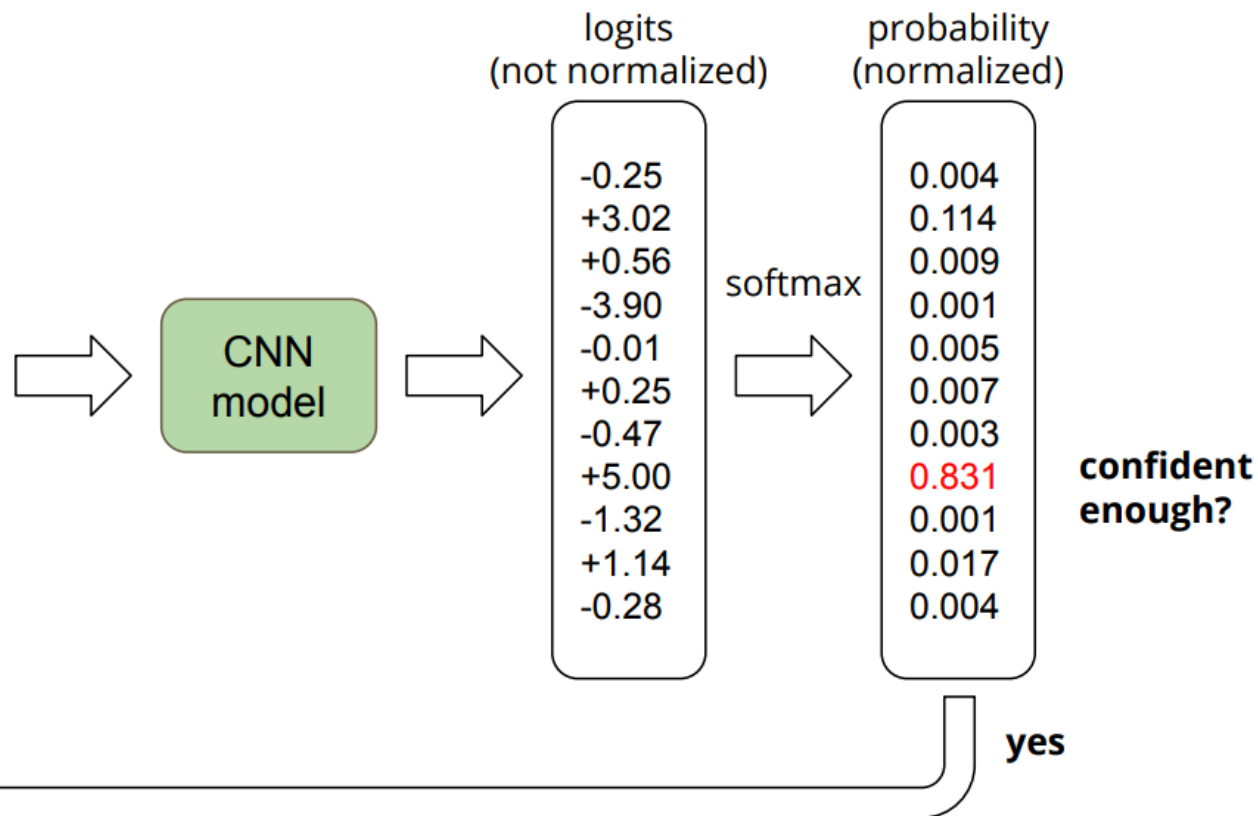


# 半监督学习

## Pseudo-labels



pseudo-label = 8



# 方案优化 – 半监督学习

方法	测试集最大准确率
初版方案 100 epochs	0.255
数据增强 + 学习率调度器 400 epochs	0.568
方案优化 + 半监督学习 400 epochs	0.597

