# 1. WORKING OF CNN ARCHITECTURE TO CLASSIFY IMAGES

|  |  |
| --- | --- |
| **EX.N0 : 1** | **WORKING OF CNN ARCHITECTURE TO CLASSIFY IMAGES** |
| **DATE : 21/01/2025** |

**AIM:**

To implement and demonstrate a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset in TensorFlow and Keras.

# ALGORITHM:

Step 1: Import necessary libraries.

Step 2: Load and pre-process the CIFAR-10 dataset.

Step 3: Build the CNN model using Keras Sequential API.

Step 4: Compile the model with appropriate loss function and optimizer. Step 5: Train the model on the training dataset.

Step 6: Evaluate the model on the test dataset.

Step 7: Visualize a few predictions to validate performance.

# PROGRAM:

import tensorflow as tf

from tensorflow.keras import layers, models import matplotlib.pyplot as plt

from tensorflow.keras.datasets import cifar10 import numpy as np

(x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data() x\_train = x\_train.astype('float32') / 255.0

x\_test = x\_test.astype('float32') / 255.0 model = models.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'), layers.Flatten(),

layers.Dense(64, activation='relu'), layers.Dense(10, activation='softmax')

])

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

history = model.fit(x\_train, y\_train, epochs=10, batch\_size=64, validation\_data=(x\_test, y\_test))

test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=2) print(f'\nTest accuracy: {test\_acc:.4f}')

class\_names = ['airplane','automobile','bird','cat','deer', 'dog','frog','horse','ship','truck']

predictions = model.predict(x\_test) plt.figure(figsize=(10,5))

for i in range(5):

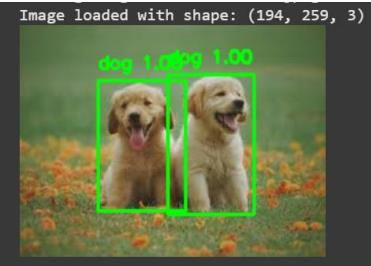
plt.subplot(1, 5, i+1) plt.xticks([])

plt.yticks([]) plt.grid(False) plt.imshow(x\_test[i])

pred\_label = class\_names[np.argmax(predictions[i])] true\_label = class\_names[y\_test[i][0]] plt.xlabel(f"Pred: {pred\_label}\nTrue: {true\_label}") plt.tight\_layout()

plt.show()

# OUTPUT:

****

**RESULT:**

Thus, to implement and demonstrate a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset in TensorFlow and Keras.