# 实验一 输入右清晰闹钟图像 使用KSVD训练 之后恢复实验

## 原图



## 测试一 out\_put=IMout./Weight; 仿造ASR

vecOfMeans = mean(blocks(:,jj:jumpSize));

blocks(:,jj:jumpSize)= Dictionary\*Coefs + ones(size(blocks,1),1) \* vecOfMeans;

IMout(row:row+bb-1,col:col+bb-1)=IMout(row:row+bb-1,col:col+bb-1)+block;

Weight(row:row+bb-1,col:col+bb-1)=Weight(row:row+bb-1,col:col+bb-1)+ones(bb);

**out\_put=IMout./Weight;**



## 测试二 这个实验做得很没有意义 因为没有牵扯训练出来的字典

vecOfMeans = mean(blocks(:,jj:jumpSize));

blocks(:,jj:jumpSize)= Dictionary\*Coefs + ones(size(blocks,1),1) \* vecOfMeans;

out\_put=IMout./Weight;

IOut = (Image+0.034\*0.85)./(1+0.034\*0.85);

errors=Image-IOut;

errors最大值是五点多

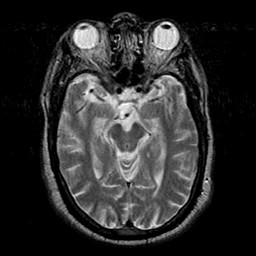


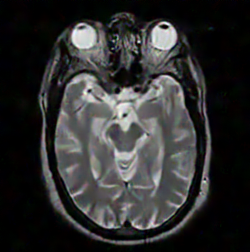
## 测试三 IOut = (Image+0.85\*IMout)./(1+0.85\*Weight);



这个测试说明原图根本就不能够恢复完整

## 测试四 使用JSR+AR直接测试 下图1为原图 使用IOut = (Image+0.85\*IMout)./(1+0.85\*Weight);恢复效果





此测试说明是可以恢复原图的 但是就是不知道是否和多聚焦原因有关 但是现在唯一不解的是为什么要通过IOut = (Image+0.85\*IMout)./(1+0.85\*Weight)获得结果 而不是通过像ASR中使用mean1 = mean(patch\_1(:));mean\_f=(mean1+mean2)/2;Patch\_f = Patch\_f + mean\_f;

## 测试五 直接使用JSR+AR dictionary.m测试闹钟·

### 闹钟

6554680.13832469



IOut=IMout./Weight;结果

error=sum(sum((image\_input1-y)^2));

值为6409012.28857142

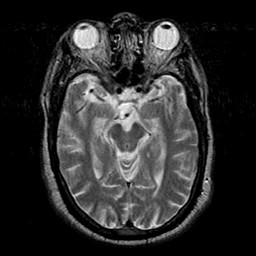
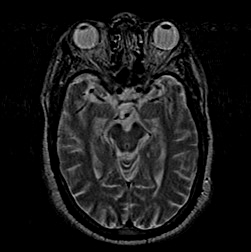


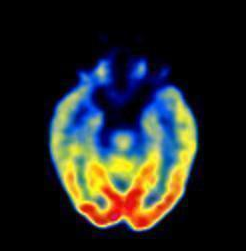
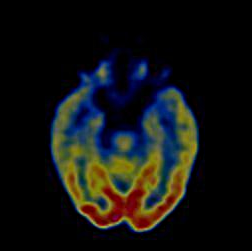
### MRI

IOut = (Image+0.034\*sigma\*IMout)./(1+0.034\*sigma\*Weight);

## 测试六 使用ASR测试恢复效果



直接对进行恢复效果都那么差说明可能原因就是一是因为字典原因 字典训练集是基于多聚焦那些数据集 二是由于字典不能精确表示原图像 下一步工作安排就是如何将图像恢复效果更好 步骤就是先进行IHS变换，进行图像块状分类 再进行分类。

# 实验二 IHS变换 论文中给出的转换矩阵无法复原

function [OUTPUT,FORRI,AA,BB]=RGB2IHS(input)

[M,N,Q]=size(input);

OUTPUT=zeros(M,N,Q);

FORRI=zeros(M,N,Q);

v11=zeros(3,1);

% AA=[1/3,1/3,1/3;

% -sqrt(2)/6,-sqrt(2)/6,-2\*(sqrt(2)/6); %RGB－>IHS正变换矩阵

% 1/sqrt(2),-1/sqrt(2),0];

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% BB=[1,-1/sqrt(2),1/sqrt(2);

% 1,-1/sqrt(2),-1/sqrt(2); %IHS->RGB逆变换矩阵

% 1,sqrt(2),0];

AA=[1/sqrt(3),1/sqrt(3),1/sqrt(3);

1/sqrt(6),1/sqrt(6),-2/sqrt(6); %RGB－>IHS正变换矩阵

1/sqrt(2),-1/sqrt(2),0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

BB=[1/sqrt(3),1/sqrt(6),1/sqrt(2);

1/sqrt(3),1/sqrt(6),-1/sqrt(2); %IHS->RGB逆变换矩阵

1/sqrt(3),-2/sqrt(6),0];

for i=1:M

for j=1:N

v11(1)=double(input(i,j,1));

v11(2)=double(input(i,j,2));

v11(3)=double(input(i,j,3));

u1=AA\*v11;

OUTPUT(i,j,1)=u1(1);

OUTPUT(i,j,2)=1/(tan(u1(3)/u1(2)));

OUTPUT(i,j,3)=sqrt(u1(2)^2+u1(3)^2);

u2=BB\*u1;

FORRI(i,j,1)=u2(1);

FORRI(i,j,2)=u2(2);

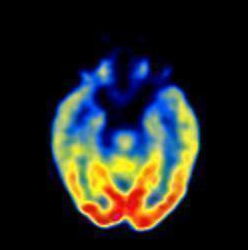
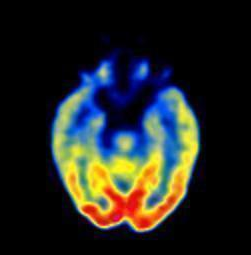
FORRI(i,j,3)=u2(3);

end

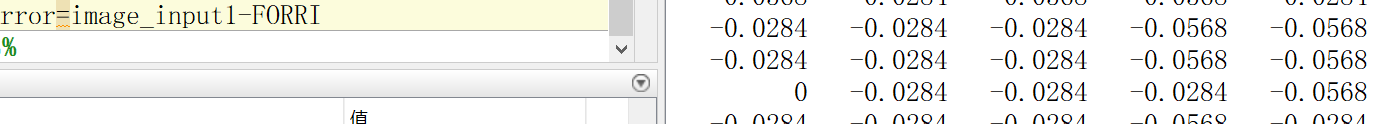
end

end

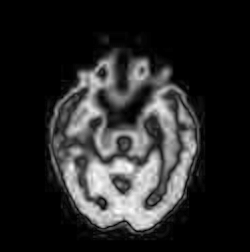
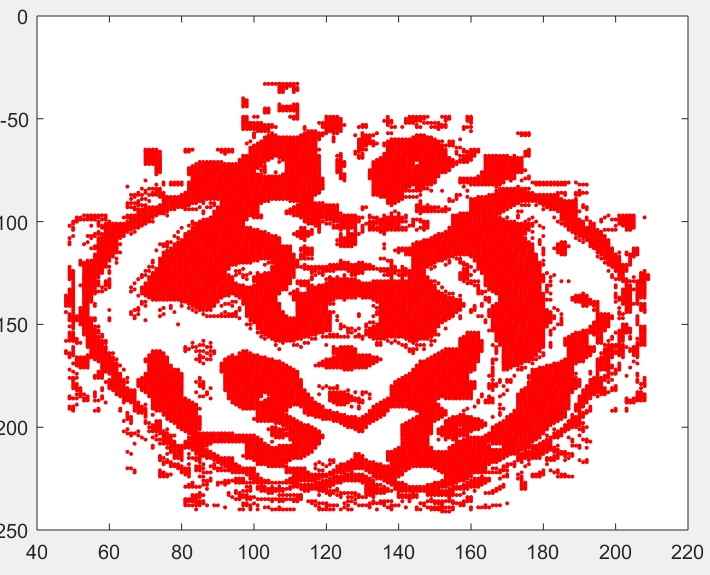
下图是复原前和复原后

还是有些略微差别 可能是由于根号原因

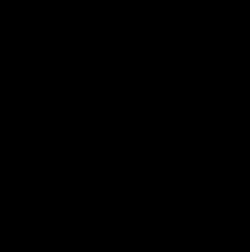


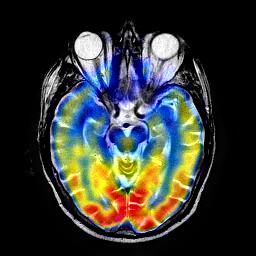
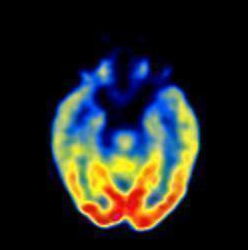
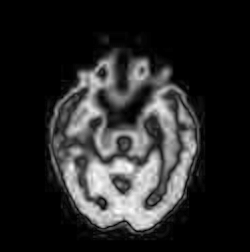
分别是IHS分量





