

# 分数阶优化的鸽群图像分割算法

李光昊 马瑜\* 郭姝琪 王仕儒

(宁夏大学物理与电子电气工程学院 宁夏回族自治区银川市 750021)

**摘要:** 本文针对原始鸽群优化的二维最大类间方差法 (Otsu) 图像分割算法收敛速度慢的缺点, 提出了基于混合分数阶鸽群优化的 Otsu 图像分割算法。算法将分数阶与鸽群优化算法结合, 利用分数阶微积分的记忆性和遗传特性, 使鸽群依据更多历史记忆获取更精确最优值, 并进行分数阶增强滤波预处理, 提高分割精度, 引入混沌搜索算法增强种群多样性, 加快收敛速度。改进算法利用 Otsu 算法的最大类间方差作为适应度函数来获取最佳阈值, 进行图像分割。实验结果验证了本文算法的正确性和有效性。

**关键词:** 鸽群算法; 最大类间方差法; 分数阶微积分; 图像分割

## 1 引言

在计算机视觉中, 图像分割是图像处理的重要组成部分, 其目的是将目标区域和背景区域分离。阈值分割是一类常用的图像分割方法, 其中最大类间方差法 (Otsu) 因其原理简单、易于实现, 被广泛应用于图像分割。二维 Otsu 算法相对原始 Otsu 算法很好地提升了分割效果<sup>[1]</sup>, 但计算耗时长, 收敛速度慢。张建波<sup>[2]</sup>等人利用二维灰度梯度直方图对算法进行优化, 有效降低了运算复杂度, 提高了分割精度。李鹏<sup>[3]</sup>等人利用麻雀算法优化 Otsu 分割算法, 大幅提升了算法效率。

仿生群体智能优化算法是通过模拟自然界中事物或生物的运动和行为规律所提出的一种算法, 常用来解决寻优、参数优化等问题, 但也存在收敛速度慢、易陷入局部最优等问题。鸽群算法是一种受鸽子归巢行为启发而产生优化算法 (pigeon-inspired optimization, PIO), 最早由段海滨<sup>[4]</sup>教授提出, 近年来在群体智能优化领域受到广泛应用<sup>[5]</sup>。林娜<sup>[6]</sup>等提出了自适应权重的鸽群算法, 一定程度上改善了收敛速度慢的缺陷。

近年来, 分数阶微积分作为一种数学工具, 被广泛应用于图像处理。2010年, Pires<sup>[7]</sup>等最早将分数阶算法引入 PSO 算法中, 验证了算法的收敛速度与分数阶阶次密切相关。魏晶茹<sup>[8]</sup>等提出了一种自适应的分数阶粒子群算法, 利用分数阶粒子群算法进行图像分割, 取得了很好的分割效果。Yashar Mousavi<sup>[9]</sup>等将分数阶与萤火虫算法结合, 同样获得了较好的实验结果。

本文针对鸽群算法在图像分割中具有收敛速度慢和寻优精度低的缺点, 将分数阶微积分引入到鸽群算法中, 对鸽群算法的速度更新进行优化, 克服其容易陷入局部最优的缺陷。并引入分数阶增强滤波对图像进行预处理, 提高分割精度。在鸽群算法寻优中引入混沌搜索算法, 提高全局搜索能力, 结合二维 Otsu 分割算法确定最佳分割阈值, 加快算法收敛速度。

## 2 分数阶混合鸽群算法

### 2.1 传统鸽群算法

鸽群优化算法是模仿鸽群归巢而建立的一类仿生优化方法。鸽群归巢过程中, 会根据情况使用不同的导航方式, 当鸽群距离巢穴较远时主要采用磁场导航, 而接近巢穴时则主要根据地标对路线进行修正。

#### 2.1.1 磁场算子

在磁场算子阶段, 鸽群的搜索具有全局最优特性。鸽群在多维搜索空间中, 位置和速度根据下式进行更新:

$$V(t)=V(t-1)\times e^{-Rt}+d\times(X_{\text{gbest}}-X(t-1)) \quad (1)$$

$$X(t)=X(t-1)+V(t) \quad (2)$$

上式中,  $R$  为磁场因子,  $R\in(0,1)$ 。  $d$  为取值范围在  $(0,1)$  的随机数。  $X_{\text{gbest}}$  为比较所有鸽子的位置后得到的全局最优位置。当该循环次数达到所要求的迭代次数后, 即停止磁场算子阶段的迭代, 进入地标算子中继续工作。

