*'''  
2021.1.28FromIvicxDS:openCV;E1图像的基本操作  
'''***'''  
图像的矩阵化  
1.图像的读取  
2.图像颜色矩阵的输出  
3.图像的显示  
4.图像的颜色通道拆分与输出  
5.图像颜色通道的合并  
6.彩图改为灰度图  
'''**import cv2*#读取格式为BGR  
#图像的读取*def ShowPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/P1.jpg"**)  
print(**"<图片的像素值>"**)  
print(img)  
  
print(**"图片的hwc,3表示RGB彩色图"**)  
print(img.shape)  
  
ShowPicture(**"this is a window"**,img)*#显示图像  
  
#读取为灰度图像*img2=cv2.imread(**"image/P1.jpg"**,cv2.IMREAD\_GRAYSCALE)  
  
print(**"<图片的像素值>"**)  
print(img2)  
  
print(**"图片的hw,2表示灰度图"**)  
print(img2.shape)  
  
ShowPicture(**"this is a no-colored picture"**,img2)  
  
*#图像的保存*cv2.imwrite(**"P3.jpg"**,img2)  
  
*#读取图像大小*print(**"<图像的大小>"**)  
print(img2.size)  
  
*#截取部分图像ROI*imgshow=img2[0:200,0:200]  
ShowPicture(**"cutting picture"**,imgshow)  
  
*#颜色通道的提取*b,g,r=cv2.split(img)  
print(**"<颜色通道提取>"**)  
print(**"<B>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**)  
print(b)  
print(**"B.shape="**)  
print(b.shape)  
print(**"<G>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**)  
print(g)  
print(**"G.shape="**)  
print(g.shape)  
print(**"<R>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**)  
print(r)  
print(**"R.shape="**)  
print(r.shape)  
print(**">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"**)  
  
*#从颜色通道合成图像*img3=cv2.merge((b,g,r))  
ShowPicture(**"an merged picture"**,img3)  
print(**"合成后图像的shape值="**)  
print(img3.shape)  
  
*#只保留R通道  
#BGR->[,,]三维数组切片->0，1，2*R\_image=img3.copy()  
R\_image[:,:,0]=0  
R\_image[:,:,1]=0  
ShowPicture(**"R"**,R\_image)

[]

<图片的像素值>

[[[242 244 245]

[242 244 245]

[243 245 246]

...

[ 33 31 37]

[ 15 13 19]

[ 16 14 20]]

[[243 245 246]

[243 245 246]

[243 245 246]

...

[ 33 31 37]

[ 12 10 16]

[ 13 11 17]]

[[241 242 246]

[241 242 246]

[241 242 246]

...

[ 34 32 38]

[ 10 8 14]

[ 11 9 15]]

...

[[118 123 138]

[117 125 138]

[119 124 139]

...

[ 79 81 135]

[ 81 82 138]

[ 82 83 139]]

[[133 140 157]

[131 140 154]

[132 139 156]

...

[ 79 78 134]

[ 81 79 138]

[ 81 79 138]]

[[139 147 164]

[139 147 164]

[139 147 164]

...

[ 77 75 134]

[ 80 78 137]

[ 80 78 138]]]

图片的hwc,3表示RGB彩色图

(448, 493, 3)

<图片的像素值>

[[244 244 245 ... 33 15 16]

[245 245 245 ... 33 12 13]

[243 243 243 ... 34 10 11]

...

[127 128 128 ... 97 99 100]

[144 143 143 ... 95 97 97]

[151 151 151 ... 93 96 96]]

图片的hw,2表示灰度图

(448, 493)

<图像的大小>

220864

<颜色通道提取>

<B>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[242 242 243 ... 33 15 16]

[243 243 243 ... 33 12 13]

[241 241 241 ... 34 10 11]

...

[118 117 119 ... 79 81 82]

[133 131 132 ... 79 81 81]

[139 139 139 ... 77 80 80]]

B.shape=

(448, 493)

<G>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[244 244 245 ... 31 13 14]

[245 245 245 ... 31 10 11]

[242 242 242 ... 32 8 9]

...

