CS4045 - Deep Learning

Assignment #02

Deadline: 21 October 2024

Instructions:

The assignment deadline is 21 October 2024. There is no extension in the deadline. Please submit on or before the deadline. You must verify that your submissions are correct.

Please ensure that you follow the proper file naming convention when submitting your assignments. The required format is as follows: **i21xxxx_A2.ipynb**

Image Compression and Reconstruction using Custom Neural Networks

Objective:

The primary aim of this assignment is to gain hands-on experience in constructing and experimenting with neural networks from scratch. Students will focus on implementing various architectures and layer configurations to explore their effects on image compression and reconstruction capabilities.

Dataset:

Students will utilize the LIVE Image Compression Dataset (LIVE 1), which contains pairs of original and JPEG-compressed images. This dataset will serve as a foundation for training and evaluating neural network models for image compression tasks.

Tasks:

1. **Data Exploration and preprocessing**
2. **Neural Network Implementation:**
    - Build a neural network architecture from scratch (NumPy or pytorch).

- Start with a simple architecture and gradually incorporate additional layers, such as convolutional layers, pooling layers, and dense layers.
- Experiment with different configurations, such as the number of layers, types of layers (e.g., convolutional, dropout, batch normalization), activation functions (e.g., ReLU, Sigmoid), and the arrangement of these layers.

3. **Training Process:**
    - Train the model using original images as inputs and their corresponding JPEG-compressed images as target outputs.
    - Implement techniques such as learning rate adjustments, batch normalization, and dropout to enhance training stability and performance.
4. **Evaluation of Results:**
    - Evaluate the performance of your neural network model by comparing the reconstructed images against the original images.
    - Use metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to quantitatively assess the quality of reconstructed images.
    - Analyze how changes in the architecture (e.g., adding more layers, modifying layer types, changing activation functions) affect the model's performance and reconstruction quality.
5. **Analysis:** Discuss your findings regarding the trade-offs between model complexity, training time, and reconstruction accuracy, offering insights into best practices for building effective neural networks for image compression tasks.

Human Exercise Recognition: Integrating YOLO for Object Detection, 3D Pose Estimation, and CNN-Based Exercise Classifier

Objective:
To build an exercise recognition pipeline that integrates object detection, 3D pose estimation, and classification using deep learning techniques. Students are required to use YOLO for detecting the bounding box around a person, a pretrained model for estimating 3D body landmarks, and a custom CNN for exercise classification based on the 3D landmark data. The final output should include a visualization of the detected 3D landmarks on the original image along with the predicted exercise.

Dataset:

- **Description:** The dataset contains data for various exercise types, including annotated 3D landmark positions and other features. It will be used to train a custom CNN for classifying exercises based on the detected 3D landmarks.
- The dataset represents 10 different physical poses that can be used to distinguish 5 exercises. The exercises are Push-up, Pull-up, Sit-up, Jumping Jack and Squat. For every exercise, 2 different classes have been used to represent the terminal positions of that exercise (e.g., “up” and “down” positions for push-ups).

Tasks:

1. Image Acquisition

- **Task:** Capture an image or frame from a video of a person performing an exercise.
- **Requirement:** The input image should show a person performing one of the five exercises mentioned in the dataset and the image should be clear enough for pose estimation.

2. Bounding Box Detection Using YOLO

- **Task:** Use a pretrained YOLO model to detect and localize the person in the image.

3. 3D Pose Estimation Using a Pretrained Model

- **Task:** Crop the image based on the bounding box detected in the previous step and use a pretrained model using **MediaPipe** (33 landmark version) to estimate 3D body landmarks.
- **Output:** Predict 3D landmark coordinates (x, y, z) for key body points (e.g., shoulders, elbows, knees).
- **Visualization:** Map the detected 3D landmarks back onto the original input image to illustrate the pose estimation results.

4. Custom CNN for Exercise Classification

- **Task:** Build a custom Convolutional Neural Network (CNN) to classify exercises based on the 3D landmark data obtained from the pose estimation model.
- **Model Architecture:** Design a CNN architecture that takes the 3D landmark coordinates as input features for classifying the exercise.

- **Training:** Train the CNN using the dataset, ensuring that the model generalizes well to different examples.
- **Evaluation:** Evaluate the CNN's performance using different metrics e.g: accuracy, precision, recall, F1-score

5. Mapping and Visualization

- **Task:** Overlay the detected 3D landmarks and the predicted exercise label on the original image for visual clarity.