#### 2.1.2 地标算子

在地标算子阶段, 鸽子的搜索具有局部快速收敛特性, 鸽群的位置按下式进行更新:

$$X_c(t)=\frac{\sum X(t)\times f(X(t))}{N_p\times\sum f(X(t))} \quad (3)$$

$$X(t)=X(t-1)+d\times(X_c(t)-X(t-1)) \quad (4)$$

上式中,  $X_c$  是位于中心的鸽子的位置,  $f(X(t))$  为当前鸽子的适应度函数值, 即目标函数。

舍弃函数为:

$$N_p(t)=\frac{N_p(t-1)}{2} \quad (5)$$

在依靠地标算子寻优的过程中, 适应度低的鸽子对地标不熟悉, 具有较差的寻优能力, 因而被舍去, 其余鸽子跟随高适应度的鸽子进行移动, 每次迭代都会舍弃一半的鸽子数量。

### 2.2 分数阶微积分

分数阶微积分即为阶次是分数的微分和积分, 是对传统

●基金项目: 国家自然科学基金项目 (62041108)。

微积分的延伸<sup>[10]</sup>，由于其具有记忆性，在数字信号处理、数字图像处理等领域具有广泛的应用。分数阶微积分有多种定义，其中最常用的是 G-L (Grumwald-Letniko) 定义<sup>[11]</sup>。

假设在区间  $[a,b]$  ( $a < b, a \in \mathbb{R}, b \in \mathbb{R}$ ) 内存在连续可导的函数  $f(x)$ ，则  $f(x)$  的一阶导数为：

$$f'(x) = \lim_{\omega \rightarrow 0} \frac{f(x+\omega) - f(x)}{\omega} \quad (6)$$

其中， $\omega$  为区间  $[a,b]$  上的增量，由上式进一步推导可得函数的  $n$  阶导数为：

$$f^n(x) = \lim_{\omega \rightarrow 0} \frac{1}{\omega^n} \sum_{k=0}^n (-1)^k \binom{n}{k} f(x - k\omega) \quad (7)$$

其中  $n \in \mathbb{N}$ 。

将上式的阶次从整数阶延伸到分数阶。当阶次  $v > 0$  时，由 G-L 定义的  $v$  阶分数阶表达式为：

$${}_a^G D_t^v f(x) = \lim_{\omega \rightarrow 0} \omega^{-v} \sum_{k=0}^{\lfloor \frac{n-m}{\omega} \rfloor} (-1)^k \frac{\Gamma(v+1)}{k! \Gamma(v-k+1)} f(x - k\omega) \quad (8)$$

上式中， $\Gamma(N) = \int_0^\infty e^{-t} t^{N-1} dt = (N-1)!$  为 Gamma 函数； $D$  表示的是分数阶微分， $G$  表示 G-L 定义。

根据式 (8)，可以得到由 G-L 定义的信号  $f(x)$  的  $v$  阶积分的近似微分表达式：

$$\begin{aligned} \frac{d^v f(x)}{dt^v} &\approx f(x) + (-v)f(x-1) + \frac{(-v)(-v+1)}{2} f(x-2) \\ &+ \frac{(-v)(-v+1)(-v+2)}{6} f(x-3) + \dots \\ &+ \frac{\Gamma(-v+1)}{n! \Gamma(-v+n+1)} f(x-n) \end{aligned} \quad (9)$$

### 2.3 分数阶优化的鸽群算法

原始鸽群算法存在收敛速度慢，容易陷入局部最优的缺点，为了提升寻优效率和精度，将分数阶微积分引入鸽群算法中，提出分数阶优化鸽群算法 (fractional-order pigeon-inspired optimization, FPIO)。该算法将分数阶微积分引入原始鸽群算法速度更新公式，利用分数阶微积分的记忆性和遗传性提升鸽群算法的收敛速度，并且能够有效的改善原始鸽群容易陷入局部最优的问题。将式 (1) 变形为：

$$V(t) - V(t-1) = V(t-1) \times e^{-Rt} + d \times (X_{gbest} - X(t-1)) - V(t-1) \quad (10)$$

仅取式 (9) 的前四项，可得：

$$V(t+1) \approx \alpha V(t) + \alpha(1-\alpha)V(t-1)$$

$$\begin{aligned} &-\frac{1}{6}\alpha(1-\alpha)(2-\alpha)V(t-2) \\ &+\frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)V(t-3) \end{aligned} \quad (11)$$

