

# 计算机视觉

---

计算机视觉是一门关于如何运用照相机和计算机来获取我们所需的，被拍摄对象的数据与信息的学问。

图像处理是用任何信号处理等数学操作处理图像的过程，输入时图像（摄影图像或者视频帧），输出是图像或者与输入图像有关的特征、参数的集合。

计算机图形学是计算机科学的一个子领域，它包括数字合成和操作可视内容（图像、视频）的方法。尽管这个术语通常指三维计算机图形学的研究，但它也包括二维图形学和图像处理。

## 数字图像处理

---

数字图像处理是对输入图像进行有目的地转化，从而提高图像质量、视觉效果等，研究的是图像的获取、传输、存储、变换、显示、理解和综合利用的学科。

### 灰度图像处理

#### 灰度线性变换

灰度线性变换是灰度变换的一种，图像的灰度变换通过建立灰度映射来调整原图像的灰度，从而达到图像增强的目的。灰度映射通常是用灰度变换曲线来表示的

#### 傅里叶变换

任何一种时域信号或空域信号

都可以转换为频域信号

有时候在频域上处理信号更加简单

### 图像去噪

噪声的产生是信号在采集、传输以及记录过程中，受到成像设备自身因素和外界环境的影响而产生的。通过去噪可以有效地增大图像信号的信噪比，提高图像质量，更好地体现原始图像所携带的信息

#### 基于滤波器的方法

利用某些人工设计的低通滤波器来去除图像噪声。包括均值滤波、中值滤波、高斯滤波

#### 基于模型的方法

试图对自然图像或噪声的分布进行建模，然后使用模型分布作为先验，试图获得清晰的图像与优化算法。

基于模型的方法通常将去噪任务定义为基于最大后验(MAP)的优化问题，其性能主要依赖于图像的先验。包括非局部自相似（NSS）模型，稀疏模型，梯度模型和马尔可夫随机场（MRF）模型。

## 基于学习的方法

学习有噪声图像到干净图像的潜在映射，可以分为传统的基于学习的方法和基于深度学习的学习方法。

## 边缘检测

### 边缘的定义

图像中灰度发生突变或不连续的微小区域，即是两个具有相对不同灰度值特性的区域的边界线。

在一幅图像中，边缘有**方向**和**幅度**两个特性。

一般认为沿边缘走向的灰度变化较为平缓，而垂直于边缘走向的灰度变化剧烈。灰度梯度指向边缘的垂直方向。

### 基于曲面拟合的边缘检测方法

基于差分检测图像边缘的算子往往对噪声敏感。因此对一些噪声比较严重的图像就难以取得满意的效果。

若用平面或高阶曲面拟合图像中某一小区域的灰度表面，求这个拟合平面或曲面的外法线方向的微分或二阶微分检测边缘，可减少噪声影响。

## 特殊形状检测

### 点检测

用空域的高通滤波器来检测孤立点

### 线检测

霍夫变换

## 图像形态学操作

### 腐蚀

腐蚀是一种消除边界点，使边界向内部收缩的过程。可以用来消除小且无意义的物体。

### 膨胀

膨胀在数学形态运算中的作用是扩展物体的边界点，在数字图像处理中，对于确定的结构元素，通过膨胀运算可以使一些相临距离较短的区域进行连接；

### 开运算

膨胀在数学形态运算中的作用是扩展物体的边界点，在数字图像处理中，对于确定的结构元素，通过膨胀运算可以使一些相临距离较短的区域进行连接；

### 闭运算

闭运算可以看作是先做膨胀处理再用腐蚀处理的方法进行恢复

## 图像分割

---

## 概念

图像分割是指根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域，使得这些特征在同一区域内表现出一致性或相似性，而在不同区域间表现出明显的不同。

**目标分割**将不同分属不同物体的像素区域分开。如前景与背景分割开，狗的区域与猫的区域与背景分割开。

**语义分割**在普通分割的基础上，分类出每一块区域的语义（即这块区域是什么物体）。如把画面中的所有物体都指出它们各自的类别。

**实例分割**在语义分割的基础上，给每个物体编号。如这个是该画面中的狗A，那个是画面中的狗B。

## 图像分割方法

### 根据阈值的分割方法

图像阈值分割是一种广泛应用的分割技术，利用图像中要提取的目标物与其背景在灰度特性上的差异，把图像视为具有不同灰度级的两类区域(目标和背景)的组合，选取一个合适的阈值，以确定图像中每个像素点应该属于目标还是背景区域，从而产生相应的二值图像。

阈值处理概述

图像二值化

直方图阈值分割

最佳阈值分割

自适应阈值分割

阈值分割法的特点是：

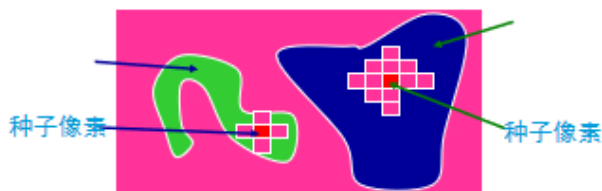
适用于物体与背景有较强对比的情况，背景或物体的灰度比较单一；  
而且总可以得到封闭且连通区域的边界。

### 基于区域的分割方法

#### 区域生长法(regiongrows)

##### ● 算法实现步骤：

- 1) 根据图像的不同应用选择一个或一组**种子**，它或者是最亮或最暗的点，或者是位于点簇中心的点；
- 2) 选择一个描述符（条件）；
- 3) 从该种子向外扩张，先把种子像素加入结果集合，然后不断将与集合中各个像素连通、且满足描述符的像素加入集合
- 4) 进行到不再有满足条件的新结点加入集合为止。



### (1) 区域分裂法

如果区域的某些特性差别比较大,即不满足一致性准则时,则区域应该采用分裂法,分裂过程从从图像的最大区域开始,一般情况下,是从整幅图像开始。

注意:

- 确定分裂准则(一致性准则)
- 确定分裂方法,即如何分裂区域,使得分裂后的子区域的特性尽可能都满足一致性准则。

### 算法

- (1) 形成初始区域
- (2) 对图像的每一个区域 $R_i$ , 计算 $P(R_i)$ , 如果 $P(R_i)=FALSE$   
则沿着某一合适的边界分裂区域
- (3) 重复步骤 (2), 当没有区域需分裂时, 算法结束。

## 基于边缘的分割方法

图割是一种交互式图像分割方法,需要用户以指定图像的部分前景与部分背景,然后算法以用户的输入作为分割的约束条件自动地计算出满足约束条件下的最佳分割。典型的交互手段包括用一把画刷在前景和背景处各画几笔以及在前景的周围画一个方框等。

Graph Cut图割算法将图像分割问题与图的最小割(min cut)问题相关联,采用最小割最大流方法进行图像分割,将图像分割为前景和背景。使用时在前景和背景处各话几笔作为输入,算法将建立各个像素点与前景背景相似度的赋权图,并通过求解最小切割区分前景和背景。

## 基于聚类的分割方法

聚类方法包括:

K-means

模糊聚类

谱聚类

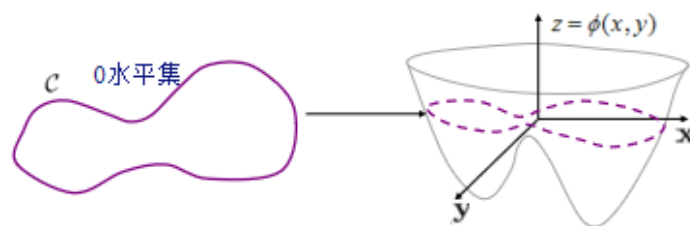
...

## 基于优化算法的分割方法

水平集方法

水平集图像分割方法是一种典型的能量最小化方法

水平集图像分割方法将平面中的曲线（或者三维中的曲面）看成高一维空间中某一函数  $\phi$ （称为水平集函数）的零水平集。通过演化水平集函数来获取目标区域的边界。



### 图割方法

图割是一种交互式图像分割方法，需要用户以指定图像的部分前景与部分背景，然后算法以用户的输入作为分割的约束条件自动地计算出满足约束条件下的最佳分割。典型的交互手段包括用一把画刷在前景和背景处各画几笔以及在前景的周围画一个方框等。

Graph Cut图割算法将图像分割问题与图的最小割（min cut）问题相关联，采用最小割最大流方法进行图像分割，将图像分割为前景和背景。使用时在前景和背景处各话几笔作为输入，算法将建立各个像素点与前景背景相似度的赋权图，并通过求解最小切割区分前景和背景。

## 目标检测

### 概念

目标检测（Object Detection）的任务是找出图像中所有感兴趣的目标（物体），确定它们的类别和位置，是计算机视觉领域的核心问题之一

目标识别（Object Recognition）的任务是基于分类（Classification）的识别（Recognition）问题，即是在所有的给定数据中，分类出哪一些sample是目标，哪一些不是。

### 目标检测算法基本流程



#### ➤ 步骤：

- ①首先在待检测图片进行**候选框的提取**。
- ②对每个候选框中的局部信息进行**特征提取**。
- ③对候选区域提取出的特征进行**分类判定**
- ④利用**NMS**来对冗余候选框进行合并
- ⑤输出得到结果