[123 125 124 ... 81 82 83]

[140 140 139 ... 78 79 79]

[147 147 147 ... 75 78 78]]

G.shape=

(448, 493)

<R>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[245 245 246 ... 37 19 20]

[246 246 246 ... 37 16 17]

[246 246 246 ... 38 14 15]

...

[138 138 139 ... 135 138 139]

[157 154 156 ... 134 138 138]

[164 164 164 ... 134 137 138]]

R.shape=

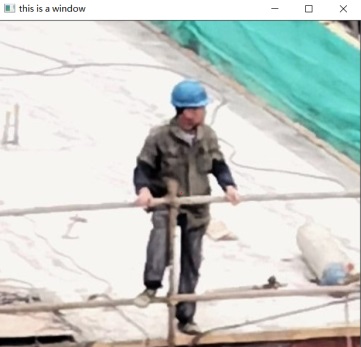
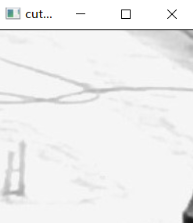
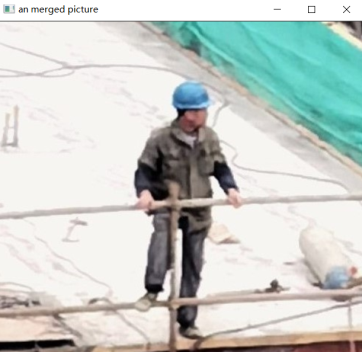
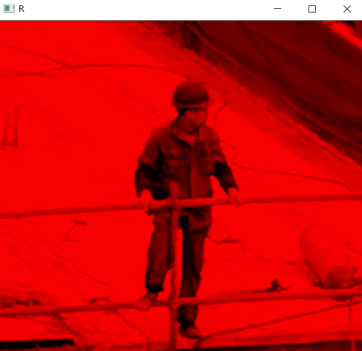
(448, 493)

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

合成后图像的shape值=

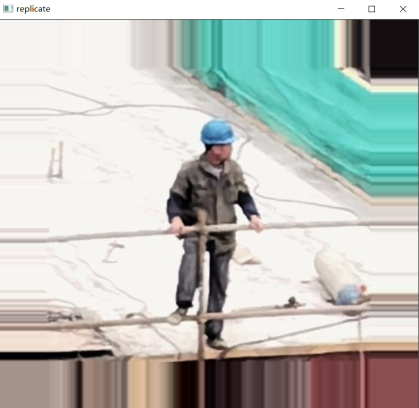
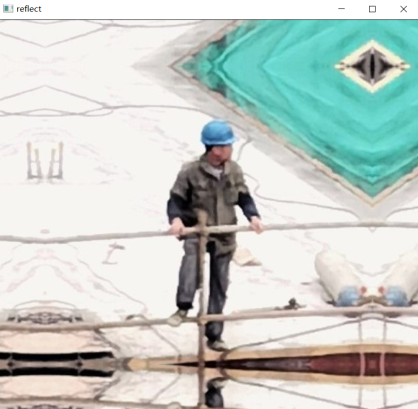
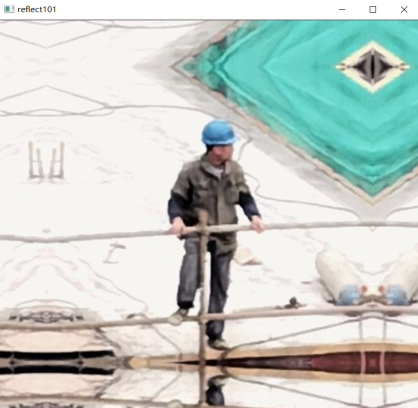
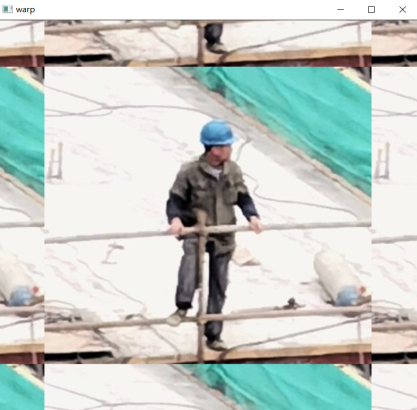
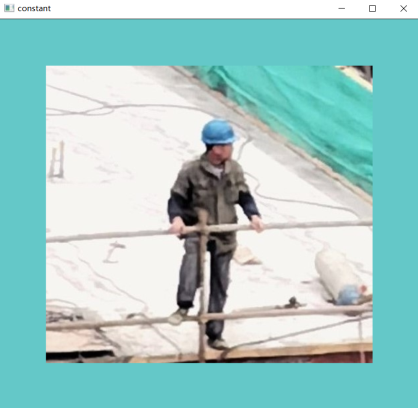
(448, 493, 3)

