prediction-using-ensemble-cnn

May 12, 2023

Importing the data from kaggle

[1]:
```
! pip install -q kaggle
from google.colab import files
files.upload()
! mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
```

```
<IPython.core.display.HTML object>
```

```
Saving kaggle.json to kaggle (1).json
mkdir: cannot create directory '/root/.kaggle': File exists
```

[2]:
```
!kaggle datasets download -d sachinkumar413/diabetic-retinopathy-dataset
```

```
diabetic-retinopathy-dataset.zip: Skipping, found more recently modified local
copy (use --force to force download)
```

[ ]:
```
!unzip diabetic-retinopathy-dataset.zip
```

Importing the dependencies

[4]:
```
import os
import numpy as np
import pandas as pd
import cv2 as cv
import xgboost as xgb
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from keras.layers import Conv2D, MaxPool2D, Dropout, Dense, Flatten,
↪GlobalAveragePooling2D
from keras.models import Sequential
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
[5]: data = []                # reading the images and the labels
     for dirs in os.listdir("/content/data"):
       new_path = os.path.join("/content/data", dirs)
       class_name = dirs
       for img in os.listdir(new_path):
         img_arr = cv.imread(os.path.join(new_path, img), cv.IMREAD_GRAYSCALE)
         re = cv.resize(img_arr, (128,128))
         data.append([re, class_name])
```

```
[6]: np.random.shuffle(data)     # shuffling the dataset
```

```
[7]: images = []                    # unpacking the data array into images and labels
     labels = []

     for features, label in data:
       images.append(features)
       labels.append(label)
```

```
[8]: encoding = {'Moderate DR' : 0, 'Healthy' : 1, 'Proliferate DR' : 2, 'Severe DR'␣
      ↪: 3, 'Mild DR' : 4}     # encoder dictionary
```

```
[9]: for i in range(len(labels)):     # encoding the labels
       labels[i] = encoding.get(labels[i])
```

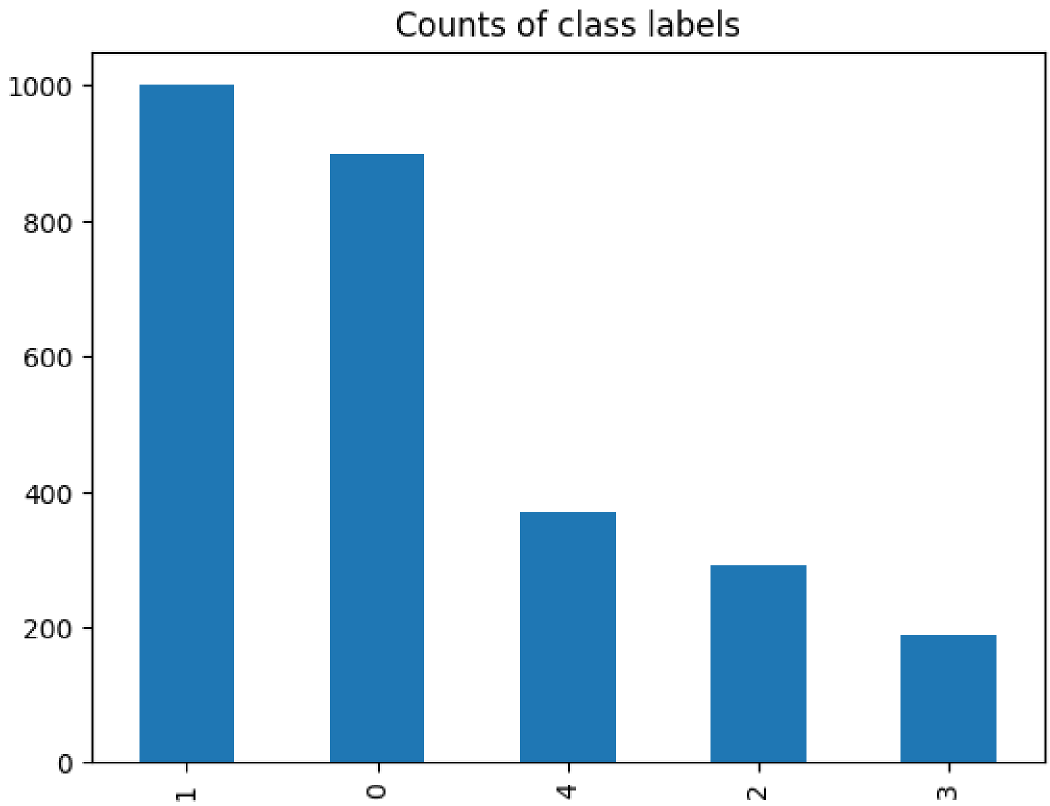
```
[10]: images = np.array(images)
      labels = np.array(labels)
```

```
[11]: images = np.expand_dims(images, axis=3)     # adding an extra dimension to the␣
       ↪images array
```

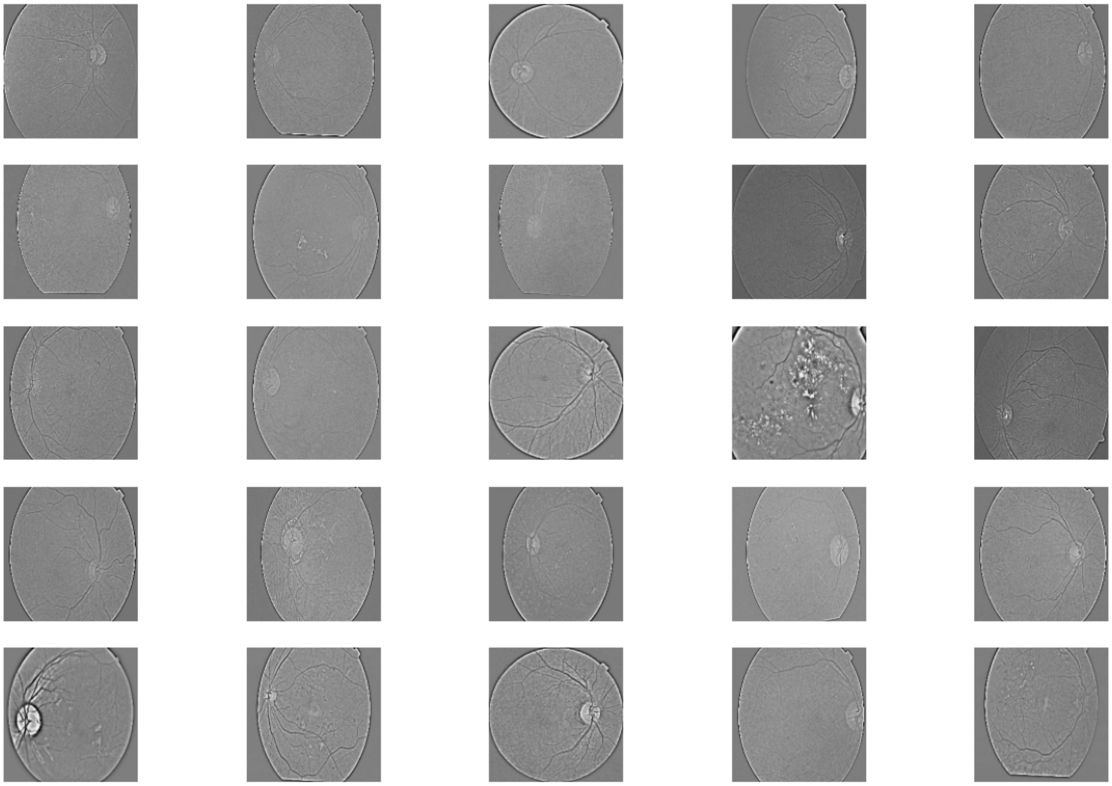
```
[12]: print(f"shape of the images are {images.shape}")
      print(f"shape of the labels are {labels.shape}")
```

```
shape of the images are (2750, 128, 128, 1)
shape of the labels are (2750,)
```

```
[13]: df = pd.DataFrame(data={"labels": labels})
      df["labels"].value_counts().plot(kind="bar")
      plt.title("Counts of class labels")
      plt.show()
```



[14]:
```
plt.figure(figsize=(15,10))
for i in range(25):
  plt.subplot(5, 5, i+1)
  plt.imshow(images[i], cmap='gray')
  plt.axis("off")
plt.show()
```



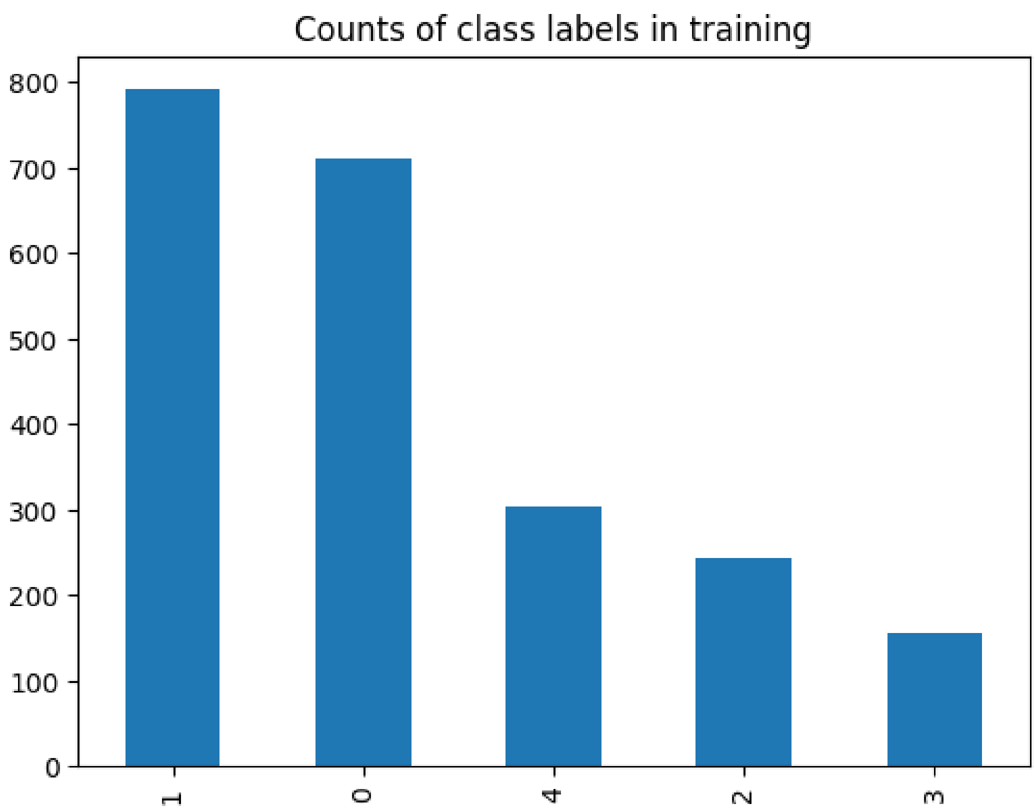
[15]:
```
images = images/255.0              # normalizing the images
```