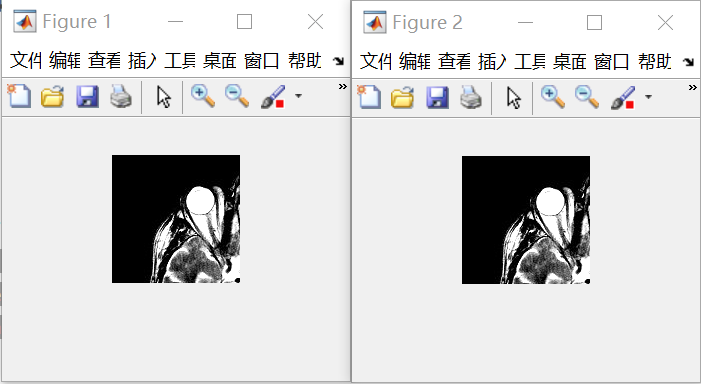


分量3全部为零 由于灰度图像三个分量对应相等



现在主要就是设计一个实验能够验证对I分量图像进行复原

# 实验三 梯度信息得到字典 然后进行恢复 kk=size(X,1) 1是列数 2是行数

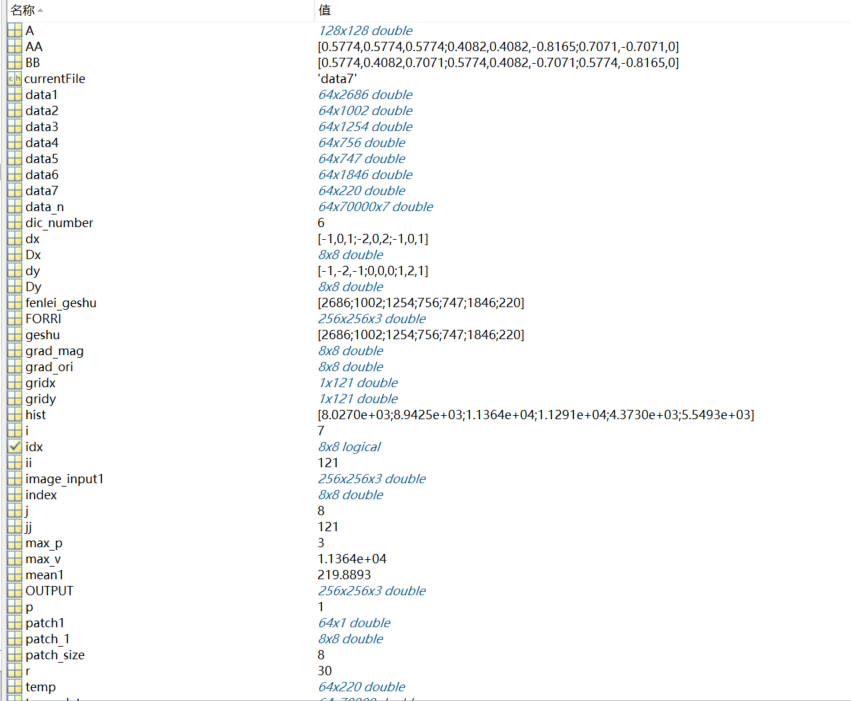


通过对MRI亮度分量进行分成8X8小块然后使用KSVD算法 发现其恢复能力较强，但是此实验唯一不足就是没有将梯度角属于第七类的加入，理论上来说，加上加上效果不会比这个差，下一步计划就是通过对多样本学习，或者将已经学习好的字典中选择效果最好的（这个步骤存疑，就是不知道是单单训练一张图像，然后训练好一张图像之后将字典替代DCT字典加入，还是将所有样本进行分类统一训练效果好），下一步实验应该是验证联合字典对PET和MRI分量的表征能力。本实验代码见TEST3TO1。

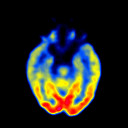
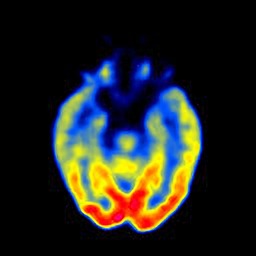
# 实验四 联合字典对PET和MRI分量表征能力测试

## 测试一 训练数据集选取 选取每种疾病SF越大的 TEST4 PET取得SF前八张 MRI取得14类SF排第一的

## 测试二 将MRI和PET分别进行HIS变换 将I分量进行字典训练 看效果



发现个问题 就是PET图像是128的 通过放大之后效果 可以说是个好方法 可以试试用在图像处理中



## 测试三 将两张PET和两张MRI的I分量合并一次性就行字典训练 看看合成效果 现在主要就是还有一点改进的地方就是将PET进行无损放大 而不是简单的imresize

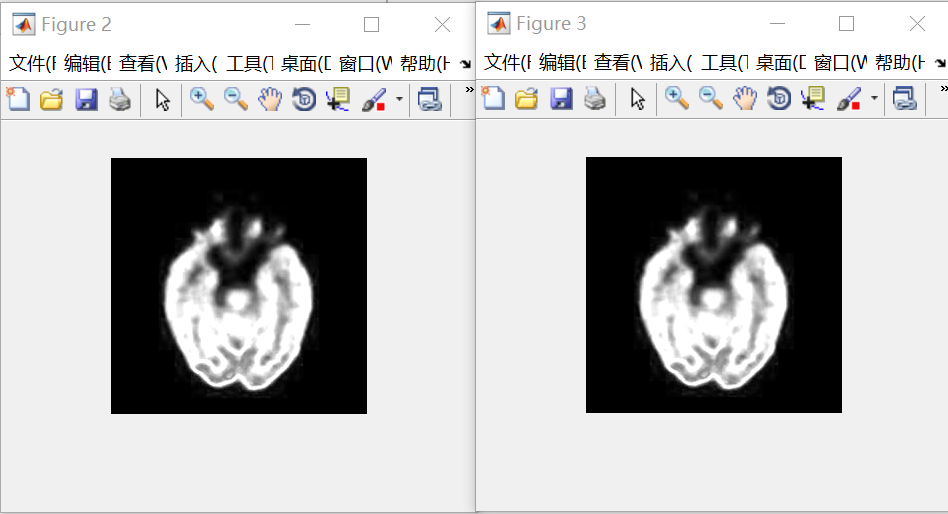


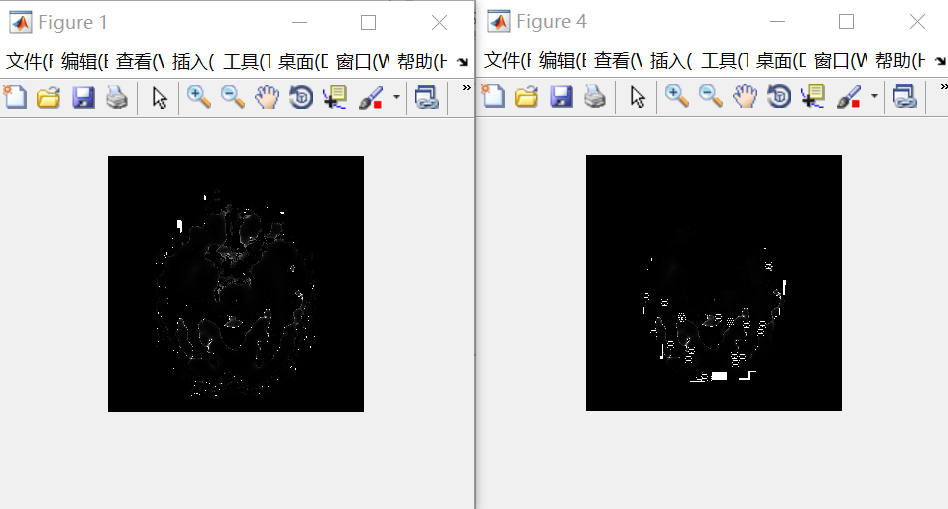


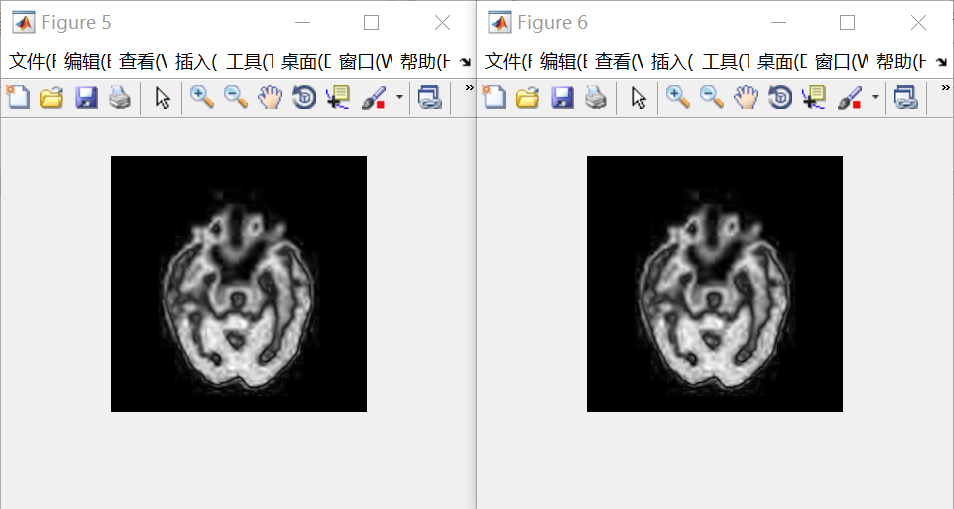
可以看到恢复效果还是不错的 这还只是在单单使用128X128训练情况下

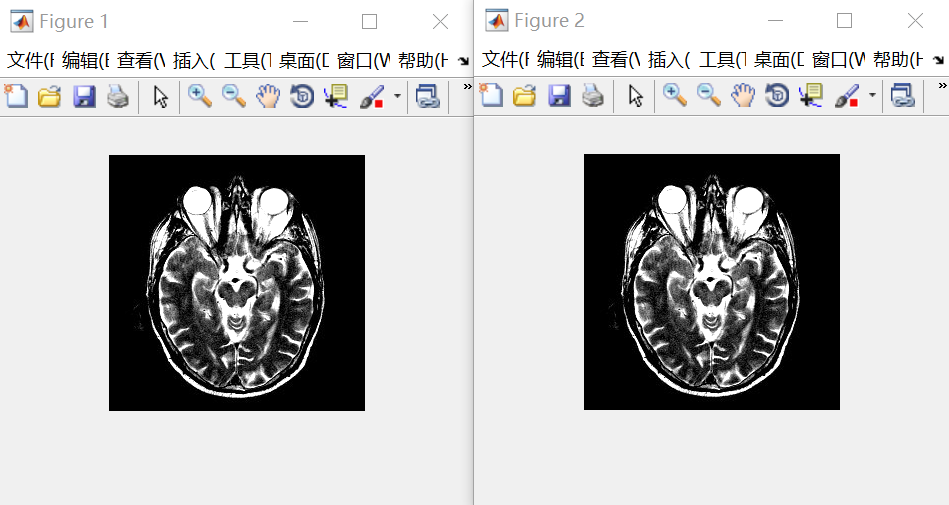
## 测试四 写一个综合程序 然后根据输入的四张图像分别得到I H S分量 分别进行存储得到MAT文件

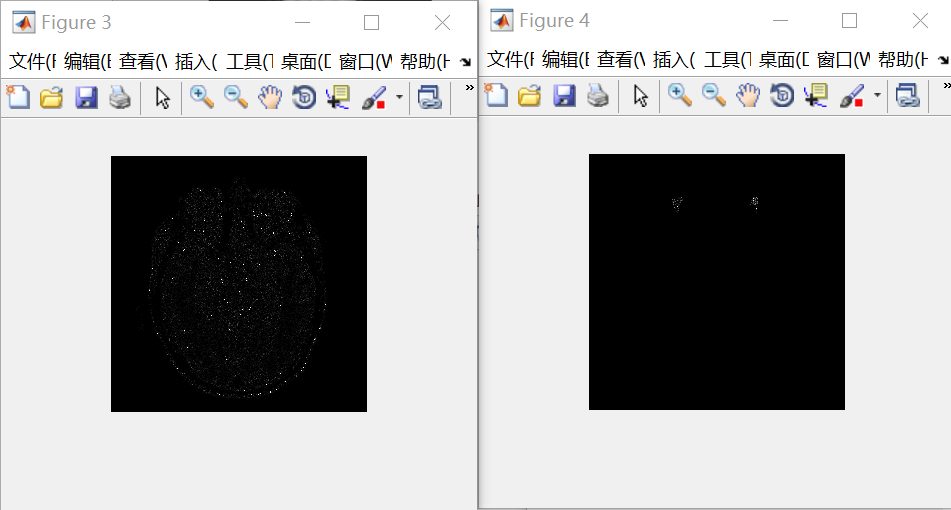
# 实验五 通过对PET图像I H S分量分别进行梯度分类 字典训练 发现其表现能力较差对于H分量 由于其稀疏性

