印花布匹瑕点分析

Haowen Xu,collabrated with Yijie Yao,姚奕捷, and Yingyi Liang, 梁樱议.
18342016002 1/6/2020

Project Description

印花布匹表面缺陷的识别有助于分析产生原因，数据集内印花布匹瑕疵被划分为15类，瑕疵以成对图片（正确的图样以及问题图样）给出，并注明了瑕疵的参考位置及类别。上述瑕疵图、模版图及瑕疵参考位置均可作为已知信息用于瑕疵类型的判别。 数据中划分的类别如下： 0: 未知, 1: 逃花, 2: 塞网, 3: 破洞, 4: 缝头, 5: 水渍, 6: 脏污, 7: 白条, 8: 花糊, 9: 坯疵, 10: 沙眼, 11: 拖色, 12: 网折印, 13: 无疵点, 14: 未对齐

Task Description

取1、2、5及13共四类数据，按4类作分类

Data Processing

In [1]:
```python
import json
import glob, os
from matplotlib import image
import numpy as np

def load_fabric_data(path):
    '''
    Loads data from fabric_data folder.
    Returns:
        (list, list)
        A list of id in the file name
        A list of dictionary file containing data from json (flaw_type, bbox)
    Sample usage: fid, fdata = load_fabric_data('fabric_data/label_json/**/**.json')
    '''
    fid = []
    fdata = []

    for filename in glob.iglob(path, recursive=True):
        #print("filename in load_fabric_data is ")
       # print(filename)
        filename = filename.replace('\\' ,'/')

        fid.append(filename.split('/')[-1].split('.')[0])
        with open(filename) as f:
            fdata.append(json.load(f))
    return (fid, fdata)

def extract_label_grouping(fdata):
    '''
    Generates lists of labels according to different groupings.
    Type 1 grouping: original
    Type 2 grouping: 6-12 as group 6, 13 as group 7, 14 as group 8
    Type 3 grouping: only take 1,2,5 and 13
    '''
    ftype1 = [] #original
    ftype2 = [] #6-12 as group 6, 13 as group 7, 14 as group 8
    ftype2_dict = {num:6 for num in range(6, 13)}
    ftype2_dict.update({num:num for num in range(6)})
    ftype2_dict[13] = 7
    ftype2_dict[14] = 8
    for i in fdata:
        ftype1.append(i['flaw_type'])
        ftype2.append(ftype2_dict[i['flaw_type']])
    return ftype1, ftype2

def load_fabric_images(path, fids, fdata, ftype):
    path += '**/**.jpg'
    labels = []
    imgs = []

    print(path)
    #random.sample(list(glob.iglob(path, recursive=True)), 50)
    for filename in glob.iglob(path, recursive=True):
        #find info about the image
        filename = filename.replace('\\' ,'/')


        fid = filename.split('/')[-1].split('.')[0]

        info = fdata[fids.index(fid)]

        filename_trgt = filename.replace("temp", "trgt")
        #filename_trgt = r"C:/Users/Administrator/Desktop/PRML/Project/fabric_data/trgt" + "/" +fid +'.
jpg'
        #get image
        size1 = os.stat(filename).st_size
        size2 = os.stat(filename_trgt).st_size
        if (size1 != 0) and (size2 != 0):
            #load image
            img_data_temp = image.imread(filename)
            img_data_temp = img_data_temp[info['bbox']['y0']:info['bbox']['y1'], info['bbox']['x0']:inf
o['bbox']['x1']]
            img_data_tgrt = image.imread(filename_trgt)
            img_data_tgrt = img_data_tgrt[info['bbox']['y0']:info['bbox']['y1'], info['bbox']['x0']:inf
o['bbox']['x1']]
            if img_data_temp.shape == img_data_tgrt.shape:
                #append image
                imgs.append(np.concatenate([img_data_temp, img_data_tgrt], axis = 2))
                #append label
                labels.append(ftype[fids.index(fid)])
    return (labels, imgs)
```

In [2]:
```python
path = r"C:/Users/Administrator/Desktop/PRML/Project/fabric_data/label_json/**/**.json"

fids, fdata = load_fabric_data(path)
ftype1, ftype2 = extract_label_grouping(fdata)
```

In [4]:
```python
path = r"C:/Users/Administrator/Desktop/PRML/Project/fabric_data/temp/"
labels, imgs = load_fabric_images(path, fids, fdata, ftype1)
```

```
C:/Users/Administrator/Desktop/PRML/Project/fabric_data/temp/**/**.jpg
```

In [5]:
```python
print(len(labels))
print(imgs[0])
```

```
3371
[[[130 130 128 155 147 171]
  [139 137 138 151 150 168]
  [147 145 148 146 154 167]
  ...
  [ 57  46  24  92  87  29]
  [ 73  51  28  83  64  34]
  [124  89  69  76  47  51]]

 [[139 141 140 162 153 174]
  [143 144 146 157 155 168]
  [148 148 150 151 155 166]
  ...
  [ 69  61  38 121 124  69]
  [ 58  41  15 113 100  56]
  [ 69  39  15  88  63  33]]

 [[143 147 150 160 149 166]
  [145 149 152 156 153 162]
  [150 151 156 153 156 161]
  ...
  [102 100  75 153 160 116]
  [ 90  79  49 140 133  79]
  [ 75  55  22 126 106  47]]

 ...

 [[ 56  19  13  76  21  24]
  [ 55  16  11  77  28  31]
  [ 57  18  13  70  31  32]
  ...
  [208 213 219 210 213 206]
  [208 218 220 222 223 228]
  [200 210 211 216 213 234]]

 [[ 74  35  28 103  31  34]
  [ 69  30  23 103  37  39]
  [ 72  31  25  95  39  40]
  ...
  [191 200 209 212 212 212]
  [220 231 237 209 209 221]
  [196 210 213 213 210 237]]

 [[ 87  49  40 122  36  37]
  [ 82  42  34 123  41  43]
  [ 85  45  37 115  43  46]
  ...
  [185 193 204 214 212 223]
  [195 208 216 215 212 233]
  [193 208 215 217 216 248]]]
```

In [6]:
```python
n_samples = len(imgs)
print("Number of samples:", n_samples)
```

```
Number of samples: 3371
```

In [7]:
```python
print(imgs[1230].shape)
```

```
(400, 400, 6)
```

In [8]:
```python
# another way of selecting categories
from numpy import array

myIndices = []
def find_indices_four_cat(listOfLabel):
    res = []
    i = 0
    for i in range(0,len(listOfLabel)):
        if (listOfLabel[i] == 1) or (listOfLabel[i]== 2) or (listOfLabel[i] == 5) or (listOfLabel[i] ==
 13):
            res.append(i)
    return res

label_four_cat_indices = find_indices_four_cat(labels)

# select the fours category out of the sample
labels_four_cat = array(labels)[label_four_cat_indices]
samples_four_cat = array(imgs)[label_four_cat_indices]
```

In [9]:
```python
labels = labels_four_cat
imgs = samples_four_cat

print(len(imgs))
```

```
1123
```

Padding

- https://blog.csdn.net/wuzqChom/article/details/74785643
- https://stackoverflow.com/questions/47697622/cnn-image-resizing-vs-padding-keeping-aspect-ratio-or-not/49882055#49882055
- https://stackoverflow.com/questions/43391205/add-padding-to-images-to-get-them-into-the-same-shape

In [10]:
```python
import cv2
```

In [11]:
```python
imgs = [cv2.resize(img,(200, 200)) for img in imgs]
```

Split data

In [12]:
```python
from sklearn.model_selection import train_test_split
```

In [13]:
```python
train_images, test_images, train_labels, test_labels = train_test_split(imgs, labels, test_size=0.2, ra
ndom_state=1)
```

Normalize data

In [14]:
```python
#first split, split to the training and testing (we will futher split training later because we need va
lidation set )
train_images, test_images, train_labels, test_labels = np.array(train_images), np.array(test_images), n
p.array(train_labels), np.array(test_labels)
train_images, test_images = train_images / 255.0, test_images / 255.0
```

In [15]:
```python
train_images.shape
print(len(test_labels))
# note all the test labels are real images at this point. We are only going to generate fake data for t
he training set
```

```
225
```