Process finished with exit code 0



*'''  
2021.1.28FromIvicxDS:openCV;E1图像的基本操作  
'''***'''  
1.图像加边框  
'''**import cv2*#读取格式为BGR*def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/P1.jpg"**)  
  
*#给图像加边框*top\_size,bottom\_size,left\_size,right\_size=(70,70,70,70)  
  
replicate=cv2.copyMakeBorder(img,top\_size,bottom\_size,left\_size,right\_size,  
 borderType=cv2.BORDER\_REPLICATE)  
showPicture(**"replicate"**,replicate)*#复制边缘像素*reflect=cv2.copyMakeBorder(img,top\_size,bottom\_size,left\_size,right\_size,  
 borderType=cv2.BORDER\_REFLECT)  
showPicture(**"reflect"**,reflect)*#镜面*reflect101=cv2.copyMakeBorder(img,top\_size,bottom\_size,left\_size,right\_size,  
 borderType=cv2.BORDER\_REFLECT\_101)  
showPicture(**"reflect101"**,reflect101)  
  
warp=cv2.copyMakeBorder(img,top\_size,bottom\_size,left\_size,right\_size,  
 borderType=cv2.BORDER\_WRAP)  
showPicture(**"warp"**,warp)  
  
constant=cv2.copyMakeBorder(img,top\_size,bottom\_size,left\_size,right\_size,  
 borderType=cv2.BORDER\_CONSTANT,value=(200,200,100))  
showPicture(**"constant"**,constant)

[]



*'''  
2021.1.28FromIvicxDS:openCV;E1图像的基本操作  
'''***'''  
1.图像运算  
2.改变图像大小  
3.图像融合  
'''**import cv2*#读取格式为BGR*def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/P3.jpg"**)  
img2=img+10  
  
print(img)  
print(**"\*\*\*\*\*\*\*\*\*\*"**)  
print(img2)  
print(**">>>>>>>>>>>>>>>>>>>>"**)  
  
*#>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  
#运算叠加两张图片与。added的区别（封顶）*img3=cv2.imread(**"image/add1.jpg"**)  
img3=cv2.resize(img3,(300,300))  
img4=cv2.imread(**"image/add2.jpg"**)*#resize强制放缩*img4=cv2.resize(img4,(300,300))  
print(img3)  
print(**"\*\*\*\*\*\*\*\*\*\*"**)  
print(img4)  
print(**">>>>>>>>>>>>>>>>>>>>"**)  
  
imgadded1=img3+img4  
print(imgadded1)  
showPicture(**"image add"**,imgadded1)  
print(**">>>>>>>>>>>>>>>>>>>>"**)  
  
imgadded2=cv2.add(img3,img4)  
print(imgadded2)  
showPicture(**"image add2"**,imgadded2)  
  
imagemesh=cv2.addWeighted(img3,0.4,img4,0.6,0)*#图像按照权重融合*showPicture(**"image mesh"**,imagemesh)*#R=a\*x1+b\*x2+c  
  
#图像按比例缩放*img5=cv2.imread(**"image/P1.jpg"**)  
img5=cv2.resize(img5,(0,0),fx=2,fy=0.3)  
showPicture(**"image process"**,img5)

[]

[[[244 244 244]

[244 244 244]

[245 245 245]

...