#### ➤ 其他：

针对深度学习中一阶段的算法，上图红色框的部分通过一个神经网络实现，不单独提取出候选框

## 1. 候选区域产生



### ➤ 以滑动窗口为例:

使用滑动窗口的方式提取候选区域, 用不同大小的图, 在图片上从左到右, 从上到下的搜索, 不同尺度的目标使用不同大小的框。

用很多不同大小的窗口进行移动, 把图片分成很多个小窗口获取候选区域。

## 2. 数据表示



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}, y_{true} = \begin{bmatrix} 1 \\ 40 \\ 45 \\ 80 \\ 60 \\ 0 \\ 1 \\ 0 \end{bmatrix}, y_{pred} = \begin{bmatrix} 0.88 \\ 41 \\ 46 \\ 82 \\ 59 \\ 0.01 \\ 0.95 \\ 0.04 \end{bmatrix}$$

- 左侧图片为标记后的样本数据, 将label信息展现在原图上的结果展示。
- 右侧图片中的 $y$ 为通过模型训练后的数据信息。  
其中 $p_c$ 为预测的概率,  $(b_x, b_y, b_w, b_h)$ 为坐标信息(中心点坐标下 $x, y$ , 宽高 $w, h$ )  
 $C_1, C_2, C_3$ 分别为类别, 以独热编码的形式展现

## 目标检测分类

### ➤ 基于特征点的目标检测

- 根据图像中的关键点, 进行目标识别和检测。
- 主要介绍三种特征点提取和描述的算法

### ➤ 基于模板匹配的目标检测

- 根据已知模板图像到另一幅图像中寻找与模板图像相似的子图像。
- 主要介绍基于灰度的模板匹配算法

### ➤ 基于深度学习的目标检测

- 分为一阶段和二阶段两大类分别介绍深度学习的目标检测算法
- 主要介绍YOLO系列算法主题思路, SSD、RCNN、Fast RCNN、Faster RCNN等算法

## 基于特征点的目标检测

## 一般过程



### ➤ 图像预处理

在进行特征检测之前,通常需要对图像进行预先的处理。包括灰度化、去噪声、生成图像金字塔等过程。

### ➤ 特征点检测

提取图像中的特征点

### ➤ 局部描述子提取

从特征点周围提取出一个小的几何区域(patch),并生成一个标识性的向量来代表这个区域的特征

### ➤ 特征匹配

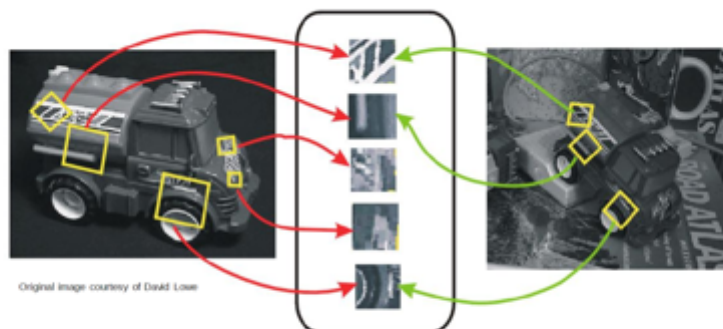
计算两个图像中局部描述子之间的相似性,找出图像间的匹配点

## SIFT算法

将一幅图像映射(变换)为一个局部特征向量集;特征向量具有平移、缩放、旋转不变性,同时对光照变化、放射及投影变换也有一定不变性。

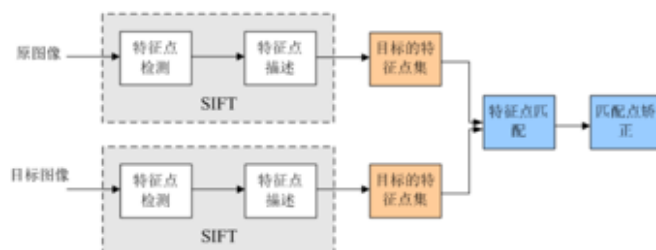
## SIFT算法

### 基于特征点的检测一般过程



- 将一幅图像映射(变换)为一个局部特征向量集;特征向量具有平移、缩放、旋转不变性,同时对光照变化、放射及投影变换也有一定不变性。

## SIFT算法



### ➤ 步骤:

1. 尺度空间极值检测: 搜索所有尺度上的图像位置。
2. 关键点定位: 在每个候选的位置上,通过一个拟合精细的模型来确定位置和尺度。
3. 方向确定: 基于图像局部的梯度方向,分配给每个关键点位置一个或多个方向。
4. 关键点描述: 在每个关键点周围的邻域内,在选定的尺度上测量图像局部的梯度

## ORB算法

ORB算法(Oriented FAST and Rotated BRIEF): 一种快速特征点提取和描述的算法,将FAST特征点的检测方法 with BRIEF特征描述子结合起来,并在它们原来的基础上做了改进与优化

### ➤步骤:

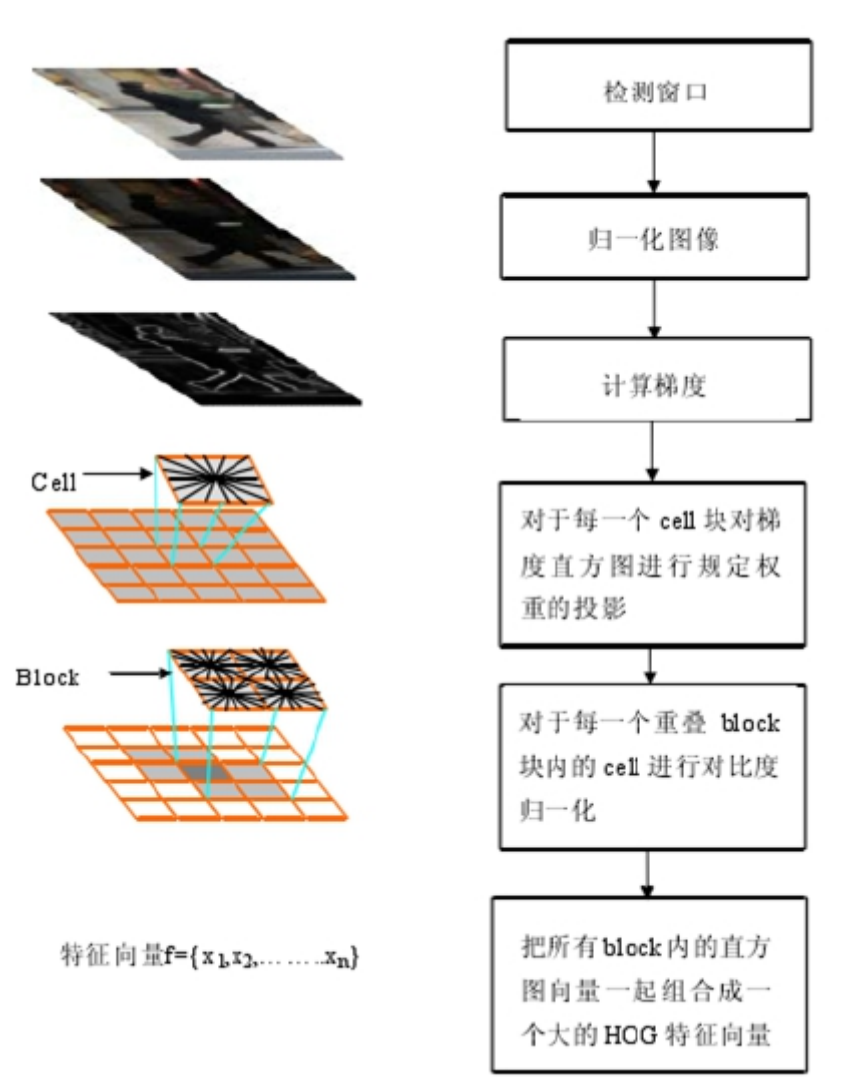
1. 利用oFAST (FAST Keypoint Orientation) 算法提取特征点 (改进的FAST算法)
2. 利用rBRIEF生成特征描述子 (改进的BRIEF算法)
3. 特征点匹配

## HOG算法

### HOG算法

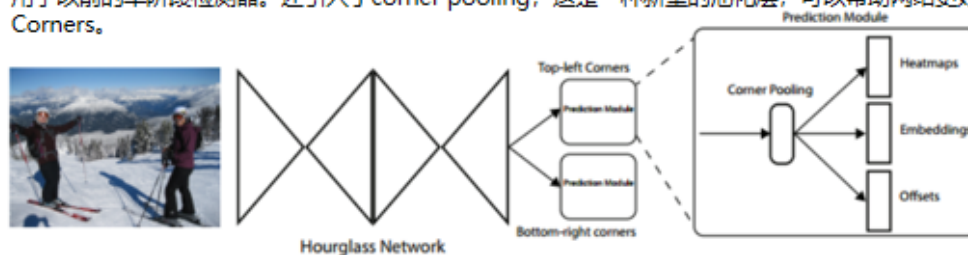
- HOG的基本思想: 在一副图像中,局部目标的外观和形状能够被梯度或边缘的方向密度分布很好地描述。
- HOG与SIFT类似,均使用梯度方向直方图来构造描述子,但HOG是在一个图像网格上密集计算描述子,而SIFT只在特征点上计算描述子。





