

# Title: Vision AI in 5 Days: From Beginner to Image Recognition Expert

Subtitle:

Image Classification using CNN in Python

Presented by:

Gayatri Sharma

### Dataset: CIFA R-10 / Cats vs Dogs

Total images:(e.g., 50,000 train, 10,000 test)

Processing Steps:

Image resizing

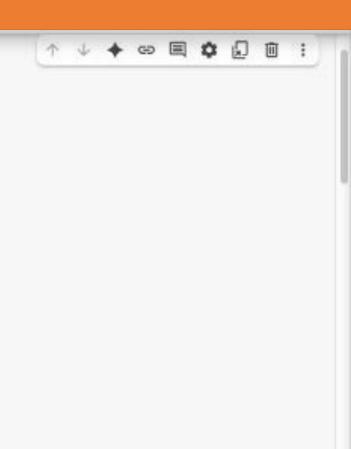
Normalization

Train-Test split

Data augmentation (rotation, flip, zoom)

#### **Dataset Preview**

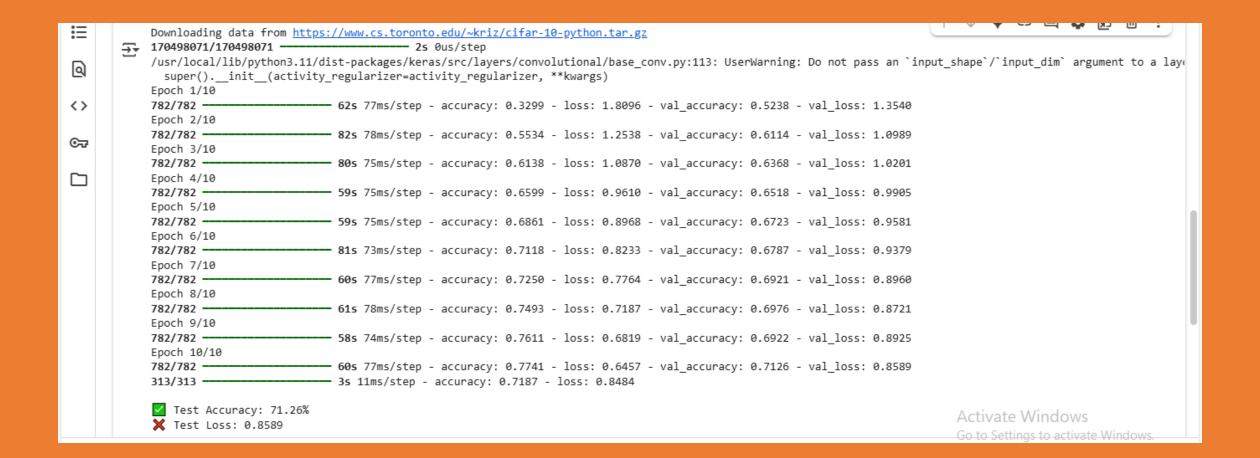
```
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
import numpy as np
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
x train, x test - x train / 255.0, x test / 255.0 # Normalize pixel values
class names - ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']
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.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)
 1)
  model.compile(optimizer='adam',
               loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
               metrics=['accuracy'])
 history = model.fit(x_train, y_train, epochs=10,
                     validation_data=(x_test, y_test), batch_size=64)
  test_loss, test_acc = model.evaluate(x_test, y_test)
 print(f"\n ✓ Test Accuracy: {test_acc * 100:.2f}%")
 print(f"X Test Loss: {test_loss:.4f}")
                                                                                                                          Activate Windows
                                                                                                                          Go to Settings to activate Windows.
     Terminal

√ 5:49 PM

bles
```



# Model Architecture:

- Model Type: Convolutional Neural Network (CNN)
- Layers: Conv → Pool → Conv → Pool → Dense → Output
- Activation Functions: ReLU,
   Softmax
- Optimizer: Adam
- Loss Function: Categorical Crossentropy

# Results & Evaluation

• Training Accuracy: ~77%

• Validation Accuracy: ~71%

• Test Accuracy: 71.26%

• **Test Loss**: 0.8589

Visuals:

Accuracy & Loss curves (Matplotlib se plot)

Confusion matrix (agar banayi ho)

Optional: Precision, Recall, F1-score table.

## Conclusion & Future Scope Conclusion:

- CNN model successfully trained for image classification.
  - Achieved 71% accuracy on test dataset.
    - Future Improvements:
  - Use Transfer Learning (MobileNetV2, ResNet)
    - More data augmentation
    - Hyperparameter tuning

#### **Demo Video:**

• <a href="https://drive.google.com/file/d/1SyWFwV90ObJnLJc9OjXRXUzF151w24y4/view?usp=sharing">https://drive.google.com/file/d/1SyWFwV90ObJnLJc9OjXRXUzF151w24y4/view?usp=sharing</a>

8/12/2025