

# 半监督图像分类

## 实验目的

### 关于半监督学习

神经网络模型通常需要大量标记好的训练数据来训练模型。然而，在许多情况下，获取大量标记好的数据可能是困难、耗时或昂贵的。这就是半监督学习的应用场景。半监督学习的核心思想是利用无标记数据的信息来改进模型的学习效果。在半监督学习中，我们使用少量标记数据进行有监督学习，同时利用大量无标记数据的信息。通过充分利用无标记数据的潜在结构和分布特征，半监督学习可以帮助模型更好地泛化和适应未标记数据。关于深度学习中半监督学习更全面的总结，可以参考深度学习半监督学习综述[1]。

### 半监督图像分类

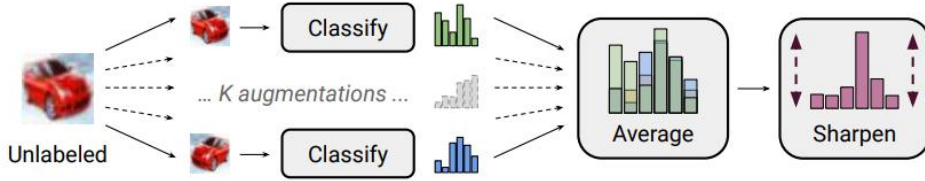
半监督学习在图像分类取得了非常大的进步，涌现了许多经典的半监督图像分类算法，如： $\pi$  Model[2]，Mean Teacher[3]，MixMatch[4]，FixMatch[5]等。这些算法都取得了非常好的结果，能够在仅使用少量标注数据的情况下，实现高精度的图像分类，在 ImageNet，CIFAR-10，CIFAR-100 等数据集上都有非常不错的效果。TorchSSL[8]是微软发布的一个用于半监督深度学习的库，其中提供了许多半监督学习算法的实现，如下图所示：

CIFAR-10 and CIFAR-100

		CIFAR-10				CIFAR100	
	40	250	4000		400	2500	10000
FullySupervised	95.38±0.05	95.39±0.04	95.38±0.05		80.7±0.09	80.7±0.09	80.73±0.05
PiModel [1]	25.66±1.76	53.76±1.29	86.87±0.59		13.04±0.8	41.2±0.66	63.35±0.0
PseudoLabel [3]	25.39±0.26	53.51±2.2	84.92±0.19		12.55±0.85	42.26±0.28	63.45±0.24
PseudoLabel_Flex [9]	26.26±1.96	53.86±1.81	85.25±0.19		14.28±0.46	43.88±0.51	64.4±0.15
MeanTeacher [2]	29.91±1.6	62.54±3.3	91.9±0.21		18.89±1.44	54.83±1.06	68.25±0.23
VAT [4]	25.34±2.12	58.97±1.79	89.49±0.12		14.8±1.4	53.16±0.79	67.86±0.19
MixMatch [5]	63.81±6.48	86.37±0.59	93.34±0.26		32.41±0.66	60.24±0.48	72.22±0.29
ReMixMatch [7]	90.12±1.03	93.7±0.05	95.16±0.01		57.25±1.05	73.97±0.35	79.98±0.27
UDA [6]	89.38±3.75	94.84±0.06	95.71±0.07		53.61±1.59	72.27±0.21	77.51±0.23
UDA_Flex [9]	94.56±0.52	94.98±0.07	95.76±0.06		54.83±1.88	72.92±0.15	78.09±0.1
FixMatch [8]	92.53±0.28	95.14±0.05	95.79±0.08		53.58±0.82	71.97±0.16	77.8±0.12
FlexMatch [9]	95.03±0.06	95.02±0.09	95.81±0.01		60.06±1.62	73.51±0.2	78.1±0.15

## 实验内容

### 1. 基于 MixMatch 的 CIFAR-10 数据集半监督图像分类



MixMatch 采用两种方式利用无标注数据，即熵最小化和一致性正则化。具体来说，MixMatch 首先对无标注数据进行  $K$  次增广，并获得对应的  $K$  个输出，并计算模型平均的预测分布，之后，MixMatch 使用一个 "sharpening" 函数来对无标注数据的平均预测进行锐化，并把锐化后的预测分布作为该无标注数据的标签。获得对无标注数据的标签后，将标注数据和无标注数据结合起来，通过 Mixup 的方式完成新数据集的构建，最后使用新的数据集进行训练。