[16]:
```
x_train, x_test, y_train, y_test = train_test_split(images, labels, test_size=0.
↪2)
```

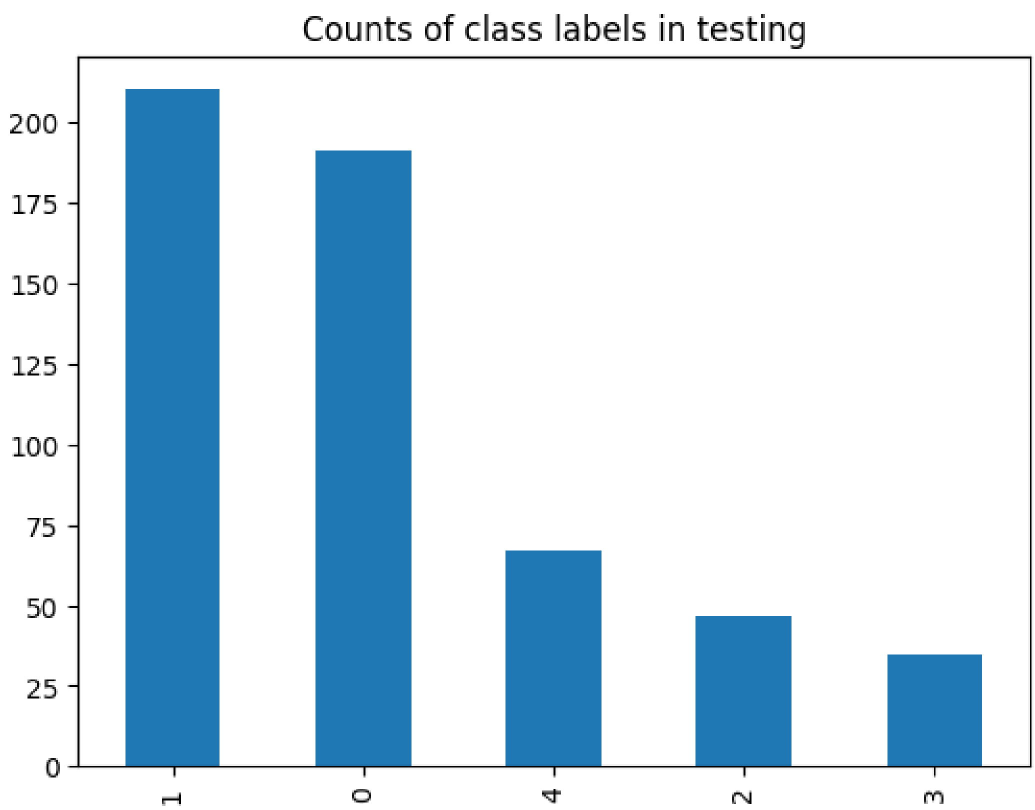
[17]:
```
print(f"shape of x_train {x_train.shape}")
print(f"shape of y_train {y_train.shape}")
print(f"shape of x_test {x_test.shape}")
print(f"shape of y_test {y_test.shape}")
```

```
shape of x_train (2200, 128, 128, 1)
shape of y_train (2200,)
shape of x_test (550, 128, 128, 1)
shape of y_test (550,)
```

[19]:
```
df = pd.DataFrame(data={"labels": y_train})
df["labels"].value_counts().plot(kind="bar")
plt.title("Counts of class labels in training")
plt.show()
```



[18]:
```
df = pd.DataFrame(data={"labels": y_test})
df["labels"].value_counts().plot(kind="bar")
plt.title("Counts of class labels in testing")
plt.show()
```



[20]:
```
class_weights = {0 : 0.3, 1 : 0.3, 2 : 0.7, 3 : 0.6, 4 : 0.4}          # giving␣
 ↪weights to class
```

CNN Model 1

[21]:
```
model1 = Sequential()

model1.add(Conv2D(32, (3,3), input_shape=(128,128,1), activation="leaky_relu"))
model1.add(MaxPool2D(2,2))
model1.add(Conv2D(64, (3,3),activation="leaky_relu"))
model1.add(MaxPool2D(2,2))
model1.add(Conv2D(128, (3,3),activation="leaky_relu"))
model1.add(MaxPool2D(2,2))
model1.add(Flatten())
model1.add(Dense(100, activation="relu"))
model1.add(Dense(5, activation="softmax"))
```

[22]:
```
model1.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      320

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 61, 61, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 30, 30, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 28, 28, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 100)               2508900

 dense_1 (Dense)             (None, 5)                 505

=================================================================
Total params: 2,602,077
Trainable params: 2,602,077
Non-trainable params: 0
_________________________________________________________________
```

[23]:
```
model1.compile(optimizer="adam", loss="sparse_categorical_crossentropy",
 ↪metrics=['accuracy'])
```

[24]:
```
history1 = model1.fit(x_train, y_train, validation_data = (x_test, y_test),
 ↪epochs=100, class_weight = class_weights)
```

```
Epoch 1/100
69/69 [==============================] - 11s 32ms/step - loss: 0.4939 -
accuracy: 0.5355 - val_loss: 1.0564 - val_accuracy: 0.5364
Epoch 2/100
69/69 [==============================] - 1s 21ms/step - loss: 0.4199 - accuracy:
0.6014 - val_loss: 1.1655 - val_accuracy: 0.4327
Epoch 3/100
69/69 [==============================] - 1s 21ms/step - loss: 0.4049 - accuracy:
0.6223 - val_loss: 0.9132 - val_accuracy: 0.6745
Epoch 4/100
69/69 [==============================] - 1s 21ms/step - loss: 0.3794 - accuracy:
0.6586 - val_loss: 0.9072 - val_accuracy: 0.6691
Epoch 5/100
69/69 [==============================] - 1s 21ms/step - loss: 0.3600 - accuracy:
```

```
0.6714 - val_loss: 0.8613 - val_accuracy: 0.6891
Epoch 6/100
69/69 [==============================] - 1s 21ms/step - loss: 0.3412 - accuracy:
0.6859 - val_loss: 0.8828 - val_accuracy: 0.6527
Epoch 7/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3147 - accuracy:
0.7132 - val_loss: 0.8812 - val_accuracy: 0.6600
Epoch 8/100
69/69 [==============================] - 1s 22ms/step - loss: 0.2794 - accuracy:
0.7518 - val_loss: 0.9225 - val_accuracy: 0.6473
Epoch 9/100
69/69 [==============================] - 1s 21ms/step - loss: 0.2606 - accuracy:
0.7695 - val_loss: 1.0825 - val_accuracy: 0.5545
Epoch 10/100
69/69 [==============================] - 2s 23ms/step - loss: 0.2211 - accuracy:
0.7950 - val_loss: 1.1058 - val_accuracy: 0.6218
Epoch 11/100
69/69 [==============================] - 1s 21ms/step - loss: 0.1926 - accuracy:
0.8341 - val_loss: 1.1218 - val_accuracy: 0.6527
Epoch 12/100
69/69 [==============================] - 2s 28ms/step - loss: 0.1701 - accuracy:
0.8623 - val_loss: 1.1723 - val_accuracy: 0.6509
Epoch 13/100
69/69 [==============================] - 2s 29ms/step - loss: 0.1426 - accuracy:
0.8855 - val_loss: 1.2054 - val_accuracy: 0.6727
Epoch 14/100
69/69 [==============================] - 2s 31ms/step - loss: 0.1244 - accuracy:
0.9018 - val_loss: 1.3199 - val_accuracy: 0.6545
Epoch 15/100
69/69 [==============================] - 2s 34ms/step - loss: 0.0995 - accuracy:
0.9177 - val_loss: 1.4119 - val_accuracy: 0.6600
Epoch 16/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0846 - accuracy:
0.9409 - val_loss: 1.6440 - val_accuracy: 0.6073
Epoch 17/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0743 - accuracy:
0.9486 - val_loss: 1.6058 - val_accuracy: 0.6527
Epoch 18/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0707 - accuracy:
0.9523 - val_loss: 1.7489 - val_accuracy: 0.6200
Epoch 19/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0584 - accuracy:
0.9595 - val_loss: 1.6385 - val_accuracy: 0.6418
Epoch 20/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0612 - accuracy:
0.9600 - val_loss: 2.0691 - val_accuracy: 0.6182
Epoch 21/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0522 - accuracy:
```

```
0.9691 - val_loss: 1.6401 - val_accuracy: 0.6673
Epoch 22/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0395 - accuracy:
0.9768 - val_loss: 1.9497 - val_accuracy: 0.6418
Epoch 23/100
69/69 [==============================] - 2s 22ms/step - loss: 0.0337 - accuracy:
0.9800 - val_loss: 1.9131 - val_accuracy: 0.6509
Epoch 24/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0357 - accuracy:
0.9795 - val_loss: 1.7339 - val_accuracy: 0.6273
Epoch 25/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0345 - accuracy:
0.9823 - val_loss: 1.8260 - val_accuracy: 0.6727
Epoch 26/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0301 - accuracy:
0.9827 - val_loss: 1.8176 - val_accuracy: 0.6455
Epoch 27/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0280 - accuracy:
0.9818 - val_loss: 1.9889 - val_accuracy: 0.6436
Epoch 28/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0247 - accuracy:
0.9827 - val_loss: 1.9257 - val_accuracy: 0.6545
Epoch 29/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0222 - accuracy:
0.9845 - val_loss: 1.8272 - val_accuracy: 0.6509
Epoch 30/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0225 - accuracy:
0.9836 - val_loss: 2.0507 - val_accuracy: 0.6236
Epoch 31/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0212 - accuracy:
0.9859 - val_loss: 2.0196 - val_accuracy: 0.6473
Epoch 32/100
69/69 [==============================] - 1s 22ms/step - loss: 0.0225 - accuracy:
0.9841 - val_loss: 1.9038 - val_accuracy: 0.6273
Epoch 33/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0234 - accuracy:
0.9832 - val_loss: 1.9637 - val_accuracy: 0.6582
Epoch 34/100
69/69 [==============================] - 2s 30ms/step - loss: 0.0233 - accuracy:
0.9845 - val_loss: 2.0272 - val_accuracy: 0.6382
Epoch 35/100
69/69 [==============================] - 2s 31ms/step - loss: 0.0236 - accuracy:
0.9814 - val_loss: 1.9292 - val_accuracy: 0.6564
Epoch 36/100
69/69 [==============================] - 2s 32ms/step - loss: 0.0207 - accuracy:
0.9841 - val_loss: 1.9597 - val_accuracy: 0.6455
Epoch 37/100
69/69 [==============================] - 2s 22ms/step - loss: 0.0255 - accuracy:
```