In [16]:
```python
# subset all the 5s
def find_indices_5(listOfLabel):
    res = []
    i = 0
    for i in range(0,len(listOfLabel)):
        if (listOfLabel[i] == 5):
            res.append(i)
    return res

label_five = find_indices_5(train_labels)
print(len(label_five))

train_is_five = array(train_images)[label_five]

# the is the training data that has labelled as "5"
print(train_is_five)
```

```
13
[[[[0.14117647 0.13333333 0.13333333 0.23529412 0.18431373 0.15686275]
   [0.21960784 0.23529412 0.19607843 0.27843137 0.37647059 0.31372549]
   [0.41568627 0.45490196 0.36078431 0.56862745 0.61176471 0.47058824]
   ...
   [0.5372549  0.52156863 0.38039216 0.60392157 0.58039216 0.44705882]
   [0.56470588 0.5372549  0.38823529 0.60392157 0.58039216 0.43137255]
   [0.58431373 0.54901961 0.39607843 0.61960784 0.58823529 0.42745098]]

  [[0.14117647 0.1372549  0.14901961 0.18823529 0.18039216 0.16862745]
   [0.3254902  0.33333333 0.30588235 0.42352941 0.49803922 0.45098039]
   [0.48235294 0.49019608 0.41960784 0.60784314 0.64313725 0.5254902 ]
   ...
   [0.53333333 0.51372549 0.39215686 0.58823529 0.56078431 0.43921569]
   [0.56470588 0.53333333 0.40784314 0.61176471 0.58039216 0.44313725]
   [0.56862745 0.5254902  0.4        0.63529412 0.59607843 0.45490196]]

  [[0.12941176 0.15294118 0.13333333 0.21960784 0.24313725 0.19607843]
   [0.42352941 0.42352941 0.37254902 0.52941176 0.5372549  0.45098039]
   [0.52941176 0.50980392 0.40784314 0.63921569 0.64313725 0.49803922]
   ...
   [0.54901961 0.53333333 0.4        0.59607843 0.57254902 0.44705882]
   [0.57254902 0.54509804 0.40784314 0.61176471 0.58039216 0.43921569]
   [0.57254902 0.53333333 0.39215686 0.62745098 0.58823529 0.44705882]]

  ...

  [[0.09019608 0.09803922 0.10980392 0.12941176 0.1254902  0.10196078]
   [0.10196078 0.09803922 0.12156863 0.11764706 0.12941176 0.10588235]
   [0.10196078 0.09411765 0.12156863 0.12156863 0.13333333 0.10196078]
   ...
   [0.07843137 0.07843137 0.07843137 0.1254902  0.11764706 0.11764706]
   [0.08627451 0.08627451 0.07843137 0.11372549 0.10980392 0.10588235]
   [0.10196078 0.10196078 0.09411765 0.1254902  0.10980392 0.10980392]]

  [[0.07843137 0.09411765 0.10196078 0.11764706 0.14509804 0.10980392]
   [0.09411765 0.09019608 0.10980392 0.11764706 0.14117647 0.1254902 ]
   [0.11372549 0.09803922 0.13333333 0.10980392 0.1254902  0.11764706]
   ...
   [0.08627451 0.08235294 0.0745098  0.12156863 0.11764706 0.11372549]
   [0.08235294 0.08235294 0.0745098  0.11764706 0.10980392 0.10980392]
   [0.09019608 0.08627451 0.07843137 0.13333333 0.11764706 0.10980392]]

  [[0.09411765 0.09411765 0.06666667 0.1254902  0.12156863 0.13333333]
   [0.10980392 0.09411765 0.07843137 0.1254902  0.11764706 0.1372549 ]
   [0.10980392 0.10196078 0.10980392 0.12941176 0.12156863 0.13333333]
   ...
   [0.09019608 0.08627451 0.07843137 0.12941176 0.1254902  0.10196078]
   [0.09019608 0.08627451 0.07843137 0.11764706 0.11372549 0.08627451]
   [0.09411765 0.09019608 0.0745098  0.13333333 0.12156863 0.09803922]]]


 [[[0.08627451 0.05490196 0.10196078 0.09019608 0.09019608 0.11764706]
   [0.09411765 0.07843137 0.09411765 0.11764706 0.05882353 0.1254902 ]
   [0.14509804 0.1372549  0.14117647 0.2627451  0.21960784 0.21960784]
   ...
   [0.07058824 0.05490196 0.09411765 0.16470588 0.09019608 0.17647059]
   [0.11764706 0.08627451 0.17254902 0.15294118 0.07843137 0.27843137]
   [0.11764706 0.08235294 0.21568627 0.1372549  0.10196078 0.28235294]]

  [[0.07843137 0.04705882 0.07843137 0.08627451 0.08235294 0.10980392]
   [0.08235294 0.05882353 0.0745098  0.0745098  0.08627451 0.10980392]
   [0.09019608 0.07058824 0.0745098  0.11764706 0.09019608 0.10588235]
   ...
   [0.08235294 0.06666667 0.08627451 0.11372549 0.09803922 0.13333333]
   [0.11372549 0.07843137 0.1254902  0.1372549  0.10588235 0.23137255]
   [0.12941176 0.08627451 0.18431373 0.12156863 0.09803922 0.2745098 ]]

  [[0.0627451  0.03921569 0.05490196 0.12941176 0.08627451 0.09019608]
   [0.07843137 0.05098039 0.07058824 0.10980392 0.10588235 0.14117647]
   [0.10196078 0.0745098  0.08627451 0.08627451 0.0745098  0.09019608]
   ...
   [0.07843137 0.06666667 0.08235294 0.10980392 0.12941176 0.13333333]
   [0.10588235 0.07058824 0.09411765 0.1254902  0.08235294 0.13333333]
   [0.1372549  0.08627451 0.16470588 0.16862745 0.11372549 0.25098039]]

  ...

  [[0.10980392 0.10588235 0.09411765 0.14117647 0.11764706 0.14901961]
   [0.10196078 0.10980392 0.09803922 0.16470588 0.13333333 0.14901961]
   [0.12156863 0.13333333 0.12941176 0.11764706 0.1254902  0.1254902 ]
   ...
   [0.59607843 0.60392157 0.43137255 0.65882353 0.61176471 0.50196078]
   [0.60392157 0.60392157 0.41960784 0.71764706 0.6627451  0.57254902]
   [0.58039216 0.61176471 0.41568627 0.75294118 0.66666667 0.57254902]]

  [[0.10196078 0.09803922 0.09803922 0.1254902  0.12941176 0.15686275]
   [0.10980392 0.10980392 0.09019608 0.14117647 0.14117647 0.14901961]
   [0.09411765 0.10196078 0.09019608 0.11764706 0.13333333 0.14901961]