[ 33 33 33]

[ 15 15 15]

[ 17 17 17]]

[[245 245 245]

[245 245 245]

[245 245 245]

...

[ 33 33 33]

[ 12 12 12]

[ 14 14 14]]

[[243 243 243]

[243 243 243]

[243 243 243]

...

[ 34 34 34]

[ 10 10 10]

[ 11 11 11]]

...

[[127 127 127]

[128 128 128]

[129 129 129]

...

[ 97 97 97]

[100 100 100]

[100 100 100]]

[[144 144 144]

[144 144 144]

[143 143 143]

...

[ 95 95 95]

[ 97 97 97]

[ 97 97 97]]

[[151 151 151]

[151 151 151]

[150 150 150]

...

[ 94 94 94]

[ 97 97 97]

[ 97 97 97]]]

\*\*\*\*\*\*\*\*\*\*

[[[254 254 254]

[254 254 254]

[255 255 255]

...

[ 43 43 43]

[ 25 25 25]

[ 27 27 27]]

[[255 255 255]

[255 255 255]

[255 255 255]

...

[ 43 43 43]

[ 22 22 22]

[ 24 24 24]]

[[253 253 253]

[253 253 253]

[253 253 253]

...

[ 44 44 44]

[ 20 20 20]

[ 21 21 21]]

...

[[137 137 137]

[138 138 138]

[139 139 139]

...

[107 107 107]

[110 110 110]

[110 110 110]]

[[154 154 154]

[154 154 154]

[153 153 153]

...

[105 105 105]

[107 107 107]

[107 107 107]]

[[161 161 161]

[161 161 161]

[160 160 160]

...

[104 104 104]

[107 107 107]

[107 107 107]]]

>>>>>>>>>>>>>>>>>>>>

[[[180 134 3]

[171 126 3]

[162 123 16]

...

[176 132 17]

[180 130 19]

[182 133 18]]

[[178 127 3]

[178 131 3]

[179 128 4]

...

[164 121 43]

[166 122 41]

[183 143 61]]

[[249 221 157]

[242 201 115]

[206 168 82]

...

[ 37 27 13]

[ 38 25 13]

[ 36 27 15]]

...

[[169 140 74]

[171 142 77]

[176 146 87]

...

[167 125 21]

[161 124 34]

[157 130 60]]

[[225 214 182]

[230 222 199]

[232 226 221]

...

[183 142 19]

[185 144 22]

[183 143 25]]

[[215 218 216]

[220 221 219]

[218 220 220]

...

[174 134 41]

[180 140 41]

[185 141 32]]]

\*\*\*\*\*\*\*\*\*\*

[[[ 11 11 11]

[ 13 13 13]

[ 13 13 13]

...

[156 247 238]

[135 246 232]

[ 73 239 218]]

[[ 14 14 14]

[ 13 13 13]

[ 13 13 13]

...

[173 248 242]

[155 251 240]

[159 245 236]]

[[ 10 10 10]

[ 10 10 10]

[ 12 12 12]

...

[173 249 242]

[165 250 240]

[163 248 239]]

...

[[ 4 2 2]

[ 3 14 13]

[ 0 6 5]

...

[ 4 153 128]

[ 32 34 34]

[ 36 36 36]]

[[ 3 43 36]

[ 4 68 55]

[ 3 29 23]

...

[ 3 163 128]

[ 33 36 36]

[ 38 38 38]]

[[ 0 17 14]

[ 5 14 13]

[ 4 7 7]

...

[ 2 152 114]

[ 26 32 31]

[ 37 37 37]]]

>>>>>>>>>>>>>>>>>>>>

[[[191 145 14]

[184 139 16]

[175 136 29]

...

