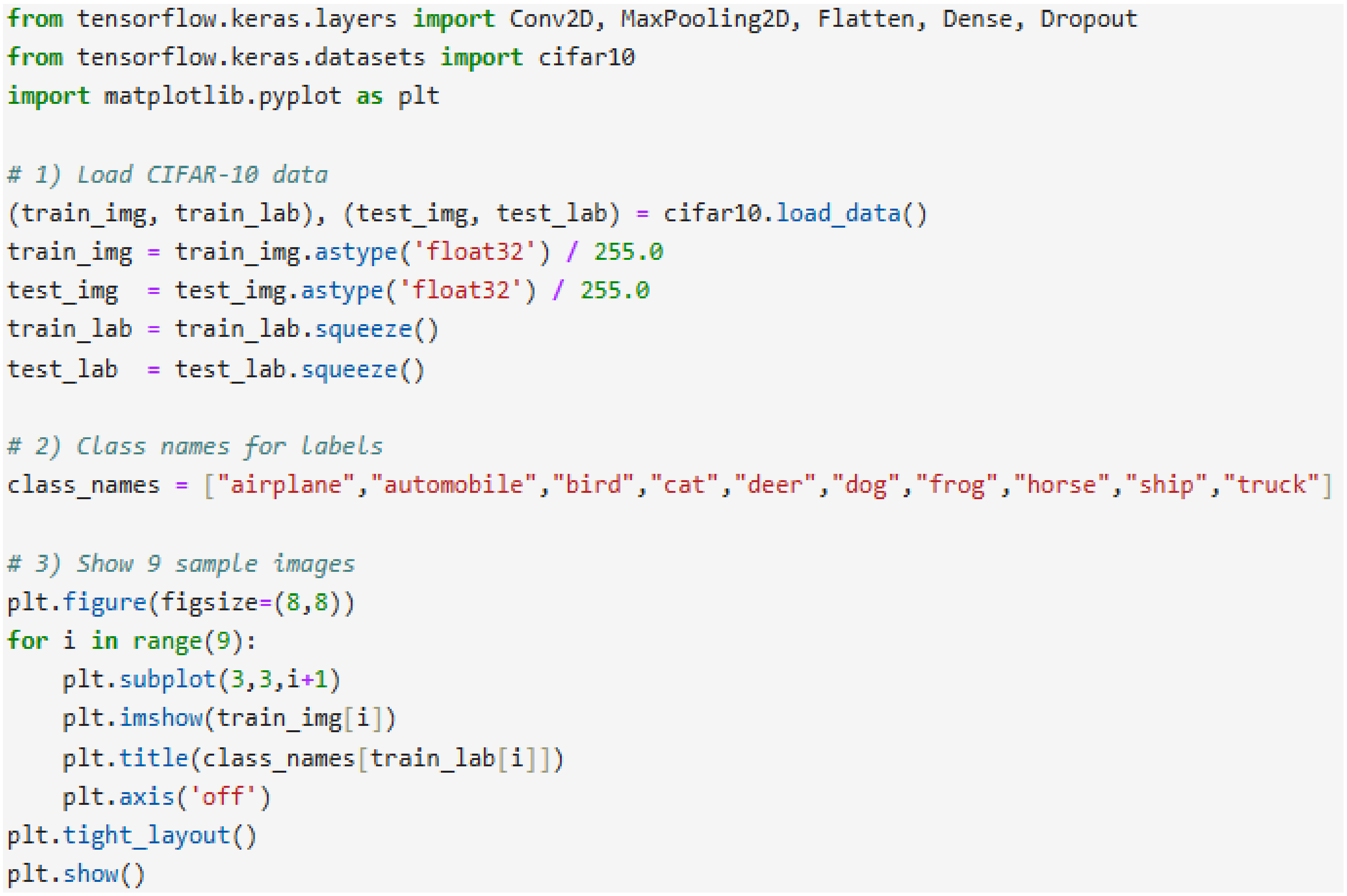
# LAB # 09

## CONVOLUTIONAL NEURAL NETWORK ALGORITHM (CNN)

## Lab Tasks:

- A dataset (CIFAR10.csv) has been provided. Implement CNN Algorithm on this dataset using Tensorflow library. Your task is to prepare the dataset and create theconvolutional base, add dense layers on top, train the model and evaluate it.

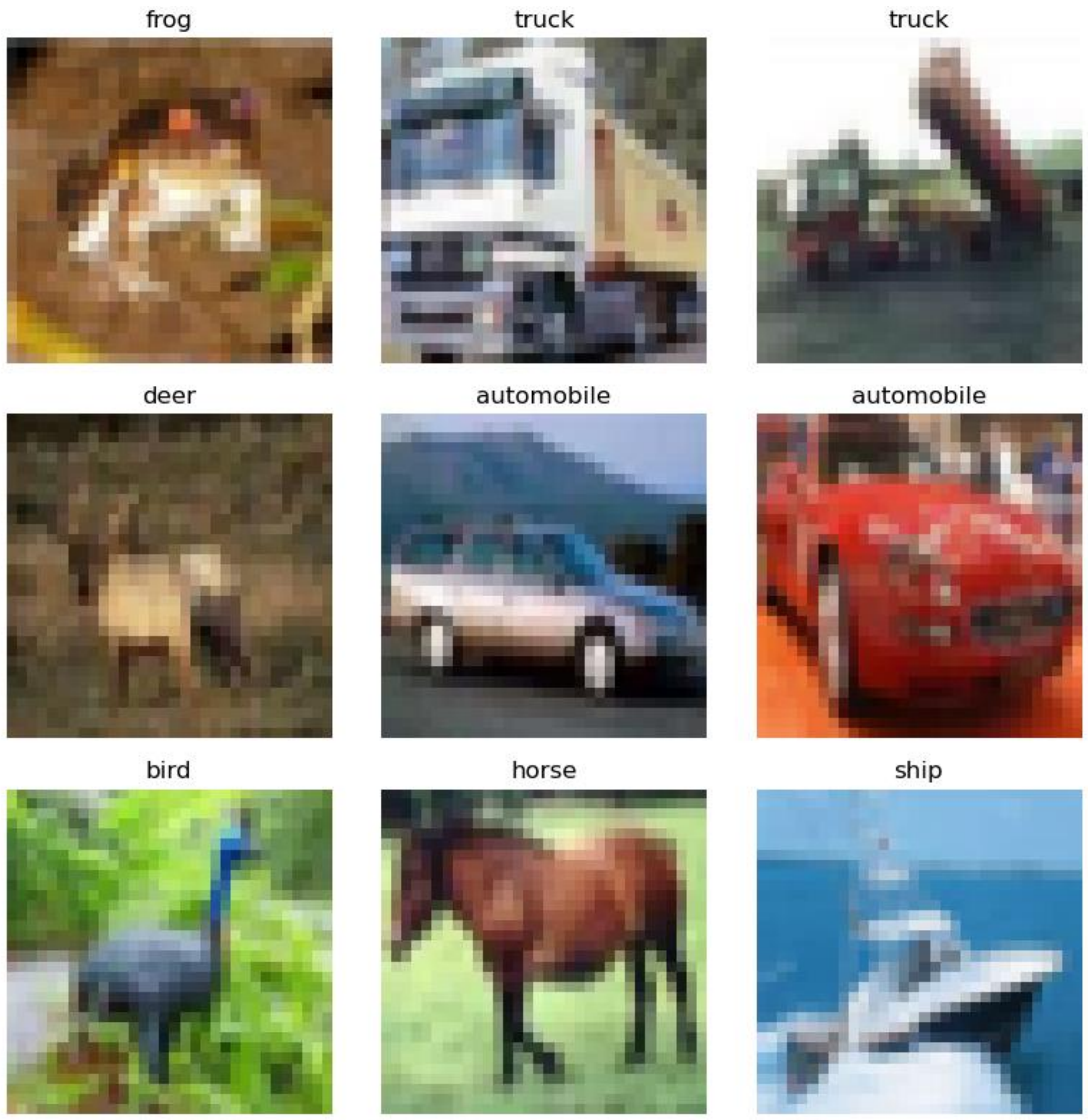
**Code:**



```
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.datasets import cifar10
import matplotlib.pyplot as plt

# 1) Load CIFAR-10 data
(train_img, train_lab), (test_img, test_lab) = cifar10.load_data()
train_img = train_img.astype('float32') / 255.0
test_img  = test_img.astype('float32') / 255.0
train_lab = train_lab.squeeze()
test_lab  = test_lab.squeeze()

# 2) Class names for labels
class_names = ["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"]

# 3) Show 9 sample images
plt.figure(figsize=(8,8))
for i in range(9):
    plt.subplot(3,3,i+1)
    plt.imshow(train_img[i])
    plt.title(class_names[train_lab[i]])
    plt.axis('off')
plt.tight_layout()
plt.show()
```

```python
# 4) Build a simple CNN
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(32,32,3)),
    MaxPooling2D((2,2)),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

# 5) Compile
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# 6) Train (few epochs to keep it fast)
history = model.fit(train_img, train_lab, epochs=5, batch_size=64,
                    validation_data=(test_img, test_lab), verbose=1)

# 7) Evaluate
loss, acc = model.evaluate(test_img, test_lab, verbose=0)
print(f"Test Accuracy: {acc:.4f}")

# 8) Plot training curves
plt.figure(figsize=(10,4))

plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], label='Train')
plt.plot(history.history['val_accuracy'], label='Val')
plt.title('Accuracy'); plt.xlabel('Epoch'); plt.ylabel('Accuracy'); plt.legend()

plt.subplot(1,2,2)
plt.plot(history.history['loss'], label='Train')
plt.plot(history.history['val_loss'], label='Val')
plt.title('Loss'); plt.xlabel('Epoch'); plt.ylabel('Loss'); plt.legend()

plt.tight_layout()
plt.show()
```

**Output:**


frog
truck
truck
deer
automobile
automobile
bird
horse
ship