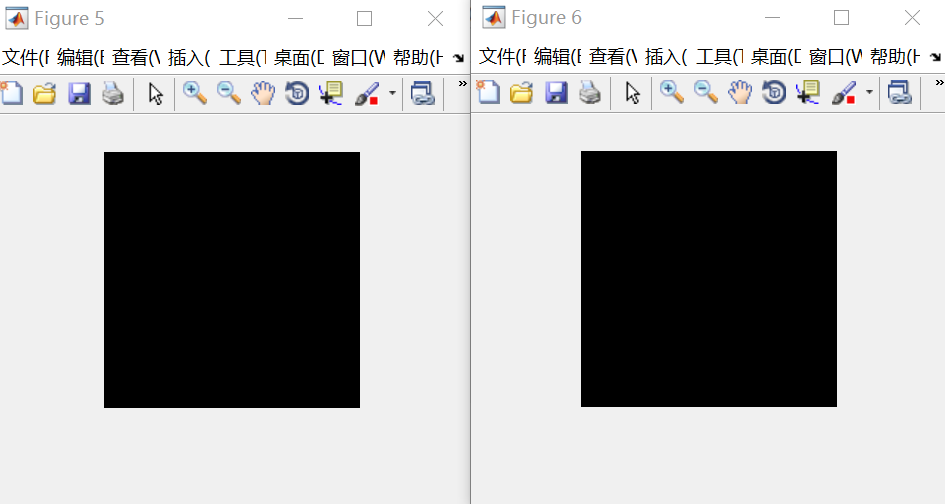


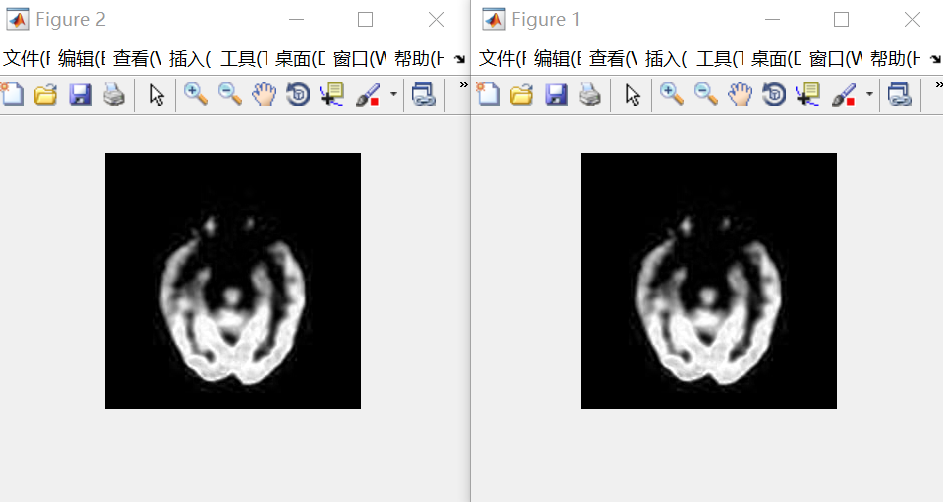


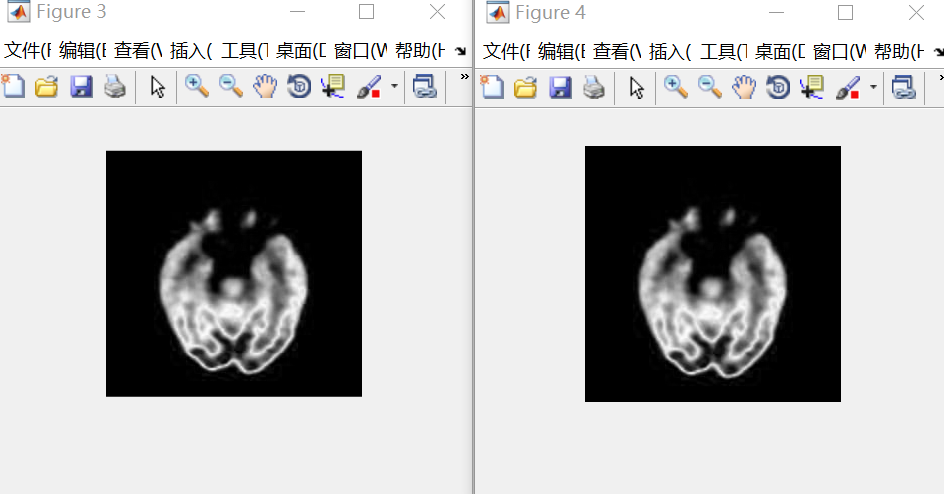


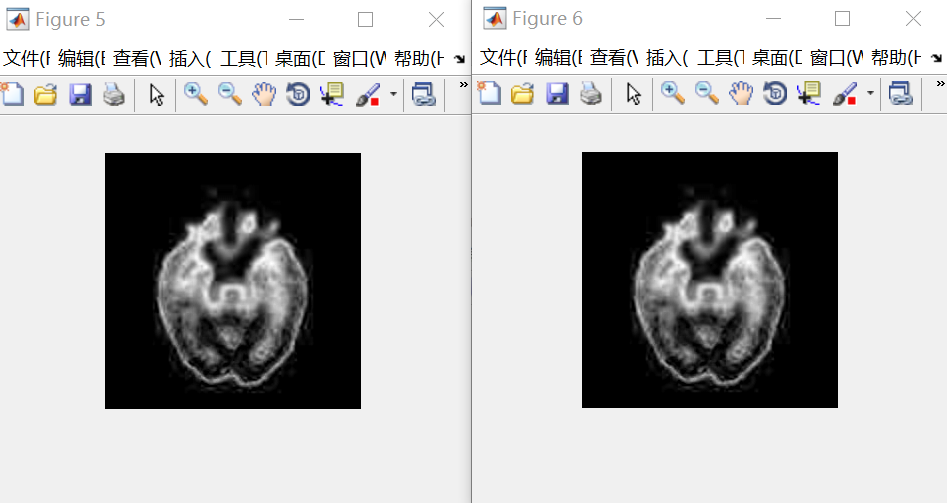






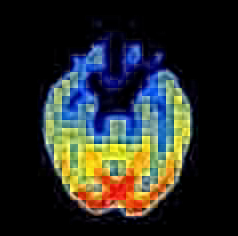


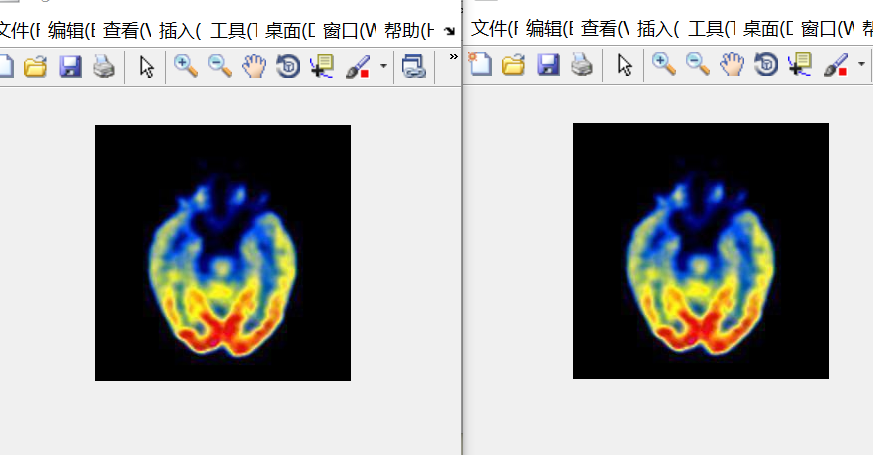




# 实验六 窗口重叠像素减少和参数预训练实验

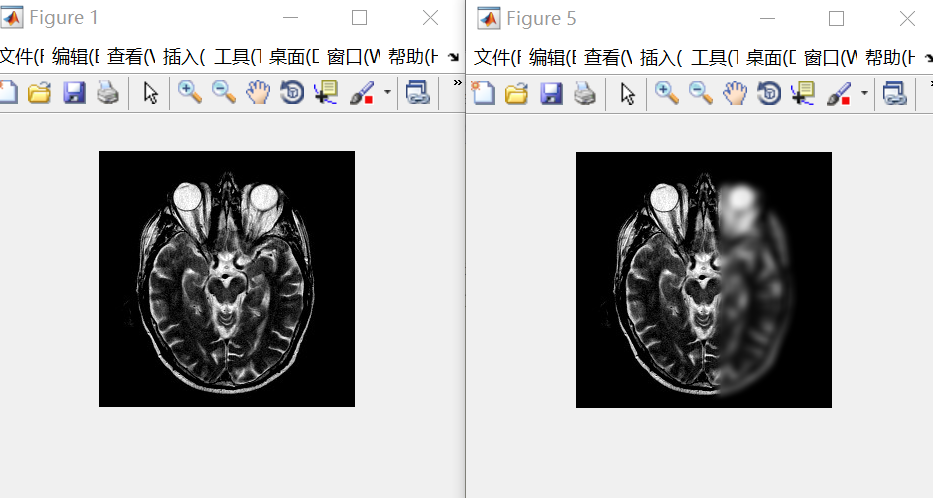
### 测试一 为了减少时间 预先分离系数 但是带来个问题就是发现彩色部分分离不了 恢复时候显示这样

 单维恢复也是这样

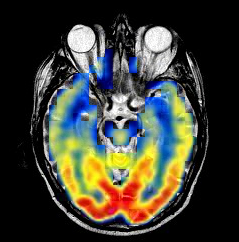


可是如果放在CUN\_ZIDIAN\_XISHU\_HUIFU\_XISHU好像又能恢复

### 测试二 为了减少时间 将采用预先训练稀疏的方法 但是由于线数不够 导致其方法还是需要加强

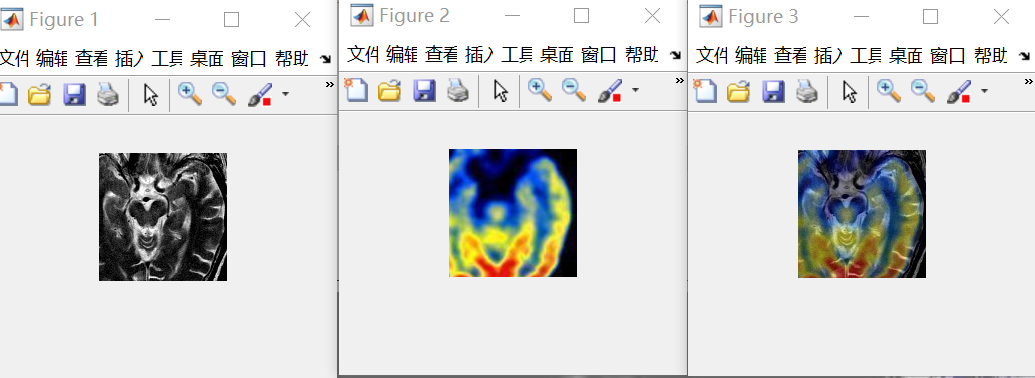


### 测试三 但是如果采用重叠像素个数为0的情况则会出现 说明重叠个数为0不太可行 详情见RGB\_FUSION\_TEST3.m



# 实验七 通过对小块进行I H S变换之后 获取亮度权重 然后进行用RGB分别进行表示合成

## 测试一 采用加权方式一的话 效果就这样 不太好



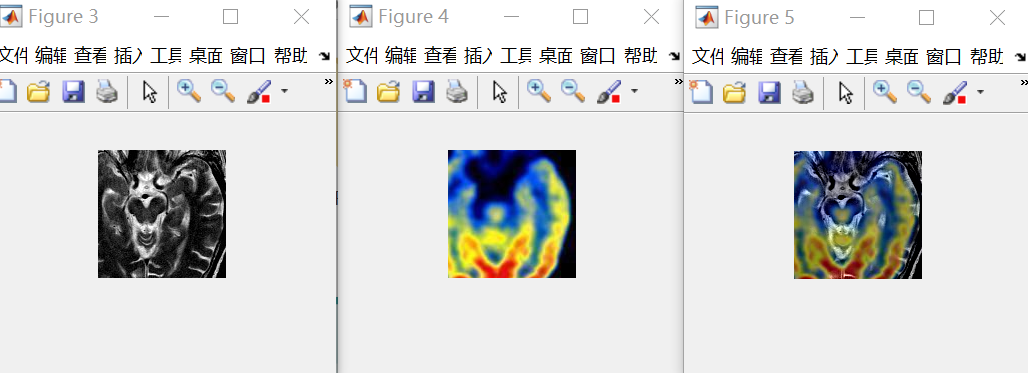
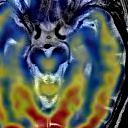
% [F1,w1,max\_p1]=USE\_DICTIONARY(patch\_1,0);

% [F2,w2,max\_p2]=USE\_DICTIONARY(patch\_1,0);

这个测试要做过一次 因为程序写错了 此处使用的都是patch\_1，但是神奇的是为什么程序写错了还是会出现融合效果

关于亮度分析就是由于是取各自的部分还进行了加权 导致就算是很大的加权一下其实也很小了

## 测试二 加权方式采用了最大值原则



这里之所以出现这种情况是因为I分量中的凸出部分和彩色I分量相比 其实没有多大的优势 因为彩色图像的I分量可能也很大 导致二者不相上下 就无法比较出来了 所以一种更好的方法需要先通过实验验证 能够将二者在一起比较时 能够准确把I分量识别出来