[ 76 123 255]

[ 59 120 251]

[255 116 236]]

[[192 141 17]

[191 144 16]

[192 141 17]

...

[ 81 113 29]

[ 65 117 25]

[ 86 132 41]]

[[ 3 231 167]

[252 211 125]

[218 180 94]

...

[210 20 255]

[203 19 253]

[199 19 254]]

...

[[173 142 76]

[174 156 90]

[176 152 92]

...

[171 22 149]

[193 158 68]

[193 166 96]]

[[228 1 218]

[234 34 254]

[235 255 244]

...

[186 49 147]

[218 180 58]

[221 181 63]]

[[215 235 230]

[225 235 232]

[222 227 227]

...

[176 30 155]

[206 172 72]

[222 178 69]]]

>>>>>>>>>>>>>>>>>>>>

[[[191 145 14]

[184 139 16]

[175 136 29]

...

[255 255 255]

[255 255 251]

[255 255 236]]

[[192 141 17]

[191 144 16]

[192 141 17]

...

[255 255 255]

[255 255 255]

[255 255 255]]

[[255 231 167]

[252 211 125]

[218 180 94]

...

[210 255 255]

[203 255 253]

[199 255 254]]

...

[[173 142 76]

[174 156 90]

[176 152 92]

...

[171 255 149]

[193 158 68]

[193 166 96]]

[[228 255 218]

[234 255 254]

[235 255 244]

...

[186 255 147]

[218 180 58]

[221 181 63]]

[[215 235 230]

[225 235 232]

[222 227 227]

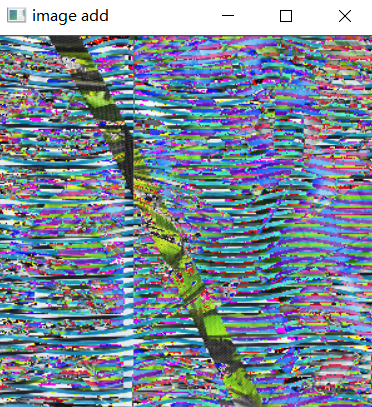
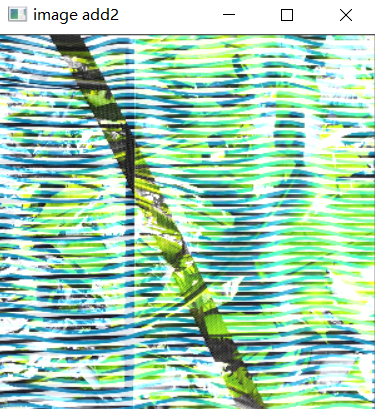
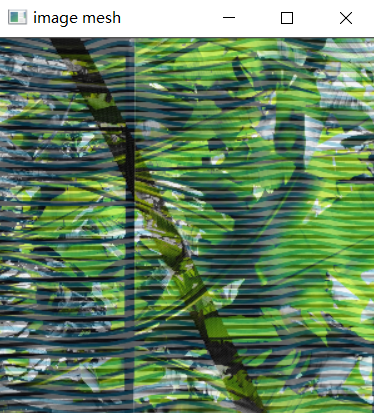
...

[176 255 155]

[206 172 72]

[222 178 69]]]

Process finished with exit code 0



*'''  
2021.1.28FromIvicxDS:openCV;E1视频的读取  
'''***'''  
2.视频的读取  
'''**import cv2*#读取格式为BGR*vc=cv2.VideoCapture(**"video/1.mp4"**)  
  
*#检查打开是否正常*if vc.isOpened():  
 open,frame=vc.read()*#执行vc.read()往后读一帧*else:  
 open=False  
  
*#遍历所有帧*while True:  
 ret,frame=vc.read()  
 if frame is None:*#为空（播放完了）* break  
 if ret==True:  
 gray=cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)*#转成灰度图* cv2.imshow(**"result"**,gray)  
 if cv2.waitKey(10) & 0xFF==27:*#设置帧率，27表示按Esc退出* break  
vc.release()  
cv2.destroyAllWindows()