

Classification using DNN:Use OCR letter recognition dataset

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

columns = ["lettr", "x-box", "y-box", "width", "height", "onpix",
"x-bar", "y-bar", "x2bar", "y2bar", "xybar", "x2ybr", "xy2br", "x-ege",
"xegvy", "y-ege", "yegvx"]

df = pd.read_csv('letter-recognition.data', names=columns)

df.head()

x = df.drop("lettr", axis=1).values
y = df["lettr"].values

x.shape

y.shape

np.unique(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

class_names = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

x_train = x_train/255
x_test = x_test/255

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y_train = encoder.fit_transform(y_train)
y_test = encoder.fit_transform(y_test)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential()
model.add(Dense(512, activation = 'relu', input_shape = (16,)))
model.add(Dropout(0.2))
model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(26, activation = 'softmax'))
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
metrics = ['accuracy'])
model.summary()

model.fit(x_train, y_train, epochs = 50, batch_size= 128,
validation_data=(x_test,y_test))

predictions = model.predict(x_test)
```

```
index=10
Pred = model.predict(x_test[index].reshape(1,-1))
final_value = np.argmax(Pred)
print("Actual label :",y_test[index])
print("Predicted label :",final_value)
print("Class (A-Z) :",class_names[final_value])
```