# Unsharp Mask (USM) 算法

## 算法原理

在任何图像中低频对应大面积缓慢变化的亮度（图像背景），高频对应像素值变化剧烈的细节（纹理、边缘、噪声）。

1. 先做一次平滑：用高斯核 Gσ 对原图 I 低通滤波，得到模糊版本 B = Gσ(I)。这一步几乎抹掉高频，只保留低频信息。

2. 形成细节掩膜：计算差分 M = I - B。由于 B 近似只含低频，相减后保留下来的正是高频分量（边缘与纹理）。

3. 按比例放大后相加：把掩膜乘以权重 k，再加回原图。公式 Is = I + k·(I - B) = (1 + k)·I - k·B，这样就得到边缘增强、整体亮度几乎不变的锐化结果。

可把整个过程看成“先削弱大块色调，再把被削弱的量按倍数返还给图像”——实质是一种高提升（high‑boost）滤波器。

## 数学表达式

设原始图像为 I，低通模糊结果为 Gσ(I)。USM 的标准公式为：

                    Iₛ = (1 + k) · I  –  k · Gσ(I)

其中 k∈[0, +∞) 为锐化强度（amount）。k=0 时无锐化；k≈0.5–2.0 为常用范围。当 k=1 时，Iₛ = 2I – Gσ(I)，亦称“高提升滤波”。

• kernel\_size / σ（半径）——决定模糊程度：半径越大，滤掉的低频范围越宽，可突出更大尺度的边缘；过大可能产生光晕。

• amount（k）——细节增强倍数：值大锐化更强，但易产生噪声与锯齿。

• threshold（M）——对 |I – Gσ(I)| 的阈值抑制：仅当局部对比度超过阈值才应用锐化，可避免在平坦区域放大噪声。

## OpenCV实现

1) 读取图像并转换为浮点类型；

2) 用 cv.GaussianBlur 进行高斯模糊，核大小可设 (0,0) 由 σ 自动推断；

3) 利用 cv.addWeighted 线性融合：dst = α·src + β·blur + γ，其中 α = 1+k, β = –k；

4) Clip 到 0–255 并转换回 uint8。