当上式分数阶次  $\alpha$  取 1 时，可将式 (10) 与式 (11) 结合，得到分数阶鸽群算法的速度更新公式：

$$V(t+1) = (\alpha-1)V(t) + \alpha(1-\alpha)V(t-1)$$

$$\begin{aligned} &-\frac{1}{6}\alpha(1-\alpha)(2-\alpha)V(t-2) \\ &+\frac{1}{24}\alpha(1-\alpha)(2-\alpha)(3-\alpha)V(t-3) \\ &+V(t) \times e^{-Rt} + d \times (X_{gbest} - X(t)) \end{aligned} \quad (12)$$

由于分数阶对算法有很大的影响，仅采用固定的分数阶次难以达到良好的效果，因此本文根据鸽群的位置和速度信息自适应地对分数阶次  $\alpha$  进行调整。

鸽子  $i$  与其他鸽子的平均距离为：

$$d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^N \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2} \quad (13)$$

上式中， $N$  为鸽群总数， $D$  为空间维数。

则进化因子  $f$  为：

$$f = \frac{d_g - d_{min}}{d_{max} - d_{min}} \quad (14)$$

上式中， $d_g$  是鸽群的全局最优位置与其他鸽子距离的平均值，鸽子间平均距离最大值为  $d_g$ ，最小值为  $d_g$ ，则  $f \in [0,1]$ 。根据文献<sup>[9]</sup>可以得到，当分数阶次  $\alpha \in [0.5, 0.8]$  时，算法效果最好，因此可将  $\alpha$  根据下式进行动态调整：

$$\alpha(f) = \frac{1}{2} e^{-0.47f} \quad (15)$$

### 2.4 混沌搜索算法

混沌是一种无规则的运动状态，因其具有随机性、遍历性和规律性，常用于寻优问题。钱伟懿<sup>[12]</sup>等人利用混沌序列引入群体智能算法的研究，提高了算法的搜索速度。李娟<sup>[13]</sup>等人利用混沌序列对种群进行初始化，提高了种群的搜索能力。

本文所采用的混沌映射为 Logistic 映射，其表达式为：

$$Z_{i+1} = \mu Z_i (1 - Z_i), \quad i=1,2,\dots \quad (16)$$

其中， $\mu$  为控制参量， $\mu \in (2, 4]$ 。当  $\mu=4$  且  $0 \leq Z_0 \leq 1$  时，系统处于完全混沌状态。

针对原始鸽群算法搜索效率较低的缺陷，本文将混沌搜索引入鸽群算法迭代过程中，提高种群多样性，提高算法效率。混沌搜索的位置变量按下式更新<sup>[14]</sup>：

$$X(t+1) = X_{min} + Z_{i+1} \times (X_{max} - X_{min}) \quad (17)$$

在每次迭代中，用混沌搜索得到的最优位置随机取代一只鸽子，这样可以有效地提升种群多样性，提升算法搜索能力。

## 3 基于分数阶混合鸽群优化的 Otsu 分割算法

### 3.1 基于灰度-梯度的二维 Otsu 阈值分割算法

Otsu 算法最早由日本学者提出，主要原理是将图像分为目标类和背景类，两类间的类间方差越大，图像分割效果越明显。基于灰度-梯度直方图的二维 Otsu 算法，一定程度上改善了原始 Otsu 算法运算缓慢的缺陷，目前是比较常用的图像分割算法。

设图像总像素点数为  $N$ ， $n_{ij}$  为灰度  $i$ 、梯度  $j$  的像点数，

$fe(m)$			$fe(m)$			$fe(m)$
	...		...		...	
		$fe(1)$	$fe(1)$	$fe(1)$		
$fe(m)$	...	$fe(1)$	$fe(0)$	$fe(1)$	...	$fe(m)$
		$fe(1)$	$fe(1)$	$fe(1)$		
	...		...		...	
$fe(m)$			$fe(m)$			$fe(m)$