# ResNet50

---

《人工智能导论》创新设计

任务概述

初期方案

方案优化 – 数据增强与学习率

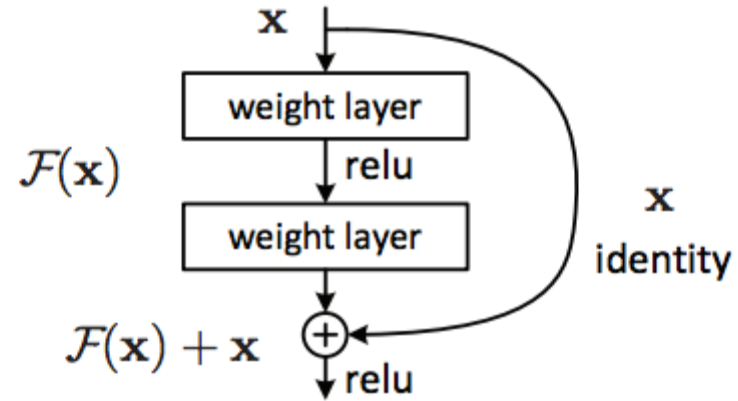
方案优化 – 半监督学习

方案优化 – ResNet50

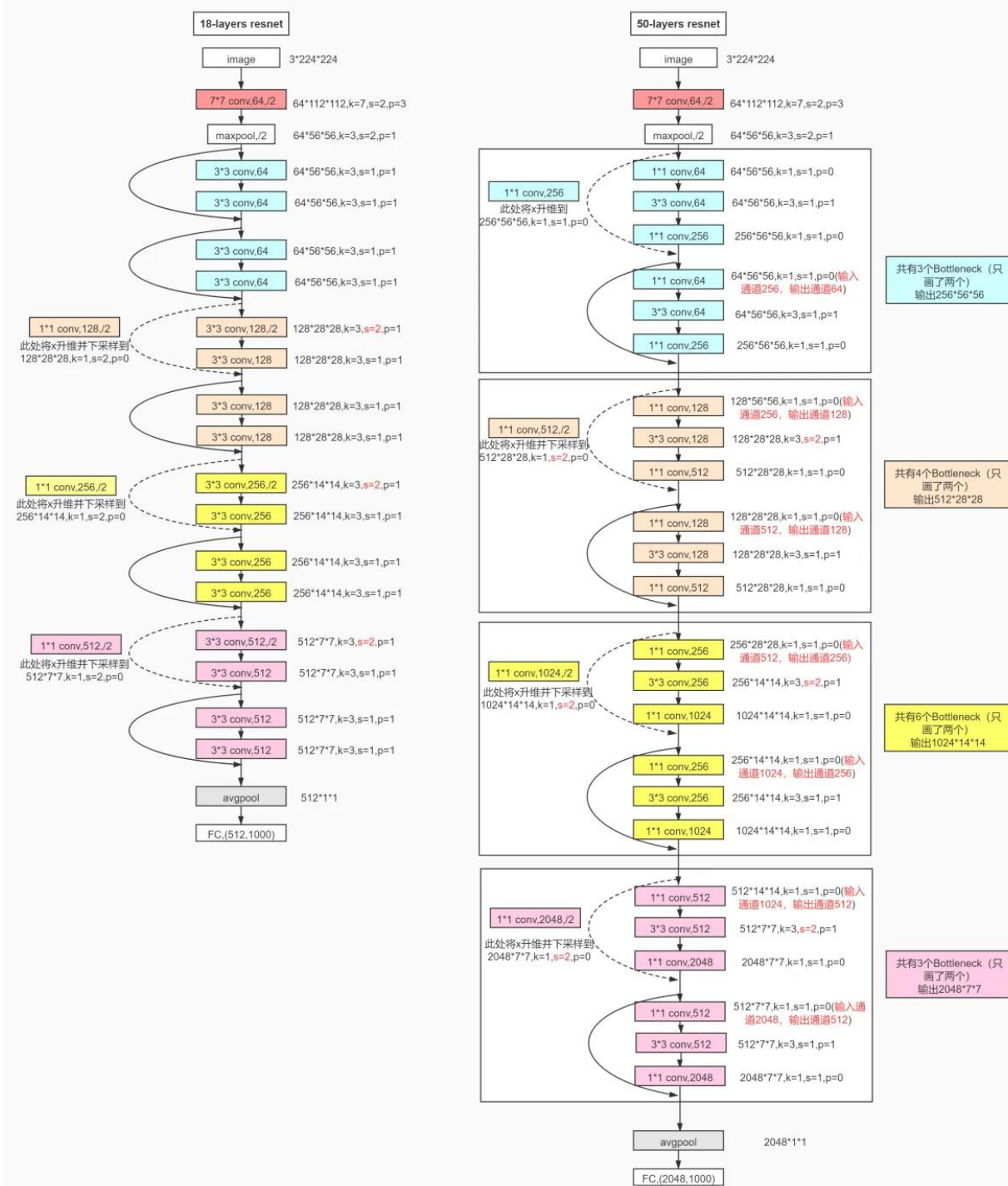
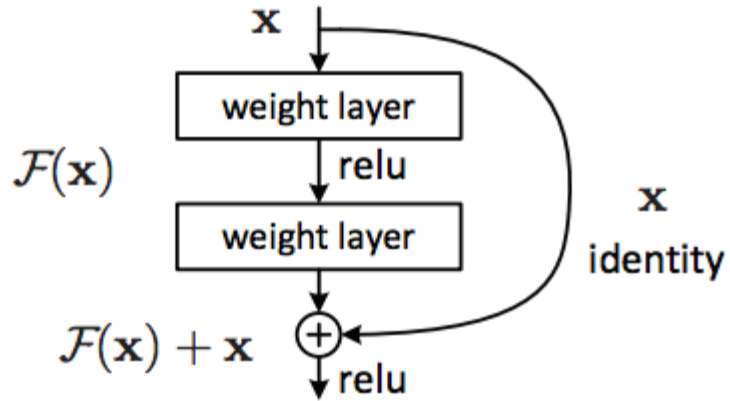
总结

# ResNet50

Deep Residual Learning for Image Recognition



# ResNet50





# ResNet50



```
model =  
resnet50().to(device)  
model.device = device  
  
model.fc.out_features = 11
```

# 方案优化 – ResNet50

方法	测试集最大准确率
初版方案 100 epochs	0.255
数据增强 + 学习率调度器 400 epochs	0.568
方案优化 + 半监督学习 400 epochs	0.597
方案优化 + ResNet50 200 epochs	0.700

# 总结

---

《人工智能导论》创新设计

任务概述

初期方案

方案优化 – 数据增强与学习率

方案优化 – 半监督学习

方案优化 – ResNet50

总结

# 总结

## CNN食物图像分类的优化：

- 图片数据本身：
  - 图像数据增广
  - 半监督，自监督学习
- 训练策略的优化
- 模型的改进
  - CNN：ResNet50
  - Attention
  - ...

方法	测试集最大准确率
初版方案 100 epochs	0.255
数据增强 + 学习率调度器 400 epochs	0.568
方案优化 + 半监督学习 400 epochs	0.597
方案优化 + ResNet50 200 epochs	0.700



# 谢谢大家

汇报人：2226114139 杨思涵