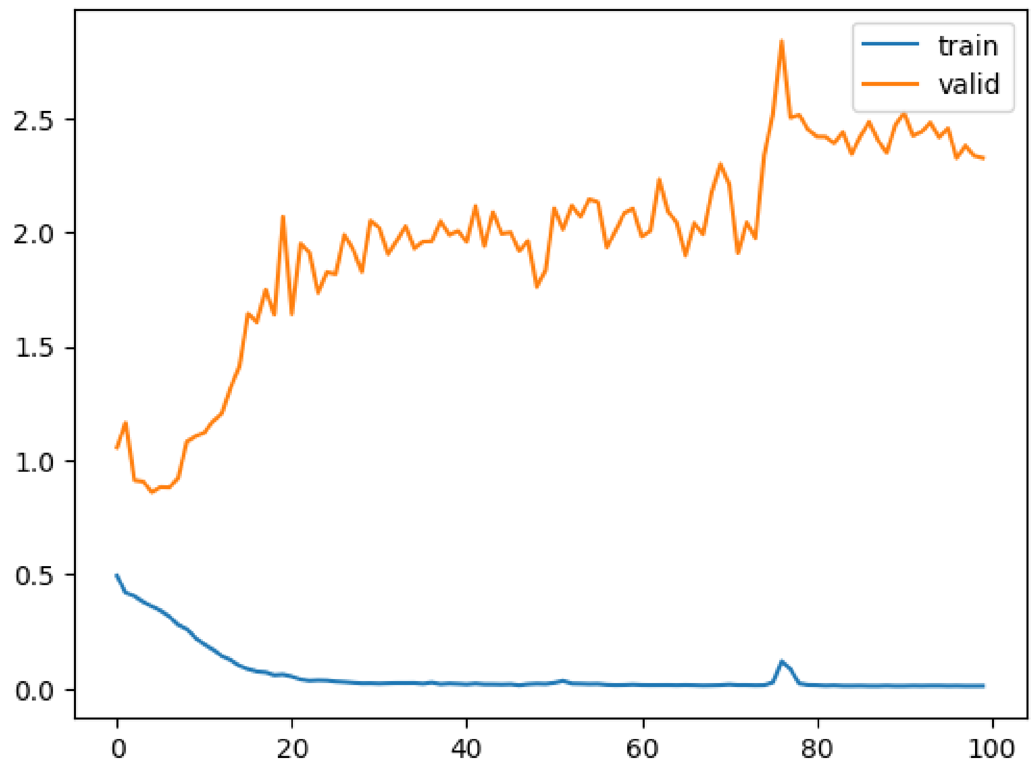
```
0.9836 - val_loss: 1.9609 - val_accuracy: 0.6600
Epoch 38/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0187 - accuracy:
0.9845 - val_loss: 2.0494 - val_accuracy: 0.6218
Epoch 39/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0216 - accuracy:
0.9845 - val_loss: 1.9887 - val_accuracy: 0.6182
Epoch 40/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0200 - accuracy:
0.9841 - val_loss: 2.0053 - val_accuracy: 0.6400
Epoch 41/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0177 - accuracy:
0.9841 - val_loss: 1.9586 - val_accuracy: 0.6455
Epoch 42/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0214 - accuracy:
0.9845 - val_loss: 2.1158 - val_accuracy: 0.6255
Epoch 43/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0179 - accuracy:
0.9836 - val_loss: 1.9402 - val_accuracy: 0.6400
Epoch 44/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0178 - accuracy:
0.9859 - val_loss: 2.0884 - val_accuracy: 0.6836
Epoch 45/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0170 - accuracy:
0.9841 - val_loss: 1.9937 - val_accuracy: 0.6509
Epoch 46/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0182 - accuracy:
0.9832 - val_loss: 1.9997 - val_accuracy: 0.6473
Epoch 47/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0138 - accuracy:
0.9855 - val_loss: 1.9185 - val_accuracy: 0.6582
Epoch 48/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0185 - accuracy:
0.9832 - val_loss: 1.9628 - val_accuracy: 0.6509
Epoch 49/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0207 - accuracy:
0.9832 - val_loss: 1.7603 - val_accuracy: 0.6709
Epoch 50/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0196 - accuracy:
0.9868 - val_loss: 1.8341 - val_accuracy: 0.6309
Epoch 51/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0243 - accuracy:
0.9795 - val_loss: 2.1052 - val_accuracy: 0.6073
Epoch 52/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0331 - accuracy:
0.9755 - val_loss: 2.0120 - val_accuracy: 0.6673
Epoch 53/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0212 - accuracy:
```

```
0.9818 - val_loss: 2.1172 - val_accuracy: 0.6691
Epoch 54/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0204 - accuracy:
0.9836 - val_loss: 2.0685 - val_accuracy: 0.6582
Epoch 55/100
69/69 [==============================] - 2s 22ms/step - loss: 0.0191 - accuracy:
0.9859 - val_loss: 2.1453 - val_accuracy: 0.6545
Epoch 56/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0197 - accuracy:
0.9845 - val_loss: 2.1331 - val_accuracy: 0.6327
Epoch 57/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0157 - accuracy:
0.9859 - val_loss: 1.9346 - val_accuracy: 0.6455
Epoch 58/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0144 - accuracy:
0.9845 - val_loss: 2.0053 - val_accuracy: 0.6764
Epoch 59/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0151 - accuracy:
0.9845 - val_loss: 2.0843 - val_accuracy: 0.6800
Epoch 60/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0167 - accuracy:
0.9832 - val_loss: 2.1044 - val_accuracy: 0.6600
Epoch 61/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0148 - accuracy:
0.9859 - val_loss: 1.9822 - val_accuracy: 0.6582
Epoch 62/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0145 - accuracy:
0.9845 - val_loss: 2.0059 - val_accuracy: 0.6582
Epoch 63/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0146 - accuracy:
0.9845 - val_loss: 2.2337 - val_accuracy: 0.6600
Epoch 64/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0147 - accuracy:
0.9855 - val_loss: 2.0910 - val_accuracy: 0.6564
Epoch 65/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0141 - accuracy:
0.9845 - val_loss: 2.0427 - val_accuracy: 0.6600
Epoch 66/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0149 - accuracy:
0.9850 - val_loss: 1.8976 - val_accuracy: 0.6636
Epoch 67/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0141 - accuracy:
0.9845 - val_loss: 2.0414 - val_accuracy: 0.6709
Epoch 68/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0129 - accuracy:
0.9864 - val_loss: 1.9917 - val_accuracy: 0.6582
Epoch 69/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0135 - accuracy:
```

```
0.9859 - val_loss: 2.1748 - val_accuracy: 0.6618
Epoch 70/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0145 - accuracy:
0.9841 - val_loss: 2.3008 - val_accuracy: 0.6418
Epoch 71/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0171 - accuracy:
0.9850 - val_loss: 2.2096 - val_accuracy: 0.6764
Epoch 72/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0147 - accuracy:
0.9841 - val_loss: 1.9085 - val_accuracy: 0.6691
Epoch 73/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0147 - accuracy:
0.9855 - val_loss: 2.0437 - val_accuracy: 0.6382
Epoch 74/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0137 - accuracy:
0.9855 - val_loss: 1.9745 - val_accuracy: 0.6655
Epoch 75/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0143 - accuracy:
0.9859 - val_loss: 2.3390 - val_accuracy: 0.6400
Epoch 76/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0258 - accuracy:
0.9768 - val_loss: 2.5172 - val_accuracy: 0.6418
Epoch 77/100
69/69 [==============================] - 1s 21ms/step - loss: 0.1198 - accuracy:
0.9186 - val_loss: 2.8398 - val_accuracy: 0.6418
Epoch 78/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0843 - accuracy:
0.9391 - val_loss: 2.5020 - val_accuracy: 0.6782
Epoch 79/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0217 - accuracy:
0.9814 - val_loss: 2.5148 - val_accuracy: 0.6491
Epoch 80/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0153 - accuracy:
0.9859 - val_loss: 2.4523 - val_accuracy: 0.6509
Epoch 81/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0143 - accuracy:
0.9855 - val_loss: 2.4201 - val_accuracy: 0.6582
Epoch 82/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0124 - accuracy:
0.9864 - val_loss: 2.4192 - val_accuracy: 0.6600
Epoch 83/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0135 - accuracy:
0.9859 - val_loss: 2.3906 - val_accuracy: 0.6455
Epoch 84/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0116 - accuracy:
0.9850 - val_loss: 2.4408 - val_accuracy: 0.6564
Epoch 85/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0115 - accuracy:
```

```
0.9850 - val_loss: 2.3430 - val_accuracy: 0.6600
Epoch 86/100
69/69 [==============================] - 2s 24ms/step - loss: 0.0120 - accuracy:
0.9859 - val_loss: 2.4218 - val_accuracy: 0.6364
Epoch 87/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0111 - accuracy:
0.9855 - val_loss: 2.4842 - val_accuracy: 0.6655
Epoch 88/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0110 - accuracy:
0.9855 - val_loss: 2.4038 - val_accuracy: 0.6455
Epoch 89/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0121 - accuracy:
0.9855 - val_loss: 2.3493 - val_accuracy: 0.6473
Epoch 90/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0109 - accuracy:
0.9859 - val_loss: 2.4700 - val_accuracy: 0.6455
Epoch 91/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0109 - accuracy:
0.9841 - val_loss: 2.5254 - val_accuracy: 0.6727
Epoch 92/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0119 - accuracy:
0.9859 - val_loss: 2.4236 - val_accuracy: 0.6418
Epoch 93/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0115 - accuracy:
0.9859 - val_loss: 2.4414 - val_accuracy: 0.6473
Epoch 94/100
69/69 [==============================] - 2s 23ms/step - loss: 0.0121 - accuracy:
0.9855 - val_loss: 2.4813 - val_accuracy: 0.6455
Epoch 95/100
69/69 [==============================] - 2s 25ms/step - loss: 0.0121 - accuracy:
0.9859 - val_loss: 2.4158 - val_accuracy: 0.6600
Epoch 96/100
69/69 [==============================] - 1s 22ms/step - loss: 0.0112 - accuracy:
0.9864 - val_loss: 2.4562 - val_accuracy: 0.6527
Epoch 97/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0116 - accuracy:
0.9841 - val_loss: 2.3254 - val_accuracy: 0.6673
Epoch 98/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0109 - accuracy:
0.9850 - val_loss: 2.3804 - val_accuracy: 0.6491
Epoch 99/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0109 - accuracy:
0.9859 - val_loss: 2.3353 - val_accuracy: 0.6673
Epoch 100/100
69/69 [==============================] - 1s 21ms/step - loss: 0.0111 - accuracy:
0.9855 - val_loss: 2.3259 - val_accuracy: 0.6345
```

