

# Genetic Programming for Automatic Global and Local Feature Extraction to Image Classification

Ying Bi, Mengjie Zhang and Bing Xue

## 摘要

提取图像中有效的特征对于图像分类至关重要，但由于图像之间巨大的差异，使提取有效的特征具有挑战性。大多数现有的方法通过预先学习专业领域知识和固定模型复杂度来执行图像分类任务。遗传编程(Genetic Programming, GP)已成为图像分类中用于进行特征学习的一种机器学习方法。然而，大多数基于GP的方法只提取了一种类型的相对简单的特征，即局部或全局特征，这对于复杂的图像分类是不高效的。本文提出了一种新的GP方法(IDGP)，以实现自动地同时提取图像的全局和局部特征进行图像分类。除此之外，还对IDGP进行了三个改进：1) 向IDGP添加其他图像相关算子，使模型能够自动提取出更多有效特征；2) 在种群演化过程中，对更新后的子代种群进行去重处理；3) 向GP模型加入自适应主成分分析方法，以减少种群评估时间。在两个数据集FEL\_1和KTH上评估了所提方法的性能。实现结果表明，向IDGP添加其他图像相关算子后，取得了比IDGP更好或相似的性能。在种群演化过程中进行去重处理，在简单任务上降低了个体平均大小。向模型加入自适应主成分分析方法虽然显著降低了训练时间，但也降低了模型精度，在简单数据集上尤其明显。将三个改进点结合，在复杂数据集上较好地平衡了训练时间和模型精度，但对于简单数据集结果仍不理想。

**关键词：**演化计算；遗传编程；特征学习；图像分类

## 1 引言

图像分类是计算机视觉和机器学习中的一项重要任务，旨在根据图像内容，从一组预定义的类标签中为数据集中的每个图像匹配一个类标签。图像分类有许多实际应用，例如医学诊断、自动驾驶和生物识别 [10]。然而，由于图像具有高维度、高变化等现实因素，使得此类问题具有挑战性。

在图像分类中，特征提取是解决问题的一个关键步骤，其将低级像素值转换为高级特征。然而，由于存在类间和类内差异，特征提取十分具有挑战性。所需的特征应包含图像的不变性和判别性信息，即能够保留不同类的判别性和同一类的相似性，从而帮助我们构造有效的分类器 [4]。目前已开发了许多特征描述符，例如方向梯度直方图(HOGs) [13]、局部二值模式(LBPs) [21]和尺度不变特征变换(SIFT) [18]用于提取图像特征。这些特征被称为手工特征，通常只对特定任务有效。例如，LBP特征在纹理分类方面表现出色，但应用于图像类型未知的任务时，其效果不一定理想。因此，人们提出了特征学习方法，自动学习图像中的特征用于分类。

然而，现有的特征学习方法，如卷积神经网络(CNNs)具有一定的局限性。它们不仅需要领域专业知识来设计用于解决特定任务的模型，还需要大量的计算资源和训练数据。为了克服这些限制，需要开发新的特征学习方法以进行图像分类。

遗传编程(Genetic Programming,GP)是一种进化计算(Evolutionary computation,EC)技术，能够利用生物进化和自然选择的原理，自动进化出一组计算机程序来解决问题 [15]。并且GP在许多任务中都得到了显著的成果，例如符号回归、分类、调度和图像分析。与其他EC技术相比，GP具有更灵活的表示方法，即基于树的表示 [22]。在基于树的GP中，每个个体用一颗树表示，其中根节点和内部节点是函数/算子，叶节点是端点。基于树的GP可以在没有预定义结构的情况下进化出解，即无论是针对简单任务的浅层模型还是针对困难任务的复杂模型，它都可以找到各种深度的解决方案/模型。

GP已成功应用于图像特征学习，并取得了显著的成果 [7]。在这些方法中，一些与图像相关的算子，例如高斯滤波器、均值滤波器、最小滤波器等，被用作GP的函数来进化提取高级图像特征的解决方案 [26]。为了有效地利用与图像相关的算子，通常需要将不同的功能和终端集成到单个树中。为此，开发了集中具有不同程序结构的GP方法。然而，这些程序结构往往受到一定的限制，即当引入新的图像相关算子时，这些程序结构可能无法适用。更重要的是，大多数这些程序结构都是固定的，并且他们在预定义的过程中提取特征，这可能会限制它们的性能。

同时，现存的GP方法大多数仅解决二元图像分类问题 [5,6]。而在实际应用中，大多数图像分类任务是多分类任务，因此这些方法不能直接使用。所以需要提出一种新的GP方法，使其具有与图像相关的算子和灵活的程序结构，以针对不同的分类任务进行特征学习。

## 2 相关工作

### 2.1 传统图像分类方法

传统的图像分类方法使用特征描述的方法从图像中提取特征，并将其输入到分类算法中进行分类。[12]等在基于直方图特征的支持向量机(SVMs)中开发了一种重尾RBF核函数用于图像分类。结果表明，在SVM中使用该核函数可以提高分类性能。[25]提取了简单的直方图特征，并采用k-最近邻(KNN)进行图像分类，取得了良好的效果。[9]等人提出了一种自动检测感兴趣区域的方法，并从区域中提取形状和外观特征用于物体分类。

然而，这些方法大多使用简单的手工特征，限制了它们在解决其他图像分类任务上的有效性。

### 2.2 基于神经网络的图像分类方法

近年来，基于神经网络的方法，如CNNs、DBNs和DBMs，在图像分类任务中取得了显著的成果。在这些方法中，CNN是最常用的图像分类技术。[23]等人通过集成CNN、PCA、BT和非时态概念器分类器的成分，开发了前馈卷积概念器NN(FCCNN)。[17]提出了一种自步的CNN (SPCN)，通过在学习过程中为训练样本分配权重来增强CNN的学习鲁棒性。该方法在六个基准数据集上达到了比一些最先进算法更好的性能。[24]通过引用新的代价函数和适当的惩罚项，提出了收缩自编码器(CAEs)。该方法在7个数据集上进行了测试，并与其

他类型的AE进行了比较, 包括3层堆叠AE(SAE-3)和带二值掩蔽噪声的3层去噪AE(DAE-b-3)。除了CNN, 作为神经网络的变体, DBN和DBM也被 [16]应用于数字识别。

然而, 神经网络方法存在一些局限性, 包括对大量计算资源和训练实例的需求, 以及需要丰富的领域知识来设计模型。

### 2.3 基于GP的图像分类方法

GP通过同时进行特征提取和特征构造, 已被广泛应用于解决图像分类问题。处理图像分类的多个子任务, 包括区域检测、特征提取、特征构建和图像分类, 可以使用强类型GP(STGP)集成到单个GP树中 [20]。GP树的最终输出是一个数字/特征, 可以使用预定义的阈值进行分类。[5]等人提出了多层GP(3TGP), 其中高层特征通过图像滤波层、聚合层和分类层进行学习。[2]等人改进了3TGP, 将其结构简化为一个聚合层和一个分类层(2TGP), 以同时执行区域检测、特征提取、特征构造和图像分类。[6]等人提出了一种多层GP方法(MLGP), 利用图像相关算子提取和构造高层特征用于图像分类。这些方法在二值图像分类上取得了较好的效果。