---

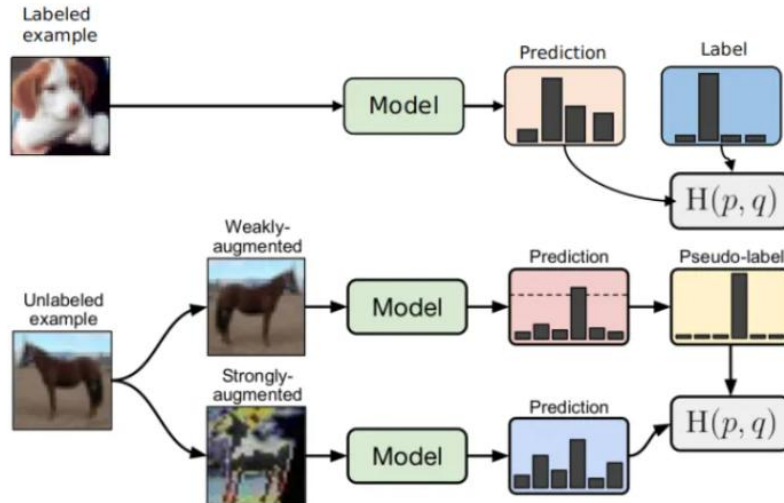
**Algorithm 1** MixMatch takes a batch of labeled data  $\mathcal{X}$  and a batch of unlabeled data  $\mathcal{U}$  and produces a collection  $\mathcal{X}'$  (resp.  $\mathcal{U}'$ ) of processed labeled examples (resp. unlabeled with guessed labels).

---

```
1: Input: Batch of labeled examples and their one-hot labels  $\mathcal{X} = ((x_b, p_b); b \in (1, \dots, B))$ , batch of unlabeled examples  $\mathcal{U} = (u_b; b \in (1, \dots, B))$ , sharpening temperature  $T$ , number of augmentations  $K$ , Beta distribution parameter  $\alpha$  for MixUp.  
2: for  $b = 1$  to  $B$  do  
3:    $\hat{x}_b = \text{Augment}(x_b)$  // Apply data augmentation to  $x_b$   
4:   for  $k = 1$  to  $K$  do  
5:      $\hat{u}_{b,k} = \text{Augment}(u_b)$  // Apply  $k^{\text{th}}$  round of data augmentation to  $u_b$   
6:   end for  
7:    $\bar{q}_b = \frac{1}{K} \sum_k p_{\text{model}}(y | \hat{u}_{b,k}; \theta)$  // Compute average predictions across all augmentations of  $u_b$   
8:    $q_b = \text{Sharpen}(\bar{q}_b, T)$  // Apply temperature sharpening to the average prediction (see eq. (7))  
9: end for  
10:  $\hat{\mathcal{X}} = ((\hat{x}_b, p_b); b \in (1, \dots, B))$  // Augmented labeled examples and their labels  
11:  $\hat{\mathcal{U}} = ((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K))$  // Augmented unlabeled examples, guessed labels  
12:  $\mathcal{W} = \text{Shuffle}(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}}))$  // Combine and shuffle labeled and unlabeled data  
13:  $\mathcal{X}' = (\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|))$  // Apply MixUp to labeled data and entries from  $\mathcal{W}$   
14:  $\mathcal{U}' = (\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|))$  // Apply MixUp to unlabeled data and the rest of  $\mathcal{W}$   
15: return  $\mathcal{X}', \mathcal{U}'$ 
```

---

### 2. 基于 FixMatch 的 CIFAR-10 数据集半监督图像分类



FixMatch 结合了伪标签和一致性正则化来实现对无标注数据的高效利用，训练过程包括两个部分，有监督训练和无监督训练。有标注的数据，执行有监督训练，和普通分类任务训练没有区别。没有标注的数据，首先将其经过弱增强之后送到模型中推理获取伪标签，然后再将其经过强增强送到模型中进行预测，并利用生成的伪标签监督模型对强增强数据的预测。模型对弱增强数据的预测，只有大于一定阈值条件时才执行伪标签的生成，并使用该伪标签来进行无标注图像的训练。