图 1: 分数阶滤波模板

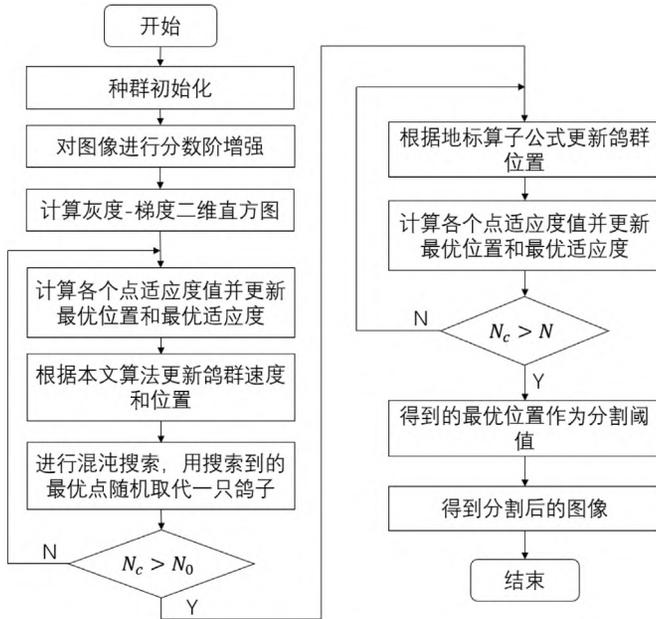


图 2: FPIO-Otsu 算法流程

其出现的概率为:

$$p_{ij} = \frac{n_{ij}}{N} \quad (18)$$

设分割阈值的平均灰度值为  $s$ , 梯度值为  $t$ , 图像被阈值  $(s, t)$  分为背景类  $B$  和目标类  $T$ , 两类出现的概率分别为:

$$p_B(s, t) = \sum_{i=1}^s \sum_{j=1}^t p_{ij} \quad (19)$$

$$p_T(s, t) = \sum_{i=1+1}^L \sum_{j=1+1}^L p_{ij} \quad (20)$$

其中  $L$  为图像的灰度级数。背景类和目标类对应的均值向量  $u_B$  和  $u_T$  及所有像素点的总均值向量  $u$  分别如下:

$$u_B(s, t) = (u_{B_i}, u_{B_j})^T = \left( \sum_{i=1}^s \sum_{j=1}^t \frac{iP_{ij}}{P_B}, \sum_{i=1}^s \sum_{j=1}^t \frac{jP_{ij}}{P_B} \right)^T \quad (21)$$

$$u_T(s, t) = (u_{T_i}, u_{T_j})^T = \left( \sum_{i=1}^s \sum_{j=1}^t \frac{iP_{ij}}{P_T}, \sum_{i=1}^s \sum_{j=1}^t \frac{jP_{ij}}{P_T} \right)^T \quad (22)$$

$$u(s, t) = (u_i, u_j)^T = \left( \sum_{i=1}^L \sum_{j=1}^L iP_{ij}, \sum_{i=1}^L \sum_{j=1}^L jP_{ij} \right)^T \quad (23)$$

离散度矩阵为:

$$S_{(s,t)} = P_B(\mu_B - \mu)(\mu_B - \mu)^T + P_T(\mu_T - \mu)(\mu_T - \mu)^T \quad (24)$$

离散度测度与背景类和目标类的差别成正比, 因此取离散度值最大时的阈值作为最佳分割阈值:

$$\text{tr}(S_{(s^*, t^*)}) = \max(\text{tr}(S_{(s,t)})) \quad (25)$$

### 3.2 分数阶增强滤波

在数字图像增强处理中, 分数阶微分增强滤波可以提升高频信号, 同时更好地保留中低频信号, 也即在增强图像纹理信息的同时, 保留住图像中的平滑部分<sup>[15]</sup>。本文引入分数阶增强滤波, 对图像进行预处理, 提升分割精度。滤波模板如图 1。

由式 (9) 可提取模板中对应位置的系数表达式:

$$fe'(m) = \frac{\Gamma(-v+1)}{m! \Gamma(-v+m+1)}, m = 0, 1, 2, \dots \quad (28)$$

实际计算中要进行归一化处理, 即:

$$fe(m) = \frac{fe'(m)}{\sum_0^k fe'(m)}, m = 0, 1, 2, \dots \quad (29)$$

式 (29) 所得到的  $fe(m)$  即为滤波模板中实际系数, 对图像进行卷积滤波, 可以实现对图像的增强。增强后的图像具有更清晰的纹理细节, 使图像分割后保留住更多细节信息