## CornerNet

CornerNet: 一种新的目标检测方法, 使用一个卷积神经网络将一个目标边界框检测视为一对关键点, 左上角和右下角。通过将目标检测看作成对的关键点, 消除了设计一组Anchor boxes的需要, 该Anchor boxes通常用于以前的单阶段检测器。还引入了corner pooling, 这是一种新型的池化层, 可以帮助网络更好地定位Corners。



CornerNet的概述: 使用沙漏网络作为CornerNet的backbone网络。沙漏网络之后是两个预测模块。一个模块用于左上角, 另一个模块用于右下角。在预测热图、embedding和offsets之前, 每个模块都有自己的corner pooling模块来从特征图上进行pooling操作。与许多其他目标检测器不同, 不使用来自不同尺度的特征来检测不同大小的目标 (即没有采用FPN)。只将这两个模块应用于沙漏网络的输出。

## CenterNet

CenterNet利用关键点三元组即中心点、左上角点和右下角点三个关键点而不是两个点来确定一个目标，使网络花费了很小的代价便具备了感知物体内部信息的能力，从而能有效抑制误检。



CornerNet的概述：网络通过 center pooling 和 cascade corner pooling 分别得到 center heatmap 和 corner heatmaps，用来预测关键点的位置。得到角点的位置和类别后，通过 offsets 将角点的位置映射到输入图片的对应位置，然后通过 embeddings 判断哪两个角点属于同一个物体，以便组成一个检测框。CenterNet 不仅预测角点，还预测中心点。对每个预测框定义一个中心区域，通过判断每个目标框的中心区域是否含有中心点，若有则保留，并且此时框的 confidence 为中心点，左上角点和右下角点的 confidence 的平均，若无则去除，使得网络具备感知目标区域内部信息的能力，能够有效去除错误的目标框。

## 基于模板匹配的目标检测

### 一般过程

#### ➤ 简介：

模板匹配是根据已知模板图像到另一幅图像中寻找与模板图像相似的子图像。

#### ➤ 步骤：

1. 输入搜索图像和模板图像
2. 对搜索图像和模板图像进行预处理操作
3. 搜索图像和模板图像中有效信息的提取
4. 通过某种搜索策略进行匹配
5. 将匹配的结果输出



## 基于灰度的模板匹配算法

### 基于灰度的模板匹配算法

#### ➤ 简介：

基于灰度的模板匹配算法是基于模板与图像中最原始的灰度值来进行匹配的，它是模板匹配中最基本的匹配思想。

#### ➤ 步骤：

在原始图像区域中所有像素的灰度值都是被搜索图像的信息，利用被搜索图像与模板图像的灰度值信息建立模板图像与被搜索图像中子图像的相似性度量，再根据采用的搜索方法，查找能够将两幅图像相似性度量值最大或最小的参数。

#### ➤ 常用的相似性度量算法：

- MAD
- SSDA
- SATD

MAD

## 基于灰度的模板匹配算法——MAD

### ➤ 简介:

平均绝对差算法 (Mean Absolute Differences, 简称MAD算法), 它是Leese在1971年提出的一种匹配算法。

### ➤ 算法思路:

在搜索图S中, 以(i, j)为左上角, 取M×N大小的子图, 计算其与模板的相似度  
遍历整个搜索图, 在所有能够取到的子图中, 找到与模板图最相似的子图作为最终匹配结果。

### ➤ MAD算法的相似性测度公式如下

$$D(i, j) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N |S(i+s-1, j+t-1) - T(s, t)|$$
$$1 \leq i \leq m-M+1, 1 \leq j \leq n-N+1$$

## SSDA

## 基于灰度的模板匹配算法——SSDA

### ➤ SSDA算法: 序贯相似性检测算法 (Sequential Similiarity Detection Algorithm)

### ➤ 算法思路:

#### ① 定义绝对误差

$$\varepsilon(i, j, s, t) = |S_{i,j}(s, t) - \overline{S_{i,j}} - T(s, t) + \bar{T}|$$

其中, 带有上划线的分别表示子图、模板的均值:

$$\overline{S_{i,j}} = E(S_{i,j}) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N S_{i,j}(s, t) \quad \bar{T} = E(T) = \frac{1}{M \times N} \sum_{s=1}^M \sum_{t=1}^N T(s, t)$$

#### ② 设定阈值Th

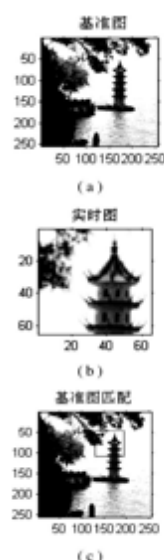
## 基于灰度的模板匹配算法——SSDA

### ➤ 算法思路:

- ③ 在模板图中随机选取不重复的像素点, 计算与当前子图的绝对误差, 将误差累加, 当误差累加值超过Th时, 记下累加次数H, 所有子图的累加次数H用一个表R(i, j)来表示。

$$R(i, j) = \left\{ H \mid \min_{1 \leq H \leq M \times N} \left[ \sum_{h=1}^H \varepsilon(i, j, s, t) \geq Th \right] \right\}$$

- ④ 在计算过程中, 随机点的累加误差和超过了阈值 (记录累加次数H) 后, 则放弃当前子图转而对下一个子图进行计算。遍历完所有子图后, 选取最大R值所对应的(i, j)子图作为匹配图像



## SATD

## 基于灰度的模板匹配算法——SATD

### ➤ 简介:

hadamard变换算法 (Sum of Absolute Transformed Difference, 简称SATD算法), 它是经hadamard变换再对绝对值求和算法

### ➤ 算法思路:

- ① 模板与子图做差后得到的矩阵Q, 再对矩阵Q求其hadamard变换 (左右同时乘以H, 即HQH), 对变换后得矩阵求其元素的绝对值之和即SATD值, 作为相似度的判别依据。对所有子图都进行如上的变换后, 找到SATD值最小的子图, 便是最佳匹配。
- ② hadamard变换等价于把原图像Q矩阵左右分别乘以一个hadamard变换矩阵H。其中, hadamard变换矩阵H的元素都是1或-1, 是一个正交矩阵。



## 基于深度学习的目标检测

### 基于深度学习算法的分类

#### ➤ One-Stage目标检测算法:

不需要Region Proposal阶段, 可以通过一个Stage直接产生物体的类别概率和位置坐标值, 比较典型的算法有YOLO、SSD和CornerNet.

#### ➤ Two-Stage目标检测算法:

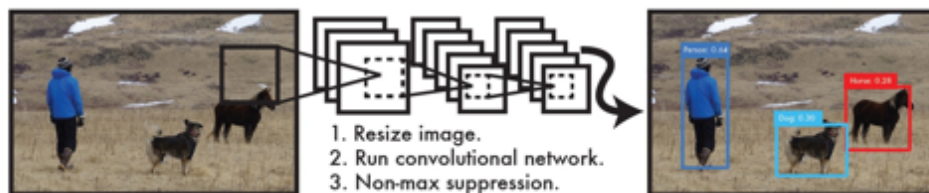
将检测问题划分为两个阶段, 第一个阶段首先产生候选区域 (Region Proposals), 包含目标大概的位置信息, 然后第二个阶段对候选区域进行分类和位置精修, 这类算法的典型代表有R-CNN, Fast R-CNN, Faster R-CNN等。

## One-Stage目标检测算法

### YOLO

#### YOLO系列算法

#### ➤ 使用单独一个CNN模型实现端到端目标检测



#### ➤ 过程:

输入的图像被调整为448x448 (为例), 然后发送到CNN网络。最后, 对网络预测结果进行处理, 得到检测目标。

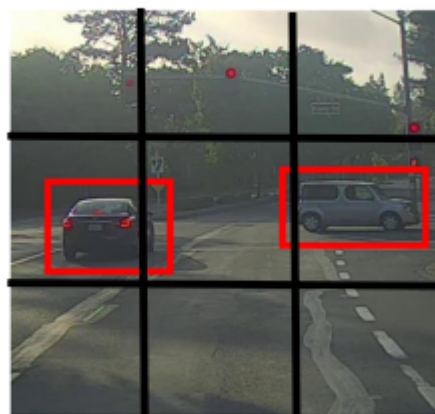
## YOLO系列算法

### ➤ 算法思路:

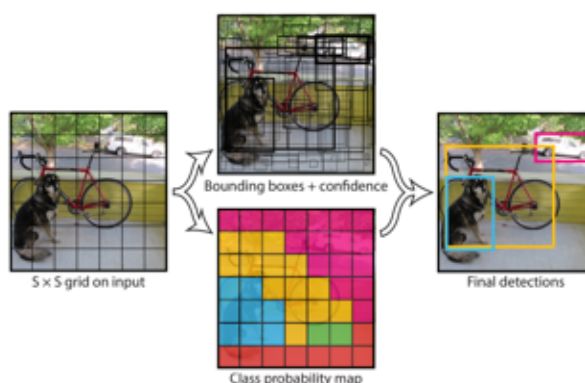
1. 将输入图像分割成一个 $[S * S]$ 图像网格, 每个cell负责检测中心点位于网格内的目标。

2. 每个cell可以预测B个边界框 (bbox) 和边界框的置信度。

- **bbox信息**  $(x, y, w, h)$  为物体的中心位置相对格子位置的偏移及宽度和高度, 均被归一化。
- **置信度** 反映是否包含物体以及包含物体情况下位置的准确性



## YOLO系列算法



➤ 每个cell需要预测  $(B*5+C)$  值。如果将输入图像划分为  $(S*S)$  网格, 则最终预测值为大小为  $S*S*(B*5+C)$  的张量

其中B为bounding box个数, 数字5分别为:  $(x, y, w, h)$  和置信度, C为类别。

## YOLO系列算法

**置信度 (confidence score)** 包含两项内容:

① bounding box包含目标的概率 (是否包含目标):

当bounding box为背景时 (不包含目标物体), 那么  $Pr(object) = 0$ .