[25]:
```
plt.plot(history1.history["loss"])
plt.plot(history1.history["val_loss"])
plt.legend(["train", "valid"])
plt.show()
```

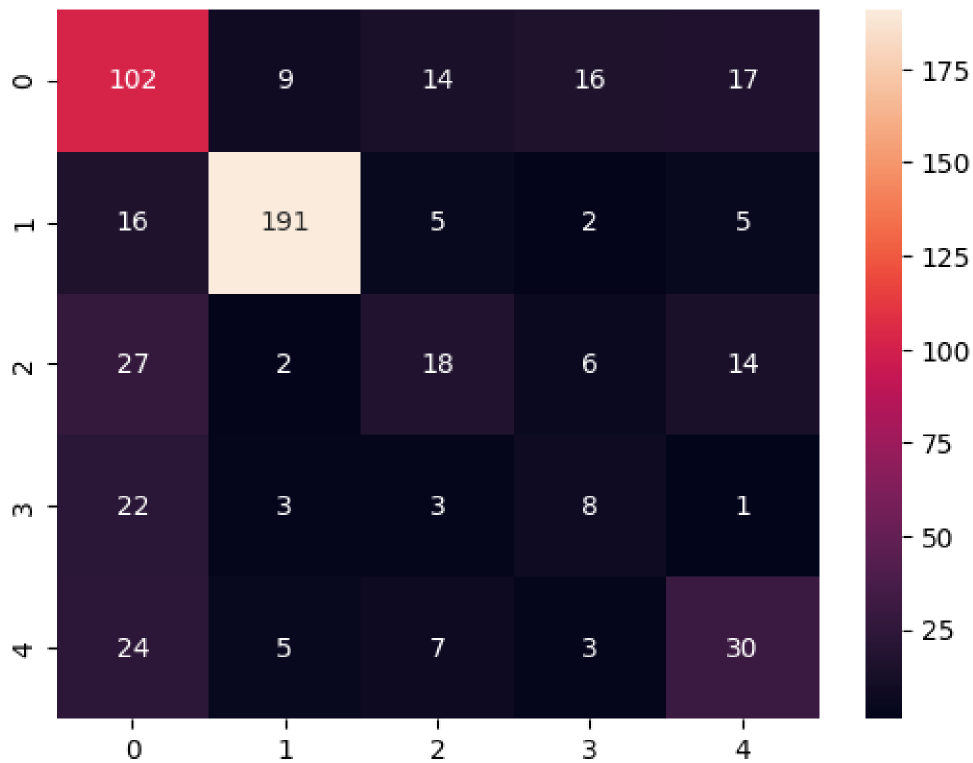


[26]:
```
y_pred1 = model1.predict(x_test)
```

```
18/18 [==============================] - 0s 7ms/step
```

[27]:
```
sns.heatmap(confusion_matrix(y_pred1.argmax(axis=1), y_test), annot=True,␣
↪fmt="g")
```

[27]: <Axes: >



CNN Model 2

[28]:
```
model2 = Sequential()

model2.add(Conv2D(32, (3,3), input_shape=(128,128,1), activation="relu"))
model2.add(MaxPool2D(2,2))
model2.add(Dropout(0.1))
model2.add(Conv2D(64, (3,3), activation="relu"))
model2.add(MaxPool2D(2,2))
model2.add(Dropout(0.2))
model2.add(Conv2D(128, (3,3), activation="relu"))
model2.add(MaxPool2D(2,2))
model2.add(Dropout(0.3))
model2.add(GlobalAveragePooling2D())
model2.add(Dense(200, activation="relu"))
model2.add(Dense(5, activation="softmax"))
```

[29]:
```
model2.summary()
```

```
Model: "sequential_1"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
```

```
 conv2d_3 (Conv2D)           (None, 126, 126, 32)      320

 max_pooling2d_3 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 dropout (Dropout)           (None, 63, 63, 32)        0

 conv2d_4 (Conv2D)           (None, 61, 61, 64)        18496

 max_pooling2d_4 (MaxPooling  (None, 30, 30, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 30, 30, 64)        0

 conv2d_5 (Conv2D)           (None, 28, 28, 128)       73856

 max_pooling2d_5 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 14, 14, 128)       0

 global_average_pooling2d (G  (None, 128)              0
 lobalAveragePooling2D)

 dense_2 (Dense)             (None, 200)               25800

 dense_3 (Dense)             (None, 5)                 1005

=================================================================
Total params: 119,477
Trainable params: 119,477
Non-trainable params: 0
_________________________________________________________________
```

[30]:

```
model2.compile(optimizer="adam", loss = "sparse_categorical_crossentropy",␣
 ↪metrics=['accuracy'])
```

[31]:

```
history2 = model2.fit(x_train, y_train, validation_data = (x_test, y_test),␣
 ↪epochs=100, class_weight = class_weights)
```

```
Epoch 1/100
69/69 [==============================] - 4s 32ms/step - loss: 0.5867 - accuracy:
0.3541 - val_loss: 1.3937 - val_accuracy: 0.3909
Epoch 2/100
69/69 [==============================] - 2s 24ms/step - loss: 0.5183 - accuracy:
0.4991 - val_loss: 1.1717 - val_accuracy: 0.6073
Epoch 3/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4919 - accuracy:
```

```
0.5727 - val_loss: 1.1529 - val_accuracy: 0.6145
Epoch 4/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4859 - accuracy:
0.5827 - val_loss: 1.1477 - val_accuracy: 0.6182
Epoch 5/100
69/69 [==============================] - 2s 25ms/step - loss: 0.4794 - accuracy:
0.5850 - val_loss: 1.1388 - val_accuracy: 0.6145
Epoch 6/100
69/69 [==============================] - 2s 22ms/step - loss: 0.4809 - accuracy:
0.5850 - val_loss: 1.1440 - val_accuracy: 0.6273
Epoch 7/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4772 - accuracy:
0.5891 - val_loss: 1.1336 - val_accuracy: 0.6236
Epoch 8/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4696 - accuracy:
0.5955 - val_loss: 1.1241 - val_accuracy: 0.6200
Epoch 9/100
69/69 [==============================] - 2s 22ms/step - loss: 0.4690 - accuracy:
0.5950 - val_loss: 1.0993 - val_accuracy: 0.6236
Epoch 10/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4641 - accuracy:
0.5936 - val_loss: 1.0959 - val_accuracy: 0.6236
Epoch 11/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4564 - accuracy:
0.5895 - val_loss: 1.1227 - val_accuracy: 0.5400
Epoch 12/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4543 - accuracy:
0.5855 - val_loss: 1.0923 - val_accuracy: 0.5727
Epoch 13/100
69/69 [==============================] - 2s 25ms/step - loss: 0.4481 - accuracy:
0.5818 - val_loss: 1.0249 - val_accuracy: 0.6455
Epoch 14/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4427 - accuracy:
0.5991 - val_loss: 1.0395 - val_accuracy: 0.6364
Epoch 15/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4414 - accuracy:
0.5823 - val_loss: 1.0439 - val_accuracy: 0.6327
Epoch 16/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4322 - accuracy:
0.5982 - val_loss: 1.0042 - val_accuracy: 0.6436
Epoch 17/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4311 - accuracy:
0.5936 - val_loss: 1.0035 - val_accuracy: 0.6255
Epoch 18/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4279 - accuracy:
0.5882 - val_loss: 1.0217 - val_accuracy: 0.6582
Epoch 19/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4211 - accuracy:
```

```
0.6227 - val_loss: 0.9689 - val_accuracy: 0.6545
Epoch 20/100
69/69 [==============================] - 2s 25ms/step - loss: 0.4209 - accuracy:
0.6282 - val_loss: 0.9937 - val_accuracy: 0.6491
Epoch 21/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4172 - accuracy:
0.6336 - val_loss: 0.9731 - val_accuracy: 0.6782
Epoch 22/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4130 - accuracy:
0.6318 - val_loss: 0.9657 - val_accuracy: 0.6709
Epoch 23/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4131 - accuracy:
0.6418 - val_loss: 0.9366 - val_accuracy: 0.6782
Epoch 24/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4122 - accuracy:
0.6391 - val_loss: 0.9562 - val_accuracy: 0.6800
Epoch 25/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4097 - accuracy:
0.6391 - val_loss: 1.0229 - val_accuracy: 0.6382
Epoch 26/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4080 - accuracy:
0.6464 - val_loss: 0.9441 - val_accuracy: 0.6800
Epoch 27/100
69/69 [==============================] - 2s 24ms/step - loss: 0.4052 - accuracy:
0.6355 - val_loss: 0.9250 - val_accuracy: 0.6873
Epoch 28/100
69/69 [==============================] - 2s 25ms/step - loss: 0.4064 - accuracy:
0.6495 - val_loss: 0.9622 - val_accuracy: 0.6818
Epoch 29/100
69/69 [==============================] - 2s 23ms/step - loss: 0.4008 - accuracy:
0.6614 - val_loss: 0.9425 - val_accuracy: 0.6818
Epoch 30/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3981 - accuracy:
0.6555 - val_loss: 0.9682 - val_accuracy: 0.6382
Epoch 31/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3963 - accuracy:
0.6545 - val_loss: 0.9239 - val_accuracy: 0.6709
Epoch 32/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3951 - accuracy:
0.6582 - val_loss: 0.9776 - val_accuracy: 0.6000
Epoch 33/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3931 - accuracy:
0.6477 - val_loss: 0.8776 - val_accuracy: 0.6982
Epoch 34/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3951 - accuracy:
0.6591 - val_loss: 0.8853 - val_accuracy: 0.6909
Epoch 35/100
69/69 [==============================] - 2s 26ms/step - loss: 0.3912 - accuracy:
```

```
0.6509 - val_loss: 0.9479 - val_accuracy: 0.6145
Epoch 36/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3871 - accuracy:
0.6605 - val_loss: 0.9247 - val_accuracy: 0.6855
Epoch 37/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3886 - accuracy:
0.6632 - val_loss: 0.9131 - val_accuracy: 0.6818
Epoch 38/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3886 - accuracy:
0.6636 - val_loss: 0.8856 - val_accuracy: 0.7055
Epoch 39/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3895 - accuracy:
0.6536 - val_loss: 0.8871 - val_accuracy: 0.6618
Epoch 40/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3887 - accuracy:
0.6536 - val_loss: 0.8955 - val_accuracy: 0.7055
Epoch 41/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3777 - accuracy:
0.6741 - val_loss: 0.8660 - val_accuracy: 0.7055
Epoch 42/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3785 - accuracy:
0.6759 - val_loss: 0.8754 - val_accuracy: 0.6982
Epoch 43/100
69/69 [==============================] - 2s 26ms/step - loss: 0.3740 - accuracy:
0.6700 - val_loss: 0.8432 - val_accuracy: 0.7018
Epoch 44/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3816 - accuracy:
0.6618 - val_loss: 0.9489 - val_accuracy: 0.5164
Epoch 45/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3739 - accuracy:
0.6645 - val_loss: 0.8758 - val_accuracy: 0.6782
Epoch 46/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3758 - accuracy:
0.6450 - val_loss: 0.9096 - val_accuracy: 0.6764
Epoch 47/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3717 - accuracy:
0.6627 - val_loss: 0.8566 - val_accuracy: 0.7055
Epoch 48/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3694 - accuracy:
0.6695 - val_loss: 0.8476 - val_accuracy: 0.7018
Epoch 49/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3671 - accuracy:
0.6668 - val_loss: 0.8401 - val_accuracy: 0.7182
Epoch 50/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3781 - accuracy:
0.6691 - val_loss: 0.8384 - val_accuracy: 0.7055
Epoch 51/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3636 - accuracy:
```