## 加权程序一 核心思想就是先完全恢复 然后再取权重

currentFile = sprintf('Dksvd%d.mat',max\_p1);

load(currentFile);

patch\_f\_f1=Dksvd\*w1+mean1;

currentFile = sprintf('Dksvd%d.mat',max\_p2);

load(currentFile);

patch\_f\_f2=Dksvd\*w2+mean2;

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=patch\_f\_f1\*(temp\_max/sum\_percent)+patch\_f\_f2\*(1-(temp\_max/sum\_percent));

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=patch\_f\_f2\*(temp\_max/sum\_percent)+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

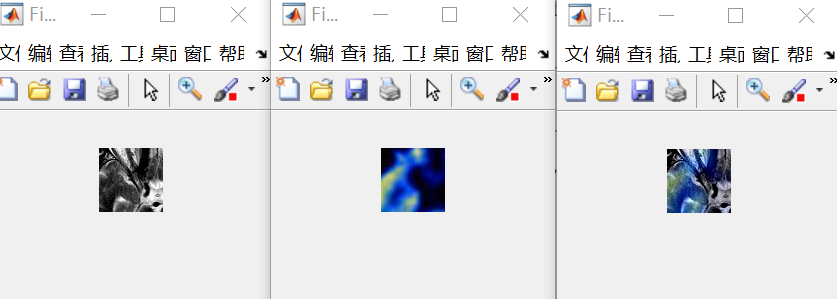
if percent\_patch1 == percent\_patch2

P1atch\_f=patch\_f\_f2\*(temp\_max/sum\_percent)+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;



## 加权程序二 测试了在重叠为0和重叠为7情况下融合效果

这部分算法就是在加权程序一基础上 通过思考应当在权重比较大时选择 彻底导向灰度图像 附带加上彩色图像的加权 patch\_f\_f2\*(1-(temp\_max/sum\_percent))

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

currentFile = sprintf('Dksvd%d.mat',max\_p1);

load(currentFile);

patch\_f\_f1=Dksvd\*w1;

currentFile = sprintf('Dksvd%d.mat',max\_p2);

load(currentFile);

patch\_f\_f2=Dksvd\*w2;

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%%

mean\_f=(mean1+mean2)/2;

sum\_percent=abs(percent\_patch1)+abs(percent\_patch2);

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2\*(1-(temp\_max/sum\_percent));

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

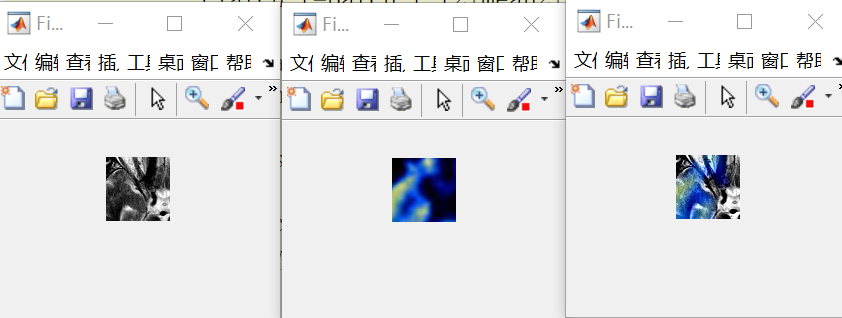
P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

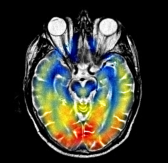
end

if percent\_patch1 == percent\_patch2

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end



跑出来效果还是不错的 但是就是部分区域难以显现

## 加权程序三 如果彩色部分I分量低于均值 则全加上

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%为了提高对彩色部分表现能力 现如今采用对彩色部分远远低于均值的时候将所有的彩色部分全部加上

if percent\_patch2<0

P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2\*(1-(temp\_max/sum\_percent));

end

end

if percent\_patch1 < percent\_patch2

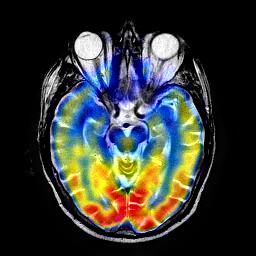
temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

 P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

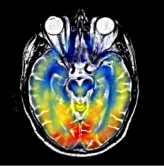
end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end



虽然是由提升 但是存在问题就是发现部分区域色彩较淡 比如红色区域

## 加权程序四 如果不是主结构信息（如果小于均值） 就把彩色图像全显示



if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%如果是彩色区域比重大 则全彩显示

if percent\_patch1<0

P1atch\_f=patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1- (temp\_max/sum\_percent));

end

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

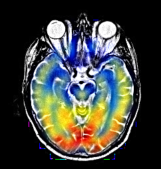
cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end

通过上图颜色淡问题分析知道一点 那就是可能有些地方虽然视觉上是均值看起来小于零 但是实际上并不会小于零 所以导致不该加权的地方还是得到了加权

如果这个还不行 说明计算均值方式存在问题 不应该取除整张图像

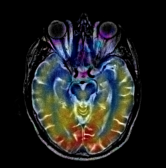
## 加权程序五 改变求均值方式



根据此图思考有两点发现：一是其实如果灰度占优情况下 不要给彩色部分赋权重最小值 因为灰度已经占有情况下 权重值最小的肯定小于一半

二是 其实在彩色占有情况下 要显示全是彩色部分 不应该只是在percent\_patch1小于零情况下 因为这种情况未必有那么容易满足

## 加权程序六 改变在彩色占有情况 显示全彩条件

想不通为什么会这样 为什么会出现红色的

%%

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch1模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_1 = A(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

if sum(patch\_1)~=0

mean1 = mean(patch\_1(:));%它这里求均值应该是求RGB整体均值

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1 I分量突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I1 = I1(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

A1=(patch\_I1-mean\_I1)/mean\_I1;%和自己比看都大多少

temp\_mean\_I1=find(patch\_I1(:)~=0);

percent\_patch1=sum(sum(A1))/(length(temp\_mean\_I1));%这里的均值应该求的是整体

else

patch1\_is\_zeros=1;

mean1=0;

end

%%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch2模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%则肯定就转不成正数了 所以意义不大

patch\_2 = B(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

if sum(patch\_2)~0

mean2 = mean(patch\_2(:));%它这里求均值应该是求RGB整体均值

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I2 = I2(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

A2=(patch\_I2-mean\_I2)/mean\_I2;%和自己比看都大多少

temp\_mean\_I2=find(patch\_I2(:)~=0);

percent\_patch2=sum(sum(A2))/(length(temp\_mean\_I2));%全部转为正数 但是如果全部为负数

else

patch2\_is\_zeros=1;

mean2=0;

end

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求彩色图像H分量突出度\*\*\*\*\*\*\*\*\*\*

patch\_S2 = S2(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

A2=(patch\_S2-mean\_S2)/mean\_S2;%和自己比看都大多少

temp\_mean\_S2=find(patch\_S2(:)~=0);

percent\_patch2\_S\_fenliang=sum(sum(A2))/(length(temp\_mean\_S2));%计算彩色图像S分量突出多少

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%\*\*\*\*加权方式一配套\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%如果采用方式二 这里就可以注释了

if jiaquan==1 | 2

if zhuangzai\_xishu==0%换系数获取方式 换成通过查找系数的方式获取

[F1,xishu,max\_p1]=USE\_DICTIONARY(patch\_1,0);

w1=xishu;

[F2,xishu,max\_p2]=USE\_DICTIONARY(patch\_2,0);

w2=xishu;

else

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch1',num2str(channel),';']);

w1=cun\_xishu\_patch1(:,windows);

max\_p1=cun\_fenlei\_patch1(windows);

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch2',num2str(channel),';']);

w2=cun\_xishu\_patch1(:,windows);

max\_p2=cun\_fenlei\_patch1(windows);

end

end

%%

%乘上字典模块 为了减少运算时间 如果为零的话 就不用那么麻烦了 直接赋值就行

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

if sum(w1)==0

patch\_f\_f1=zeros(64,1);

else

currentFile = sprintf('Dksvd%d.mat',max\_p1);

load(currentFile);

patch\_f\_f1=Dksvd\*w1;

end

if sum(w2)==0

patch\_f\_f2=zeros(64,1);

else

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

currentFile = sprintf('Dksvd%d.mat',max\_p2);

load(currentFile);

patch\_f\_f2=Dksvd\*w2;

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%%

if patch1\_is\_zeros==1 | patch2\_is\_zeros==1

if patch1\_is\_zeros==1 & patch2\_is\_zeros==1

%如果都为0则直接赋值0

P1atch\_f=zeros(64,1);

else

P1atch\_f=patch1\_is\_zeros\*(patch\_f\_f1+mean1)+patch2\_is\_zeros\*(patch\_f\_f2+mean2);

end

else

mean\_f=(mean1+mean2)/2;

sum\_percent=abs(percent\_patch1)+abs(percent\_patch2);