当bounding box中包含目标时,  $Pr(object) = 1$

② bounding box中目标的准确度:

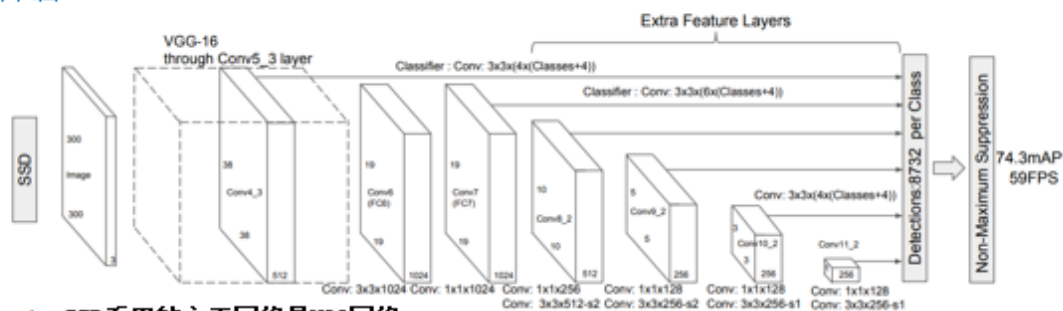
预测的 bounding box和实际的ground truth之间的 IoU 值

$$\text{置信度: } Pr(object) * IOU_{pred}^{truth}$$





## SSD算法



- SSD采用的主干网络是VGG网络
- 相比普通的VGG网络有一定的修改：
  - ①将VGG16的FC6和FC7层转化为卷积层。
  - ②去掉所有的Dropout层和FC8层；
  - ③新增了Conv6、Conv7、Conv8、Conv9。

## SSD算法

SSD的创新点和优缺点

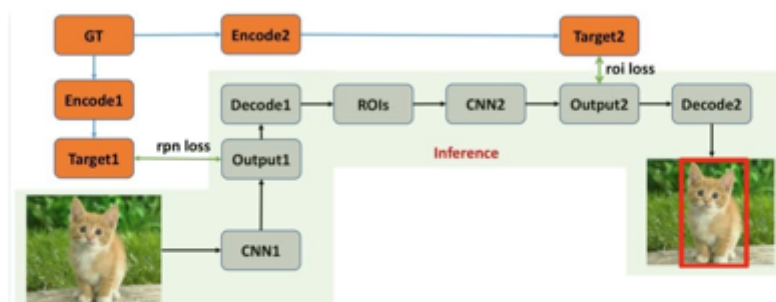
**SSD创新性地**在多个不同深度的特征图上进行**PriorBox**生成、分类、回归，最终融合得到最终的预测结果。这样做的好处是考虑到了不同深度的特征图具有大小不同的感受野，浅层特征图感受野小，适合做小目标检测，但同时浅层特征的语义层次较低，不能帮助网络很好地识别；深层特征具备高层次的语义，但是其感受野较大，适合做大目标检测，容易忽略小目标。SSD在深浅不同的特征图上进行回归，在保证大目标的检测准确率同时也兼顾了小目标的检测；

**提出PriorBox作为强先验知识。**PriorBox是在特征图上生成的一系列长宽比不一的长方形方框，用来进行物体位置的预测。作用相当于Anchor，在预测的过程中无需从零开始选择性搜索生成很多个Proposals，而是直接在设定好的PriorBox上进行调整，极大地加快了网络训练和推理的速度。

**SSD做了充分的数据增强**，包括光学变换和几何变换，为网络提供了充分的训练数据，在保证训练数据分布的前提下扩充了其丰富性，从而有效提升了检测精度。

## Two-Stage目标检测算法

### Two-Stage目标检测算法



- 在Testing的时候输入图片经过卷积神经网络产生第一阶段输出，对输出进行解码处理生成候选区域，然后获取对应候选区域的特征表示 (ROIs)，然后对ROIs进一步精化产生第二阶段的输出，解码 (后处理) 生成最终结果，解码生成对应检测框即可；在Training的时候需要将Ground Truth编码成CNN输出对应的格式以便计算对应损失loss。

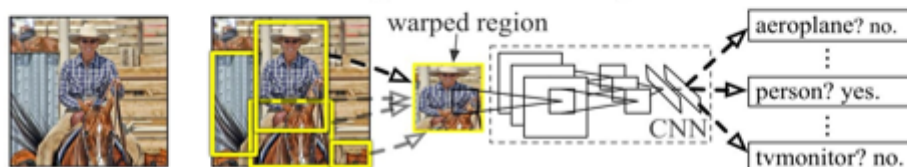
## RCNN

## RCNN算法

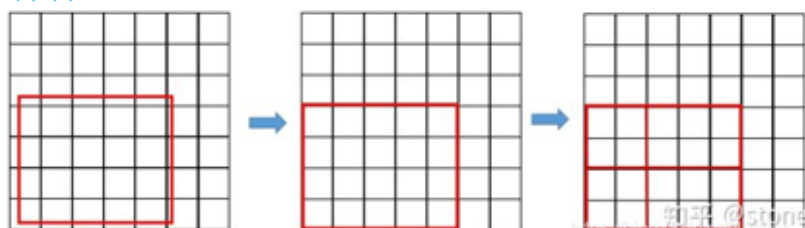
### 步骤：

- ① 一张图像生成1K~2K个候选区域(使用Selective Search方法)
- ② 对每个候选区域，使用深度网络提取特征
- ③ 特征送入每一类的SVM 分类器，判别是否属于该类
- ④ 使用回归器精细修正候选框位置

### R-CNN: Regions with CNN features



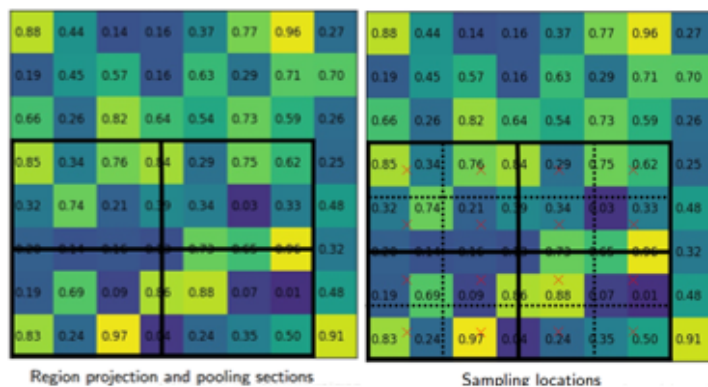
## Fast-RCNN算法



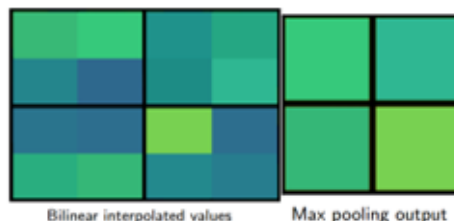
region proposal的xywh通常是小数，但是为了方便操作会把它整数化（取得最近的边界点）。将整数化后的边界区域平均分割成  $k \times k$  个单元，对每一个单元的边界进行整数化。两次整数化的过程如下图所示：事实上，经过上述两次整数化，此时的候选框已经和最开始回归出来的位置有一定的偏差，这个偏差会影响检测或者分割的准确度。在论文里，作者把它总结为“不匹配问题”（misalignment）。

为了解决这个问题，Mask RCNN（<https://arxiv.org/pdf/1703.06870.pdf>）提出了ROI Align方法，取消整数化操作，保留了小数，使用双线性插值的方法获得坐标为浮点数的像素点上的图像数值。

## Fast-RCNN算法



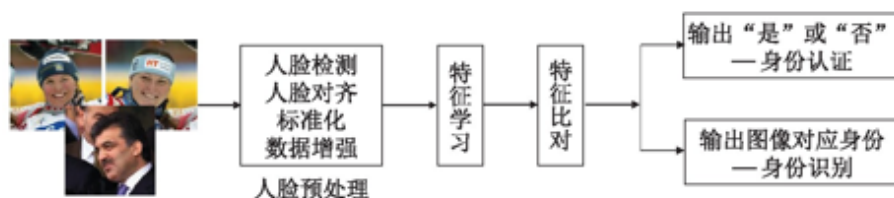
将ROI切成2x2的单元格（实线框）。如果采样点数是4，那我们首先将每个单元格子均分成四个小方格（虚线框），每个小方格中心就是采样点。这些采样点的坐标通常是浮点数，所以需要对该像素点进行双线性插值，就可以得到该像素点的值了。然后对每个单元格内的四个采样点进行maxpooling，就可以得到最终的ROIAlign的结果。



## 人脸识别



## 步骤:



### ➤ 图像预处理

- 人脸检测指检测出人脸图像中人脸的具体位置
- 人脸对齐指检测人脸特征点, 提取图像中感兴趣的点
- 对图像进行一些数值标准化的处理
- 数据增强

### ➤ 特征学习

- 对图像进行特征提取

### ➤ 特征比对

- 对比相似度, 进行人脸匹配

## 人脸识别——LBP

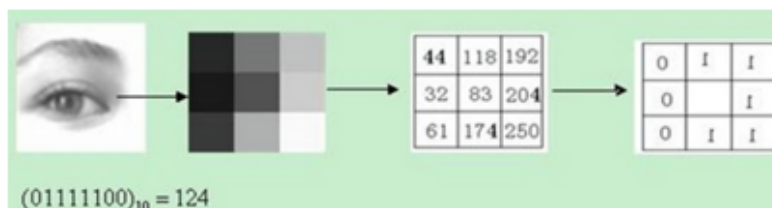


### ➤ LBP (Local Binary Pattern):

一种简单, 有效的纹理分类的特征提取算法。构造了一种衡量一个像素点和它周围像素点的关系。

#### ➤ 定义:

以邻域中心像素为阈值, 将相邻的8个像素的灰度值与其进行比较, 若周围像素值大于中心像素值, 则该像素点的位置被标记为1, 否则为0。这样,  $3 \times 3$ 邻域内的8个点经比较可产生8位二进制数(通常转换为十进制数即LBP码, 共256种), 即得到该邻域中心像素点的LBP值, 并用这个值来反映该区域的纹理信息。



## 图像行人再识别

### 图像行人再识别

#### ➤ 有监督方法:

通常被认为是使用有标签的训练数据对模型进行监督训练, 该类方法在行人再识别研究领域即为使用有ID标签的行人图片来训练行人再识别模型; 随后用训练好的模型从行人图片中提取出行人不变特征, 再对这些特征进行距离度量计算, 从而判断出哪些行人图片属于同一身份, 哪些行人图片属于不同身份。

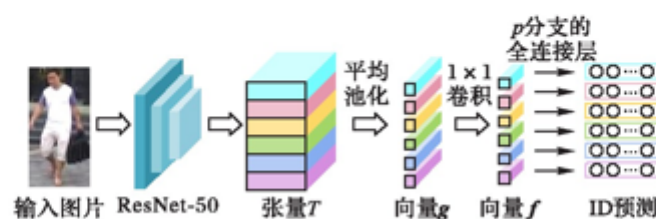
#### ➤ 无监督方法:

在实际的应用场景中, 待检索的目标域多为无标签的数据, 且标注工作异常困难。



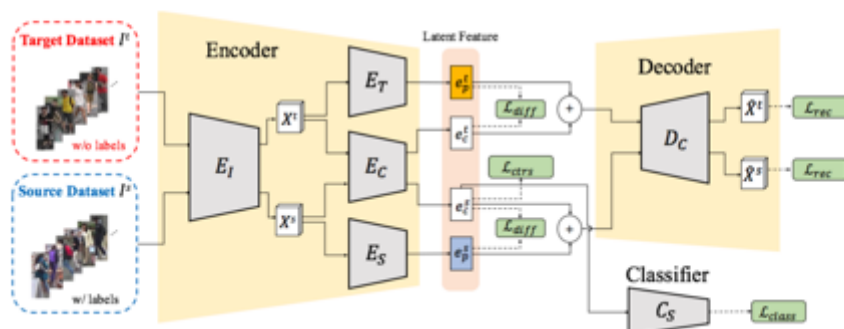
## 图像行人再识别——有监督方法

- 一种基于局部特征学习的卷积基线网络 (Part-based Convolutional Baseline, PCB)  
PCB以整幅图像作为基线网络Res Net-50输入，将所得的特征映射在卷积层上均匀划分为 $p$ 个部分，经过Softmax多分类损失函数训练得到 $p$ 个分类器。



## 图像行人再识别——无监督方法

- 提出的ARN (Adaptation and Reidentification Network) 框架  
利用数据集之间的信息来学习域不变特性，通过域不变特性来进行跨域行人匹配



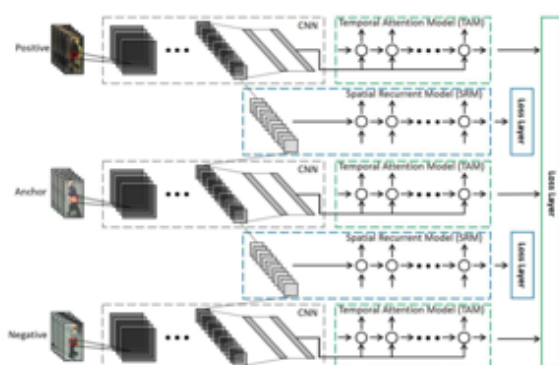
## 视频行人再识别

- 图像行人再识别 VS 视频行人再识别
- 基于图像的行人再识别方法只能从单个图像中获取有限的信息，并且在解决单个图像中遮挡、姿态变化和摄像机视角等问题上效果有限。
- 由于图像序列通常包含丰富的时间信息，因此更适合在复杂的环境和较大的几何变化下对行人进行再识别，而且基于视频序列的方法更符合实际监控系统中的行人再识别任务的需求，这样会避免一些监控视频的预处理过程。

## 视频行人再识别

### ➤ TAM(Temporal Attention Model):

一种时间注意力机制模型，用来提取表示视频中特定行人的判别帧。此外，还采用RNN模型SRM(Spatial Recurrent Model)从六个方向结合上下文信息，增强行人特征映射中每个行人的位置表示



## 跨模态行人再识别

### ➤ 问题:

- 在实际生活应用中，仅用可见光相机采集的单模态行人数据会出现画质模糊、分辨率低、光照变化大等问题，这将影响可见光相机捕捉的行人图像的质量，导致实际应用中检索精度不高。

### ➤ 方法:

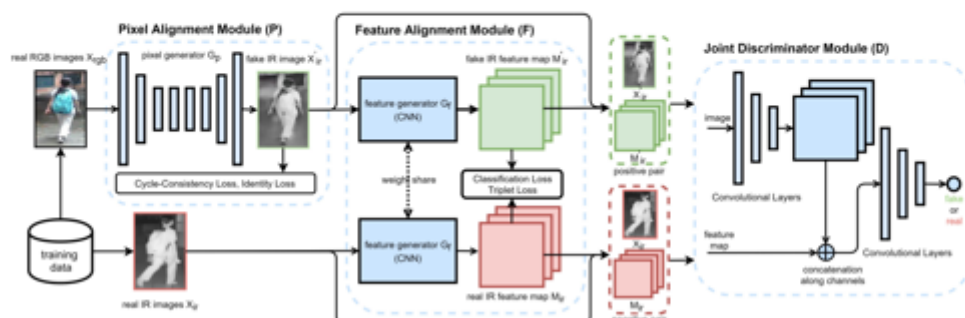
- 引入多模态行人数据。(如右图所示)



## 跨模态行人再识别

### ➤ 端到端对齐生成对抗网络Align GAN:

该模型可以联合利用像素对齐和特征对齐，缓解跨模态数据的变化，学习跨模态行人图像的不变特征。



## 图像增强

有目的地强调图像的整体或局部特性，将原来不清晰的图像变得清晰或强调某些感兴趣的特征，扩大图像中不同物体特征之间的差别，抑制不感兴趣的特征，使之改善图像质量、丰富信息量，加强图像判读和识别效果，满足某些特殊分析的需要

# 图像去雨算法

## 基于模型驱动算法

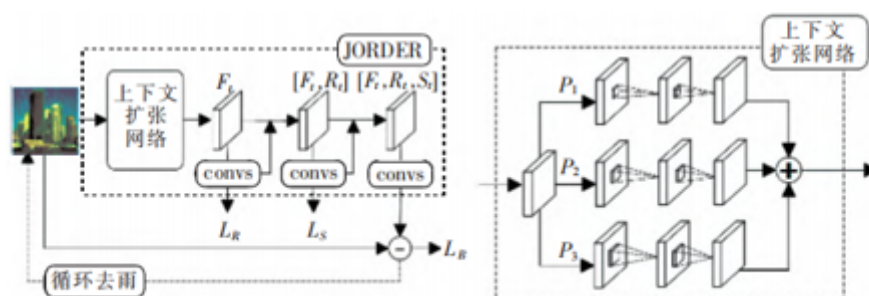
利用图像的先验知识，如雨条纹的方向、密度和尺寸等约束去雨问题及去雨模型，再通过设计优化算法进行求解，从而获得干净无雨的图像。