### 3.3 基于分数阶混合鸽群优化的Otsu算法

针对原始的鸽群优化的二维 Otsu 图像分割算法 (PIO-Otsu) 收敛速度慢的问题, 将分数阶引入鸽群算法速度更新公式, 提出了分数阶混合鸽群的二维 Otsu 图像分割算法 (FPIO-Otsu), 令二维 Otsu 算法的离散度矩阵作为 FPIO 的适应度函数, 算法的流程图及步骤如图 2 所示。

步骤 1 输入图像, 对图像进行分数阶增强, 基于二维 Otsu 算法生成目标函数。

步骤 2 确定初始的位置  $X_0$ , 鸽子总数  $G$ , 初始速度  $V_0$ , 总的迭代次数  $N$  以及磁场模型的迭代次数  $N_0$ 。

步骤 3 计算得到每个点的适应度  $f(x)$ 。

步骤 4 将该点的  $f(x)$  与该点的历史最佳位置对比, 取较大者作为该点的历史最佳位置。

步骤 5 将该点的  $f(x)$  与全局最优位置对比, 取较大者作为当前全局最优位置。

步骤 6 在前  $N_0$  次迭代中, 使用式 (1) 和式 (2) 来进行更新点的速度和位置; 在  $N_0$  次后使用式 (4) 来进行位置的更新。根据式 (16) 和式 (17) 进行混沌搜索, 用搜索到的最优位置随机取代一只鸽子。

表 1: 三种算法对 Baboon 图像分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	35	0.2597	8.5042
FPSO-Otsu	10	0.2571	8.4950
本文算法	6	0.2695	8.5384

表 2: 三种算法对 Lena 图像分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	10	0.3510	8.8400
FPSO-Otsu	14	0.3618	8.8931
本文算法	6	0.3677	8.8943

表 3: 三种算法对 lung 图像分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	59	0.6202	12.8796
FPSO-Otsu	16	0.6189	12.8374
本文算法	3	0.6302	13.0459

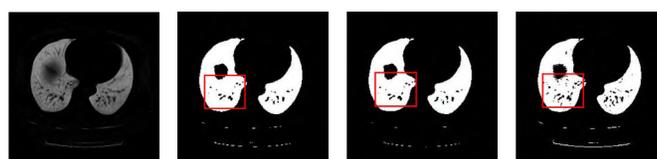
表 4: 三种算法对 Flower 图像分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	23	0.2011	8.0199
FPSO-Otsu	14	0.2011	8.0199
本文算法	4	0.2071	8.0957



(a) 原图像 (b) PIO-Otsu 算法 (c) FPSO-Otsu 算法 (d) 本文算法

图 3: Baboon 图像及分割结果



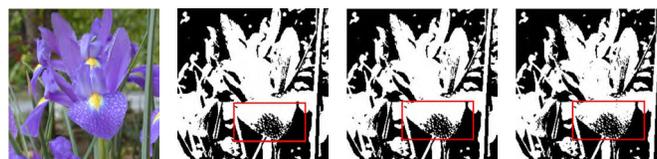
(a) 原图像 (b) PIO-Otsu 算法 (c) FPSO-Otsu 算法 (d) 本文算法