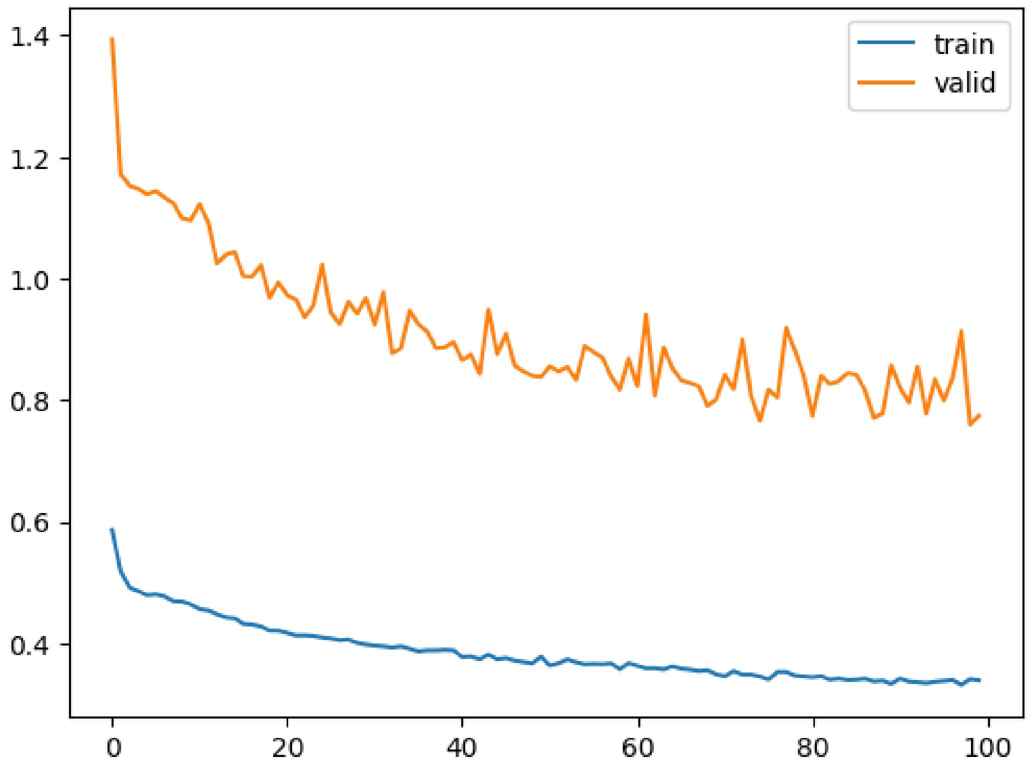
```
0.6786 - val_loss: 0.8558 - val_accuracy: 0.6964
Epoch 52/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3673 - accuracy:
0.6736 - val_loss: 0.8476 - val_accuracy: 0.7109
Epoch 53/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3737 - accuracy:
0.6659 - val_loss: 0.8547 - val_accuracy: 0.7055
Epoch 54/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3690 - accuracy:
0.6786 - val_loss: 0.8338 - val_accuracy: 0.7073
Epoch 55/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3656 - accuracy:
0.6795 - val_loss: 0.8893 - val_accuracy: 0.6436
Epoch 56/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3662 - accuracy:
0.6695 - val_loss: 0.8792 - val_accuracy: 0.6655
Epoch 57/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3657 - accuracy:
0.6627 - val_loss: 0.8704 - val_accuracy: 0.7091
Epoch 58/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3670 - accuracy:
0.6705 - val_loss: 0.8393 - val_accuracy: 0.6945
Epoch 59/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3575 - accuracy:
0.6814 - val_loss: 0.8171 - val_accuracy: 0.7164
Epoch 60/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3676 - accuracy:
0.6691 - val_loss: 0.8691 - val_accuracy: 0.6782
Epoch 61/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3630 - accuracy:
0.6782 - val_loss: 0.8233 - val_accuracy: 0.7091
Epoch 62/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3590 - accuracy:
0.6714 - val_loss: 0.9408 - val_accuracy: 0.6636
Epoch 63/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3592 - accuracy:
0.6777 - val_loss: 0.8071 - val_accuracy: 0.7145
Epoch 64/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3571 - accuracy:
0.6805 - val_loss: 0.8869 - val_accuracy: 0.6636
Epoch 65/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3624 - accuracy:
0.6577 - val_loss: 0.8528 - val_accuracy: 0.7073
Epoch 66/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3583 - accuracy:
0.6786 - val_loss: 0.8330 - val_accuracy: 0.6927
Epoch 67/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3569 - accuracy:
```

```
0.6818 - val_loss: 0.8285 - val_accuracy: 0.6873
Epoch 68/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3545 - accuracy:
0.6709 - val_loss: 0.8232 - val_accuracy: 0.6964
Epoch 69/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3558 - accuracy:
0.6591 - val_loss: 0.7910 - val_accuracy: 0.7182
Epoch 70/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3490 - accuracy:
0.6632 - val_loss: 0.8010 - val_accuracy: 0.7182
Epoch 71/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3459 - accuracy:
0.6786 - val_loss: 0.8419 - val_accuracy: 0.6855
Epoch 72/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3542 - accuracy:
0.6723 - val_loss: 0.8184 - val_accuracy: 0.6982
Epoch 73/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3486 - accuracy:
0.6736 - val_loss: 0.8997 - val_accuracy: 0.6800
Epoch 74/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3486 - accuracy:
0.6718 - val_loss: 0.8065 - val_accuracy: 0.7164
Epoch 75/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3455 - accuracy:
0.6818 - val_loss: 0.7663 - val_accuracy: 0.7255
Epoch 76/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3406 - accuracy:
0.6805 - val_loss: 0.8173 - val_accuracy: 0.6764
Epoch 77/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3531 - accuracy:
0.6709 - val_loss: 0.8040 - val_accuracy: 0.7182
Epoch 78/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3529 - accuracy:
0.6636 - val_loss: 0.9190 - val_accuracy: 0.6436
Epoch 79/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3470 - accuracy:
0.6773 - val_loss: 0.8818 - val_accuracy: 0.6164
Epoch 80/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3455 - accuracy:
0.6809 - val_loss: 0.8411 - val_accuracy: 0.6545
Epoch 81/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3443 - accuracy:
0.6700 - val_loss: 0.7741 - val_accuracy: 0.7182
Epoch 82/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3459 - accuracy:
0.6723 - val_loss: 0.8398 - val_accuracy: 0.6782
Epoch 83/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3402 - accuracy:
```

```
0.6764 - val_loss: 0.8269 - val_accuracy: 0.6527
Epoch 84/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3422 - accuracy:
0.6691 - val_loss: 0.8317 - val_accuracy: 0.6655
Epoch 85/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3397 - accuracy:
0.6773 - val_loss: 0.8447 - val_accuracy: 0.6618
Epoch 86/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3401 - accuracy:
0.6768 - val_loss: 0.8422 - val_accuracy: 0.6600
Epoch 87/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3419 - accuracy:
0.6736 - val_loss: 0.8168 - val_accuracy: 0.6636
Epoch 88/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3374 - accuracy:
0.6718 - val_loss: 0.7711 - val_accuracy: 0.7236
Epoch 89/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3386 - accuracy:
0.6736 - val_loss: 0.7781 - val_accuracy: 0.7164
Epoch 90/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3333 - accuracy:
0.6764 - val_loss: 0.8571 - val_accuracy: 0.6218
Epoch 91/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3421 - accuracy:
0.6686 - val_loss: 0.8212 - val_accuracy: 0.6745
Epoch 92/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3371 - accuracy:
0.6759 - val_loss: 0.7962 - val_accuracy: 0.6964
Epoch 93/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3361 - accuracy:
0.6773 - val_loss: 0.8550 - val_accuracy: 0.6745
Epoch 94/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3346 - accuracy:
0.6886 - val_loss: 0.7779 - val_accuracy: 0.7036
Epoch 95/100
69/69 [==============================] - 2s 25ms/step - loss: 0.3368 - accuracy:
0.6705 - val_loss: 0.8349 - val_accuracy: 0.7000
Epoch 96/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3381 - accuracy:
0.6736 - val_loss: 0.7997 - val_accuracy: 0.6982
Epoch 97/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3398 - accuracy:
0.6750 - val_loss: 0.8357 - val_accuracy: 0.6909
Epoch 98/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3317 - accuracy:
0.6855 - val_loss: 0.9139 - val_accuracy: 0.5636
Epoch 99/100
69/69 [==============================] - 2s 23ms/step - loss: 0.3410 - accuracy:
```

```
0.6600 - val_loss: 0.7597 - val_accuracy: 0.7273
Epoch 100/100
69/69 [==============================] - 2s 24ms/step - loss: 0.3392 - accuracy:
0.6700 - val_loss: 0.7746 - val_accuracy: 0.7236
```

[32]:
```
plt.plot(history2.history["loss"])
plt.plot(history2.history["val_loss"])
plt.legend(["train", "valid"])
plt.show()
```

