University Echahid Hama Lakhdar, El-oued
Institute of Exact Sciences
Department of Computer Science
2^{ed}*Master* :
Artificial Intelligence and Data Science
Semester : 3.2025

Lab2 Image Understanding and Convolutional Neural Networks

Master Level – Computer Vision

Objective

This laboratory aims to help students understand image representation as numerical data, how to process image matrices, and how to design and evaluate Convolutional Neural Networks (CNNs) using the MNIST and CIFAR-10 datasets.

Part 1 : Image as a Matrix

1. Load a grayscale image (for example `car.jpg`) using OpenCV. Display it using Matplotlib and print its matrix shape.

```
import cv2
import matplotlib.pyplot as plt

img = cv2.imread('car.jpg', cv2.IMREAD_GRAYSCALE)
print("Image shape:", img.shape)

plt.imshow(img, cmap='gray')
plt.title("Grayscale Image")
plt.show()
```

2. Flatten the image into a 1D vector and print its shape.

```
flattened = img.flatten()
print("Flattened shape:", flattened.shape)
```

What information is lost when the image is flattened ?

3. Normalize pixel values to the range $[0, 1]$ and visualize the histogram of pixel intensities.

```
import numpy as np

normalized = img.astype('float32') / 255.0
plt.hist(normalized.ravel(), bins=50)
plt.title("Normalized Pixel Intensity Distribution")
plt.show()
```

Explain why normalization is necessary before training a neural network.

Part 2 : Convolutional Neural Network with MNIST

1. Load and visualize a few MNIST samples.

```
from tensorflow.keras.datasets import mnist
import matplotlib.pyplot as plt

(x_train, y_train), (x_test, y_test) = mnist.load_data()
plt.figure(figsize=(5,5))
for i in range(9):
    plt.subplot(3,3,i+1)
    plt.imshow(x_train[i], cmap='gray')
    plt.axis('off')
plt.show()
```

Describe what each pixel value represents.

2. Reshape and normalize the data for CNN input.

```
x_train = x_train.reshape(-1, 28, 28, 1).astype('float32') / 255
x_test  = x_test.reshape(-1, 28, 28, 1).astype('float32') / 255
```

Why do we add the extra dimension (the channel) ?

3. Build a CNN model using Keras.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
model.summary()
```

Explain the role of each layer in this architecture.

4. Compile, train, and evaluate the model.

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=5, validation_split=0.1)
test_loss, test_acc = model.evaluate(x_test, y_test)
print("Test accuracy:", test_acc)
```

Plot the training and validation accuracy curves and discuss overfitting or underfitting behavior.

Part 3 : Deep CNN for CIFAR-10

1. Load the CIFAR-10 dataset and visualize one image from each class.

```
from tensorflow.keras.datasets import cifar10
import matplotlib.pyplot as plt
import numpy as np

(x_train, y_train), (x_test, y_test) = cifar10.load_data()
class_names = ['airplane','automobile','bird','cat','deer',
               'dog','frog','horse','ship','truck']

plt.figure(figsize=(10,4))
for i in range(10):
    idx = np.where(y_train==i)[0][0]
    plt.subplot(2,5,i+1)
    plt.imshow(x_train[idx])
    plt.title(class_names[i])
    plt.axis('off')
plt.show()
```

Describe how CIFAR-10 images differ from MNIST images in structure and complexity.

2. Normalize the images and design a CNN architecture.

```
x_train = x_train.astype('float32') / 255.0
x_test  = x_test.astype('float32') / 255.0

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),
    Conv2D(128, (3,3), activation='relu'),
    Flatten(),
    Dense(256, activation='relu'),
    Dense(10, activation='softmax')
])
model.summary()
```

How does increasing the number of filters affect feature extraction ?

3. Add dropout and retrain the model to reduce overfitting. Compare the accuracy before and after adding dropout.

Part 4 : Reflection Questions

— Compare the performance of CNNs on MNIST and CIFAR-10 datasets. Which dataset is harder to classify and why ?

— Explain how convolution layers learn spatial hierarchies of features.

— Suggest one modification to improve CIFAR-10 model accuracy.

— Discuss why flattening an image is not suitable for complex visual tasks compared to convolutional processing.