为了从图像中学习多种特征, [1]等人提出了一种基于GP的方法, 自动生成一组使用LBP相似方式的特征用于纹理图像分类, 但该方法生成的特征数量是固定的。为了改进这一点, [3]提出了一种动态方法来学习用于纹理分类的动态数量的图像特征。在该方法中, 开发了一个接受灵活数量的子节点的根函数。然而, 这两种方法都是针对纹理描述的, 它们的性能还没有在其它类型的数据集上进行检验。[26]提出了一种具有输入层、滤波层、池化层和级联层的多目标GP(Multiobjective GP, MOGP)方法, 将图像转化为高层特征。该方法在四个不同的数据集上取得了比使用手工特征和简单CNNs的方法更好的性能。但是这种方法具有固定的程序结构来产生滤波和池化的特征, 无法产生在困难数据集上取得良好性能的不变特征。[8]开发了一种GP方法, 用于自动地同时学习特征和进化集成用于图像分类。该方法构建了一个分类器集合来解决图像分类问题。然而, 该方法在大型图像分类数据集上表现较差。

虽然已经开发了许多基于GP的方法用于图像分类, 由于GP程序的性质, 大多数方法仅用于二值图像分类任务。然而, 实际中大多数图像分类任务都是多分类任务, 因此需要不同的算法设计来解决这一问题。

## 3 本文方法

提出的IDGP方法能够自动演化解决方案/树, 这些解决方案/树能够使用输入图像中的现有图像描述符提取有区别的全局/局部特征。

IDGP的整体特征学习过程如图 1所示。IDGP从种群初始化开始, 根据模型的程序结构、函数和终端集随机生成第一代种群。然后使用适应度评估过程对每个个体进行评估。适应度评估后, 采用一种选择方法和三种遗传算子获得子代种群。需要注意的是, 选择方法选择适应度较高的个体进行交叉和突变。演化学习过程在满足中止条件时终止。在IDGP中, 终止条件是达到最大演化代数。如果不满足终止条件, 则重新进行适应度评估和种群更新过程。否则, 演化过程结束, 返回最佳个体。

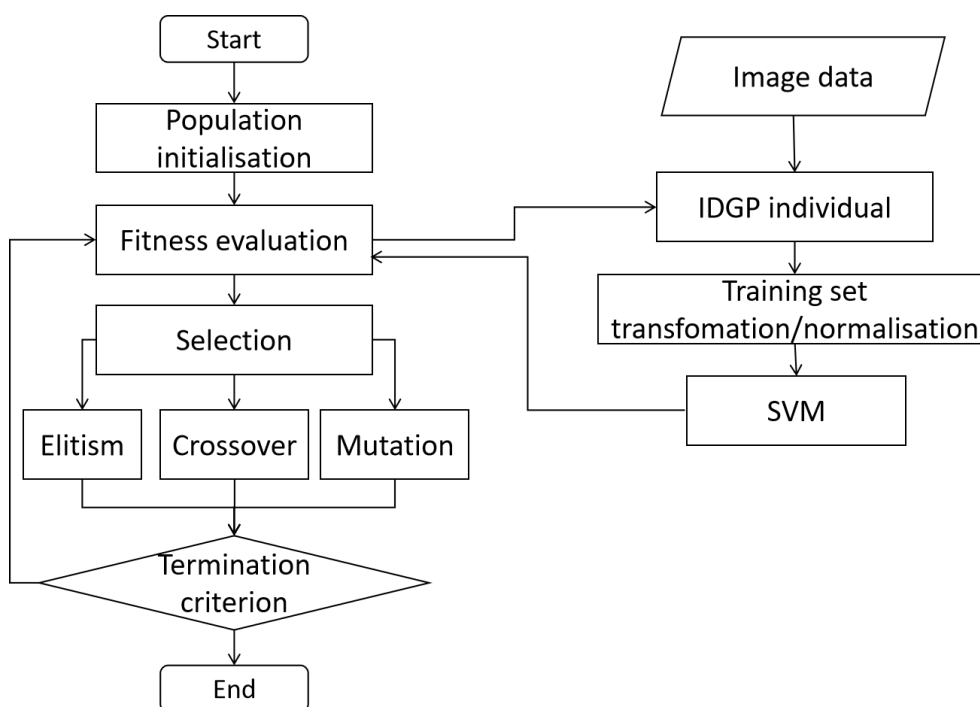


图 1. IDGP整体特征学习过程

利用IDGP找到的最佳个体进行图像分类，如图 2所示。

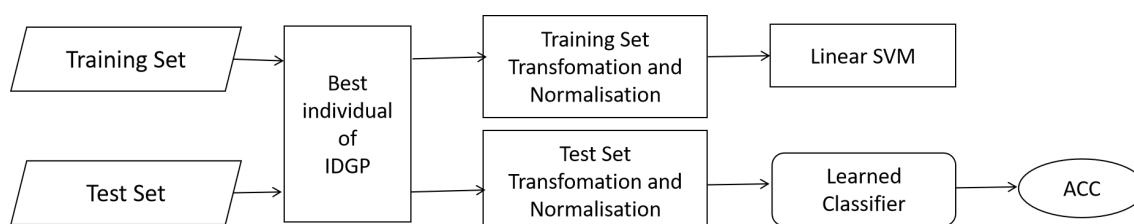


图 2. IDGP图像分类流程

### 3.1 Genetic Programming结构

IDGP开发了一种新的程序结构，使其能够以灵活的方式实现全局/局部特征学习。基于强类型GP的新程序结构如图 3所示。它包含输入层、区域检测层、特征提取层和输出层，其中不同的功能层使用不同的功能。

区域检测层从大输入图像中检测感兴趣的小区域。IDGP树中可能存在区域检测层，也可能不存在，这表明IDGP的解/树可以在不进行区域检测的情况下构造从而使其能够生成全局特征。特征提取层从输入图像中提取全局特征或从检测区域中提取局部特征。因此，在全局和局部场景下开发了特征提取功能。特征连接层将其子节点的特征连接到特征向量。

图 1给出了IDGP的典型程序/解决方案实例，其中不同的颜色表示输入、输出和不同的功能。在这个程序中，有区域检测功能，即Region\_S和Region\_R；有特征提取功能，即G\_SIFT、L\_DIF和L\_uLBP；有特征拼接功能，即Feacon2。