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%为了提高对彩色部分表现能力 现如今采用对彩色部分远远低于均值的时候将所有的彩色部分全部加上

if percent\_patch2<0

P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f1+mean1+(patch\_f\_f2+mean2)\*((temp\_max/sum\_percent));

end

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%如果是彩色区域比重大 则全彩显示

if percent\_patch1<0 | percent\_patch2\_S\_fenliang>=0

P1atch\_f=patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

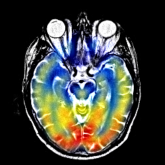
F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end

## 加权程序七

### 效果部分



### 程序部分

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*I分量为引导 RGB融合实验\*\*\*\*\*\*\*\*\*\*\*\*

%注意点1 如果改变求均值方式的时候 还要改变percent\_patch2地方的求值方式

%注意点2 此处发现个很大的bug 那就是在求percent或者其他时 如果a=0 那么无论是判断大于零还是小于零 都是不会有

% 结果的

% a=0/0;

% % a=1;

% if a>-0.2

% k=3;

% end

% 注意点3 load操作其实很费时间

close all

%本程序是第三种加权方法的融合实验

clc;clear;

test\_ronghe=1;

shangxian=1;

xiaxian=0;

bufen=1;

zhuangzai\_xishu=1;%采用预先训练字典获取系数之后再融合

jiaquan=1;%加权方式选择

if test\_ronghe==1

for ii=1:7

% cd('.\MAT\_DATA');%此处如果报错 则采用将字典文件复制到外面的方法 装载其实很费时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

currentFile = sprintf('Dksvd%d.mat',ii);

load(currentFile);

eval(['Dksvd',num2str(ii),'=','Dksvd',';']);

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%

end

if zhuangzai\_xishu==1%预先装载节省时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

for channel=1:3

currentFile = sprintf('%dfenlei\_patch1.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch1',num2str(channel),'=','cun\_fenlei\_patch1',';']);

currentFile = sprintf('%dzidian\_xishu\_patch1.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch1',num2str(channel),'=','cun\_xishu\_patch1',';']);

currentFile = sprintf('%dfenlei\_patch2.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

currentFile = sprintf('%dzidian\_xishu\_patch2.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

end

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

image\_input1=double(imread('./2.jpg'));

image\_input2=double(imread('./3.jpg'));

% image\_input1=imresize(image\_input1,[256 256]);

% image\_input2=imresize(image\_input2,[256 256]);

A=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

B=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

Fusion\_image=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

for i=1:3

A(:,:,i)=image\_input1(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input1,2)/bufen,i);

B(:,:,i)=image\_input2(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input2,2)/bufen,i);

end

figure;

imshow(uint8(A),[]);

figure;

imshow(uint8(B),[]);

[Ifenliang\_1,FORRI,AA,BB]=RGB2IHS(A);

[Ifenliang\_2,FORRI,AA,BB]=RGB2IHS(B);

I1=Ifenliang\_1(:,:,1);

I2=Ifenliang\_2(:,:,1);

S2=Ifenliang\_2(:,:,3);

[I1\_zero1,I1\_zero2]=find(I1>0);

[I2\_zero1,I2\_zero2]=find(I2>0);

[S2\_zero1,S2\_zero2]=find(I2>0);

%此处改变算均值的方式

mean\_I1=sum(sum(I1))/(length(I1\_zero1));

mean\_I2=sum(sum(I2))/(length(I2\_zero1));

mean\_S2=sum(sum(S2))/(length(S2\_zero1));

% mean\_I1=sum(sum(I1))/(size(I1,1)\*size(I1,2));

% mean\_I2=sum(sum(I2))/(size(I2,1)\*size(I2,2));

dic\_number=6;

r=180/dic\_number;

patch\_size=8;

overlap=7;

A100=A;

B100=B;

for channel=1:3

windows=0;

A=A100(:,:,channel);

B=B100(:,:,channel)

[kuan,chang]=size(B);

F=zeros(kuan,chang);

cntMat=zeros(kuan,chang);

gridx = 1:patch\_size - overlap : kuan-patch\_size+1;

gridy = 1:patch\_size - overlap : chang-patch\_size+1;

dx=[-1 0 1;-2 0 2;-1 0 1];

dy=dx';

for ii = 1:length(gridx)

for jj = 1:length(gridy)

patch1\_is\_zeros=0;

patch2\_is\_zeros=0;

windows=windows+1;

xx = gridx(ii);

yy = gridy(jj);

%注意点：为了避免开头注意点2陷阱 现进行预处理 先判断是不是为0

%%

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch1模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_1 = A(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

mean1 = mean(patch\_1(:));

patch1 = patch\_1(:) - mean1;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I1 = I1(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I1~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A1=(patch\_I1-mean\_I1)/mean\_I1;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch1=sum(sum(A1))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch1=sum(sum(A1))/(size(A1,1)\*size(A1,2));%全部转为正数 但是如果全部为负数

%%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch2模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_2 = B(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

mean2 = mean(patch\_2(:));

patch2 = patch\_2(:) - mean2;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I2 = I2(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I2~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A2=(patch\_I2-mean\_I2)/mean\_I2;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch2=sum(sum(A2))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch2=sum(sum(A2))/(size(A2,1)\*size(A2,2));%全部转为正数 但是如果全部为负数

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%\*\*\*\*加权方式一配套\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%如果采用方式二 这里就可以注释了

if jiaquan==1 | 2

if zhuangzai\_xishu==0%换系数获取方式 换成通过查找系数的方式获取

[F1,xishu,max\_p1]=USE\_DICTIONARY(patch\_1,0);

w1=xishu;

[F2,xishu,max\_p2]=USE\_DICTIONARY(patch\_2,0);

w2=xishu;

else

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch1',num2str(channel),';']);

w1=cun\_xishu\_patch1(:,windows);

max\_p1=cun\_fenlei\_patch1(windows);

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch2',num2str(channel),';']);

w2=cun\_xishu\_patch1(:,windows);

max\_p2=cun\_fenlei\_patch1(windows);

end

end

%%

%乘上字典模块 为了减少运算时间 如果为零的话 就不用那么麻烦了 直接赋值就行

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

if sum(w1)==0

patch\_f\_f1=zeros(64,1);

else

% currentFile = sprintf('Dksvd%d.mat',max\_p1);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p1),';']);

patch\_f\_f1=Dksvd\*w1;

end

if sum(w2)==0

patch\_f\_f2=zeros(64,1);

else

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

% currentFile = sprintf('Dksvd%d.mat',max\_p2);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p2),';']);

patch\_f\_f2=Dksvd\*w2;

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%%

mean\_f=(mean1+mean2)/2;

sum\_percent=abs(percent\_patch1)+abs(percent\_patch2);

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%为了提高对彩色部分表现能力 现如今采用对彩色部分远远低于均值的时候将所有的彩色部分全部加上

if percent\_patch2<0

P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f1+mean1+(patch\_f\_f2+mean2)\*((temp\_max/sum\_percent));

end

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%如果是彩色区域比重大 则全彩显示

if percent\_patch1<0

P1atch\_f=patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end

%cnt

fprintf([num2str(ii), ' rows have been processed\n']);

if mod(ii,5)==0

disp(['The channel ', num2str(channel),' was working!']);

end

end

%idx是逻辑值 所以知道其如果所在位置逻辑值不为1 则

%不会进行任何操作

idx = (cntMat < 1);

F(idx) = (A(idx)+B(idx))./2;

cntMat(idx) = 1;

F = F./cntMat;

Fusion\_image(:,:,channel)=F;

end

figure;

imshow(uint8(Fusion\_image),[]);

imwrite(uint8(Fusion\_image),'./fusion\_image.jpg');

end

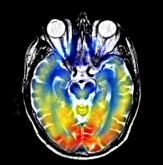
%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%

## 加权程序八

### 思路：从程序七效果来看 对于彩色图像弱小部分（小于均值） 其实效果还是可以的 特别是蓝色部分较深 但是对于彩色图像不弱小部分 但是又不占有部分 加权效果不佳 现改变这部分加权 既然灰度部分已经占有 就不怕你掩盖了

### 效果部分



### 程序部分

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*I分量为引导 RGB融合实验\*\*\*\*\*\*\*\*\*\*\*\*

%注意点1 如果改变求均值方式的时候 还要改变percent\_patch2地方的求值方式

%注意点2 此处发现个很大的bug 那就是在求percent或者其他时 如果a=0 那么无论是判断大于零还是小于零 都是不会有

% 结果的

% a=0/0;

% % a=1;

% if a>-0.2

% k=3;

% end

% 注意点3 load操作其实很费时间

close all

%本程序是第三种加权方法的融合实验

clc;clear;

test\_ronghe=1;

shangxian=1;

xiaxian=0;

bufen=1;

zhuangzai\_xishu=1;%采用预先训练字典获取系数之后再融合

jiaquan=1;%加权方式选择

if test\_ronghe==1

for ii=1:7

% cd('.\MAT\_DATA');%此处如果报错 则采用将字典文件复制到外面的方法 装载其实很费时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

currentFile = sprintf('Dksvd%d.mat',ii);

load(currentFile);

eval(['Dksvd',num2str(ii),'=','Dksvd',';']);