   ...
   [0.54509804 0.54509804 0.40392157 0.68235294 0.65098039 0.56078431]
   [0.60392157 0.60392157 0.44313725 0.72156863 0.68235294 0.59215686]
   [0.57647059 0.61176471 0.43137255 0.76078431 0.69803922 0.59215686]]

  [[0.11372549 0.10980392 0.10196078 0.12941176 0.12941176 0.15294118]
   [0.11372549 0.10980392 0.08235294 0.1254902  0.13333333 0.11764706]
   [0.11372549 0.11764706 0.09411765 0.12156863 0.13333333 0.12156863]
   ...
   [0.30196078 0.29803922 0.18039216 0.60784314 0.54901961 0.46666667]
   [0.56470588 0.55686275 0.41568627 0.70196078 0.65882353 0.57254902]
   [0.54117647 0.57647059 0.40784314 0.74117647 0.68627451 0.58039216]]]


 [[[0.0627451  0.05490196 0.05490196 0.0627451  0.04705882 0.05882353]
   [0.14509804 0.10196078 0.11764706 0.25490196 0.15686275 0.21176471]
   [0.38431373 0.38431373 0.39215686 0.42352941 0.44313725 0.46666667]
   ...
   [0.03921569 0.03137255 0.04313725 0.07058824 0.05098039 0.05490196]
   [0.0627451  0.04705882 0.0627451  0.05490196 0.03529412 0.03921569]
   [0.0627451  0.04705882 0.0627451  0.0627451  0.05882353 0.0627451 ]]

  [[0.07058824 0.03137255 0.05098039 0.0627451  0.07058824 0.07058824]
   [0.23529412 0.17254902 0.20392157 0.3372549  0.21176471 0.2745098 ]
   [0.55686275 0.5372549  0.56470588 0.59607843 0.56470588 0.61176471]
   ...
   [0.04705882 0.03921569 0.05098039 0.07058824 0.04705882 0.05490196]
   [0.06666667 0.04705882 0.0627451  0.07843137 0.05490196 0.0627451 ]
   [0.05882353 0.03921569 0.05490196 0.05098039 0.04705882 0.05098039]]

  [[0.07843137 0.02352941 0.05098039 0.06666667 0.04705882 0.05882353]
   [0.24705882 0.16470588 0.21176471 0.35294118 0.21960784 0.28627451]
   [0.54901961 0.51372549 0.56470588 0.61568627 0.57647059 0.63529412]
   ...
   [0.05490196 0.03529412 0.05098039 0.05490196 0.02352941 0.03137255]
   [0.07058824 0.05098039 0.06666667 0.07058824 0.04313725 0.05490196]
   [0.06666667 0.04705882 0.0627451  0.06666667 0.05490196 0.05882353]]

  ...

  [[0.08627451 0.05490196 0.05098039 0.04313725 0.03921569 0.03137255]
   [0.06666667 0.05098039 0.05098039 0.0627451  0.03137255 0.03529412]
   [0.07843137 0.05490196 0.07058824 0.11372549 0.06666667 0.08235294]
   ...
   [0.10980392 0.02352941 0.02352941 0.0745098  0.05490196 0.04313725]
   [0.11372549 0.06666667 0.07058824 0.07843137 0.05490196 0.0627451 ]
   [0.05490196 0.04313725 0.05098039 0.05490196 0.04705882 0.04313725]]

  [[0.0745098  0.03921569 0.03137255 0.04705882 0.03921569 0.04313725]
   [0.08627451 0.07058824 0.07058824 0.08627451 0.05490196 0.06666667]
   [0.14117647 0.11764706 0.13333333 0.18823529 0.14117647 0.15686275]
   ...
   [0.04705882 0.02352941 0.02352941 0.07058824 0.0627451  0.08235294]
   [0.06666667 0.04705882 0.05882353 0.05882353 0.05490196 0.0627451 ]
   [0.06666667 0.05490196 0.0745098  0.0745098  0.04313725 0.04313725]]

  [[0.08235294 0.04705882 0.04313725 0.06666667 0.03921569 0.05882353]
   [0.0745098  0.05882353 0.05882353 0.07058824 0.05490196 0.07058824]
   [0.18823529 0.16470588 0.18039216 0.10196078 0.07058824 0.08627451]
   ...
   [0.04313725 0.03921569 0.03137255 0.0627451  0.05098039 0.09019608]
   [0.05490196 0.04705882 0.05490196 0.08627451 0.05098039 0.06666667]
   [0.04705882 0.03921569 0.05882353 0.0745098  0.03921569 0.03137255]]]


 ...


 [[[0.56078431 0.48627451 0.35686275 0.69019608 0.57254902 0.41568627]
   [0.54509804 0.4627451  0.3372549  0.6745098  0.56078431 0.40784314]
   [0.56078431 0.47058824 0.34509804 0.65490196 0.54901961 0.4       ]
   ...
   [0.07058824 0.18039216 0.19607843 0.07058824 0.19607843 0.19215686]
   [0.05882353 0.17254902 0.18039216 0.0627451  0.19607843 0.19215686]
   [0.04313725 0.16862745 0.16862745 0.08627451 0.22745098 0.21960784]]

  [[0.54901961 0.47843137 0.35294118 0.71764706 0.60784314 0.44705882]
   [0.54117647 0.4627451  0.34117647 0.6745098  0.56470588 0.41176471]
   [0.56862745 0.48627451 0.36470588 0.68235294 0.58039216 0.42745098]
   ...
   [0.0745098  0.18431373 0.19215686 0.0627451  0.2        0.2       ]
   [0.05098039 0.16470588 0.16470588 0.05882353 0.20392157 0.19607843]
   [0.05098039 0.17647059 0.16470588 0.06666667 0.21960784 0.20784314]]

  [[0.54901961 0.48627451 0.36078431 0.68235294 0.57647059 0.40784314]
   [0.50980392 0.44313725 0.31372549 0.65098039 0.54901961 0.38823529]
   [0.51764706 0.43921569 0.31372549 0.65098039 0.55686275 0.39607843]
   ...
   [0.07843137 0.19215686 0.18431373 0.05882353 0.20392157 0.20392157]
   [0.05882353 0.17647059 0.16078431 0.05490196 0.20784314 0.20392157]
   [0.04313725 0.16862745 0.14901961 0.03529412 0.19607843 0.18039216]]

  ...

  [[0.57254902 0.49411765 0.37254902 0.62745098 0.53333333 0.41568627]
   [0.55686275 0.47058824 0.36078431 0.6627451  0.56862745 0.43921569]
   [0.54901961 0.46666667 0.35294118 0.67058824 0.58039216 0.43529412]
   ...
   [0.83137255 0.83137255 0.7372549  0.8745098  0.65882353 0.07058824]