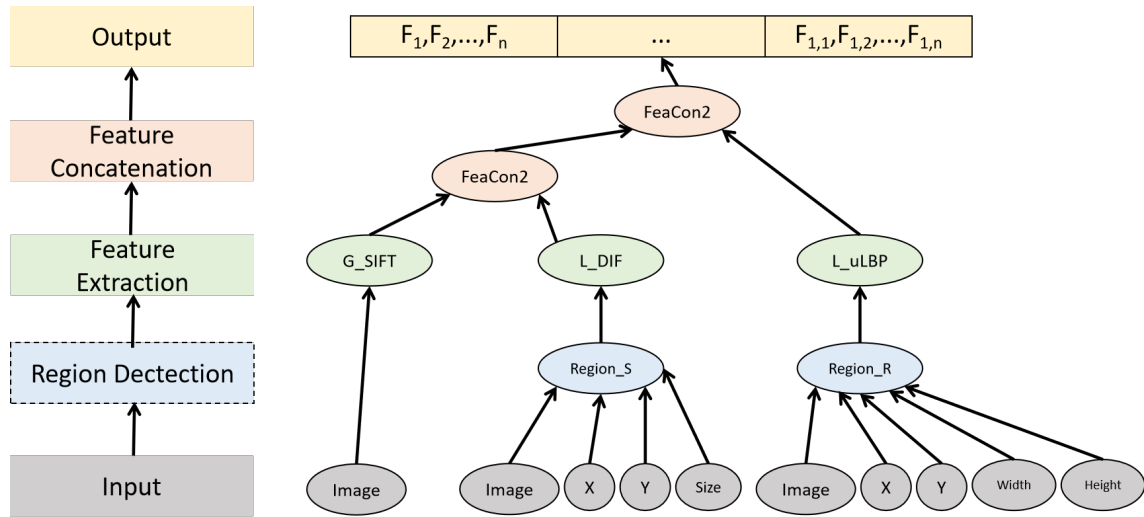


图 3. IDGP模型结构

新的程序结构允许IDGP具有灵活的树深度，并生成各种数量的全局/局部特征。全局和局部特征可以通过使用全局和局部场景中的图像描述符作为内部节点来构建GP树以学习。这种灵活性使得IDGP能够适应不同的任务和数据集，并在全局和局部特征学习方面取得优异的性能。

### 3.2 操作算子定义

IDGP的终端集有6个不同的终端：Image、X、Y、Size、Width、Height，如表 1所示。Image表示输入图像，它是一个二维数组，数组中的值通过除以255被归一化为[0,1]。其他终端为IDGP的随机常数，仅出现在使用相应函数的树中。这些终端的值在初始化步骤时从预定的范围内随机选择，并且可以在演化过程中通过变异算子进行修改。

表 1. IDGP终端集

终端	输出类型	详细描述
Image	Img	输入图像：一个范围为[0,1]的数组
X	Int1	输入图像中检测区域左上角点的水平坐标：范围为[0,1-20]
Y	Int2	输入图像中检测区域左上角点的垂直坐标：范围为[0,1-20]
Size	Int3	一个正方形区域的大小：范围为[20,70]
Width	Int3	(矩形)区域的宽度：范围为[20,70]
Height	Int3	(矩形)区域的高度：范围为[20,70]

IDGP的函数集有三种不同类型的功能：区域检测函数、特征提取函数和特征拼接函数，详细介绍如表 2所示。

### 3.3 适应度评估

在演化学习过程中，适应度函数被用来指导寻找最佳个体。算法1给出了IDGP的适应度评估过程。



表 2. IDGP函数集

函数	输入类型	输出类型	详细描述
区域检测函数			
Region_S	Img,Int1,Int2,Int3	Region	从一个大的输入图像中检测一个正方形区域
Region_S	Img,Int1,Int2,Int3,Int3	Region	从一个大的输入图像中检测一个矩形区域
特征提取函数			
G_DIF	Img	Vector	从输入图像中提取20个域无关特征
G_Hist	Img	Vector	从输入图像中提取32个直方图特征(箱子的数量设置为32)
G_SIFT	Img	Vector	从输入图像中提取128个SIFT特征作为关键点
G_HOG	Img	Vector	以10步长从HOG图像中提取每个网格的平均值
G_uLBP	Img	Vector	提取59个均匀的LBP直方图特征(半径为1.5, 邻居数量为8)
L_DIF	Region	Vector	从输入图像中提取20个域无关特征
L_Hist	Region	Vector	从输入图像中提取32个直方图特征(箱子的数量设置为32)
L_SIFT	Region	Vector	从输入图像中提取128个SIFT特征作为关键点
L_HOG	Region	Vector	以10步长从HOG图像中提取每个网格的平均值
L_uLBP	Region	Vector	提取59个均匀的LBP直方图特征(半径为1.5, 邻居数量为8)
特征拼接函数			
FeaCon2	Vector,Vector	Vector	将两个特征向量连接成一个特征向量
FeaCon3	Vector,Vector,Vector	Vector	将三个特征向量连接成一个特征向量

在算法 1 中，使用一个包含  $N$  个  $m \times l$  图像和标签的数据集  $D_{train}$ 。每个 IDGP 个体作为特征提取的解决方案，将每张图像  $I_i$  转换为大小为  $S$  的特征向量  $F_i$ 。 $S$  是 IDGP 模型提取的特征的个数。然后使用最大最小标准化方法对变换后的训练集进行归一化，如公式 (1) 所示。归一化的训练集使用分层  $k\_fold$  交叉验证输入线性支持向量机(SVM)。

$$\overline{F}_i = \frac{F_i - \min(F_{i=1}^N)}{\max(F_{i=1}^N) - \min(F_{i=1}^N)} \quad (1)$$

在 IDGP 中，不同图像描述符产生的特征值范围不同。例如，u\_LBP 特征可能在  $[0,100]$  范围内，而 SIFT 特征可能在  $[0,1]$  范围内。当使用这些特征的组合进行分类时，可能会导致对特征类型特征的偏见。因此，在 IDGP 中，使用最大最小标准化方法将输出特征  $F_i$  重新缩放为  $\overline{F}_i$ ，如公式 (1) 所示。

---

#### Algorithm 1 适应度评估

---

```

1: Input:  $D_{train}$ :训练实例; $p$ :需要评估的个体
2: Output:  $f(p)$ :需要评估的个体的适应度值
3: for each image  $I_i$  in  $D_{train}$  do
4:    $F_i \leftarrow$  使用  $p$  将图像  $I_i$  转换成一组特征
5: end for
6:  $D_{train}^{tr} \leftarrow$  使用 IDGP 方法学习到的特征
7:  $D_{train}^{norm} \leftarrow$  使用最大最小标准化方法处理  $D_{train}^{tr}$ 
8: 使用五折交叉验证分割  $D_{train}^{norm}$ 
9: for  $k = 0; k < 5; k++$  do
10:   使用第  $k$  折作为测试集，剩余的  $k - 1$  折作为训练集
11:   使用训练集训练一个线性 SVM 分类器并用测试集测试分类器的性能
12:    $F_k \leftarrow$  第  $k$  折测试集的分类精度
13: end for
14:  $f(p) = \frac{1}{5} \sum_{k=1}^5 f_k$ 
15: Return  $f(p)$ 

```

---

## 4 复现细节

### 4.1 与已有开源代码对比

本次复现工作主要引用了 <https://github.com/YingBi92/BookCode> 中的代码。下面是改进部分的关键代码。

```

1 def varAnd(population, toolbox, cxpb, mutpb):
2     offspring = [toolbox.clone(ind) for ind in population]
3     new_cxpb=cxpb/(cxpb+mutpb)
4     i = 1
5     while i < len(population):
6         if random.random() < new_cxpb:
7             if (offspring[i - 1] == offspring[i]) or pop_compare(offspring[i - 1], offspring[i]):
8                 offspring[i - 1], = toolbox.mutate(offspring[i - 1])