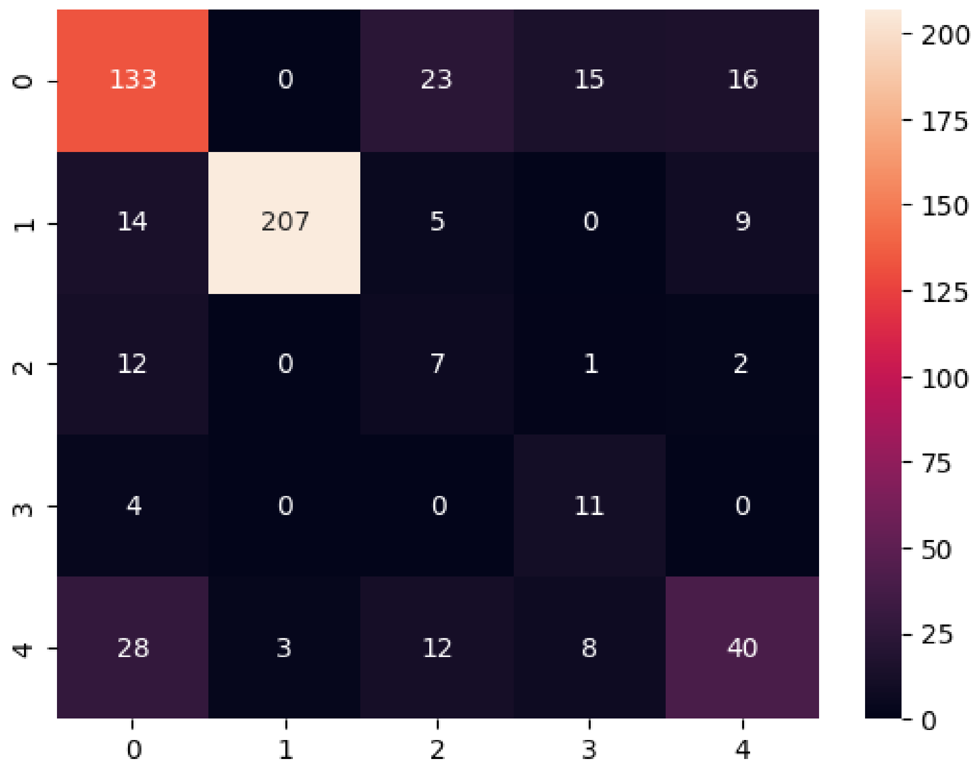


[33]:
```
y_pred2 = model2.predict(x_test)
```

```
18/18 [==============================] - 0s 5ms/step
```

[34]:
```
sns.heatmap(confusion_matrix(y_pred2.argmax(axis=1), y_test), annot=True,␣
 ↪fmt="g")
```

[34]: <Axes: >



CNN Model 3

[35]:
```
model3 = Sequential()

model3.add(Conv2D(32, (3,3), input_shape=(128,128,1), activation="relu"))
model3.add(Conv2D(32, (3,3), activation="relu"))
model3.add(MaxPool2D(2,2))

model3.add(Conv2D(64, (3,3), input_shape=(128,128,1), activation="relu"))
model3.add(Conv2D(64, (3,3), activation="relu"))
model3.add(MaxPool2D(2,2))
model3.add(Dropout(0.1))

model3.add(Conv2D(128, (3,3), input_shape=(128,128,1), activation="relu"))
model3.add(MaxPool2D(2,2))
model3.add(Dropout(0.2))

model3.add(GlobalAveragePooling2D())
model3.add(Dense(128, activation="relu"))
model3.add(Dense(5, activation="softmax"))
```

[36]:
```
model3.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 126, 126, 32)      320

 conv2d_7 (Conv2D)           (None, 124, 124, 32)      9248

 max_pooling2d_6 (MaxPooling  (None, 62, 62, 32)       0
 2D)

 conv2d_8 (Conv2D)           (None, 60, 60, 64)        18496

 conv2d_9 (Conv2D)           (None, 58, 58, 64)        36928

 max_pooling2d_7 (MaxPooling  (None, 29, 29, 64)       0
 2D)

 dropout_3 (Dropout)         (None, 29, 29, 64)        0

 conv2d_10 (Conv2D)          (None, 27, 27, 128)       73856

 max_pooling2d_8 (MaxPooling  (None, 13, 13, 128)      0
 2D)

 dropout_4 (Dropout)         (None, 13, 13, 128)       0

 global_average_pooling2d_1   (None, 128)              0
 (GlobalAveragePooling2D)

 dense_4 (Dense)             (None, 128)               16512

 dense_5 (Dense)             (None, 5)                 645

=================================================================
Total params: 156,005
Trainable params: 156,005
Non-trainable params: 0
_________________________________________________________________
```

[37]:
```python
model3.compile(optimizer="adam", loss="sparse_categorical_crossentropy",␣
 ↪metrics=['accuracy'])
```

[38]:
```python
history3 = model3.fit(x_train, y_train, validation_data = (x_test, y_test),␣
 ↪epochs=100, class_weight = class_weights)
```

```
Epoch 1/100
69/69 [==============================] - 9s 61ms/step - loss: 0.5841 - accuracy:
```

```
0.3582 - val_loss: 1.3725 - val_accuracy: 0.4473
Epoch 2/100
69/69 [==============================] - 3s 44ms/step - loss: 0.5167 - accuracy:
0.5405 - val_loss: 1.2058 - val_accuracy: 0.6055
Epoch 3/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4913 - accuracy:
0.5795 - val_loss: 1.1805 - val_accuracy: 0.6109
Epoch 4/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4845 - accuracy:
0.5845 - val_loss: 1.2057 - val_accuracy: 0.6127
Epoch 5/100
69/69 [==============================] - 3s 43ms/step - loss: 0.4795 - accuracy:
0.5868 - val_loss: 1.1459 - val_accuracy: 0.6182
Epoch 6/100
69/69 [==============================] - 3s 43ms/step - loss: 0.4722 - accuracy:
0.5986 - val_loss: 1.1063 - val_accuracy: 0.6236
Epoch 7/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4719 - accuracy:
0.5968 - val_loss: 1.1361 - val_accuracy: 0.6145
Epoch 8/100
69/69 [==============================] - 3s 47ms/step - loss: 0.4663 - accuracy:
0.6014 - val_loss: 1.1000 - val_accuracy: 0.6218
Epoch 9/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4680 - accuracy:
0.5991 - val_loss: 1.0907 - val_accuracy: 0.6200
Epoch 10/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4641 - accuracy:
0.5991 - val_loss: 1.1525 - val_accuracy: 0.6255
Epoch 11/100
69/69 [==============================] - 3s 47ms/step - loss: 0.4657 - accuracy:
0.6036 - val_loss: 1.1078 - val_accuracy: 0.6273
Epoch 12/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4579 - accuracy:
0.6086 - val_loss: 1.1162 - val_accuracy: 0.6309
Epoch 13/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4450 - accuracy:
0.5905 - val_loss: 1.0315 - val_accuracy: 0.6345
Epoch 14/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4373 - accuracy:
0.6168 - val_loss: 1.0360 - val_accuracy: 0.6364
Epoch 15/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4339 - accuracy:
0.5945 - val_loss: 1.0724 - val_accuracy: 0.6055
Epoch 16/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4364 - accuracy:
0.6150 - val_loss: 0.9949 - val_accuracy: 0.6455
Epoch 17/100
69/69 [==============================] - 3s 43ms/step - loss: 0.4284 - accuracy:
```

```
0.6336 - val_loss: 0.9812 - val_accuracy: 0.6491
Epoch 18/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4239 - accuracy:
0.6295 - val_loss: 1.0071 - val_accuracy: 0.6564
Epoch 19/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4228 - accuracy:
0.6414 - val_loss: 0.9885 - val_accuracy: 0.6436
Epoch 20/100
69/69 [==============================] - 4s 52ms/step - loss: 0.4171 - accuracy:
0.6405 - val_loss: 0.9889 - val_accuracy: 0.6455
Epoch 21/100
69/69 [==============================] - 3s 47ms/step - loss: 0.4145 - accuracy:
0.6332 - val_loss: 0.9827 - val_accuracy: 0.6582
Epoch 22/100
69/69 [==============================] - 3s 44ms/step - loss: 0.4146 - accuracy:
0.6468 - val_loss: 0.9794 - val_accuracy: 0.6673
Epoch 23/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4133 - accuracy:
0.6391 - val_loss: 0.9888 - val_accuracy: 0.6455
Epoch 24/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4125 - accuracy:
0.6409 - val_loss: 0.9996 - val_accuracy: 0.6582
Epoch 25/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4056 - accuracy:
0.6573 - val_loss: 0.9844 - val_accuracy: 0.6582
Epoch 26/100
69/69 [==============================] - 3s 46ms/step - loss: 0.4072 - accuracy:
0.6495 - val_loss: 0.9479 - val_accuracy: 0.6764
Epoch 27/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3998 - accuracy:
0.6568 - val_loss: 0.9304 - val_accuracy: 0.6782
Epoch 28/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3971 - accuracy:
0.6605 - val_loss: 0.9354 - val_accuracy: 0.6818
Epoch 29/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3938 - accuracy:
0.6541 - val_loss: 0.9127 - val_accuracy: 0.6945
Epoch 30/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3943 - accuracy:
0.6314 - val_loss: 0.9464 - val_accuracy: 0.6691
Epoch 31/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3861 - accuracy:
0.6673 - val_loss: 0.9130 - val_accuracy: 0.6909
Epoch 32/100
69/69 [==============================] - 3s 45ms/step - loss: 0.3840 - accuracy:
0.6564 - val_loss: 0.8670 - val_accuracy: 0.6909
Epoch 33/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3787 - accuracy:
```

```
0.6736 - val_loss: 0.9325 - val_accuracy: 0.6291
Epoch 34/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3750 - accuracy:
0.6805 - val_loss: 0.9037 - val_accuracy: 0.6855
Epoch 35/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3772 - accuracy:
0.6764 - val_loss: 0.9026 - val_accuracy: 0.6836
Epoch 36/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3744 - accuracy:
0.6645 - val_loss: 0.8940 - val_accuracy: 0.6818
Epoch 37/100
69/69 [==============================] - 3s 43ms/step - loss: 0.3615 - accuracy:
0.6773 - val_loss: 0.8376 - val_accuracy: 0.6964
Epoch 38/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3617 - accuracy:
0.6827 - val_loss: 0.8589 - val_accuracy: 0.6691
Epoch 39/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3624 - accuracy:
0.6859 - val_loss: 0.8969 - val_accuracy: 0.6364
Epoch 40/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3549 - accuracy:
0.6773 - val_loss: 0.8247 - val_accuracy: 0.7018
Epoch 41/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3542 - accuracy:
0.6864 - val_loss: 0.8179 - val_accuracy: 0.7164
Epoch 42/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3531 - accuracy:
0.6736 - val_loss: 0.8065 - val_accuracy: 0.7164
Epoch 43/100
69/69 [==============================] - 3s 48ms/step - loss: 0.3446 - accuracy:
0.6791 - val_loss: 0.8354 - val_accuracy: 0.7000
Epoch 44/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3541 - accuracy:
0.6964 - val_loss: 0.8354 - val_accuracy: 0.6945
Epoch 45/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3469 - accuracy:
0.6727 - val_loss: 0.8579 - val_accuracy: 0.6709
Epoch 46/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3468 - accuracy:
0.7009 - val_loss: 0.8703 - val_accuracy: 0.6927
Epoch 47/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3449 - accuracy:
0.6823 - val_loss: 0.8604 - val_accuracy: 0.6782
Epoch 48/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3359 - accuracy:
0.6823 - val_loss: 0.8382 - val_accuracy: 0.6927
Epoch 49/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3308 - accuracy:
```