   [0.82745098 0.85098039 0.81176471 0.86666667 0.72941176 0.2627451 ]
   [0.81960784 0.85490196 0.81568627 0.8627451  0.8627451  0.65098039]]

  [[0.58039216 0.49803922 0.36862745 0.62745098 0.53333333 0.41568627]
   [0.58039216 0.49411765 0.37254902 0.63921569 0.54901961 0.40784314]
   [0.59215686 0.50588235 0.38431373 0.65098039 0.55686275 0.40392157]
   ...
   [0.8        0.82352941 0.77647059 0.84705882 0.65882353 0.14117647]
   [0.82352941 0.84313725 0.81176471 0.83529412 0.78431373 0.47843137]
   [0.82352941 0.84705882 0.81568627 0.82352941 0.8627451  0.78431373]]

  [[0.6        0.51372549 0.36862745 0.62745098 0.55294118 0.41960784]
   [0.61176471 0.5254902  0.38823529 0.63529412 0.55686275 0.40784314]
   [0.59215686 0.50196078 0.37254902 0.65882353 0.58039216 0.41176471]
   ...
   [0.80784314 0.85490196 0.81960784 0.85490196 0.68235294 0.2745098 ]
   [0.83529412 0.85098039 0.82745098 0.82745098 0.83921569 0.6745098 ]
   [0.83921569 0.84705882 0.82745098 0.79607843 0.84313725 0.84313725]]]


 [[[0.8627451  0.87843137 0.92156863 0.91372549 0.9254902  0.95294118]
   [0.83921569 0.85490196 0.89803922 0.90196078 0.9254902  0.95294118]
   [0.81568627 0.82745098 0.87058824 0.87843137 0.92156863 0.95686275]
   ...
   [0.9254902  0.92941176 0.96470588 0.85490196 0.8745098  0.9372549 ]
   [0.90588235 0.90196078 0.9372549  0.90980392 0.91372549 0.95294118]
   [0.90980392 0.90588235 0.92941176 0.83137255 0.85098039 0.89411765]]

  [[0.85490196 0.88235294 0.91764706 0.87843137 0.89411765 0.9254902 ]
   [0.84705882 0.8627451  0.90196078 0.90980392 0.9254902  0.96470588]
   [0.8627451  0.87843137 0.91372549 0.88235294 0.89411765 0.9372549 ]
   ...
   [0.87843137 0.88627451 0.91764706 0.85882353 0.87843137 0.92156863]
   [0.91764706 0.91372549 0.94509804 0.90196078 0.92156863 0.94509804]
   [0.92156863 0.91764706 0.94117647 0.88627451 0.90980392 0.94117647]]

  [[0.82352941 0.85098039 0.87843137 0.87843137 0.90196078 0.9254902 ]
   [0.78431373 0.80784314 0.83137255 0.88235294 0.87843137 0.93333333]
   [0.84313725 0.8627451  0.88627451 0.9372549  0.94901961 0.97254902]
   ...
   [0.92156863 0.92941176 0.94901961 0.80392157 0.83921569 0.87058824]
   [0.89019608 0.89411765 0.91372549 0.89019608 0.88627451 0.90588235]
   [0.92941176 0.9254902  0.94901961 0.85490196 0.87058824 0.89411765]]

  ...

  [[0.98823529 0.99607843 0.99607843 0.98431373 0.99607843 1.        ]
   [0.91372549 0.9254902  0.93333333 0.90196078 0.9254902  0.96078431]
   [0.82352941 0.83921569 0.87058824 0.8745098  0.90588235 0.94901961]
   ...
   [0.85490196 0.87058824 0.87843137 0.89411765 0.89019608 0.9254902 ]
   [0.86666667 0.87843137 0.88627451 0.93333333 0.9254902  0.94901961]
   [0.88627451 0.89019608 0.90980392 0.87058824 0.86666667 0.88627451]]

  [[0.99215686 1.         0.99607843 0.98431373 0.99607843 0.99215686]
   [0.97254902 0.98823529 0.99215686 0.94509804 0.96470588 0.98823529]
   [0.85490196 0.8745098  0.89019608 0.82352941 0.85490196 0.89019608]
   ...
   [0.96470588 0.98039216 0.98039216 0.92941176 0.92941176 0.94509804]
   [0.97254902 0.98823529 0.99215686 0.90980392 0.91372549 0.93333333]
   [0.98039216 0.98431373 0.99607843 0.93333333 0.94509804 0.96862745]]

  [[0.98823529 0.99607843 0.99607843 0.99215686 1.         0.98823529]
   [0.98431373 1.         1.         0.97254902 0.98823529 0.99215686]
   [0.98039216 1.         1.         0.89019608 0.91764706 0.9372549 ]
   ...
   [0.98039216 0.99607843 0.99215686 0.98823529 0.99607843 0.98823529]
   [0.94509804 0.96078431 0.96470588 0.97254902 0.99215686 0.98823529]
   [0.96862745 0.97647059 0.98823529 0.89411765 0.93333333 0.9254902 ]]]


 [[[0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]
   [0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]
   [0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]
   ...
   [0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]
   [0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]
   [0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]]

  [[0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   ...
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]]

  [[0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   ...
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   [0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]]

  ...

  [[0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   ...
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   [0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]]

  [[0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   ...
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]]

  [[0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]
   [0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]
   [0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]
   ...
   [0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]
   [0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]
   [0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]]]]
```