图 5: lung 图像及分割结果



(a) 原图像 (b) PIO-Otsu 算法 (c) FPSO-Otsu 算法 (d) 本文算法

图 4: Lena 图像及分割结果



(a) 原图像 (b) PIO-Otsu 算法 (c) FPSO-Otsu 算法 (d) 本文算法

图 6: Flower 图像及分割结果

步骤 7 重复步骤 3~步骤 6, 直到迭代 N 代后, 迭代停止, 得到最佳阈值进行图像分割。

步骤 8 输出分割后的图像。

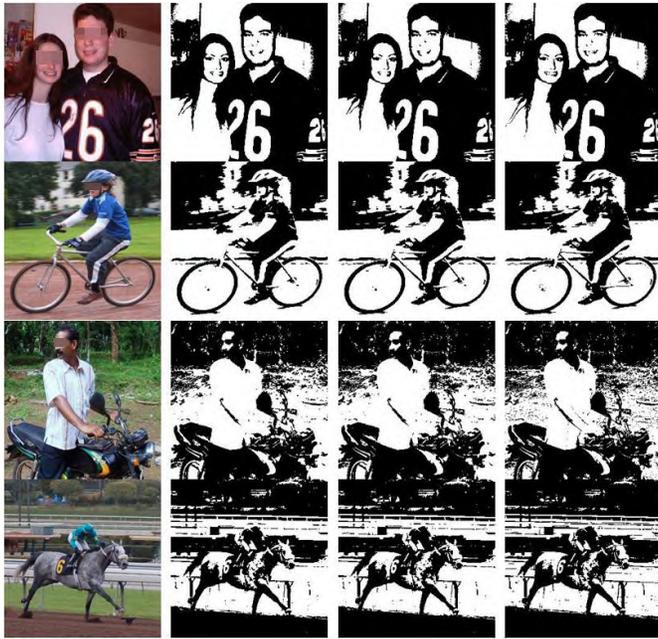
#### 4 实验结果及分析

为了验证本文算法的有效性, 本文对不同类型的经典图像进行了仿真实验, 并将本文算法 (FPIO-Otsu)、分数阶粒子群优化的二维 Otsu 算法 (FPSO-Otsu)、原始鸽群优化的二维 Otsu 算法 (PIO-Otsu) 在分割效果上进行对比。算法的硬件运行环境为 Inter (R) Core (TM) i7-10875H CPU

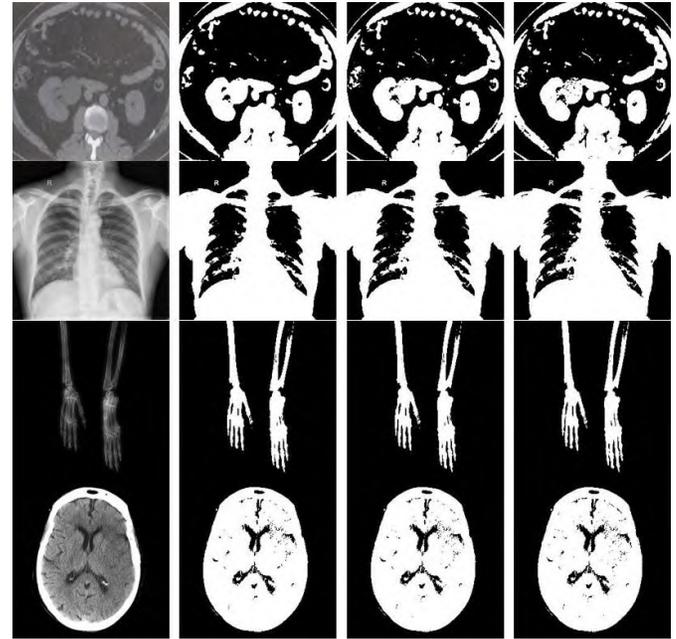
2.30GHz, 16GB 内存, 软件编程语言为 Matlab R2019a。

##### 4.1 主观视觉分割结果

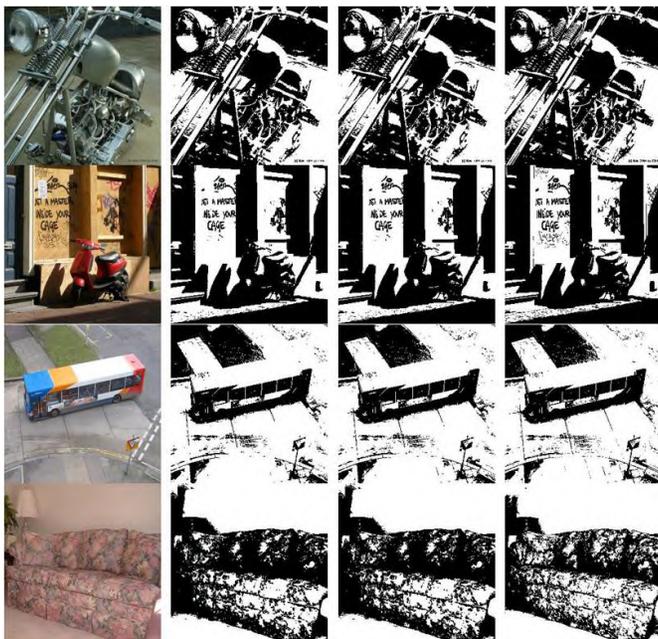
从图 3 至图 6 的分割结果对比可以看出, Baboon 图像中, 本文算法对胡须部分的分割细节明显好于另外两种算法。Lena 图像中, 本文算法对人物头发部分的纹理细节分割更加精细。lung 图像中, 本文算法对肺组织病变区域的分割更清晰。Flower 图像中, 本文算法在花瓣部分能分割出更多细节。



