# **Practical – 8**

**Aim:** **Data/Object classification using CNN**

* Code:

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Dropout

from tensorflow.keras.utils import to\_categorical

data\_set = pd.read\_csv('/content/drive/MyDrive/temp/practical\_4\_2.csv')

X = data\_set[['Match\_Duration', 'Loot\_Collected', 'Enemies\_Defeated']].values

y = data\_set['Player\_Score'].values

y\_class = pd.qcut(y, q=3, labels=[0, 1, 2])

y\_class = y\_class.astype(int)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_class, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

X\_train = X\_train.reshape((X\_train.shape[0], X\_train.shape[1], 1))

X\_test = X\_test.reshape((X\_test.shape[0], X\_test.shape[1], 1))

y\_train\_cat = to\_categorical(y\_train, num\_classes=3)

y\_test\_cat = to\_categorical(y\_test, num\_classes=3)

model = Sequential()

model.add(Conv1D(32, kernel\_size=2, activation='relu', input\_shape=(3, 1)))

model.add(MaxPooling1D(pool\_size=1))

model.add(Flatten())

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

model.add(Dropout(0.2))

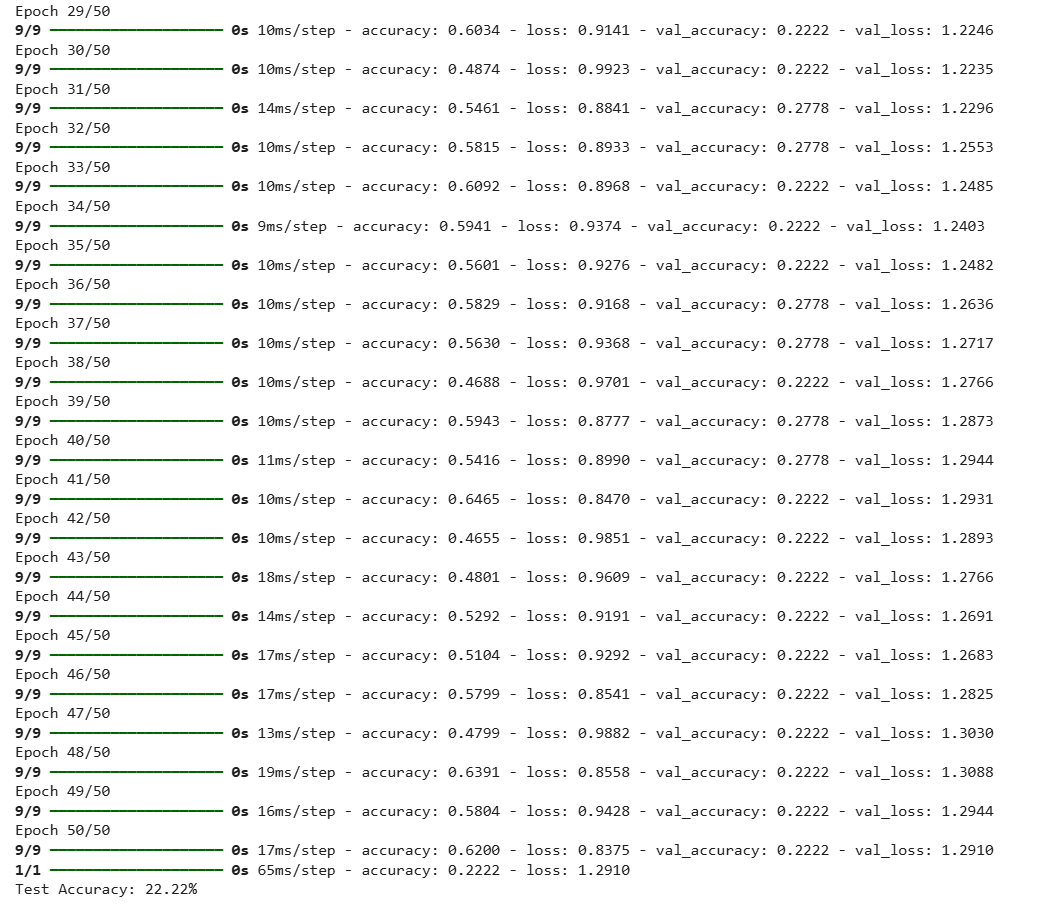
model.add(Dense(3, activation='softmax'))

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

model.fit(X\_train, y\_train\_cat, epochs=50, batch\_size=8, validation\_data=(X\_test, y\_test\_cat))

loss, accuracy = model.evaluate(X\_test, y\_test\_cat)

print(f"Test Accuracy: {accuracy\*100:.2f}%")

* Output