---

**Algorithm 1** FixMatch algorithm.

---

- 1: **Input:** Labeled batch  $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$ , unlabeled batch  $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$ , confidence threshold  $\tau$ , unlabeled data ratio  $\mu$ , unlabeled loss weight  $\lambda_u$ .
  - 2:  $\ell_s = \frac{1}{B} \sum_{b=1}^B H(p_b, \alpha(x_b))$  {Cross-entropy loss for labeled data}
  - 3: **for**  $b = 1$  **to**  $\mu B$  **do**
  - 4:    $q_b = p_m(y \mid \alpha(u_b); \theta)$  {Compute prediction after applying weak data augmentation of  $u_b$ }
  - 5: **end for**
  - 6:  $\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} H(\arg \max(q_b), p_m(y \mid \mathcal{A}(u_b)))$  {Cross-entropy loss with pseudo-label and confidence for unlabeled data}
  - 7: **return**  $\ell_s + \lambda_u \ell_u$
- 

## 实验要求

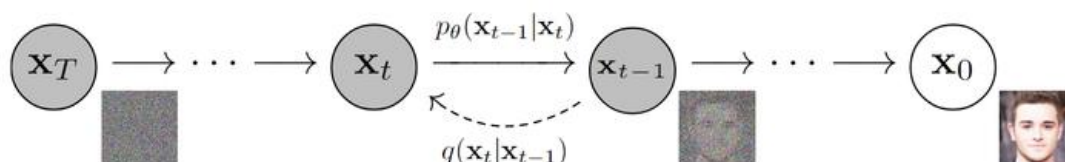
1. 阅读原始论文和相关参考资料，基于 Pytorch 实现 MixMatch 和 FixMatch 半监督图像分类算法，按照原始论文的设置，MixMatch 和 FixMatch 均使用 WideResNet-28-2 作为 Backbone 网络，即深度为 28，扩展因子为 2，在 CIFAR-10 数据集上进行半监督图像分类实验，**报告算法在分别使用 40, 250, 4000 张标注数据的情况下的图像分类效果（标注数据随机选取指定数量）**
2. 使用 TorchSSL 中提供的 MixMatch 和 FixMatch 的实现进行半监督训练和测试，**对比自己实现的算法和 TorchSSL 中的实现的效果**
3. 提交源代码，并提交实验报告，描述主要的实现步骤，并**分析对比 MixMatch 和 FixMatch 的相同点和不同点。**

# 扩散生成模型(选做)

## 实验目的

### 关于扩散模型

扩散模型(Diffusion Model)是近年来提出的一种新的生成模型。一般我们讨论的生成模型是一种概率模型，我们希望学习一个简单的分布比如高斯分布，来学习映射到一个我们想要生成的真实图像的分布，推理时，我们只需要从简单分布中取一个样本，比如高斯分布中的一个噪声样本，经过模型计算就能映射到真实图像类中的一个样本，即生成一张真实的图像。与之前的对抗生成式模型通过对抗隐式学习从噪声分布到真实图像分布不同，扩散模型基于马尔科夫链，将生成过程建模为一个加噪，去噪的过程。训练时，先对一张真实图像，不断往其中添加噪声使其变成标准高斯噪声（前向过程），然后训练网络学习前向过程中加噪声的逆过程，从标准高斯噪声中不断去噪，还原回原始数据（逆向过程）。学习完成后，就能够从已知的标准高斯分布中采样一个噪声数据，利用学到的去噪过程生成符合原始数据分布的新数据。更多原理的推理和相关细节可以阅读 DDPM[6] 论文以及相关综述[7]。