```

```

9         offspring[i], = toolbox.mutate(offspring[i])
10     else:
11         offspring[i - 1], offspring[i] = toolbox.mate(offspring[i - 1], offspring[i])
12     del offspring[i - 1].fitness.values, offspring[i].fitness.values
13     i = i + 2
14     else:
15         offspring[i], = toolbox.mutate(offspring[i])
16         del offspring[i].fitness.values
17         i = i + 1
18     if i == len(population)-1:
19         offspring_value = []
20         for inv in offspring:
21             inv_str = str(inv)
22             offspring_value.append(inv_str)
23         unique_population = list(set(offspring_value))
24         i = len(unique_population) - 1
25 return offspring

```

```

1 # Add an operator
2 pset.addPrimitive(felgp_fs.PCA_ori, [Array4, Int8], Array5, name='PCA')
3 pset.addPrimitive(felgp_fs.kpca, [Array4, Int7], Array5, name='kPCA')
4 # Operator parameters are added
5 pset.addEphemeralConstant('k-length', lambda: random.randint(1, 5), Int7)
6 pset.addEphemeralConstant('length', lambda: random.randint(2, 10), Int8)

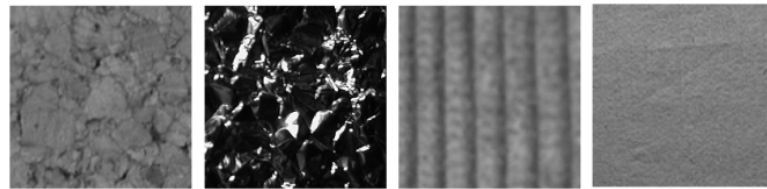
```

## 4.2 实验环境搭建与基准数据集介绍

本文实验的基本环境为Python3.11，Deap库版本为1.4。记录运行时间时仅记录当前进程的累计系统和用户CPU时间。



(a) FEL\_1



(b) KTH

图 4. 来自FEL\_1和KTH基准数据集的实例图像展示

FEL\_1数据集是面部表情分类任务 [11]，包含从200个不同的人身上采集的带有微笑或自然表情的面部图像。KTH数据集 [19]是一个纹理分类数据集，包含10类共810张图像。在4中光照条件下，对图像进行了9种不同尺度、3种姿态的采样。由于FEL\_1和KTH数据集没有公共的训练集和测试集，因此使用常用的比例进行分割，如表 3所示。



表 3. Benchmark Datasets

No.	Dataset	Image Size	Training Set Size	Test Set Size	Class
1	FEL_1	$180 \times 130$	150	50	2
2	KTH	$100 \times 100$	600	210	10

所有实验的参数设置是基于GP模型方法常用的设置[14]。本文中，最大演化代数 $G$ 为50代，种群规模 $N$ 为100代。交叉率 $P_c$ 为0.8，突变率 $P_m$ 为0.19，精英率 $P_e$ 为0.01。选择方法为规模为7的锦标赛选择。在初始化阶段，使用爬坡减半法，且树的深度在2-6之间，最大深度为8。

### 4.3 改进点

#### 4.3.1 改进点1

GP已被广泛应用于图像特征学习，并取得了很好的效果。但IDGP只使用了全局/局部特征提取算子，提取到的有效特征有限。受到FGP的启发，可以向GP模型中加入其他图像相关算子，使模型能够自动提取出更多样的有效特征以进行图像分类任务。本文向IDGP模型中加入过滤函数和池化函数。新提出的模型结构如图5所示。新增的滤波函数详细介绍如表4所示。

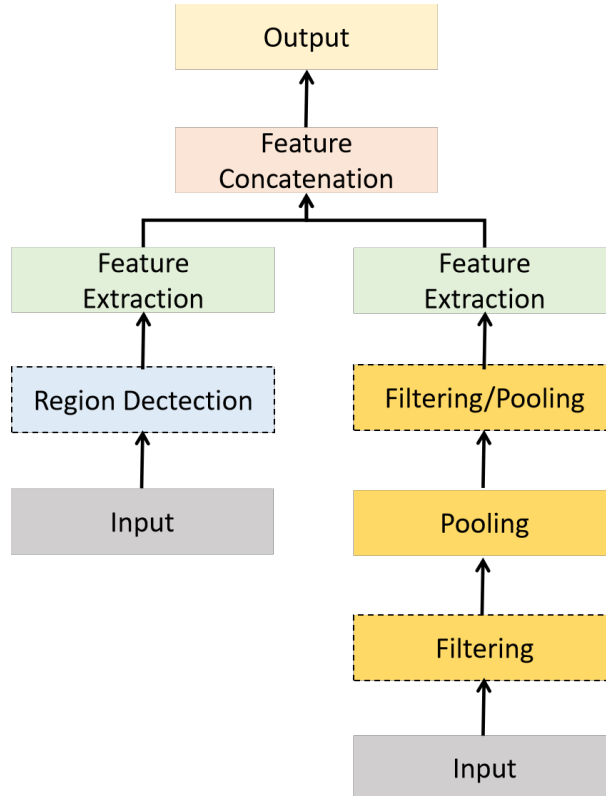


图 5. New\_IDGP示意图

New\_IDGP中，左分支提取图像的全局/局部特征，右分支提取图像的过滤/池化特征。需要注意的是，New\_IDGP具有可变长表示，其中实线框表示固定层，虚线框表示可变层，即个体中可能进行这个操作，也可能不进行这个操作。

在新增的操作函数中，Gau函数将图像和标准差 $\sigma$ 作为输入，并返回由高斯核卷积的图像。GauD有3个参数，即标准差 $\sigma$ 、 $o_1$ 和 $o_2$ 。参数 $o_1$ 和 $o_2$ 分别表示沿X轴和Y轴方向导数的阶数。Gabor滤波器由一个Gabor小波函数生成。它具有参数 $\theta$ 和 $f$ ，分别表示Gabor小波函数中的正弦波函数的核方向和波长( $\lambda = \frac{1}{f}$ )。Lap函数是通过Laplacian算子进行离散和近似得到的，它可以检测出平坦的区域或边缘。LoG1和LoG2函数将Laplacian滤波器与高斯函数卷积，可以降低图像中的噪声。LoG1和LoG2中高斯函数的标准差分别设为1和2。其中，Gau、Med和Mean滤波器常被用于图像去噪和平滑。GauD、Lap、LoG1、LoG2、Sobel、SobelX、SobelY等滤波器可以检测图像中的边缘或平坦区域。Mean、Med、Min和Max函数的核尺寸均为 $3 \times 3$ 。

除此之外，还使用了W-Add、W-Sub、Sqrt和Relu函数。利用W-Add和W-Sub函数对大小相同或不同的两幅加权图像进行相加或相减，其中权重分别为 $n_1$ 和 $n_2$ 。在两幅图像大小不相同的情况下，W-Add和W-Sub将坐标(0,0)处的图像像素进行重叠，将较大图像的超出部分进行裁剪，并进行加减运算。ReLU是经过校正的线性单元。Sqrt计算图像中每个像素值的平方根，如果像素值为负，则通过返回1来保护。ReLU和Sqrt可以通过将像素值从负值转换为非负值来对输入图像进行重新缩放。