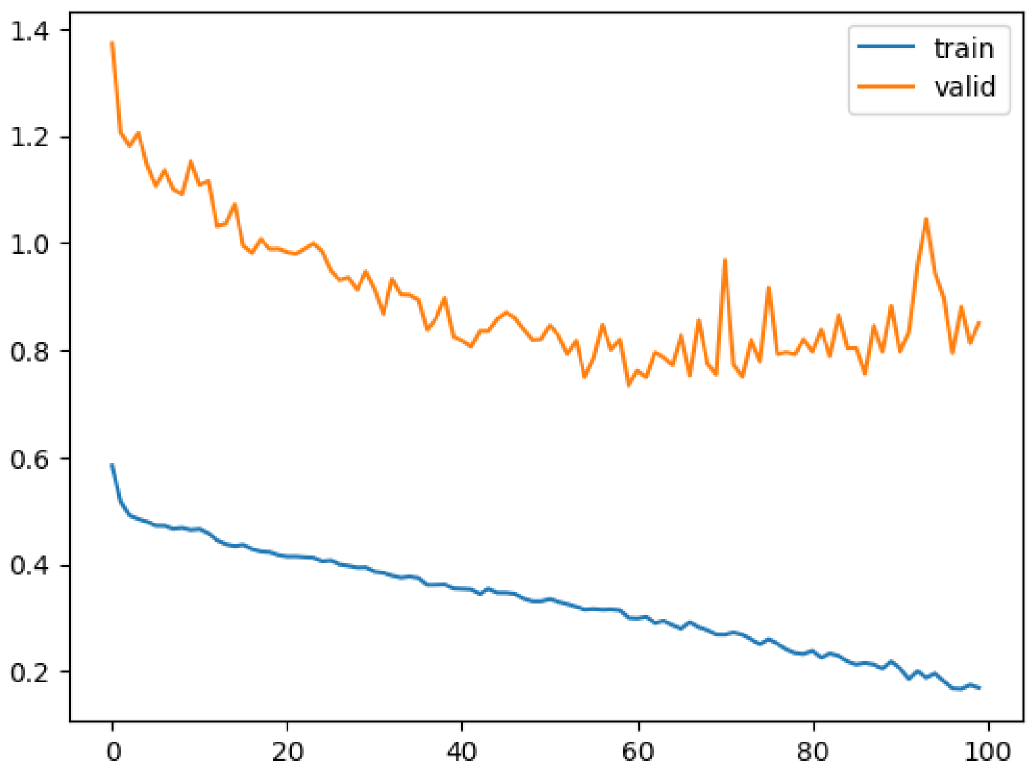
```
0.6941 - val_loss: 0.8187 - val_accuracy: 0.6818
Epoch 50/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3308 - accuracy:
0.6900 - val_loss: 0.8200 - val_accuracy: 0.6891
Epoch 51/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3350 - accuracy:
0.6905 - val_loss: 0.8460 - val_accuracy: 0.6745
Epoch 52/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3299 - accuracy:
0.6795 - val_loss: 0.8271 - val_accuracy: 0.7127
Epoch 53/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3256 - accuracy:
0.7132 - val_loss: 0.7925 - val_accuracy: 0.7073
Epoch 54/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3206 - accuracy:
0.7091 - val_loss: 0.8174 - val_accuracy: 0.6782
Epoch 55/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3153 - accuracy:
0.6959 - val_loss: 0.7493 - val_accuracy: 0.7418
Epoch 56/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3166 - accuracy:
0.7050 - val_loss: 0.7849 - val_accuracy: 0.7036
Epoch 57/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3152 - accuracy:
0.7086 - val_loss: 0.8470 - val_accuracy: 0.5945
Epoch 58/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3158 - accuracy:
0.7159 - val_loss: 0.8009 - val_accuracy: 0.6818
Epoch 59/100
69/69 [==============================] - 3s 46ms/step - loss: 0.3141 - accuracy:
0.7055 - val_loss: 0.8195 - val_accuracy: 0.6564
Epoch 60/100
69/69 [==============================] - 3s 44ms/step - loss: 0.3000 - accuracy:
0.7123 - val_loss: 0.7337 - val_accuracy: 0.7327
Epoch 61/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2989 - accuracy:
0.7027 - val_loss: 0.7618 - val_accuracy: 0.7382
Epoch 62/100
69/69 [==============================] - 3s 47ms/step - loss: 0.3022 - accuracy:
0.7159 - val_loss: 0.7492 - val_accuracy: 0.7200
Epoch 63/100
69/69 [==============================] - 3s 48ms/step - loss: 0.2899 - accuracy:
0.7309 - val_loss: 0.7959 - val_accuracy: 0.7145
Epoch 64/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2945 - accuracy:
0.7186 - val_loss: 0.7861 - val_accuracy: 0.6655
Epoch 65/100
69/69 [==============================] - 3s 47ms/step - loss: 0.2863 - accuracy:
```

```
0.7300 - val_loss: 0.7721 - val_accuracy: 0.6927
Epoch 66/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2796 - accuracy:
0.7405 - val_loss: 0.8275 - val_accuracy: 0.6345
Epoch 67/100
69/69 [==============================] - 3s 47ms/step - loss: 0.2920 - accuracy:
0.7182 - val_loss: 0.7523 - val_accuracy: 0.7109
Epoch 68/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2823 - accuracy:
0.7295 - val_loss: 0.8558 - val_accuracy: 0.6509
Epoch 69/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2772 - accuracy:
0.7355 - val_loss: 0.7749 - val_accuracy: 0.6873
Epoch 70/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2690 - accuracy:
0.7505 - val_loss: 0.7544 - val_accuracy: 0.7073
Epoch 71/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2688 - accuracy:
0.7382 - val_loss: 0.9678 - val_accuracy: 0.5855
Epoch 72/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2728 - accuracy:
0.7495 - val_loss: 0.7729 - val_accuracy: 0.7018
Epoch 73/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2686 - accuracy:
0.7409 - val_loss: 0.7501 - val_accuracy: 0.7109
Epoch 74/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2596 - accuracy:
0.7600 - val_loss: 0.8187 - val_accuracy: 0.6691
Epoch 75/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2506 - accuracy:
0.7641 - val_loss: 0.7789 - val_accuracy: 0.7036
Epoch 76/100
69/69 [==============================] - 3s 47ms/step - loss: 0.2599 - accuracy:
0.7527 - val_loss: 0.9163 - val_accuracy: 0.5891
Epoch 77/100
69/69 [==============================] - 3s 45ms/step - loss: 0.2515 - accuracy:
0.7623 - val_loss: 0.7920 - val_accuracy: 0.7000
Epoch 78/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2412 - accuracy:
0.7714 - val_loss: 0.7958 - val_accuracy: 0.7218
Epoch 79/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2343 - accuracy:
0.7700 - val_loss: 0.7920 - val_accuracy: 0.7127
Epoch 80/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2326 - accuracy:
0.7727 - val_loss: 0.8201 - val_accuracy: 0.6673
Epoch 81/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2384 - accuracy:
```

```
0.7800 - val_loss: 0.7967 - val_accuracy: 0.7364
Epoch 82/100
69/69 [==============================] - 3s 47ms/step - loss: 0.2257 - accuracy:
0.7855 - val_loss: 0.8379 - val_accuracy: 0.6691
Epoch 83/100
69/69 [==============================] - 3s 48ms/step - loss: 0.2335 - accuracy:
0.7809 - val_loss: 0.7884 - val_accuracy: 0.6873
Epoch 84/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2290 - accuracy:
0.7864 - val_loss: 0.8651 - val_accuracy: 0.6382
Epoch 85/100
69/69 [==============================] - 3s 45ms/step - loss: 0.2184 - accuracy:
0.7941 - val_loss: 0.8034 - val_accuracy: 0.7164
Epoch 86/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2129 - accuracy:
0.7977 - val_loss: 0.8040 - val_accuracy: 0.7091
Epoch 87/100
69/69 [==============================] - 3s 45ms/step - loss: 0.2158 - accuracy:
0.7891 - val_loss: 0.7558 - val_accuracy: 0.6927
Epoch 88/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2127 - accuracy:
0.8082 - val_loss: 0.8454 - val_accuracy: 0.7073
Epoch 89/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2053 - accuracy:
0.8018 - val_loss: 0.7967 - val_accuracy: 0.7000
Epoch 90/100
69/69 [==============================] - 3s 46ms/step - loss: 0.2184 - accuracy:
0.7827 - val_loss: 0.8825 - val_accuracy: 0.7382
Epoch 91/100
69/69 [==============================] - 3s 48ms/step - loss: 0.2058 - accuracy:
0.8041 - val_loss: 0.7966 - val_accuracy: 0.7309
Epoch 92/100
69/69 [==============================] - 3s 47ms/step - loss: 0.1858 - accuracy:
0.8223 - val_loss: 0.8332 - val_accuracy: 0.7145
Epoch 93/100
69/69 [==============================] - 3s 44ms/step - loss: 0.2003 - accuracy:
0.8168 - val_loss: 0.9592 - val_accuracy: 0.6218
Epoch 94/100
69/69 [==============================] - 3s 46ms/step - loss: 0.1883 - accuracy:
0.8177 - val_loss: 1.0451 - val_accuracy: 0.5945
Epoch 95/100
69/69 [==============================] - 3s 45ms/step - loss: 0.1962 - accuracy:
0.8136 - val_loss: 0.9445 - val_accuracy: 0.6200
Epoch 96/100
69/69 [==============================] - 3s 46ms/step - loss: 0.1814 - accuracy:
0.8232 - val_loss: 0.8970 - val_accuracy: 0.7018
Epoch 97/100
69/69 [==============================] - 3s 46ms/step - loss: 0.1684 - accuracy:
```

```
0.8464 - val_loss: 0.7943 - val_accuracy: 0.7255
Epoch 98/100
69/69 [==============================] - 3s 47ms/step - loss: 0.1673 - accuracy:
0.8514 - val_loss: 0.8808 - val_accuracy: 0.7291
Epoch 99/100
69/69 [==============================] - 3s 47ms/step - loss: 0.1749 - accuracy:
0.8336 - val_loss: 0.8128 - val_accuracy: 0.7364
Epoch 100/100
69/69 [==============================] - 3s 47ms/step - loss: 0.1695 - accuracy:
0.8368 - val_loss: 0.8504 - val_accuracy: 0.7327
```

[39]:
```
plt.plot(history3.history["loss"])
plt.plot(history3.history["val_loss"])
plt.legend(["train", "valid"])
plt.show()
```