Generate fake data

Data augementation. Because category five has only 13 samples. We decide to rotate them and flip them and etc.

In [17]:

```python
# Data augementation
# generate more category FIVE

def rotate_a_bunch(a_list_of_image):
    res = []
    for i in range(0, len(a_list_of_image)):
        rotate_once1 = np.rot90(a_list_of_image[i], k=1, axes=(0, 1))
        a = rotate_once1[::-1]
        rotate_twice1 = np.rot90(rotate_once1, k=1, axes=(0, 1))
        b = rotate_twice1[::-1]
        rotate_third1 = np.rot90(rotate_twice1, k=1, axes=(0, 1))
        c = rotate_third1[::-1]
        rotate_once2 = np.rot90(a_list_of_image[i], k=1, axes=(1, 0))
        d= rotate_once2[::-1]
        rotate_twice2 = np.rot90(rotate_once2, k=1, axes=(1, 0))
        e = rotate_twice2[::-1]
        rotate_third2 = np.rot90(rotate_twice2, k=1, axes=(1, 0))
        f = rotate_third2[::-1]

        new_gen = [rotate_once1,rotate_once2, rotate_twice1,rotate_twice2,rotate_third1, rotate_third2
]
        new_gen2 = [a,b,c,d,e,f]
        res= new_gen + res + new_gen2
    return np.array(res)

generated_5 = rotate_a_bunch(train_is_five)

print(generated_5)

generated_5_label = [5] * len(generated_5)

# create the new training data set (note test data set is already seperated out before hand)
train_images = np.concatenate((train_images , generated_5))
train_labels =  np.concatenate((train_labels , generated_5_label))
```

```
[[[[0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]
   [0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   ...
   [0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   [0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]]

  [[0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   ...
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]]

  [[0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   ...
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   [0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]]

  ...

  [[0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   ...
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]]

  [[0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   ...
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]]

  [[0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]
   ...
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]
   [0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]]]


 [[[0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]
   ...
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]
   [0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]]

  [[0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   ...
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]]

  [[0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   ...
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]]

  ...

  [[0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   ...
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   [0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]]

  [[0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   ...
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]]

  [[0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]
   [0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   ...
   [0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   [0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]]]


 [[[0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]
   [0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]
   [0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]
   ...
   [0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]
   [0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]
   [0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]]

  [[0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   ...
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]]

  [[0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   ...
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]]

  ...

  [[0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   ...
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]]

  [[0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   ...
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]]

  [[0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]
   [0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]
   [0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]
   ...
   [0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]
   [0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]
   [0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]]]


 ...


 [[[0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]
   [0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   ...
   [0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   [0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]]

  [[0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   ...
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]]

  [[0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   ...
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   [0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]]

  ...

  [[0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   ...
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]]

  [[0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   ...
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]]

  [[0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]
   ...
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]
   [0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]]]


 [[[0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]
   [0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]
   [0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]
   ...
   [0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]
   [0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]
   [0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]]

  [[0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   ...
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]]

  [[0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   ...
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]]

  ...

  [[0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   ...
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]]

  [[0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   ...
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]]

  [[0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]
   [0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]
   [0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]
   ...
   [0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]
   [0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]
   [0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]]]


 [[[0.0627451  0.03921569 0.0627451  0.05098039 0.04313725 0.04705882]
   [0.0745098  0.05490196 0.07058824 0.05098039 0.04705882 0.04313725]
   [0.05490196 0.03529412 0.04705882 0.05882353 0.05490196 0.04705882]
   ...
   [0.05882353 0.03921569 0.0627451  0.09411765 0.0745098  0.0745098 ]
   [0.05490196 0.02352941 0.05490196 0.09019608 0.0627451  0.0627451 ]
   [0.0627451  0.03529412 0.0627451  0.05882353 0.03529412 0.02745098]]

  [[0.05882353 0.03921569 0.03137255 0.07058824 0.05490196 0.05882353]
   [0.05490196 0.03529412 0.03529412 0.0627451  0.04705882 0.04705882]
   [0.05882353 0.04313725 0.04705882 0.04705882 0.03137255 0.03137255]
   ...
   [0.07058824 0.05490196 0.07058824 0.07058824 0.03921569 0.05098039]
   [0.06666667 0.04705882 0.06666667 0.09411765 0.05882353 0.06666667]
   [0.05098039 0.03137255 0.04705882 0.06666667 0.03137255 0.03921569]]

  [[0.11372549 0.09411765 0.07843137 0.09019608 0.06666667 0.07843137]
   [0.07058824 0.05490196 0.04705882 0.07058824 0.04705882 0.05490196]
   [0.05882353 0.04313725 0.04313725 0.06666667 0.04313725 0.04313725]
   ...
   [0.16470588 0.14509804 0.16078431 0.1372549  0.09803922 0.10980392]
   [0.16862745 0.14901961 0.16470588 0.22745098 0.18039216 0.19607843]
   [0.16862745 0.14901961 0.16470588 0.25098039 0.19607843 0.21176471]]

  ...

  [[0.05882353 0.05098039 0.04313725 0.07843137 0.05882353 0.0745098 ]
   [0.07058824 0.05882353 0.05490196 0.05490196 0.03529412 0.05098039]
   [0.07843137 0.0627451  0.05490196 0.09411765 0.08235294 0.09411765]
   ...
   [0.0745098  0.05882353 0.05882353 0.07058824 0.0627451  0.0745098 ]
   [0.07058824 0.05490196 0.05098039 0.06666667 0.05098039 0.06666667]
   [0.08235294 0.06666667 0.05490196 0.0627451  0.04313725 0.05882353]]

  [[0.05490196 0.05098039 0.04313725 0.09411765 0.07843137 0.09411765]
   [0.07843137 0.0745098  0.06666667 0.0627451  0.04313725 0.05882353]
   [0.0627451  0.04705882 0.04313725 0.09019608 0.07058824 0.08627451]
   ...
   [0.0627451  0.04705882 0.04705882 0.04705882 0.03529412 0.04705882]
   [0.0745098  0.05882353 0.05490196 0.09411765 0.07058824 0.08627451]
   [0.07843137 0.0627451  0.05098039 0.0745098  0.05098039 0.06666667]]

  [[0.05490196 0.05098039 0.04313725 0.0627451  0.03921569 0.05490196]
   [0.05882353 0.05490196 0.04705882 0.05490196 0.03137255 0.04705882]
   [0.07058824 0.0627451  0.05490196 0.06666667 0.04705882 0.0627451 ]
   ...
   [0.05882353 0.04313725 0.04313725 0.03921569 0.02745098 0.03921569]
   [0.05882353 0.04313725 0.03921569 0.0745098  0.05490196 0.07058824]
   [0.08627451 0.07058824 0.05882353 0.08627451 0.0627451  0.07843137]]]]
```

In [18]:

```python
print(len(train_images))
print(len(train_labels))
print(train_labels)
```

```
1054
1054
[13  2  1 ...  5  5  5]
```

Further split training data to actual training data and validation data set.

In [19]:

```python
train1_images , val_images, train1_labels , val_labels =  train_test_split(train_images, train_labels,
test_size=0.2, random_state=1)
```

The number of each data set after the data augementation

In [20]:

```python
print("#Training data: {}\n #Validation data: {}\n #Testing data: {}\n#Class: {}".format(len(train1_ima
ges), len(val_images), len(test_images), len(set(train_labels))))
```

```
#Training data: 843
 #Validation data: 211
 #Testing data: 225
#Class: 4
```

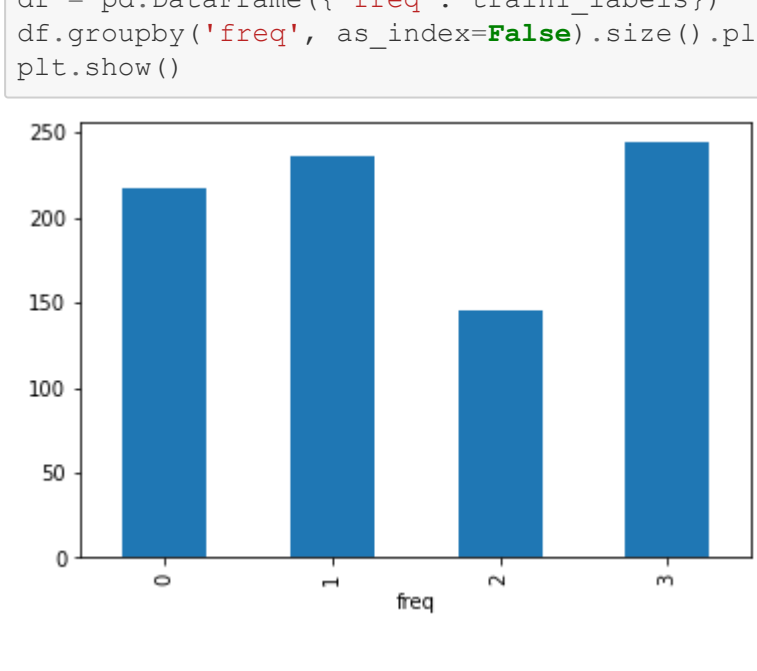
In [21]:

```python
# rename 1,2,5,12 category number to 0 1 2 3
def change_cat_number(list_1):
    for i in range(0,len(list_1)):
        if list_1[i] == 1:
            list_1[i] = 0
        elif list_1[i] == 2:
            list_1[i] = 1
        elif list_1[i] == 5:
            list_1[i] = 2
        else:
            list_1[i] = 3
    return list_1

train1_labels = change_cat_number(train1_labels)
val_labels= change_cat_number(val_labels)
test_labels = change_cat_number(test_labels)
```

take a look at the frequency distribution of our four categories in the training data set

In [22]:

```python
import pandas as pd
import matplotlib.pyplot as plt

# output the new frequency
df = pd.DataFrame({'freq': train1_labels})
df.groupby('freq', as_index=False).size().plot(kind='bar')
plt.show()
```



Take a look at transformed labels. There should only by 0, 1,2 ,3.