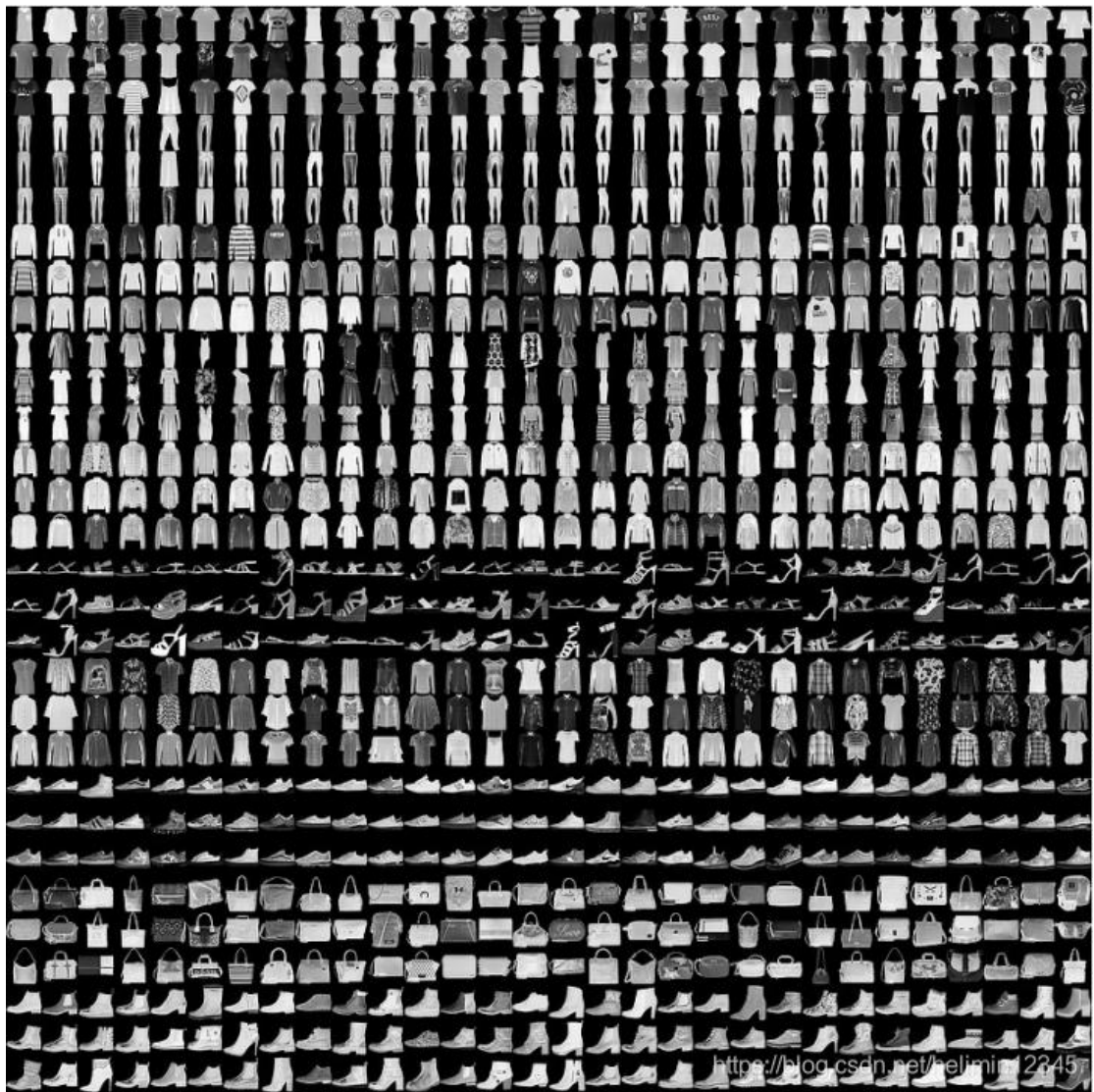


## 实验内容

### 1. 基于 DDPM 的 MNIST 和 Fasion-MNIST 图像生成

MNIST 手写数字体数据集是由 LeCun 创建的用来进行深度学习训练的简单数据集，被广泛用来图像识别图像生成中，由于 MNIST 过于简单，Fashion MNIST 数据集被提出，该数据集由衣服、鞋子等服饰组成，包含 70000 张图像，其中 60000 张训练图像加 10000 张测试图像，图像大小为 28x28，单通道，共分 10 个类，如下图，每 3 行表示一个类。



