1

model.add(Conv2D(32, (3,3), activation='relu', input\_shape = (256,256,3)))

model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64, (3,3), activation='relu'))

model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64, (3,3), activation='relu'))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, diagram

Opis wygenerowany automatycznie

2

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(256, 256, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

# Flatten the output before the dense layers

model.add(Flatten())

# Add dense layers

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5)) # Add dropout for regularization

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5)) # Add dropout for regularization

# Output layer

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, Wykres, zrzut ekranu, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

3

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(256, 256, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

# Flatten the output before the dense layers

model.add(Flatten())

# Add dense layers

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5)) # Add dropout for regularization

model.add(Dense(64, activation='relu'))

# Output layer

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

4

Batch\_size =4

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(256, 256, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

# Flatten the output before the dense layers

model.add(Flatten())

# Add dense layers

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5)) # Add dropout for regularization

model.add(Dense(64, activation='relu'))

# Output layer

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, zrzut ekranu, Wykres, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, Wykres, zrzut ekranu, diagram

Opis wygenerowany automatycznie

5

Batch\_size =4

model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=(256, 256, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

# Flatten the output before the dense layers

model.add(Flatten())

# Add dense layers

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5)) # Add dropout for regularization

model.add(Dense(64, activation='relu'))

# Output layer

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

5

Batch\_size =32

model.add(Conv2D(32, (3,3), activation='relu', input\_shape = (256,256,3)))

model.add(MaxPooling2D((2,2)))

model.add(Conv2D(64, (3,3), activation='relu'))

model.add(MaxPooling2D((2,2)))

model.add(Conv2D(128, (3,3), activation='relu'))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.3))

model.add(Dense(32, activation='relu'))

model.add(Dense(10, activation='softmax'))

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

Obraz zawierający tekst, Wykres, zrzut ekranu, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

6

Batch\_size =64

model = models.Sequential()  
model.add(Conv2D(32, (3,3), activation='relu', input\_shape = (256,256,3)))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(64, (3,3), activation='relu'))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(128, (3,3), activation='relu'))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(128, (3,3), activation='relu'))  
model.add(Flatten())  
model.add(Dense(128, activation='relu'))  
model.add(Dropout(0.3))  
model.add(Dense(64, activation='relu'))  
model.add(Dense(10, activation='softmax'))  
model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

7

Batch\_size =32

model = models.Sequential()  
model.add(Conv2D(32, (3,3), activation='relu', input\_shape = (256,256,3)))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(64, (3,3), activation='relu'))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(128, (3,3), activation='relu'))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(128, (3,3), activation='relu'))  
model.add(MaxPooling2D((2,2)))  
model.add(Conv2D(256, (3,3), activation='relu'))  
model.add(Flatten())  
model.add(Dense(128, activation='relu'))  
model.add(Dropout(0.4))  
model.add(Dense(64, activation='relu'))  
model.add(Dense(10, activation='softmax'))  
model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

Obraz zawierający tekst, zrzut ekranu, diagram, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

8Obraz zawierający tekst, Wykres, zrzut ekranu, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznie

9

Obraz zawierający tekst, linia, Wykres, diagram

Opis wygenerowany automatycznie