In [23]:

```python
print(val_labels)
```

```
[3 3 1 3 1 2 2 3 0 2 1 2 1 0 3 3 1 1 0 1 1 0 3 0 1 1 1 1 3 0 3 3 0 1 3 0 0
 3 0 2 1 1 1 3 1 2 1 0 1 0 0 1 1 1 2 1 0 1 0 3 1 2 3 0 0 3 2 3 0 0 3 3 1 1
 3 1 0 3 3 0 1 0 1 3 1 1 0 2 1 3 0 1 3 1 1 0 0 3 1 1 0 2 0 1 1 3 0 0 1 2 0
 1 3 1 0 3 2 0 0 1 0 3 3 0 3 0 3 2 3 2 3 0 3 0 3 3 0 3 1 1 0 1 2 0 2 1 3 0
 3 0 1 3 0 3 3 1 0 2 2 1 2 0 2 3 0 3 0 1 3 3 3 0 3 1 0 1 1 1 1 0 0 1 1 2 1
 3 3 1 0 1 1 1 3 0 3 1 1 0 2 0 0 0 1 0 1 1 3 1 3 0 1]
```

In [24]:

```python
# Load the TensorBoard notebook extension
%load_ext tensorboard

import datetime
import numpy as np
import tensorflow as tf


from tensorflow.keras import Model
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D, Dropout
```

Create model

- https://www.tensorflow.org/tutorials/images/classification

In [25]:

```python
num_classes = len(set(train_labels))
```

In [26]:

```python
# AlexNet model
class AlexNet(Sequential):
    def __init__(self, input_shape, num_classes):
        super().__init__()

        self.add(Conv2D(96, kernel_size=(11,11), strides= 4,
                        padding= 'valid', activation= 'relu',
                        input_shape= input_shape,
                        kernel_initializer= 'he_normal'))
        self.add(MaxPooling2D(pool_size=(3,3), strides= (2,2),
                              padding= 'valid', data_format= None))

        self.add(Conv2D(256, kernel_size=(5,5), strides= 1,
                        padding= 'same', activation= 'relu',
                        kernel_initializer= 'he_normal'))
        self.add(MaxPooling2D(pool_size=(3,3), strides= (2,2),
                              padding= 'valid', data_format= None))

        self.add(Conv2D(384, kernel_size=(3,3), strides= 1,
                        padding= 'same', activation= 'relu',
                        kernel_initializer= 'he_normal'))

        self.add(Conv2D(384, kernel_size=(3,3), strides= 1,
                        padding= 'same', activation= 'relu',
                        kernel_initializer= 'he_normal'))

        self.add(Conv2D(256, kernel_size=(3,3), strides= 1,
                        padding= 'same', activation= 'relu',
                        kernel_initializer= 'he_normal'))

        self.add(MaxPooling2D(pool_size=(3,3), strides= (2,2),
                              padding= 'valid', data_format= None))

        #drop out at 0.4  achieve 79% accuracy in prediction
        #drop out at 0.5 achieve 73% accuracy in prediction
        self.add(Flatten())
        self.add(Dense(4096, activation= 'relu'))
        self.add(Dropout(0.3))
        self.add(Dense(4096, activation= 'relu'))
        self.add(Dropout(0.3))
        self.add(Dense(1000, activation= 'relu'))
        self.add(Dropout(0.3))
        self.add(Dense(num_classes, activation= 'softmax'))

        self.compile(optimizer= tf.keras.optimizers.Adam(0.0001),
                    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
                    metrics=['accuracy'])
```

In [27]:

```python
model = AlexNet((200, 200, 6), num_classes)
```

In [28]:

```python
model.summary()
```

```
Model: "alex_net"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 96)        69792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 23, 23, 96)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 23, 23, 256)       614656
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 384)       885120
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 11, 11, 384)       1327488
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 11, 11, 256)       884992
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0
_________________________________________________________________
dense (Dense)                (None, 4096)              26218496
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1000)              4097000
_________________________________________________________________
dropout_2 (Dropout)          (None, 1000)              0
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 4004
=================================================================
Total params: 50,882,860
Trainable params: 50,882,860
Non-trainable params: 0
_________________________________________________________________
```

In [29]:

```python
# some training parameters
EPOCHS = 10
image_height = 200
image_width = 200
train_dir = "train"
valid_dir = "validation"
model_dir = "my_model.h5"
```