(a) 原始图像 (b) PIO-0tsu (c) FPSO-0tsu (d) 本文算法  
算法 算法  
图 7: 对人物图像分割结果



(a) 原始图像 (b) PIO-0tsu (c) FPSO-0tsu (d) 本文算法  
算法 算法  
图 9: 对医学图像分割结果



(a) 原始图像 (b) PIO-0tsu (c) FPSO-0tsu (d) 本文算法  
算法 算法  
图 8: 对景物图像分割结果

#### 4.2 客观评价指标

本文选取算法收敛时的迭代次数作为算法收敛速度的评价指标, 迭代次数少则算法收敛快。同时采用峰值信噪比 (PSNR) 和结构相似性 (SSIM) 作为评价图像分割效果的客观评价指标。PSNR 值越大, 算法抗噪性能越好。SSIM

值越大, 说明分割图像与原始图像更相似, 分割精度更好。

由表 1 的数据可以看出, 本文算法对 Baboon 图像的分割经过 6 次迭代达到收敛, 比另外两种算法收敛更快。而 SSIM 以及 PSNR 值均高于另外两种算法, 可见本文算法提升了图像分割的准确度, 抗噪性能也得到了保障。

由表 2 的数据可以看出, 对 Lena 图像本文算法仅经过 6 次迭代找到了最佳阈值, 而另外两种算法分别经过了 10 次以及 14 次迭代, 说明本文算法收敛速度更快。本文算法的 SSIM 以及 PSNR 值均高于另外两种算法, 说明本文算法分割图像的准确度和抗噪性能都得到了提升。

由表 3 的数据可以看出, 对 lung 图像, 本文算法经过 3 次迭代达到收敛, 另外两种算法分别经过 59 次和 16 次, 说明本文算法收敛速度得到了较大提升。本文算法的 SSIM 和 PSNR 值同样高于另外两种算法, 说明抗噪性能较强, 分割细节较好。

由表 4 的数据可知, 对 Flower 图像的分割结果, 本文算法的 PSNR 值比另外两种算法高 0.0758, SSIM 值比另外两种算法高 0.006, 说明本文算法分割效果好于另外两种算法。迭代次数上, 本文算法经过 4 次迭代达到收敛, 另外两种算法分别经过 23 次和 14 次, 说明本文算法收敛速度较快。

为了进一步验证本文算法对不同图像的分割效果, 选取另外 36 幅人物图像、32 幅景物图像以及 20 幅医学图像, 采用上述三种算法对其进行分割, 将结果取平均值, 受篇幅限制, 部分分割结果如图 7 所示。

从图 7 至图 9 的分割结果可以看出, 本文算法对人物、景物、医学图像均从主观视觉上保证了分割的精度。从表 5 至表 7 的数据可以看出, 对人物图像, 本文算法迭代次数较 PIO-Otsu 和 FPSO-Otsu 分别减少了 68.99% 和 34.99%,

表 5: 36 幅人物图像平均分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	25.7949	0.2407	9.1654
FPSO-Otsu	12.3056	0.2616	9.9302
本文算法	8.0000	0.2712	10.0663

表 6: 32 幅景物图像平均分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	27.3438	0.3116	9.8859
FPSO-Otsu	12.8125	0.3134	9.8970
本文算法	8.1563	0.3301	10.0633

表 7: 20 幅医学图像平均分割结果对比

算法	迭代次数	SSIM	PSNR
PIO-Otsu	24.6500	0.3905	9.8597
FPSO-Otsu	12.1000	0.3909	9.8632
本文算法	6.3000	0.3963	9.9426

SSIM 值分别提升了 12.67% 和 3.67%，PSNR 值分别提升了 9.83% 和 1.37%。对景物图像，本文算法迭代次数较 PIO-Otsu 和 FPSO-Otsu 分别减少了 70.17% 和 36.34%，SSIM 值分别提升了 5.94% 和 5.33%，PSNR 值分别提升了 1.79% 和 1.68%。对医学图像，本文算法迭代次数较 PIO-Otsu 和 FPSO-Otsu 分别减少了 74.44% 和 47.93%，SSIM 值分别提升了 1.49% 和 1.39%，PSNR 值分别提升了 0.84% 和 0.81%。