## 基于数据驱动算法

以深度学习为代表，通过构建神经网络，利用成对的雨条纹标签和干净无雨图像来学习有雨到无雨的非线性映射

### 2.1 卷积神经网络——JORDER (2017年)

基于暴雨合成模型构建一个用于雨水的检测与去除网络，使用上下文扩展网络来保留更多细节信息。



# 图像去雾

## 基于图像增强的去雾算法

通过图像增强技术突出图像细节，提升对比度。具体算法有：直方图均衡化算法、Retinex算法、小波变换算法、偏微分方程算法等；

## 基于图像复原的去雾算法

基于大气散射物理学模型，根据有雾图像的形成过程来进行逆运算，从而恢复清晰图像

## 基于深度学习的去雾算法

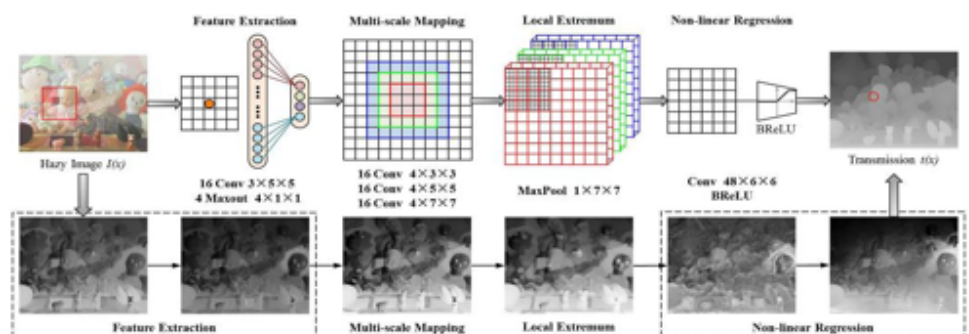
①基于大气散射模型，使用神经网络生成大气散射模型中的某些参数，从而复原出清晰无雾的图像；②直接端到端的生成清晰的无雾图像。

### 3、基于深度学习的去雾算法

#### 3.1 DehazeNet 算法, 2016年

$$I(x) = J(x)t(x) + A(1 - t(x))$$

将有雾的模糊图像作为输入, 输出其透射率 $t(x)$ , 基于大气散射模型理论恢复出无雾的清晰图像。



## 图像超分辨率

### 基于插值的图像重建方法

根据 LR 图像已知的像素点灰度信息, 运用插值公式增强像素点间的灰度信息来实现图像放大问题。

### 基于重构的图像重建方法

### 基于学习的重建方法

基于 CNN 网络模型的超分方法

基于生成对抗网络GAN网络模型的超分方法

## 视频目标跟踪

视频目标跟踪的目的是定位感兴趣目标在视频每一帧中的位置, 常分为单目标跟踪和多目标跟踪。多目标跟踪算法需要实时跟踪多个目标在视频帧中的位置, 需要确保同一个物体在视频中获得固定的、唯一的数字标识。单目标跟踪算法要求处理任意类别的物体, 即不知道任何关于目标的先验信息。

目前, 目标跟踪技术主要应用于以下领域:

- 无人驾驶
- 智能视频监控
- 机器人视觉导航
- 虚拟现实 (人体跟踪)
- 医学诊断 (细胞状态跟踪)

### 背景差分法:

背景差分法首先需要构建好背景图像, 然后将当前图像与背景图像作差, 之后再处理差值图像, 从而得到最终的前景图像。该方法的关键在于构建合适的背景模型, 自然场景给背景建模带来了很大挑战, 因为它们是动态的, 例如, 光照变化、极端天气条件等, 即便在静态环境下, 由于噪声、摄像机抖动等, 图像之间也存在着差异, 所以很多学者对此进行了研究。Stauffer与Grimson提出了混合高斯背景建模算法, 为每个像素点建立多个高斯分布, 从而能够应对背景多峰分布的情况, 但是也存在一些缺点, 如: 参数的选择问题、更新速率问题、光照突变造成的大面积误检问题等; Barnich与Droogenbroeck提出了Visual Background Extractor (ViBe), 利用单帧图像来构建背景模型, 并且采用随机选择策略、空间信息传播方法来更新该模型, 该方法的实时性比较好, 但是容易出现“鬼影”、阴影等问题。



### 光流法：

光流法的基本思想可以概括为在不变的光流场中寻找相对改变的光流场，而改变的部分即为运动目标所在部分。前提是假设场景中的亮度信息不改变，然后求出连续几幅图像帧间的像素点的运动矢量，即光流场。对于背景区，由于改变较少，一般会产生比较均匀的运动矢量场，而对于运动对象，其运动矢量场则不是很规则，从而据此检测出运动目标。光流法的优点是可以在不预先知道场景的任何信息的情况下进行运动目标检测，缺点是由于涉及大量的矩阵和迭代运算，难以实现实时的运动目标检测。

### 相邻帧间差分法：

相邻帧差分法又称为时间差分法，其基本思想是利用相邻的两帧或者几帧作差，求得的差值图像即为检测结果。时间差分法是在运动目标检测中使用的比较多的算法。由其基本思想可知，这种方法适用于相邻的几帧变化不大的情况，对于光照或者树枝摆动等变化效果并不好。

## 图像合成与生成

---

图像合成 (image composition) 是图像处理的常用操作，把前景从一张图上剪贴下来粘贴到另一张图上，获得一张合成图。合成图可以用来获取感兴趣的目标图像，也可以用于数据增广，有着广泛的应用的前景。

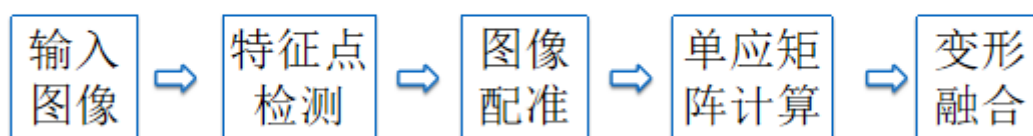
图像生成 (image generation) 是指根据给定的信息，生成特定的图像。这里的信息可以使噪声、图像、文字、音频等各种类型的信息。

### 图像拼接

图像拼接是将同一场景的多个重叠图像拼接成较大的图像的一种方法，在医学成像、计算机视觉、卫星数据、军事目标自动识别等领域具有重要意义。



## 图像拼接的一般流程



### 特征点检测

特征点检测：在所有输入图像中检测特征点，是要匹配的两个输入图像中的元素，它们是在图像块的内部像素组。如下图所示，fig1和fig2给出了一个很好的patch匹配，因为fig2中有一个patch看起来和fig1中的patch非常相似。当我们考虑到fig3和fig4时，这里的patch并不匹配，因为fig4中有很多类似的patch，它们看起来与fig3中的patch很相似。由于像素强度很相近，所以无法进行精确的特征匹配。



为了给图像对提供更好的特征匹配，采用角点匹配，进行定量测量。角点是很好的匹配特性。在视点变化时，角点特征是稳定的。此外，角点的邻域具有强度突变。利用角点检测算法对图像进行角点检测。角点检测算法有Harris角点检测算法、SIFT特征点检测算法(Scale Invariant Feature Transform),FAST算法角点检测算法，SURF特征点检测算法(Speeded-up robust feature)

### 图像配准

图像配准：

在特征点被检测出来之后，需要以某种方式将不同图像中的特征点关联起来,可以通过NCC或者SDD（Sum of Squared Difference）方法来确定其对应关系。

归一化互相关（normalized cross correlation, NCC）：互相关的工作原理是分析第一幅图像中每个点周围的像素窗口，并将它们与第二幅图像中每个点周围的像素窗口关联起来。将双向相关性最大的点作为对应的对。基于图像强度值计算在两个图像中的每个位移（shifts）的“窗口”之间的相似性

$$NCC(u) = \frac{\sum_i [I_1(x_i) - \bar{I}_1][I_2(x_i + u) - \bar{I}_2]}{\sqrt{\sum_i [I_1(x_i) - \bar{I}_1]^2 [I_2(x_i + u) - \bar{I}_2]^2}}$$

NCC峰值相对应的位移参数表示两个图像之间的几何变换。此方法的优点是计算简单，但是速度特别慢。此外，此类算法要求源图像之间必须有显著的重叠。

图像配准:

互信息 (Mutual Information, MI), 测量基于两个图像之间共享信息数量的相似性。

$$M(I_1, I_2) = E(I_1) + E(I_2) - E(I_1, I_2)$$

$E(I_1)$ 、 $E(I_2)$ 为两幅图像的熵,  $E(I_1, I_2)$ 为两幅图像的联合熵,

$$E(I_1) = - \sum_g p_{I_1}(g) \log(p_{I_1}(g))$$

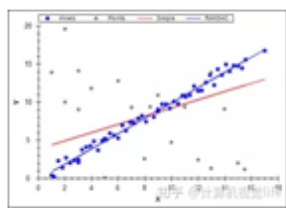
$$E(I_1, I_2) = - \sum_{g,h} p_{I_1, I_2}(g, h) \log(p_{I_1, I_2}(g, h))$$