In [30]:

```python
# start training
history = model.fit(train1_images, train1_labels, epochs=10,
                    validation_data=(val_images, val_labels))
```

```
Epoch 1/10
27/27 [==============================] - 74s 3s/step - loss: 1.4820 - accuracy: 0.3096 - val_loss: 1.
2669 - val_accuracy: 0.3697
Epoch 2/10
27/27 [==============================] - 71s 3s/step - loss: 1.2044 - accuracy: 0.4733 - val_loss: 1.
1371 - val_accuracy: 0.4834
Epoch 3/10
27/27 [==============================] - 72s 3s/step - loss: 1.0722 - accuracy: 0.5208 - val_loss: 1.
0860 - val_accuracy: 0.4645
Epoch 4/10
27/27 [==============================] - 73s 3s/step - loss: 0.9143 - accuracy: 0.6109 - val_loss: 0.
9937 - val_accuracy: 0.5498
Epoch 5/10
27/27 [==============================] - 72s 3s/step - loss: 0.8659 - accuracy: 0.6109 - val_loss: 1.
0044 - val_accuracy: 0.6019
Epoch 6/10
27/27 [==============================] - 73s 3s/step - loss: 0.7583 - accuracy: 0.7034 - val_loss: 0.
8828 - val_accuracy: 0.6730
Epoch 7/10
27/27 [==============================] - 70s 3s/step - loss: 0.5533 - accuracy: 0.7912 - val_loss: 0.
9398 - val_accuracy: 0.6825
Epoch 8/10
27/27 [==============================] - 75s 3s/step - loss: 0.5751 - accuracy: 0.7841 - val_loss: 0.
8716 - val_accuracy: 0.7251
Epoch 9/10
27/27 [==============================] - 70s 3s/step - loss: 0.4186 - accuracy: 0.8529 - val_loss: 0.
6999 - val_accuracy: 0.7962
Epoch 10/10
27/27 [==============================] - 73s 3s/step - loss: 0.3258 - accuracy: 0.8802 - val_loss: 1.
0708 - val_accuracy: 0.7393
```

In [31]:

```python
# make prediction
predicted_label = model.predict_classes(test_images)
```

```
WARNING:tensorflow:From <ipython-input-31-0b36f4dbf883>:2: Sequential.predict_classes (from tensorflo
w.python.keras.engine.sequential) is deprecated and will be removed after 2021-01-01.
Instructions for updating:
Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classif
ication   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int
32")`,   if your model does binary classification   (e.g. if it uses a `sigmoid` last-layer activatio
n).
```

In [32]:

```python
print(predicted_label)
```

```
[0 3 3 3 3 3 3 0 1 3 0 3 0 2 0 3 1 3 0 0 3 2 0 1 0 1 2 3 0 1 0 3 3 0 0 3 2
 0 0 3 3 0 1 3 3 0 0 3 1 3 3 0 0 3 1 3 0 3 3 3 0 3 3 0 0 3 0 3 1 3 0 3 1 3
 0 0 0 1 3 3 3 3 3 1 1 1 3 3 0 3 0 1 1 0 3 1 0 1 0 0 3 1 3 2 0 3 3 1 0 2 3
 0 3 2 3 3 3 3 1 3 3 3 3 1 0 0 0 3 3 1 0 0 0 3 0 1 3 0 0 0 1 1 0 1 0 3 1
 3 3 3 0 0 1 0 0 0 1 3 0 2 3 3 0 1 3 3 0 3 2 0 1 3 3 1 1 3 0 0 3 0 0 3 0 0
 0 3 0 1 3 0 0 1 3 3 0 2 0 1 3 3 3 1 3 3 3 0 3 0 0 3 0 0 1 1 1 0 1 0 3 2 0
 1 0 0]
```

In [33]:

```python
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(test_labels, predicted_label))
print(classification_report(test_labels, predicted_label))
```

```
[[64  3  0  7]
 [13 40  1 10]
 [ 0  0  6  2]
 [ 4  0  4 71]]
              precision    recall  f1-score   support

           0       0.79      0.86      0.83        74
           1       0.93      0.62      0.75        64
           2       0.55      0.75      0.63         8
           3       0.79      0.90      0.84        79

    accuracy                           0.80       225
   macro avg       0.76      0.78      0.76       225
weighted avg       0.82      0.80      0.80       225
```

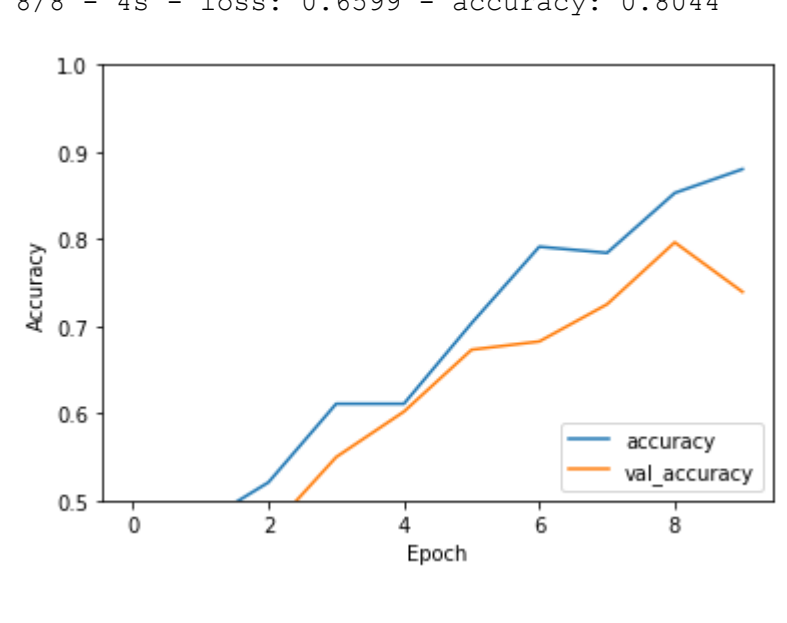
Reason why accuracy doesn't further increase

1. imbalance of trianing set
2. learning rate too large

In [34]:

```python
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)
```

```
8/8 - 4s - loss: 0.6599 - accuracy: 0.8044
```



Evaluation

During the process of find the best algorithm, we tried out many variations of neural networks. AlexNet turned out to have the most accuracy. LeNet was the second best algorithm. Being the first major CNN model that used GPU's for training, AlexNet is a deeper architecture with 8 layers and is better at extracting features comparing to LeNet. Other more complicated neural networks, Google net and VGG, turned to have a lower accuracy, perhaps because the nature of the task is not suitable for them, which means we still require futhur exploration on what algorithm is most suitable for the task.

1. LeNet-5: 65%
2. Alex Net with BN: 66%
3. Google Net: failed (<40%)
4. VGG: failed (<40%)

Collaboration Acknowledgement: This project was collaborated with Yijie Yao,姚奕捷, and Yingyi Liang, 梁樱议.

For complete work on this project, go to https://github.com/503600265/PRML