综上所述，本文算法无论是从主观视觉还是客观评价指标都提升了分割效果，收敛速度获得了较大提升。

## 5 结束语

本文提出的基于分数阶混合鸽群优化的二维 Otsu 图像阈值分割算法，通过将分数阶与鸽群优化算法结合，并引入分数阶增强滤波和混沌搜索算法，使得鸽群个体更快地找到最优解，提高了收敛速度。本文算法从主观视觉和客观评价指标两方面的分割效果均优于传统算法，与传统鸽群优化的二维 Otsu 算法相比，收敛速度提升约 71.20%，与分数阶粒子群结合的二维 Otsu 算法相比，收敛速度提升约 39.75%，说明本文算法收敛速度更快。本文算法的 PSNR 值相较于另外两种算法提升约 4.15% 和 1.48%，SSIM 值提升约 6.7% 和 3.46%，说明本文算法在提升收敛速度的同时，也取得了良好的分割效果。

## 参考文献

[1] 罗钧, 杨永松, 侍宝玉. 基于改进的自适应差分演化算法的二维 Otsu 多阈值图像分割 [J]. 电子与信息学报, 2019, 41 (08): 2017-2024.

[2] 张建波, 杨恢先, 周彤彤, 谭正华, 李焱. 一种改进的 2 维 Otsu 红外图像分割法研究 [J]. 激光技术, 2014, 38 (05): 713-717.

[3] 李鹏, 丁倩雯. 基于麻雀算法优化的 OSTU 分割算法 [J]. 电子测量技术, 2021, 44 (19): 148-154.

[4] Duan H B, Qiao P X. Pigeon-inspired

optimization: a new swarm intelligence optimizer for air robot path planning [J]. International Journal of Intelligent Computing and Cybernetics, 2014, 7 (1): 24-37.

[5] 胡春鹤, 王依帆, 朱书豪, 刘文定. 基于鸽群优化算法的图像分割方法研究 [J]. 郑州大学学报 (工学版), 2019, 40 (04): 42-47.

[6] 林娜, 黄思铭, 拱长青. 基于自适应权重鸽群算法的无人机航路规划 [J]. 计算机仿真, 2018, 35 (01): 38-42+125.

[7] E. J. Solteiro Pires, J. A. Tenreiro Machado, P. B. Moura Oliveira, J. Boaventura Cunha, Luis Mendes. Particle swarm optimization with fractional-order velocity [J]. Nonlinear Dynamics, 2010, 61 (1-2).

[8] 魏晶茹, 马瑜, 夏瑞, 蒋海波, 周亭亭. 基于分数阶粒子群的 Otsu 图像分割算法 [J]. 计算机工程与设计, 2017, 38 (12): 3284-3290.

[9] Mousavi Y, Alfi A. Fractional calculus based firefly algorithm applied to parameter estimation of chaotic systems [J]. Chaos, Solitons and Fractals, 2018, 114 (7): 202-215.

[10] 宋超. 从整数阶微积分到分数阶微积分 [J]. 高师理科学刊, 2016, 36 (9): 15-17

[11] Khudair A R, Haddad S A M, Khalaf S L. Restricted fractional differential transform for solving irrational lorder fractional differential equations [J]. Chaos, Solitons & Fractals, 2017, 101 (8): 81-85

[12] 钱伟懿, 李阿军, 杨宁宁. 基于混沌的多目标粒子群优化算法 [J]. 计算机工程与设计, 2008 (18): 4794-4796+4800.

[13] 李娟, 杨琳, 刘金龙, 杨德龙, 张晨. 基于自适应混沌粒子群优化算法的多目标无功优化 [J]. 电力系统保护与控制, 2011, 39 (09): 26-31.

[14] 赵志刚, 常成. 自适应混沌粒子群优化算法 [J]. 计算机工程, 2011, 37 (15): 128-130.

[15] 赵九龙. 基于分数阶微积分的三维图像去噪增强算法研究 [D]. 银川: 宁夏大学, 2015.

## 作者简介

李光昊 (1996-), 男, 硕士研究生。研究方向为模式识别与图像处理。

马瑜 (1974-) (通讯作者), 男, 宁夏回族自治区固原市人。工学博士, 教授。研究方向为计算机视觉与模式识别。

郭姝琪 (1997-), 女, 陕西省渭南市人。硕士研究生。研究方向为模式识别与图像处理。

王仕儒 (1995-), 男, 山东省临沂市人。硕士研究生。研究方向为模式识别与图像处理。