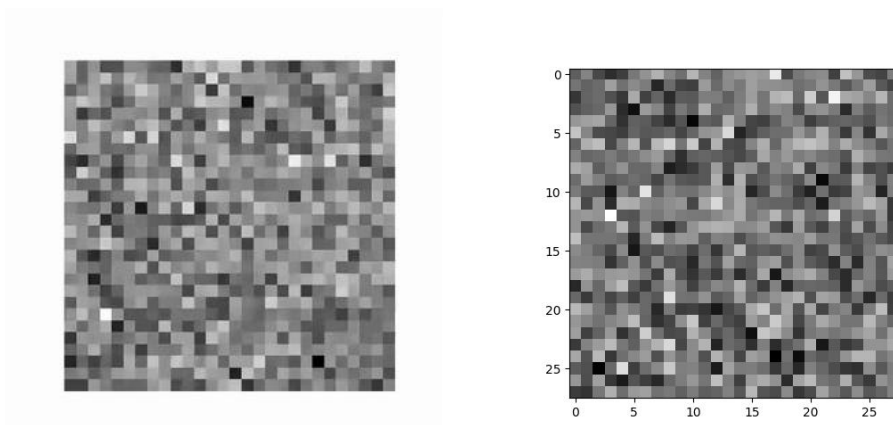


数据集信息如下：

Name	Content	Examples	Size	Link	MD5 Checksum	Label	Description
train-images-idx3-ubyte.gz	training set images	60,000	26 MBytes	<a href="#">Download</a>	8d4fb7e6c68d5	0	T-shirt/top
					91d4c3dfe9ec88bf0d	1	Trouser
						2	Pullover
train-labels-idx1-ubyte.gz	training set labels	60,000	29 KBytes	<a href="#">Download</a>	25c81989df183	3	Dress
					df01b3e8a0aad5dffb	4	Coat
						5	Sandal
t10k-images-idx3-ubyte.gz	test set images	10,000	4.3 MBytes	<a href="#">Download</a>	bef4ecab320f0	6	Shirt
					6d8554ea6380940ec79	7	Sneaker
						8	Bag
t10k-labels-idx1-ubyte.gz	test set labels	10,000	5.1 KBytes	<a href="#">Download</a>	bb300cfdad3c1	9	Ankle boot
					6e7a12a480ee83cd310		

## 实验要求

1. 阅读原始论文和相关参考资料，基于 Pytorch 实现简单版 DDPM（不要求改进），并在 MNIST 和 Fashion MNIST 上进行训练，测试生成效果
2. 在 MNIST 和 Fashion MNIST 上完成 DDPM 的训练，**每个数据集展示 10 个生成的图片**，并以**动图形式在 MNIST 和 Fashion MNIST 上各展示一个生成效果图**，如下图所示：