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%

end

if zhuangzai\_xishu==1%预先装载节省时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

for channel=1:3

currentFile = sprintf('%dfenlei\_patch1.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch1',num2str(channel),'=','cun\_fenlei\_patch1',';']);

currentFile = sprintf('%dzidian\_xishu\_patch1.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch1',num2str(channel),'=','cun\_xishu\_patch1',';']);

currentFile = sprintf('%dfenlei\_patch2.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

currentFile = sprintf('%dzidian\_xishu\_patch2.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

end

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

image\_input1=double(imread('./2.jpg'));

image\_input2=double(imread('./3.jpg'));

% image\_input1=imresize(image\_input1,[256 256]);

% image\_input2=imresize(image\_input2,[256 256]);

A=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

B=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

Fusion\_image=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

for i=1:3

A(:,:,i)=image\_input1(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input1,2)/bufen,i);

B(:,:,i)=image\_input2(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input2,2)/bufen,i);

end

figure;

imshow(uint8(A),[]);

figure;

imshow(uint8(B),[]);

[Ifenliang\_1,FORRI,AA,BB]=RGB2IHS(A);

[Ifenliang\_2,FORRI,AA,BB]=RGB2IHS(B);

I1=Ifenliang\_1(:,:,1);

I2=Ifenliang\_2(:,:,1);

S2=Ifenliang\_2(:,:,3);

[I1\_zero1,I1\_zero2]=find(I1>0);

[I2\_zero1,I2\_zero2]=find(I2>0);

[S2\_zero1,S2\_zero2]=find(I2>0);

%此处改变算均值的方式

mean\_I1=sum(sum(I1))/(length(I1\_zero1));

mean\_I2=sum(sum(I2))/(length(I2\_zero1));

mean\_S2=sum(sum(S2))/(length(S2\_zero1));

% mean\_I1=sum(sum(I1))/(size(I1,1)\*size(I1,2));

% mean\_I2=sum(sum(I2))/(size(I2,1)\*size(I2,2));

dic\_number=6;

r=180/dic\_number;

patch\_size=8;

overlap=7;

A100=A;

B100=B;

for channel=1:3

windows=0;

A=A100(:,:,channel);

B=B100(:,:,channel)

[kuan,chang]=size(B);

F=zeros(kuan,chang);

cntMat=zeros(kuan,chang);

gridx = 1:patch\_size - overlap : kuan-patch\_size+1;

gridy = 1:patch\_size - overlap : chang-patch\_size+1;

dx=[-1 0 1;-2 0 2;-1 0 1];

dy=dx';

for ii = 1:length(gridx)

for jj = 1:length(gridy)

patch1\_is\_zeros=0;

patch2\_is\_zeros=0;

windows=windows+1;

xx = gridx(ii);

yy = gridy(jj);

%注意点：为了避免开头注意点2陷阱 现进行预处理 先判断是不是为0

%%

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch1模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_1 = A(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

mean1 = mean(patch\_1(:));

patch1 = patch\_1(:) - mean1;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I1 = I1(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I1~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A1=(patch\_I1-mean\_I1)/mean\_I1;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch1=sum(sum(A1))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch1=sum(sum(A1))/(size(A1,1)\*size(A1,2));%全部转为正数 但是如果全部为负数

%%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch2模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_2 = B(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

mean2 = mean(patch\_2(:));

patch2 = patch\_2(:) - mean2;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I2 = I2(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I2~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A2=(patch\_I2-mean\_I2)/mean\_I2;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch2=sum(sum(A2))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch2=sum(sum(A2))/(size(A2,1)\*size(A2,2));%全部转为正数 但是如果全部为负数

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%\*\*\*\*加权方式一配套\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%如果采用方式二 这里就可以注释了

if jiaquan==1 | 2

if zhuangzai\_xishu==0%换系数获取方式 换成通过查找系数的方式获取

[F1,xishu,max\_p1]=USE\_DICTIONARY(patch\_1,0);

w1=xishu;

[F2,xishu,max\_p2]=USE\_DICTIONARY(patch\_2,0);

w2=xishu;

else

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch1',num2str(channel),';']);

w1=cun\_xishu\_patch1(:,windows);

max\_p1=cun\_fenlei\_patch1(windows);

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch2',num2str(channel),';']);

w2=cun\_xishu\_patch1(:,windows);

max\_p2=cun\_fenlei\_patch1(windows);

end

end

%%

%乘上字典模块 为了减少运算时间 如果为零的话 就不用那么麻烦了 直接赋值就行

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

if sum(w1)==0

patch\_f\_f1=zeros(64,1);

else

% currentFile = sprintf('Dksvd%d.mat',max\_p1);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p1),';']);

patch\_f\_f1=Dksvd\*w1;

end

if sum(w2)==0

patch\_f\_f2=zeros(64,1);

else

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

% currentFile = sprintf('Dksvd%d.mat',max\_p2);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p2),';']);

patch\_f\_f2=Dksvd\*w2;

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%%

mean\_f=(mean1+mean2)/2;

sum\_percent=abs(percent\_patch1)+abs(percent\_patch2);

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%为了提高对彩色部分表现能力 现如今采用对彩色部分远远低于均值的时候将所有的彩色部分全部加上

% P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2+mean2;

P1atch\_f=patch\_f\_f1+mean1+(patch\_f\_f2+mean2);

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%如果是彩色区域比重大 则全彩显示

if percent\_patch1<0

P1atch\_f=patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end

%cnt

fprintf([num2str(ii), ' rows have been processed\n']);

if mod(ii,5)==0

disp(['The channel ', num2str(channel),' was working!']);

end

end

%idx是逻辑值 所以知道其如果所在位置逻辑值不为1 则

%不会进行任何操作

idx = (cntMat < 1);

F(idx) = (A(idx)+B(idx))./2;

cntMat(idx) = 1;

F = F./cntMat;

Fusion\_image(:,:,channel)=F;

end

figure;

imshow(uint8(Fusion\_image),[]);

imwrite(uint8(Fusion\_image),'./fusion\_image.jpg');

end

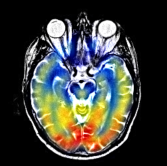
%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%

## 加权程序九

### 思路：怀疑是不是在算P1atch\_f=patch\_f\_f1+mean1+(patch\_f\_f2+mean2);的时候mean2其实算均值方式不合理 没有去除实际点数 而是除的8x8

### 效果部分



### 程序部分

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*I分量为引导 RGB融合实验\*\*\*\*\*\*\*\*\*\*\*\*

%注意点1 如果改变求均值方式的时候 还要改变percent\_patch2地方的求值方式

%注意点2 此处发现个很大的bug 那就是在求percent或者其他时 如果a=0 那么无论是判断大于零还是小于零 都是不会有

% 结果的

% a=0/0;

% % a=1;

% if a>-0.2

% k=3;

% end

% 注意点3 load操作其实很费时间

close all

%本程序是第三种加权方法的融合实验

clc;clear;

test\_ronghe=1;

shangxian=1;

xiaxian=0;

bufen=1;

zhuangzai\_xishu=1;%采用预先训练字典获取系数之后再融合

jiaquan=1;%加权方式选择

if test\_ronghe==1

for ii=1:7

% cd('.\MAT\_DATA');%此处如果报错 则采用将字典文件复制到外面的方法 装载其实很费时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

currentFile = sprintf('Dksvd%d.mat',ii);

load(currentFile);

eval(['Dksvd',num2str(ii),'=','Dksvd',';']);

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%

end

if zhuangzai\_xishu==1%预先装载节省时间

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

for channel=1:3

currentFile = sprintf('%dfenlei\_patch1.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch1',num2str(channel),'=','cun\_fenlei\_patch1',';']);

currentFile = sprintf('%dzidian\_xishu\_patch1.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch1',num2str(channel),'=','cun\_xishu\_patch1',';']);

currentFile = sprintf('%dfenlei\_patch2.mat',channel);

load(currentFile);

eval(['temp\_fenlei\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

currentFile = sprintf('%dzidian\_xishu\_patch2.mat',channel);

load(currentFile);

eval(['temp\_zidian\_patch2',num2str(channel),'=','cun\_fenlei\_patch2',';']);

end

cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

image\_input1=double(imread('./2.jpg'));

image\_input2=double(imread('./3.jpg'));

% image\_input1=imresize(image\_input1,[256 256]);

% image\_input2=imresize(image\_input2,[256 256]);

A=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

B=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

Fusion\_image=zeros(size(image\_input1,1)/bufen,size(image\_input1,2)/bufen,3);

for i=1:3

A(:,:,i)=image\_input1(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input1,2)/bufen,i);

B(:,:,i)=image\_input2(shangxian:size(image\_input1,1)/bufen+xiaxian,shangxian:xiaxian+size(image\_input2,2)/bufen,i);

end

figure;

imshow(uint8(A),[]);

figure;

imshow(uint8(B),[]);