## 单应矩阵

计算单应矩阵:

单应矩阵估计是图像拼接的第三步。在单应矩阵估计中, 不属于重叠区域的不需要的角被删除。采用RANSAC算法进行单应。

RANSAC算法从可能含有异常值的观测数据集中拟合数学模型, 是一种鲁棒参数估计的迭代方法。该算法是不确定性的, 因为它只在一定的概率下产生一个合理的结果, 随着执行更多的迭代, 这个概率会增加。RANSAC算法用于在存在大量可用数据外行的情况下以鲁棒的方式拟合模型。



单应矩阵 (Homography):

有了两组相关点, 接下来就需要建立两组点的转换关系, 也就是图像变换关系。单应性是两个空间之间的映射, 常用于表示同一场景的两个图像之间的对应关系, 可以匹配大部分相关的特征点, 并且能实现图像投影, 使一张图通过投影和另一张图实现大面积的重合。

设两个图像的匹配点分别是  $X = [u_1, v_1]^T$ ,  $X' = [u_2, v_2]^T$ , 则必须满足

$$X' = HX$$

由于两个向量共线, 所以

$$X' \times HX = 0$$

其中, H为单应矩阵, 由8个参数组成

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \propto \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

## 图像变形和融合



图像变形和融合：

最后一步是将所有输入图像变形并融合到一个符合的输出图像中。可以简单地将所有输入的图像变形到一个平面上，这个平面名为复合全景平面。

图像变形步骤

首先计算每个输入图像的变形图像坐标范围，得到输出图像大小，可以很容易地通过映射每个源图像的四个角并且计算坐标(x,y)的最小值和最大值确定输出图像的大小。最后，需要计算指定参考图像原点相对于输出全景图的偏移量的偏移量xoffset和偏移量yoffset。

下一步是使用上面所述的反向变形，将每个输入图像的像素映射到参考图像定义的平面上，分别执行点的正向变形和反向变形。

图形融合

最后一步是在重叠区域融合像素颜色，以避免接缝。最简单的可用形式是使用羽化（feathering），它使用加权平均颜色值融合重叠的像素。使用alpha因子，通常称为alpha通道，它在中心像素处的值为1，在与边界像素线性递减后变为0。当输出拼接图像中至少有两幅重叠图像时，将使用如下的alpha值来计算其中一个像素处的颜色：假设两个图像  $I_1, I_2$ ，在输出图像中重叠的每个像素点

$$I_i(x, y) = (\alpha_i R, \alpha_i G, \alpha_i B, \alpha_i)$$

其中 (R,G,B) 是像素的颜色值，在缝合后的输出图像中计算(x, y)的像素值

$$[(\alpha_1 R, \alpha_1 G, \alpha_1 B, \alpha_1) + (\alpha_2 R, \alpha_2 G, \alpha_2 B, \alpha_2)] / (\alpha_1 + \alpha_2).$$

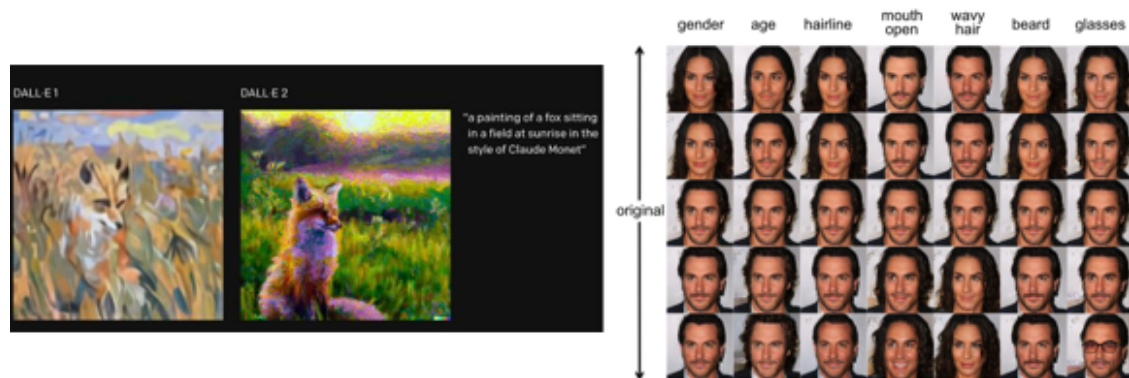
## 图像合成

图像合成是根据输入的两幅或多幅图像，生成一幅新的虚拟图像。



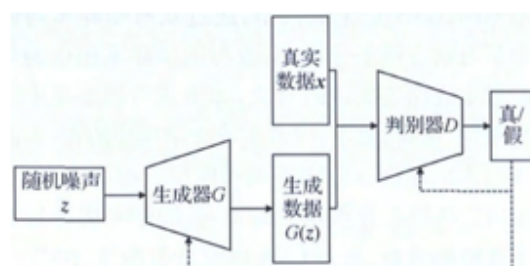
## 图像生成

图像生成是从现有数据集生成新图像的任务，与图像合成类似，但其适用范围更广，其输入可以是噪声、图像、文本，也可以是多模态数据。



当前图像生成主要是以生成对抗网络为主（GAN），GAN 包括两个模型，生成模型(G)和判别模型(D)。生成器 G 和判别器 D 本质上都是函数，通常用深层神经网络来实现。GAN 模型结构如图所示，G 从真实样本中捕获数据分布映射到某个新的数据空间，输出生成的数据记作  $G(z)$ ，其分布记作  $p_g(z)$ ，并尽量使其看上去和训练集中样本  $p_r(x)$  一样。D 的输入包括真实数据  $x$  与生成数据  $G(z)$ ，输出是一个概率值或一个标量值，表示 D 认定输入是真实分布的概率，数值越大，是真实数据的概率越大，反之认为输入的是生成样本，D 根据输出反馈给 G，使 G 生成的数据与真实数据逐渐一致

$$\min_G \max_D V(D, G) = E_{x \sim p_r(x)} [\ln D(x)] + E_{z \sim p_g(z)} [\ln (1 - D(G(z)))]$$



## 前沿

### 小样本学习

#### FSL问题定义

- **定义：**小样本学习是指，给定一个有特定于任务  $T$  的包含少量可用的有监督信息的数据集  $D_T$  和与  $T$  不相关的辅助数据集  $D_A$ ，小样本学习的目标是为任务  $T$  构造函数  $f$ ，该任务的完成利用了  $D_T$  中很少的监督信息和  $D_A$  中的知识，完成将输入映射到目标的任务。
- 上述定义中与  $T$  不相关的术语表示  $D_T$  和  $D_A$  中的类别是正交的，即  $Y_A \cap Y_T = \emptyset$ 。如果  $D_A$  覆盖了  $T$  中的任务，即  $Y_A \cap Y_T = Y_T$ ，则FSL问题将崩溃为传统的大样本学习问题。

判别模型、生成模型

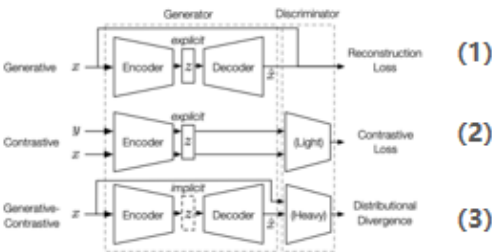
背景和意义：

深度监督学习在过去十年中取得了巨大的成功。然而，它严重依赖手动标签和易受攻击的缺陷促使人们寻找其他模式。作为一种替代方法，自监督学习（SSL）在过去几年中以其在表征学习方面的出色表现吸引了许多研究人员。

自监督表征学习利用输入数据本身作为监督，并使几乎所有类型的下游任务受益。在过去几年中以其在表征学习方面的出色表现吸引了许多研究人员。自监督表征学习利用输入数据本身作为监督，并使几乎所有类型的下游任务受益。

自我监督学习的成功最关键的一点是，它找到了一种利用大数据时代大量未标记数据的方法！

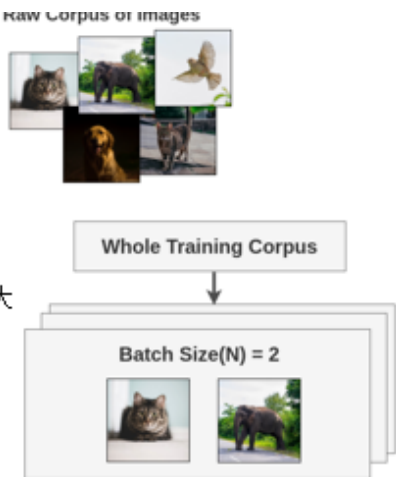
主流的自监督学习分为三类，如下图所示：



- (1) 生成式：训练编码器将输入编码为**显式向量**，训练解码器从 $z$ 重建 $x$ （例如完形填空测试、图形生成）。
- (2) 对比式：训练编码器将输入编码为**显式向量**，以衡量相似性（例如，互信息最大化、实例区分）。
- (3) 生成对比式（对抗性）：训练编码器生成假样本，并训练鉴别器将其与真实样本（如GAN）区分开来。