表 4. 新增滤波函数集

函数	输入类型	输出类型	详细描述
Gau	1 image, $\sigma$	1 image	标准差为 $\sigma$ 的高斯滤波器
GauD	1 image, $\sigma, o_1, o_2$	1 image	高斯滤波器的导数
Gabor	1 image, $\theta, f$	1 image	具有 $\theta$ 方向和 $f$ 频率的 $\frac{1}{\lambda}$ 的Gabor滤波器
Lap	1 image	1 image	Laplacian滤波器
LoG1	1 image	1 image	$\sigma = 1$ 的高斯滤波器的拉普拉斯算子
LoG2	1 image	1 image	$\sigma = 2$ 的高斯滤波器的拉普拉斯算子
Sobel	1 image	1 image	Sobel边缘检测
SobelX	1 image	1 image	沿X轴方向的Sobel滤波
SobelY	1 image	1 image	沿Y轴方向的Sobel滤波
Med	1 image	1 image	$3 \times 3$ 中值滤波
Max	1 image	1 image	$3 \times 3$ 最大值滤波
Min	1 image	1 image	$3 \times 3$ 最小值滤波
Mean	1 image	1 image	$3 \times 3$ 均值滤波
W-Add	2 image, $n_1, n_2$	1 image	添加两幅加权图像
W-Sub	2 image, $n_1, n_2$	1 image	将两幅加权图像相减
Sqrt	1 image	1 image	对图像中的每个像素取平方根
Relu	1 image	1 image	激活后的线性单元

常用的池化函数有最大池化和平均池化。最大池化返回每个滑动窗口的最大值，而平均池化返回滑动窗口的平均值。重要的特征，例如边缘，可以通过最大池化函数提取，但可以通过平均池化函数平滑。因此，只使用最大池化(简称MaxP)函数。

#### 4.3.2 改进点2

IDGP在生成子代种群时,采用的是对适应度较高的个体进行随机交叉和随机突变。这会导致在一个种群中可能产生重复的个体,降低模型的搜索能力。为了使GP模型能够在搜索空间中进行有效搜索,保持种群多样性,本文在改进点1提出的New\_IDGP中使用算法 2对更新后的子代种群进行去重(Unique.Pop),并重新生成新的个体,直到子代种群的大小达到预设种群的大小,算法终止。

---

**Algorithm 2** Unique\_Pop

---

```
1: Input: Population:父代种群;P_size:生成的种群大小;cross_px:交叉率;mut_px:突变率
2: Output: Offspring:子代种群
3:  $i = 0$ 
4: while  $i < P\_size$  do
5:    $Random\_state \leftarrow$ 任取一个[0,1]的随机数
6:   if  $Random\_state < cross\_px$  then
7:     进行交叉操作
8:      $i \leftarrow i + 2$ 
9:   else
10:    进行突变操作
11:     $i \leftarrow i + 1$ 
12:   end if
13:   if  $i == P\_size - 1$  then
14:      $unique\_offspring \leftarrow$ 对更新后的种群进行去重,提取不包含重复个体的Offspring
15:   end if
16:    $i \leftarrow unique\_offspring$ 的长度-1
17: end while
18: Return Offspring
```

---

#### 4.3.3 改进点3

主成分分析方法可以在尽量不降低任务分类精度的同时减少任务的训练时间。但对于不同的数据集和个体结构,采用的方法与参数也不同,所以需要自适应的选择适合的主成分分析方法及其参数。

因此,可以向GP模型中加入主成分分析方法算子,使模型能够针对特定的数据集及个体结构自动选择合适的方法与参数。本文在改进点1提出的New\_IDGP中添加主成分分析方法算子,得到的新模型结构如图 6所示,新加层的介绍如图 7所示。

在Classifiers层中使用线性SVM在训练集上对个体进行评估。选择线性SVM是因为它常用于图像分类任务。在Principal component analysis中使用主成分分析方法(PCA)和核主成分分析方法(Kernel.PCA)。PCA通过降低数据维度来减少特征数量,同时保留数据中的主要信息。Kernel.PCA是一种在非线性数据集上进行降维的方法,将数据映射到高维特征空间中,并在高维特征空间中执行PCA来处理非线性关系。

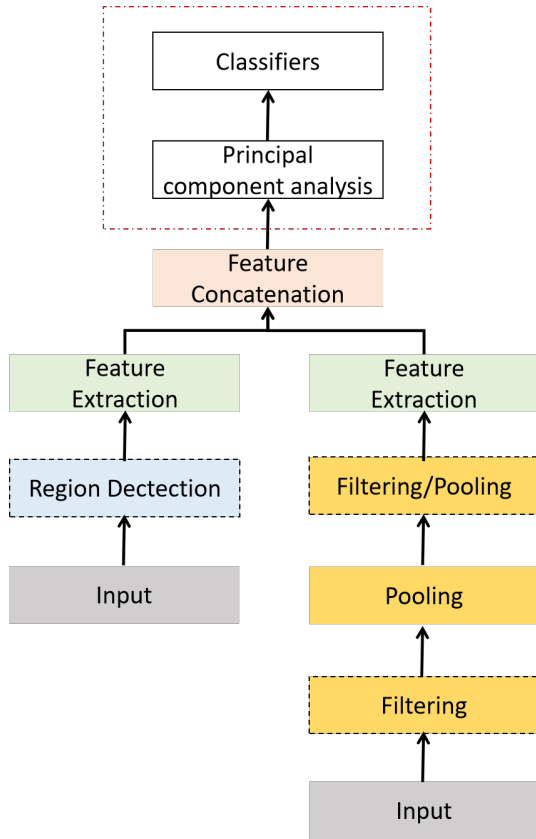


图 6. 加入主成分分析算子的New\_IDGP示意图

结合三个改进点得到的模型称为PUIDGP模型，即向模型中添加多个图像相关算子提取更多图像特征，在子代种群生成过程中加入Unique\_Pop以提高空间搜索效率，添加PCA过程以提高算法效率。

## 5 实验结果分析

### 5.1 复现结果分析

首先对IDGP进行复现，得到复现结果如表 5 所示，并在进行分类之前，尝试对每一个个体得到的特征向量运用主成分分析以减少特征向量的长度，观察主成分分析对任务分类精度、运行时间的影响，详细结果列在表 5 中，符号‘+’和‘-’表示采用不同主成分分析方法后取得了明显优于和差于比较方法的结果。符号‘=’表示取得了与文章结果类似的结果。Overall总结了测试的整体结果。

从表 5 可以发现，对文章进行复现时，在FEL\_1数据集上取得了与文章结果一致的分类精度，在KTH数据集上比文章结果低1.71%。观察复现结果训练集的收敛行为，如图 8 所示，种群在一开始时就达到了较高的精度，并且从第5代开始，种群中最佳个体的适应度值已经稳定。分析复现结果与原文结果有细微差别有两个原因：(1)本文的实验结果只运行了1次，而原文独立运行了30次并取均值。(2)原文在FEL\_1数据集上取图像尺寸为 $60 \times 40$ ，在KTH数据集上取图像尺寸为 $50 \times 50$ 。而本文取图像尺寸如图 3 所示。