3. 提交源代码和实验报告，对实现过程中的关键步骤进行必要说明

# 物体检测模型（选做）

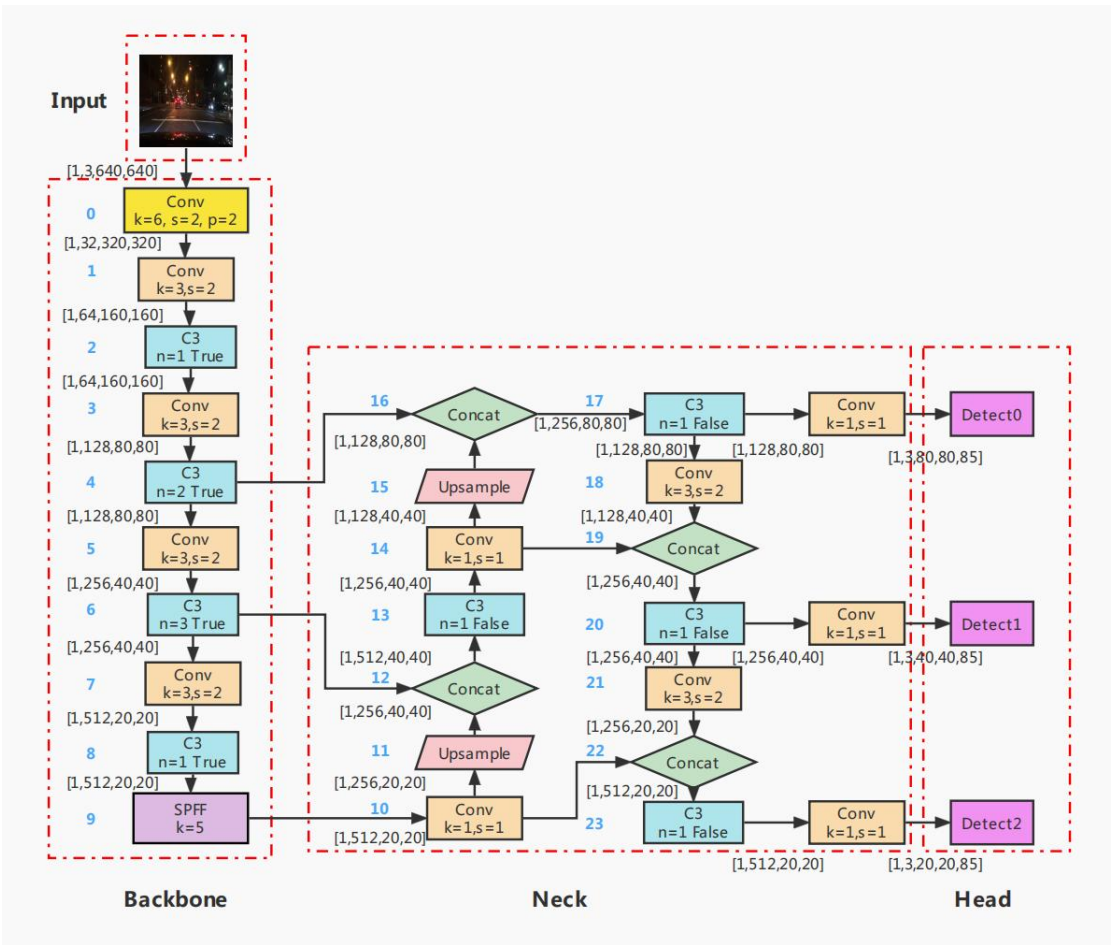
## 实验目的

### 关于物体检测模型

物体检测（Object Detection）是计算机视觉和数字图像处理的热门方向，意在判断一幅图像上是否存在感兴趣物体，并给出物体分类及位置等（What and Where）。本文主要进行物体检测研究背景、发展脉络、相关算法及评价指标的概述。从研究角度来看，物体检测是计算机视觉根本问题之一，是很多高层视觉任务的基础（目标跟踪、动作识别、行为描述等）；从应用角度来看，在目前诸如视频监控、自动驾驶等领域，物体检测也表现出了广泛的应用需求。在本次作业中我们将尝试使用物体检测方法进行图片的脸部识别。

## 实验内容

### 1、基于 yoloV5 进行物体检测



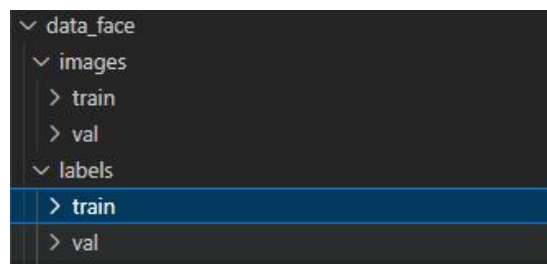
yolov5[9]于 2020 年由 glenn-jocher 首次提出，直至今日 yolov5 仍然在不断进行升级迭代。YOLO 将对象检测重新定义为一个回归问题。它将单个卷积神经网络

(CNN)应用于整个图像，将图像分成网格，并预测每个网格的类概率和边界框。可以通过该网址进行学习[10]。

使用的数据集是 Wider Face 数据集，该数据集挑选出了 32,203 图片并进行了人脸标注，总共标注了 393,703 个人脸数据。并且对于每张人脸都附带有更加详细的信息，由于其图片过多，我们选择其中一个类别进行实验。数据集下载通过该链接[11]下载