① 相似和不同图像的样本

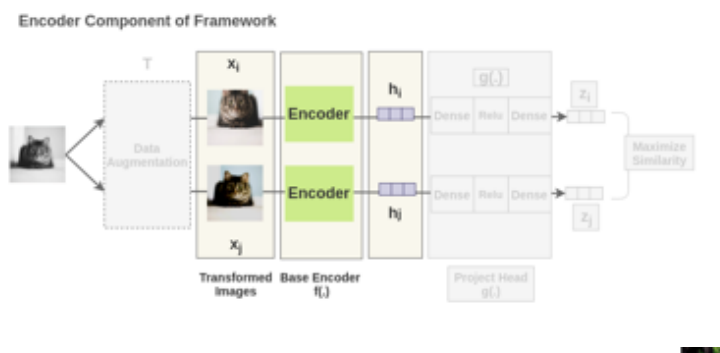
数据：假设有一个包含数百万未标记图像的训练库。



从原始图像生成批大小为 $N$ 的batch。为了简单起见，取一批大小为 $N = 2$ 的数据。

## ② 了解图像所表示的内容的能力

每一对中的增强过的图像都通过一个编码器。如（如下图）来获得图像表示。所使用的编码器是通用的，可与其他架构替换。下面显示的两个编码器有共享的权值，我们得到向量 $h_i$ 和 $h_j$ 。



## ③ 量化两个图像是否相似的能力

- a) 用余弦相似度计算图像的两个增强的图像之间的相似度。对于两个增强的图像 $x_i$ 和 $x_j$ ，在其投影表示 $z_i$ 和 $z_j$ 上计算余弦相似度。

Similarity Calculation of Augmented Images

$$\text{similarity}(x_i, x_j) = \text{cosine similarity}(z_i, z_j)$$

## (2) 基于生成对比式（对抗性）的自监督学习方法：

生成对比式特征学习，或者用一个更熟悉的名字叫对抗性特征学习。

在对比学习方面，对抗性方法仍然保留了由编码器和解码器组成的生成器结构，相反，对比方法放弃解码器组件。生成器是至关重要的，一方面，生成器赋予对抗性学习以生成模型特有的强大表达能力；另一方面，它也使得对抗性方法的目标比对比性方法的目标更具挑战性，导致不稳定的趋同。

在对抗性方面，解码器的存在要求特征是“重建的”，换句话说，它包含构建输入所需的所有信息。然而，在对比环境中，我们只需要学习“可分辨”信息来区分不同的样本。

综上所述，对抗性方法吸收了生成性方法和对比性方法的优点，同时也存在一些缺点。在需要适应隐式分布的情况下，这是一个更好的选择。

## 图像描述

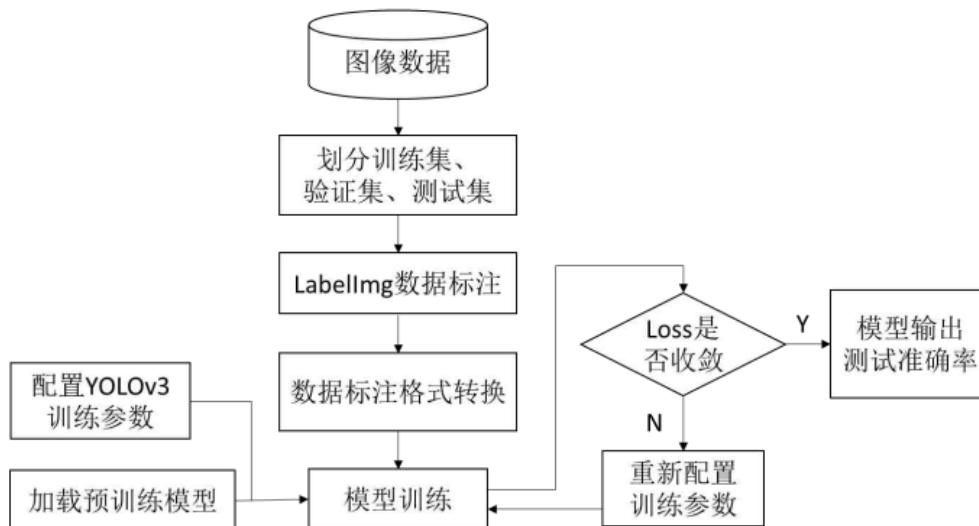


图 4-5 基于 YOLOv3 算法的目标检测流程

YOLOv3 算法检测流程如图 4-5 所示，首先将构建的图像数据划分为训练集、验证集和测试集；然后采用 LabelImg 图形化标注软件对数据集目标物体的位置信息与所属分类进行标注，并保存在 xml 文件中；接着将生成的 xml 文件数据格式转换成 YOLOv3 自定义的 txt 格式；然后，加载预训练模型和配置训练参数即可对训练集和验证集进行模型训练。在训练时观察损失函数输出是否收敛，最后对输出的模型进行测试。实验中将训练两个数据集模型，分别用于提取读数区域和识别表盘读数。

#### 4.3.2 数据集制作

本文采用 LabelImg 标注工具<sup>[41]</sup>来进行数据集制作，该工具可以通过图形化操作框出目标物体的所在矩形区域，同时标注目标所属分类，并把矩形位置和分类等数据保存在 xml 文件中以便读取与解析。

如图 4-6 所示为制作用于提取读数区域的数据集，通过 LabelImg 软件标出目标区域的位置，位置信息以左上角横纵坐标 (xmin, ymin), 右下角横纵坐标 (xmax, ymax) 的形式定义，并对区域打上 box 标签，最后将数值保存在 xml 文件中。同理，如图 4-7 所示

为制作用于读数识别的数据集，通过软件标出数字区域并对其打上 0-9 的分类标签，最后将数值存储到文件中。

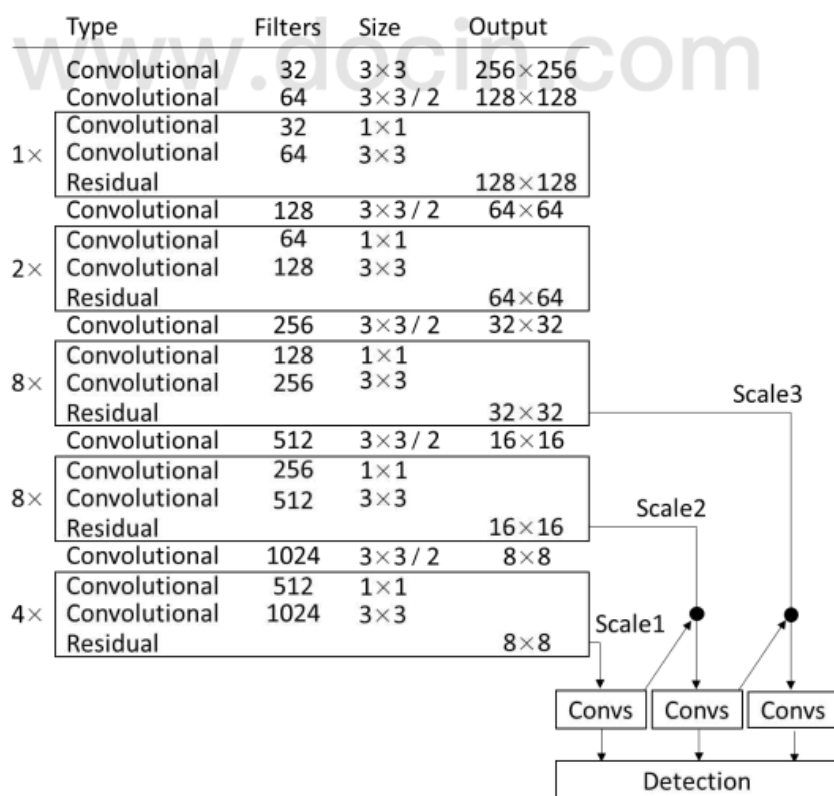


图 4-3 YOLOv3 网络结构图

YOLOv3 算法通过先验框对不同尺寸的特征图上每个单元格进行逻辑回归时，每个格子均会预测三个边界框。边框属性包括边框位置坐标（中心点横纵坐标以及边框宽高）、边框置信度、边框分类概率。算法通过逻辑回归预测每个边界框的目标分数，其损失函数如公式（4-4）所示。其中， $S \times S$  是特征图单元格个数， $B$  是每个单元格的预测边界框个数， $I_{ij}^{obj}$  代表第  $i$  个单元格的第  $j$  个边界框是否负责目标物体，其中  $IoU$  最大的边界框视作负责，其值置为 1，否则置为 0。 $I_{ij}^{noobj}$  的取值则相反。最终，损失函数由中心坐标误差  $\left[ \left( x_i^j - \hat{x}_i^j \right)^2 + \left( y_i^j - \hat{y}_i^j \right)^2 \right]$ 、宽高坐标误差  $\left[ \left( \sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left( \sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right]$ 、边框置信度误差  $\left( C_i^j - \hat{C}_i^j \right)^2$  和分类误差  $\left( P_i^j(c) - \hat{P}_i^j(c) \right)^2$  组成。