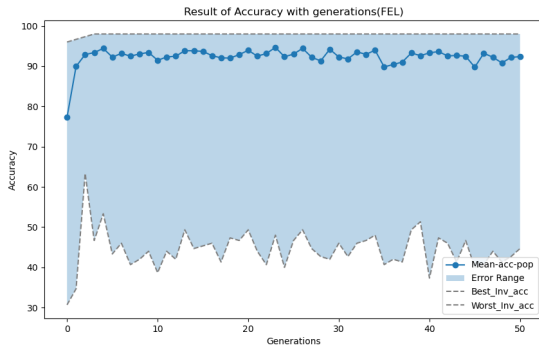
### 新加层的介绍

Classifiers	
输入类型	Array1, Array2, Int
输出类型	Array
详细描述	传入经过主成分分析后的特征矩阵、label和算法迭代数量，返回一个矩阵
Principal component analysis	
输入类型	Array, Int
输出类型	Array
详细描述	传入特征矩阵和降维参数，返回经过主成分分析后的特征矩阵

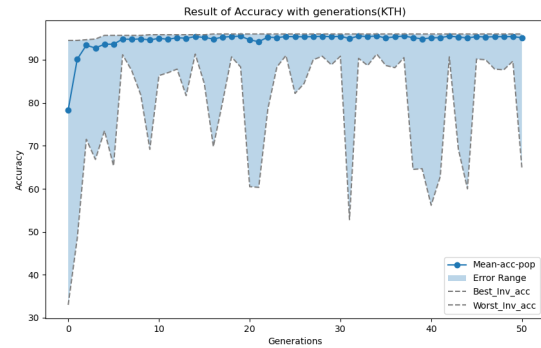
图 7. Classifiers层与Principal component analysis层的介绍

表 5. ACC1

	FEL_1	KTH
文章结果	98.00	95.71
复现结果	98.00	94.00
PCA	96.00	94.67
Kernel-PCA(l=1/2)	<b>98.00</b>	94.67
Kernel-PCA(l=1/3)	96.00	<b>96.00</b>
Kernel-PCA(l=1/4)	96.00	92.67
Overall	1=,3-	1+,3-



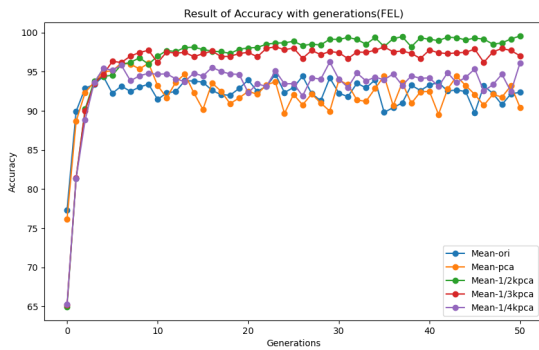
(a) FEL训练集



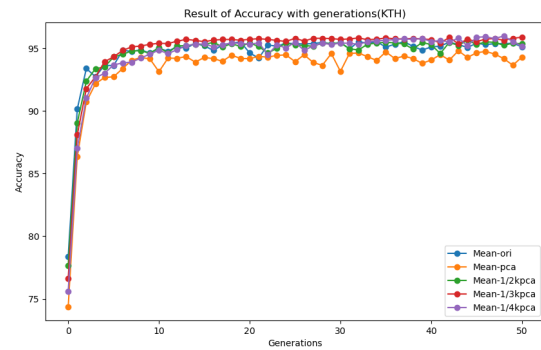
(b) KTH训练集

图 8. 复现结果训练集收敛行为

在运用主成分分析的4个比较中，FEL\_1数据集上有1个方法达到了与文章相似的结果，3个方法达到了较差的效果；KTH数据集上有1个方法优于文章结果，3个方法达到了较差的效果。观察不同主成分分析方法收敛行为，如图 9所示。在FEL中运用Kernel-PCA(1/2)方法和Kernel-PCA(1/3)方法得到的种群平均适应度值较为稳定，在KTH中，4种方法都有相似的收敛行为。



(a) FEL训练集



(b) KTH训练集

图 9. 不同主成分分析方法收敛行为

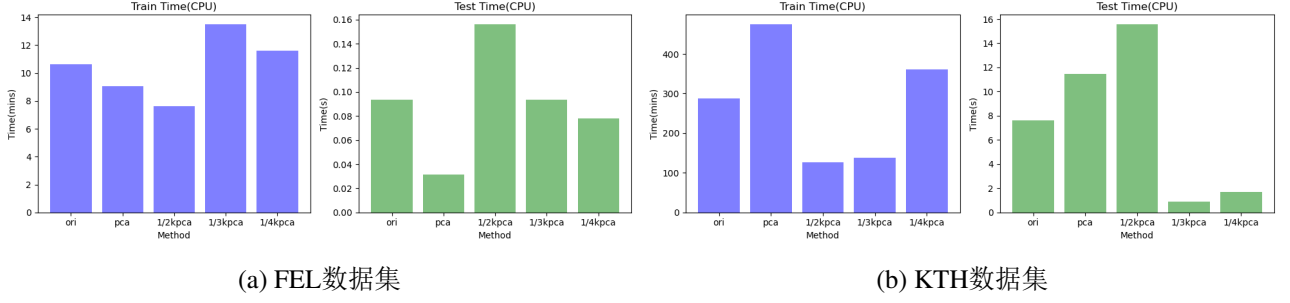


图 10. 不同方法的时间展示

比较两个数据集采用不同方法得到的训练时间和测试时间，如图 10所示。在两个数据集中，训练时间以 $mins$ 为单位，测试时间以 $s$ 为单位，尤其对于KTH数据集，训练时间是测试时间的2200x。这可能是由于FEL是图像二分类任务，而KTH是图像多分类任务，所以在相同参数设置下，KTH需要更多时间进行演化过程。在FEL数据集中，PCA和Kernel-PCA(1/2)分别降低了原训练时间的1.17x和1.4x。在KTH数据集中，Kernel-PCA(1/2)和Kernel-PCA(1/3)分别降低了原训练时间的2.28x和2.09x。

实验表明，主成分分析方法可以在尽量不降低任务分类精度的同时减少任务的训练时间。但并不是所有主成分分析方法都能达到如上目的，因为对于不同的数据集和个体结构，采用的方法与参数也不同，所以，需要自适应的选择适合的主成分分析方法及其参数。除此之外，通过尝试我们可以发现，当任务复杂时，主成分分析方法对于时间的加速比也会增加，这给我们处理大规模数据集提供了思路。

## 5.2 改进结果分析

如表 6的1、2行所示，New\_IDGP在FEL\_1和KTH数据集上的精度，分别是98.00%和98.67%。可以发现在FEL\_1上达到了与IDGP相似的性能，但是KTH上表现比IDGP更佳，提高了2.96%。观察复现结果训练集的收敛行为，如图 11所示，种群在一开始就达到了较高的精度。一方面，New\_IDGP展现出了更高得稳定性，收敛曲线的波动明显较小，如图 11a, 11d所示。另一方面，由于添加了更多的图像相关算子，使得模型提取特征的能力增强，其模型精度也有所提高。