[Ifenliang\_1,FORRI,AA,BB]=RGB2IHS(A);

[Ifenliang\_2,FORRI,AA,BB]=RGB2IHS(B);

I1=Ifenliang\_1(:,:,1);

I2=Ifenliang\_2(:,:,1);

S2=Ifenliang\_2(:,:,3);

[I1\_zero1,I1\_zero2]=find(I1>0);

[I2\_zero1,I2\_zero2]=find(I2>0);

[S2\_zero1,S2\_zero2]=find(I2>0);

%此处改变算均值的方式

mean\_I1=sum(sum(I1))/(length(I1\_zero1));

mean\_I2=sum(sum(I2))/(length(I2\_zero1));

mean\_S2=sum(sum(S2))/(length(S2\_zero1));

% mean\_I1=sum(sum(I1))/(size(I1,1)\*size(I1,2));

% mean\_I2=sum(sum(I2))/(size(I2,1)\*size(I2,2));

dic\_number=6;

r=180/dic\_number;

patch\_size=8;

overlap=7;

A100=A;

B100=B;

for channel=1:3

windows=0;

A=A100(:,:,channel);

B=B100(:,:,channel)

[kuan,chang]=size(B);

F=zeros(kuan,chang);

cntMat=zeros(kuan,chang);

gridx = 1:patch\_size - overlap : kuan-patch\_size+1;

gridy = 1:patch\_size - overlap : chang-patch\_size+1;

dx=[-1 0 1;-2 0 2;-1 0 1];

dy=dx';

for ii = 1:length(gridx)

for jj = 1:length(gridy)

patch1\_is\_zeros=0;

patch2\_is\_zeros=0;

windows=windows+1;

xx = gridx(ii);

yy = gridy(jj);

%注意点：为了避免开头注意点2陷阱 现进行预处理 先判断是不是为0

%%

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch1模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_1 = A(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_1~=0);

if (length(temp\_no\_zero\_hang\_zuobiao))==0

mean1=zeros(64,1);

else

mean1 = sum(sum(patch\_1))/(length(temp\_no\_zero\_hang\_zuobiao));

end

% patch1 = patch\_1(:) - mean1;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I1 = I1(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I1~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A1=(patch\_I1-mean\_I1)/mean\_I1;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch1=sum(sum(A1))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch1=sum(sum(A1))/(size(A1,1)\*size(A1,2));%全部转为正数 但是如果全部为负数

%%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*处理patch2模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_2 = B(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_2~=0);

if (length(temp\_no\_zero\_hang\_zuobiao))==0

mean2=zeros(64,1);

else

mean2 = sum(sum(patch\_2))/(length(temp\_no\_zero\_hang\_zuobiao));

end

% patch2 = patch\_2(:) - mean2;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*求对应patch1突出度模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*

patch\_I2 = I2(yy:yy+patch\_size-1, xx:xx+patch\_size-1);

[temp\_no\_zero\_hang\_zuobiao,temp\_no\_zero\_liebiao]=find(patch\_I2~=0);

% mean\_patch\_I1=sum(sum(patch\_I1))/(size(patch\_I1,1)\*size(patch\_I1,2));

% mean\_b=sum(sum(B))/(size(B,1)\*size(B,2));

A2=(patch\_I2-mean\_I2)/mean\_I2;%和自己比看都大多少

% [k1,k2]=max(max(A1));

percent\_patch2=sum(sum(A2))/(length(temp\_no\_zero\_hang\_zuobiao));%全部转为正数 但是如果全部为负数

% percent\_patch2=sum(sum(A2))/(size(A2,1)\*size(A2,2));%全部转为正数 但是如果全部为负数

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%\*\*\*\*加权方式一配套\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%如果采用方式二 这里就可以注释了

if jiaquan==1 | 2

if zhuangzai\_xishu==0%换系数获取方式 换成通过查找系数的方式获取

[F1,xishu,max\_p1]=USE\_DICTIONARY(patch\_1,0);

w1=xishu;

[F2,xishu,max\_p2]=USE\_DICTIONARY(patch\_2,0);

w2=xishu;

else

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch1',num2str(channel),';']);

w1=cun\_xishu\_patch1(:,windows);

max\_p1=cun\_fenlei\_patch1(windows);

eval(['cun\_fenlei\_patch1','=','temp\_fenlei\_patch2',num2str(channel),';']);

w2=cun\_xishu\_patch1(:,windows);

max\_p2=cun\_fenlei\_patch1(windows);

end

end

%%

%乘上字典模块 为了减少运算时间 如果为零的话 就不用那么麻烦了 直接赋值就行

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

if sum(w1)==0

patch\_f\_f1=zeros(64,1);

else

% currentFile = sprintf('Dksvd%d.mat',max\_p1);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p1),';']);

patch\_f\_f1=Dksvd\*w1;

end

if sum(w2)==0

patch\_f\_f2=zeros(64,1);

else

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS\MAT\_DATA');

% currentFile = sprintf('Dksvd%d.mat',max\_p2);

% load(currentFile);

eval(['Dksvd','=','Dksvd',num2str(max\_p2),';']);

patch\_f\_f2=Dksvd\*w2;

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

end

% cd('C:\E-DATA-GROUNP\github\02-research\06 practice-myself\_codes\_notes\03 SCI\_CODE\_PAPER\_FUSION\01 code\02\_proposed\_method\05 TEST-IHS');

%%

mean\_f=(mean1+mean2)/2;

sum\_percent=abs(percent\_patch1)+abs(percent\_patch2);

if percent\_patch1 > percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%为了提高对彩色部分表现能力 现如今采用对彩色部分远远低于均值的时候将所有的彩色部分全部加上

% P1atch\_f=patch\_f\_f1+mean1+patch\_f\_f2+mean2;

P1atch\_f=patch\_f\_f1+mean1+(patch\_f\_f2+mean2);

end

if percent\_patch1 < percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

%如果是彩色区域比重大 则全彩显示

if percent\_patch1<0

P1atch\_f=patch\_f\_f2+mean2;

else

P1atch\_f=patch\_f\_f2+mean2+patch\_f\_f1\*(1-(temp\_max/sum\_percent));

end

end

if percent\_patch1 == percent\_patch2

temp\_max=max(abs(percent\_patch1),abs(percent\_patch2));

P1atch\_f=(patch\_f\_f2+mean2)\*(temp\_max/sum\_percent)+(patch\_f\_f1+mean1)\*(1-(temp\_max/sum\_percent));

end

Patch\_f= reshape(P1atch\_f, [patch\_size, patch\_size]);

F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = F(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + Patch\_f;

cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) = cntMat(yy:yy+patch\_size-1, xx:xx+patch\_size-1) + 1;

end

%cnt

fprintf([num2str(ii), ' rows have been processed\n']);

if mod(ii,5)==0

disp(['The channel ', num2str(channel),' was working!']);

end

end

%idx是逻辑值 所以知道其如果所在位置逻辑值不为1 则

%不会进行任何操作

idx = (cntMat < 1);

F(idx) = (A(idx)+B(idx))./2;

cntMat(idx) = 1;

F = F./cntMat;

Fusion\_image(:,:,channel)=F;

end

figure;

imshow(uint8(Fusion\_image),[]);

imwrite(uint8(Fusion\_image),'./fusion\_image.jpg');

end

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%

## 加权程序十

思路：

效果部分

程序部分

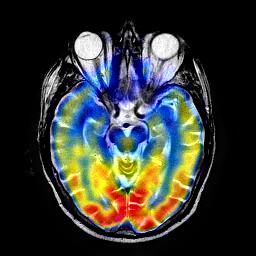
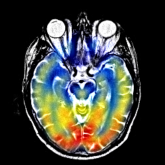
加权程序九

思路：

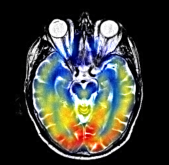
效果部分

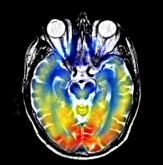
程序部分

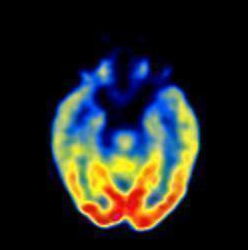
# 融合加权效果研究区

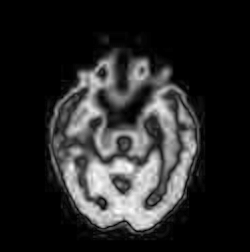












# 改进点

## 1 字典训练可以不用每次都是DCT重新生成 可以用训练好的字典代入

## 2 PET图像放大效果不适用imresize

## 3 A=A(32:size(A,1)/2+32,32:size(A,2)/2+32);为了减少运算量梯度分类选取了部分

## 4 currentFile = sprintf('fenlei\_data%d.mat',i); load(currentFile);不宜放在函数中被调用

## 5 对源数据求最大无关组 分别获取稀疏表示系数 之后再进行解坐标向量方式 加权稀疏系数 这样就可以获取系数 减少时间

## 6 变量命名一定要有秩序 不然就是要走很多回头路