## 实验要求

1. 充分阅读和理解原论文的算法和相关资料，并且将数据集进行下载，使用文件夹内“59--people--driving--car”的数据进行训练。
2. 从链接[12]下载 yolov5 的代码，先自主编写脚本将数据的标签从 wider\_face\_val\_bbx\_gt.txt 转换成代码可以采用的模式（为每张图片生成相应的 label txt，里面的内容是类别、中心点 x、中心点 y、框宽度 w、框高度 h），数据的格式如下：



将数据中的 789~1038 作为测试集，其余作为训练集进行训练，并且在 data 文件夹下创建相关的 yaml 文件进行数据的处理。

3. 阅读 GitHub 的 readme 文件开始训练，我们使用 yolov5s 模型进行训练，记得下载预训练模型，并且在 data/hyps/文件夹下新建你的超参数文件，模仿其给的例子进行训练，你可以通过调整你的参数来优化你的训练结果。
4. 将你不同参数条件下的训练结果和可视化结果保存并且在实验报告中进行分析，并将你的代码和数据处理脚本打包上交。如果你对更多的数据集感兴趣，你可以将 Wider Face 更多的数据投入训练，自主划分训练集、验证集和测试集进行实验。若你对实验的评价指标有所疑问，可以查看链接[13]

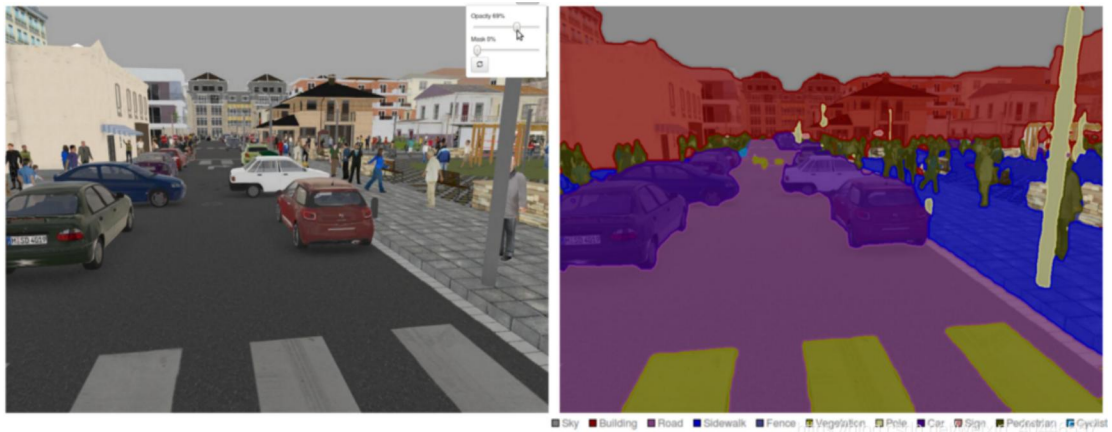


# 语义分割模型（选做）

## 实验目的

### 关于语义分割模型

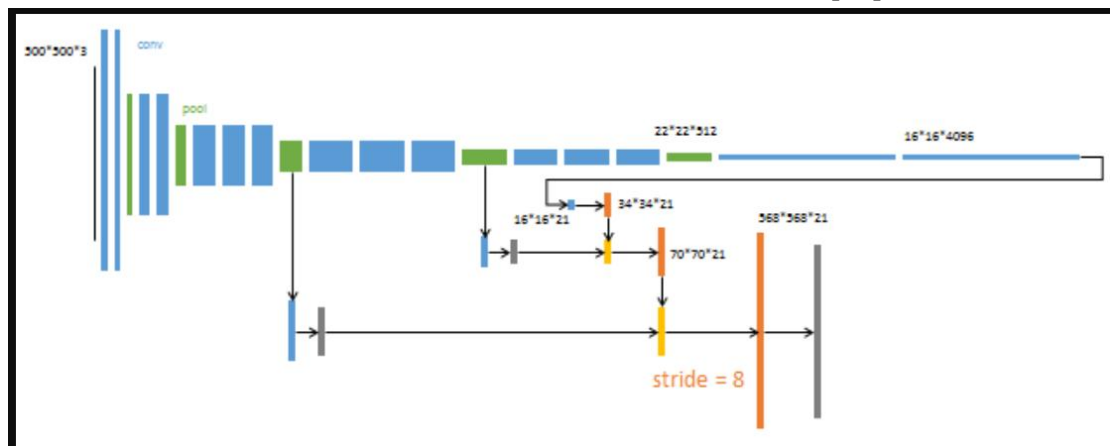
语义分割是计算机视觉中很重要的一个方向。不同于目标检测和识别，语义分割实现了图像像素级的分类。它能够将一张图片或者视频（视频以帧来提取的话其实就是图片），按照类别的异同，将图像分为多个块。如下图所示：