表 6. New\_IDGP与PUIDGP运行结果

No.	Method	Dataset	Test Accuracy	Training Time(s)	Test Time(s)
1	New_IDGP	FEL_1	98.00(=)	3462.90	1.04
2	New_IDGP	KTH	98.67(+)	34862.07	15.56
3	PUIDGP	FEL_1	78.00(-)	1056.50(+)	0.64
4	PUIDGP	KTH	90.00(-)	16885.18(+)	5.59

表 6的3、4行展示了PUFGP的结果，可以发现与IDGP的模型精度相比，在FEL\_1上降低了20.00%的精度，KTH上降低了5.71%的精度。但在训练时间上，在FEL\_1上实现了69.49%x的加速，KTH上实现51.56%x的加速。



从图 11b, 11e 可以发现, PUIDGP 在两个数据集的训练集上演化到 50 代时还未收敛, 所以在测试精度上表现较差。但在 KTH 数据集上的表现比 FEL\_1 的稳定, 所以, 相比之下 PUIDGP 在复杂数据集上的效果更好。从图 11c, 11f 可以发现, PCA 增大了训练过程中每代种群的平均个体大小, 但在 KTH 上却与 IDGP 的平均个体大小相似, 这也说明了为什么 PCA 在 KTH 数据集上的效果没有在 FEL\_1 数据集上的差, 因为虽然提取到了更多的特征, 但其个体大小与 IDGP 相似。

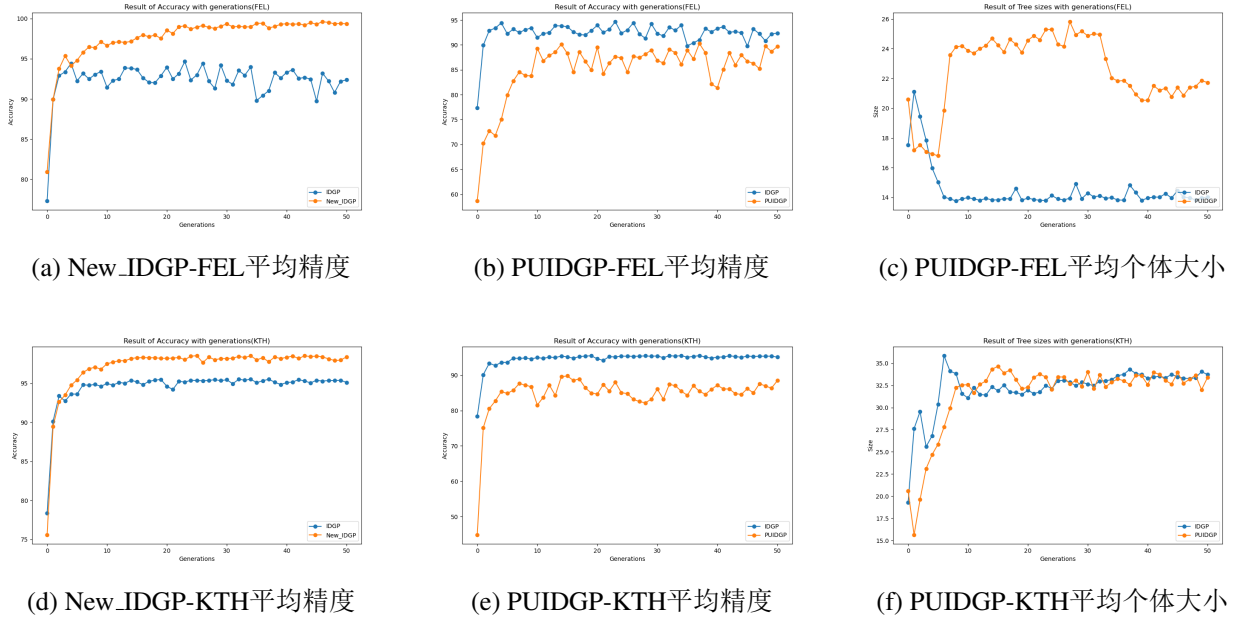


图 11. 复现结果训练集收敛行为

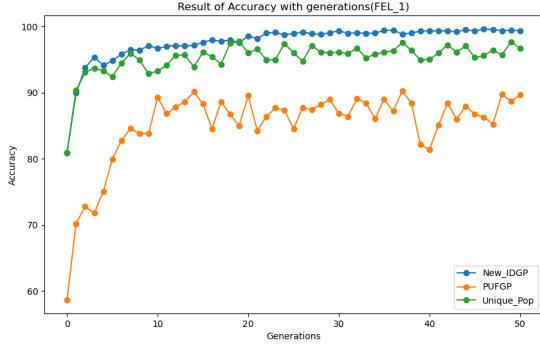
根据上述结果可以发现 New\_IDGP 对 IDGP 起到正向作用, 所以我们以 New\_IDGP 为基础, 分析向 New\_IDGP 添加 Unique\_Pop 和 PCA 对所构成的 PUIDGP 有什么贡献。

表 7. 进一步分析运行结果

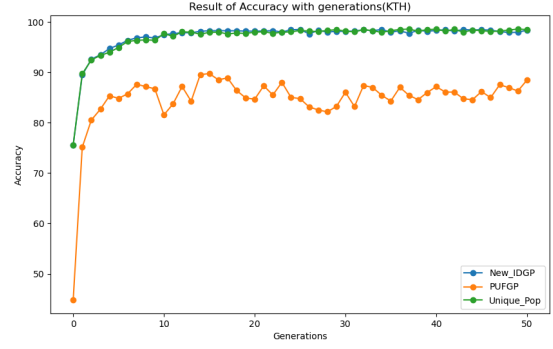
No.	Method	Dataset	Test Accuracy	Training Time(s)	Test Time(s)
1	Unique_Pop	FEL_1	92.00(-)	2673.68(+)	0.62
2	Unique_Pop	KTH	98.67(+)	28112.89(+)	11.71
3	PCA	FEL_1	68.00(-)	1035.92(+)	0.46
4	PCA	KTH	91.33(-)	6749.23(+)	1.00

如表 7 的 1、2 所示, 添加 Unique\_Pop 操作后, 与 New\_IDGP 得到的结果相比, 在 FEL\_1 数据集上降低了 6% 的精度, 在 KTH 数据集上降低了 0.12% 的精度。但在训练时间上均有减少, 在 FEL\_1 数据集上减少了 22.79% $\times$ , 在 KTH 数据集上减少了 19.36% $\times$ 。

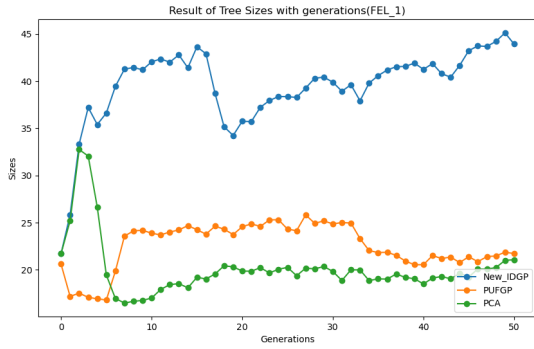
表 7 的 3、4 所示对个体得到的结果进行 PCA, 与 New\_IDGP 得到的结果相比, 在 FEL\_1 数据集上降低了 20% 的精度, 在 KTH 数据集上降低了 7.46% 的精度, 但在训练时间上达到了显著的减少, 在 FEL\_1 数据集上减少了 70.08% $\times$ , 在 KTH 数据集上减少了 80.64% $\times$ 。



