NO IMAGE GENERATOR

model= Sequential()

model.add(Convolution2D(32, (4, 4), activation='relu', input\_shape=(64, 64, 1), kernel\_regularizer=l2(0.01)))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(64, (4, 4), activation='relu'))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(MaxPool2D((2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(16, activation='softmax'))

from tensorflow.keras import optimizers

#model.compile(loss='binary\_crossentropy', optimizer=optimizers.RMSprop(lr=1e-4),metrics=["accuracy"])

model.compile(loss='binary\_crossentropy', optimizer = 'adam', metrics=["accuracy"])

# Starting the model training

history = model.fit\_generator(train\_generator,

                    steps\_per\_epoch=5,

                    epochs=75,

                    validation\_data=test\_generator,

                    validation\_steps=4)

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Wykres

Opis wygenerowany automatycznie

train\_datagen = ImageDataGenerator(

                                   rotation\_range = 40,

                                   shear\_range = 0.2,

                                   brightness\_range = (0.2,1.5),

                                   zoom\_range = 0.2,

                                   horizontal\_flip = True)

model= Sequential()

model.add(Convolution2D(32, (4, 4), activation='relu', input\_shape=(64, 64, 1), kernel\_regularizer=l2(0.01)))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(64, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(16, activation='softmax'))

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

model= Sequential()

model.add(Convolution2D(32, (4, 4), activation='relu', input\_shape=(64, 64, 1), kernel\_regularizer=l2(0.01)))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(256, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(256, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(16, activation='softmax'))

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Wykres

Opis wygenerowany automatycznie

model= Sequential()

model.add(Convolution2D(32, (4, 4), activation='relu', input\_shape=(64, 64, 1), kernel\_regularizer=l2(0.01)))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(64, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(16, activation='softmax'))

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, wyświetlacz

Opis wygenerowany automatycznie

model= Sequential()

model.add(Convolution2D(32, (4, 4), activation='relu', input\_shape=(64, 64, 1), kernel\_regularizer=l2(0.01)))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(64, (4, 4), activation='relu'))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(MaxPool2D((2, 2)))

model.add(Convolution2D(128, (4, 4), activation='relu'))

model.add(Dropout(0.5))

model.add(MaxPool2D((2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dense(16, activation='softmax'))

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznieCATEGORICAL CROSSENTRTOPY

IMAGE GENERATOR   
  
  
  
Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Czcionka

Opis wygenerowany automatycznie