## 实验内容

### 1、基于 FCN 进行语义分割

全卷积网络（Fully Convolutional Networks, FCN）是 Jonathan Long 等人于 2015 年在 Fully Convolutional Networks for Semantic Segmentation[14]一文中提出的用于图像语义分割的一种框架，是深度学习用于语义分割领域的开山之作。FCN 将传统 CNN 后面的全连接层换成了卷积层，这样网络的输出将是热力图而非类别；同时，为解决卷积和池化导致图像尺寸的变小，使用上采样方式对图像尺寸进行恢复。本次实验采用 FCN-8s 的结构，具体可通过链接[15]学习。



## 2、VOC2007 数据集

VOC 数据集的详细介绍可通过链接[16]进行了解

## 实验要求

1. 充分阅读和理解原论文的算法和相关资料，并且将数据集通过链接[12]进行下载。
2. 在本次实验中，我们将使用 `pytorch` 简单的编写 FCN-8s 的模型代码，数据预处理和相关代码可以进行参考，模型代码自行编写，并且使用 VOC2007 数据集进行训练和测试，可以尝试调整参数。
3. 将主要的模型代码分析和结果分析，结果的可视化结果请使用测试集中的 000068、000175、000243、000333、000346、000364、000392、000452 进行展示。提交源代码和实验报告，对实现过程中的关键步骤进行必要说明。

- [1]. A Survey on Deep Semi-supervised Learning 2021
- [2]. Temporal Ensembling for Semi-Supervised Learning
- [3]. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results
- [4]. Mixmatch: A holistic approach to semi-supervised learning
- [5]. Fixmatch: Simplifying semi-supervised learning with consistency and confidence
- [6]. Denoising Diffusion Probabilistic Models
- [7]. Diffusion Models: A Comprehensive Survey of Methods and Applications
- [8]. <https://github.com/StephenStorm/TorchSSL>
- [9]. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios
- [10]. [https://blog.csdn.net/weixin\\_45842280/article/details/123132320](https://blog.csdn.net/weixin_45842280/article/details/123132320)
- [11]. <https://pan.baidu.com/s/1kEwRsR9jr0wHfLh9jOxIqg?pwd=qptn>
- [12]. <https://github.com/ultralytics/yolov5.git>
- [13]. <https://github.com/rafaelpadilla/Object-Detection-Metrics>
- [14]. Fully Convolutional Networks for Semantic Segmentation
- [15]. [https://blog.csdn.net/qq\\_41731861/article/details/120511148](https://blog.csdn.net/qq_41731861/article/details/120511148)
- [16]. [https://blog.csdn.net/HUAI\\_BI\\_TONG/article/details/121499060](https://blog.csdn.net/HUAI_BI_TONG/article/details/121499060)

**作业要求：第一道题目必做，后三道题目选一道进行实验。**

**作业提交时间：**2023 年 6 月 20 日 24:00 前，推迟一天成绩减 5 分  
(不足一天按一天计算)，以有效作业提交的邮件时间戳为准。

**提交方式：**作业命名 “学号-姓名-计算机视觉第三次作业.zip” ，交到课程邮箱（邮箱：prcv\_homework@163.com）

**注意：**不得抄袭，包括源代码!!!