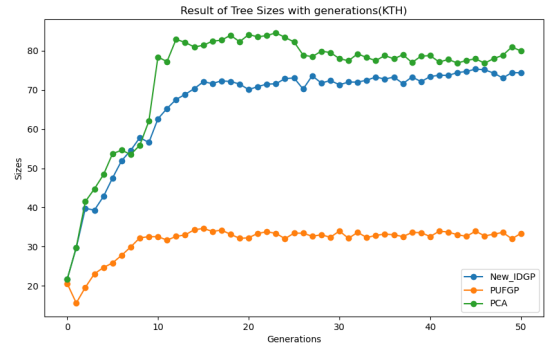
(a) FEL\_1训练集平均精度



(b) KTH训练集平均精度



(c) FEL\_1训练集平均个体大小



(d) KTH训练集平均个体大小

图 12. 进一步分析收敛曲线

观察Unique\_Pop的收敛行为曲线如图 11a, 11b所示, 可以发现Unique\_Pop在训练集上的收敛行为与FGP相似, 说明其对PUFGP的预测效果起到了正向作用, 同时也减少了整个算法的训练时间。特别可以发现, 任务越复杂, Unique\_Pop的表现越好。

由于PCA在预测效果上降低, 在训练时间上加速的表现, 我们分析其在演化过程中每一代平均个体大小, 如图 11c, 11d所示。可以发现在FEL\_1上, PCA显著降低了训练过程中每代种群的平均个体大小, 但在KTH上却增大了平均个体大小, 这也说明了为什么PCA在KTH数据集上的效果没有在FEL\_1数据集上的差, 因为提取到了更多的特征。对个体得到的特征进行PCA导致在FEL\_1数据集上精度降低较多是因为该分类任务较为简单, 经过PCA后可能虽然减少了冗余的特征, 但也有可能把有效的特征去除了。总之可以发现, 任务越复杂, PCA的表现越好。

总的来说, 分类任务越复杂, PUFGP的训练和预测效果越好。

## 6 总结与展望

本文复现了IDGP, 并通过结合添加图像相关算子, Unique\_Pop算法和主成分分析方法开发了一种新的基于GP的方法, 使用图像相关算子对不同类型的图像分类任务进行特征学习, 并在两个数据集上进行验证。提出的New\_IDGP提高了IDGP的性能, 提出的PUFGP可以为目标任务进化出不同深度的解, 并且可以让GP模型在搜索空间中进行有效搜索, 保持种群多样性。同时, 加入主成分分析方法可以有效减少任务的训练时间。实验结果表明, PUFGP结合

了New\_IDGP, Unique\_1和主成分分析方法的优点,能够在尽可能保持模型精度的前提下,减少模型训练时间。特别注意的是,分类任务越复杂,PUFGP的性能越好。PUFGP中只集成了两种PCA方法,而每个任务使用的方法不尽相同。在未来,将研究基于GP的图像方法集成不同的PCA方法,提升模型的泛化能力。

## 参考文献

- [1] Harith Al-Sahaf, Ausama Al-Sahaf, Bing Xue, Mark Johnston, and Mengjie Zhang. Automatically evolving rotation-invariant texture image descriptors by genetic programming. *IEEE Transactions on Evolutionary Computation*, 21(1):83–101, 2016.
- [2] Harith Al-Sahaf, Andy Song, Kourosh Neshatian, and Mengjie Zhang. Two-tier genetic programming: Towards raw pixel-based image classification. *Expert Systems with Applications*, 39(16):12291–12301, 2012.
- [3] Harith Al-Sahaf, Mengjie Zhang, Ausama Al-Sahaf, and Mark Johnston. Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving rotation-invariant texture image descriptors. *IEEE Transactions on Evolutionary Computation*, 21(6):825–844, 2017.
- [4] Wissam A. Albukhanajer, Johann A. Briffa, and Yaochu Jin. Evolutionary multiobjective image feature extraction in the presence of noise. *IEEE Transactions on Cybernetics*, 45(9):1757–1768, 2015.
- [5] Daniel Atkins, Kourosh Neshatian, and Mengjie Zhang. A domain independent genetic programming approach to automatic feature extraction for image classification. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 238–245. IEEE, 2011.
- [6] Ying Bi, Bing Xue, and Mengjie Zhang. An automatic feature extraction approach to image classification using genetic programming. In *Applications of Evolutionary Computation: 21st International Conference, EvoApplications 2018, Parma, Italy, April 4-6, 2018, Proceedings 21*, pages 421–438. Springer, 2018.
- [7] Ying Bi, Bing Xue, and Mengjie Zhang. A survey on genetic programming to image analysis. *Journal of Zhengzhou University (Engineering Science)*, 39(06):3–13, 2018.
- [8] Ying Bi, Bing Xue, and Mengjie Zhang. An automated ensemble learning framework using genetic programming for image classification. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 365–373, 2019.
- [9] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee, 2007.

- [10] Armand R. Burks and William F. Punch. Genetic programming for tuberculosis screening from raw x-ray images. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018.
- [11] C.E.Thomaz. Fei face database. 2012. <https://fei.edu.br/~cet/facedatabase.html>.
- [12] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.
- [13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.
- [14] Muhammad Iqbal, Bing Xue, Harith Al-Sahaf, and Mengjie Zhang. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Transactions on Evolutionary Computation*, 21(4):569–587, 2017.
- [15] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4:87–112, 1994.
- [16] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480, 2007.
- [17] Hao Li and Maoguo Gong. Self-paced convolutional neural networks. In *IJCAI*, pages 2110–2116, 2017.
- [18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, (2):60, 2004.
- [19] P Mallikarjuna, Alireza Tavakoli Targhi, Mario Fritz, Eric Hayman, Barbara Caputo, and Jan-Olof Eklundh. The kth-tips2 database. *Computational Vision and Active Perception Laboratory, Stockholm, Sweden*, 11:12, 2006.
- [20] David J Montana. Strongly typed genetic programming. *Evolutionary computation*, 3(2):199–230, 1995.
- [21] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
- [22] Michael O’ Neill. Riccardo poli, william b. langdon, nicholas f. mcphée: A field guide to genetic programming: Lulu. com, 2008, 250 pp, isbn 978-1-4092-0073-4, 2009.
- [23] Guangwu Qian and Lei Zhang. A simple feedforward convolutional conceptor neural network for classification. *Applied Soft Computing*, 70:1034–1041, 2018.

- [24] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning*, pages 833–840, 2011.
- [25] Szabolcs Serdyan. Color histogram features based image classification in content-based image retrieval systems. In *2008 6th international symposium on applied machine intelligence and informatics*, pages 221–224. IEEE, 2008.
- [26] Ling Shao, Li Liu, and Xuelong Li. Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1359–1371, 2014.