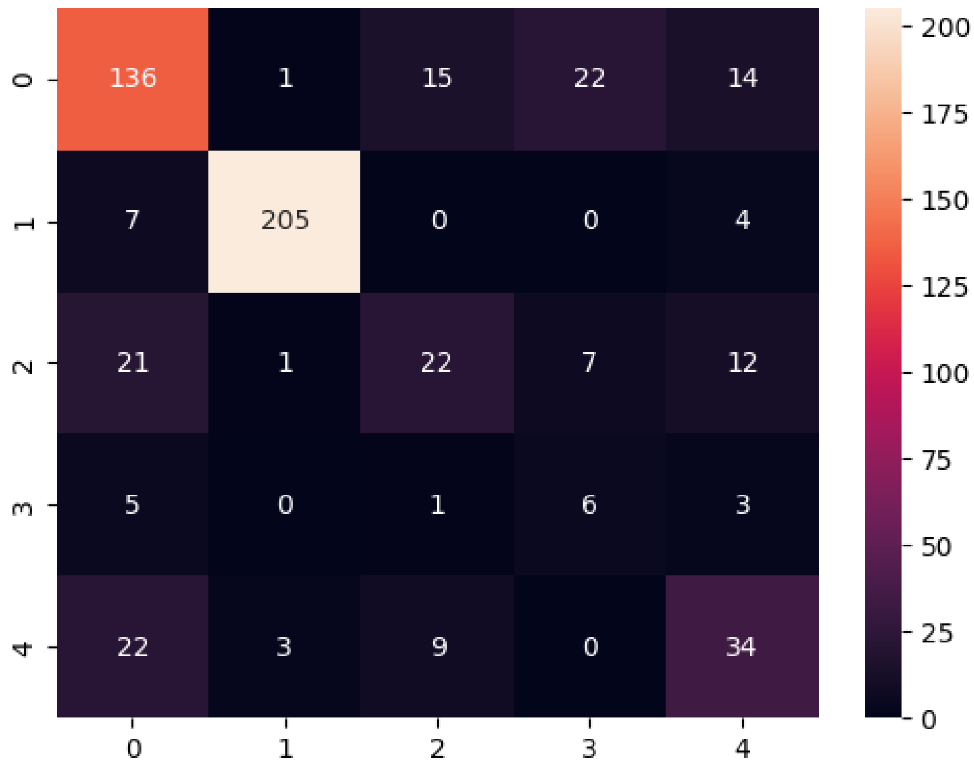


[40]:
```
y_pred3 = model3.predict(x_test)
```

```
18/18 [==============================] - 0s 11ms/step
```

[41]:
```
sns.heatmap(confusion_matrix(y_pred3.argmax(axis=1), y_test), annot=True,␣
↪fmt="g")
```

[41]: <Axes: >



Ensemble Prediction

[42]:
```
model = [model1, model2, model3]            # creating list of the 3 cnn models
preds = [model.predict(x_test) for model in model]
```

```
18/18 [==============================] - 0s 7ms/step
18/18 [==============================] - 0s 5ms/step
18/18 [==============================] - 0s 9ms/step
```

[43]:
```
preds = np.array(preds)       # getting the prediction
```

[44]:
```
preds.shape
```

[44]: (3, 550, 5)

[45]:
```
preds = np.sum(preds, axis=0)        # summing up the prediction of the 3 models
```

[46]:
```
preds.shape
```

[46]: (550, 5)

[47]:
```
ensemble_result = np.argmax(preds, axis=1)      # ensemble prediction
```

[48]:
```
print(f"accuracy of model 1 {accuracy_score(y_pred1.argmax(axis=1), y_test)}")
print(f"accuracy of model 2 {accuracy_score(y_pred2.argmax(axis=1), y_test)}")
print(f"accuracy of model 3 {accuracy_score(y_pred3.argmax(axis=1), y_test)}")
print(f"ensemble accuracy {accuracy_score(ensemble_result, y_test)}")
```

```
accuracy of model 1 0.6345454545454545
accuracy of model 2 0.7236363636363636
accuracy of model 3 0.7327272727272728
ensemble accuracy 0.7254545454545455
```

Predicting with giving weights to the models

[49]:
```
models = [model1, model2, model3]
preds = [model.predict(x_test) for model in models]
```

```
18/18 [==============================] - 0s 8ms/step
18/18 [==============================] - 0s 6ms/step
18/18 [==============================] - 0s 8ms/step
```

[50]:
```
preds = np.array(preds)
```

[51]:
```
weights = [0.6, 0.6, 0.3]      # giving weights to the models , weights can be␣
 ↪assigned according to the importance of the model
```

[52]:
```
weighted_preds = np.tensordot(preds, weights, axes=((0), (0)))
```

[53]:
```
weighted_preds.shape
```

[53]: (550, 5)

[54]:
```
weighted_ensemble_prediction = np.argmax(weighted_preds, axis=1)
```

[55]:
```
print(f"the accuracy of weighted ensemble prediction is␣
 ↪{accuracy_score(weighted_ensemble_prediction, y_test)}")
```

```
the accuracy of weighted ensemble prediction is 0.7036363636363636
```

Grid Search for Weighted Ensemble Prediction

[56]:
```
models = [model1, model2, model3]
preds = [model.predict(x_test) for model in models]
preds = np.array(preds)
```

```
18/18 [==============================] - 0s 8ms/step
18/18 [==============================] - 0s 5ms/step
18/18 [==============================] - 0s 8ms/step
```

[57]:
```
weights1 = []
weights2 = []
weights3 = []
acc = []
for w1 in range(0, 5):
  for w2 in range(0, 5):
    for w3 in range(0, 5):
      wts = [w1/10., w2/10., w3/10.]
      wt_pred = np.tensordot(preds, wts, axes=((0), (0)))
      wtd_ensemble_pred = np.argmax(wt_pred, axis=1)
      weighted_accuracy = accuracy_score(wtd_ensemble_pred, y_test)
      weights1.append(wts[0])
      weights2.append(wts[1])
      weights3.append(wts[2])
      acc.append(weighted_accuracy)
```

[58]:
```
df = pd.DataFrame({"w1" : weights1, "w2" : weights2, "w3" : weights3, "acc" :␣
↪acc})
```

[59]:
```
df.head()
```

[59]:

	w1	w2	w3	acc
0	0.0	0.0	0.0	0.347273
1	0.0	0.0	0.1	0.732727
2	0.0	0.0	0.2	0.732727
3	0.0	0.0	0.3	0.732727
4	0.0	0.0	0.4	0.732727

[60]:
```
max_acc_row = df.iloc[df["acc"].idxmax()]
```

[61]:
```
print(f"the highest ensemble accuracy is {max_acc_row[3]}")
```

```
the highest ensemble accuracy is 0.7527272727272727
```

Random Forest Classifier

[62]:
```
rf_clf = RandomForestClassifier()
```

[63]:
```
x_train_reshaped = x_train.reshape(x_train.shape[0],-1)
x_test_reshaped = x_test.reshape(x_test.shape[0], -1)
```

[64]:
```
print(f"shape of x_train_reshaped {x_train_reshaped.shape}")
print(f"shape of x_test_reshaped {x_test_reshaped.shape}")
```

```
shape of x_train_reshaped (2200, 16384)
shape of x_test_reshaped (550, 16384)
```

[65]:
```
rf_clf.fit(x_train_reshaped, y_train)
```

```
[65]: RandomForestClassifier()
```

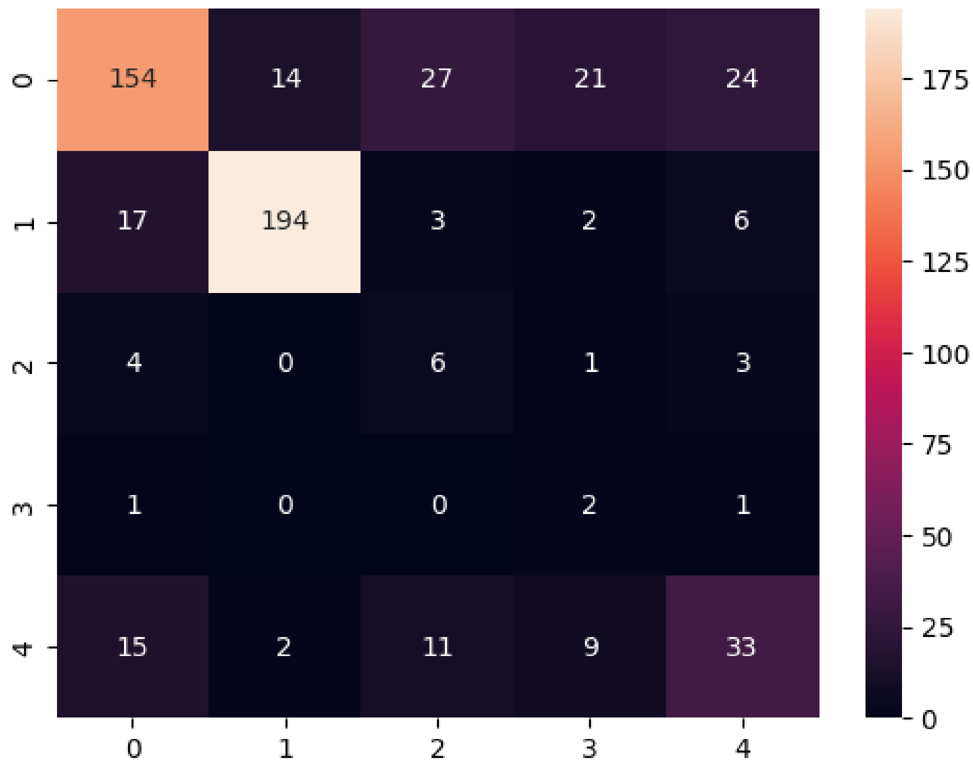
```
[66]: y_pred_rf = rf_clf.predict(x_test_reshaped)
```

```
[67]: accuracy_score(y_pred_rf, y_test)
```

```
[67]: 0.7072727272727273
```

```
[68]: sns.heatmap(confusion_matrix(y_pred_rf, y_test), annot=True, fmt="g")
```

```
[68]: <Axes: >
```



XGBoost Classifier

```
[69]: xgb_clf = xgb.XGBClassifier()
```

```
[70]: xgb_clf.fit(x_train_reshaped, y_train)
```

```
[70]: XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
```

```
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, …)
```

[71]:
```
y_pred_xgb = xgb_clf.predict(x_test_reshaped)
```

[72]:
```
accuracy_score(y_pred_xgb, y_test)
```

[72]: 0.6945454545454546

[73]:
```
sns.heatmap(confusion_matrix(y_pred_xgb, y_test), annot=True, fmt="g")
```

[73]: <Axes: >