**Lab 11: Write a program that uses Neural networks for image classification using Keras Iris dataset.**

**Program:**

import numpy as np

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

from sklearn.preprocessing import StandardScaler, LabelEncoder

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.utils import to\_categorical

from sklearn.datasets import load\_iris

# Load the Iris dataset

iris = load\_iris()

X = iris.data

y = iris.target

# Standardize the features

scaler = StandardScaler()

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

# Encode the labels as one-hot vectors

encoder = LabelEncoder()

y\_encoded = encoder.fit\_transform(y)

y\_dummy = to\_categorical(y\_encoded)

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y\_dummy, test\_size=0.3, random\_state=42)

# Create a simple neural network model

model = Sequential()

model.add(Dense(8, input\_shape=(4,), activation='relu'))

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

# Compile the model

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

# Train the model

model.fit(X\_train, y\_train, epochs=10, batch\_size=15)

# Evaluate the model

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

print(f'Accuracy: {accuracy:.2f}')

output:

