DEPENDABLE AND SECURE AI-ML (AI60006)

Session 2023-2024

Assignment 1

Submission Deadline: (2 weeks from the date of upload)

You need to be alert to (usually minor) changes that may be made to the assignment statement or to the guidelines after the assignment is first put up. Refresh this frame and re-read the assignment carefully before you make your final submission.

Reference papers:

Visualizing and Understanding Convolutional Networks (https://arxiv.org/pdf/1311.2901.pdf)

Visualizing and Comparing Convolutional Neural Networks (https://ar5iv.labs.arxiv.org/html/1412.6631)

Interpretable Convolutional Neural Networks (https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.pdf)

Notebook links

https://drive.google.com/file/d/1jpSUEZYBD1h2yQrvAJCOao6k9s7vWKDl/view?usp=drive_link

https://drive.google.com/file/d/1iW9XJPl5B2mLmXWPHNfe65EXO5JMXfXd/view?usp=drive_link

Advanced Image Processing: Pixel Expansion and Noise Addition

Objective:

- Understand how pixel-based transformations affect image quality.
- Implement algorithms to expand single pixels into multiple pixels.
- Introduce different types of noise (Gaussian, Salt & Pepper, Poisson) into images.
- Evaluate the effect of transformations using image quality metrics

Assignment Tasks:

Task 1: Single Pixel Expansion to Multiple Pixels

1. Load a grayscale image (e.g., 512×512 Lena or any sample image).
2. Implement a function that expands a single pixel into a block of pixels. The function should:
    - Take an image and a scaling factor (e.g., 2×2, 3×3, 4×4).
    - Replace each pixel with a block of similar or interpolated values.
    - Support different expansion strategies:
        - Constant Expansion: Every new block has the same value as the original pixel.
        - Gradient Expansion: Uses interpolation to create a smooth transition between adjacent pixels.
        - Random Expansion: Uses random variations in the new block pixels based on a predefined range.
3. Display the original and transformed images side by side.

Task 2: Adding Noise to an Image

1. Implement functions to add the following types of noise to the image:
    - Gaussian Noise: Add random variations with a normal distribution.
    - Salt & Pepper Noise: Randomly turn some pixels completely black (0) or white (255).
    - Poisson Noise: Generate noise based on the image pixel intensity distribution.
2. Allow the user to select the noise type and adjust the intensity level.
3. Display the noisy images along with the original image.

Task 3: Evaluating Image Quality

1. Compute and compare the following quality metrics before and after applying noise:
    - Mean Squared Error (MSE)
    - Peak Signal-to-Noise Ratio (PSNR)
    - Structural Similarity Index (SSIM)
2. Analyze the impact of different noise types on image quality.

Bonus Task (Optional):

- Implement a denoising algorithm (e.g., Median Filter, Bilateral Filter, Non-Local Means) to remove noise and recover the original image.
- Compare the denoised image with the original using PSNR and SSIM.

Understanding CNN Layers: Filters, Activations, and Noise Impact

Objective:

- Understand the anatomy of CNN layers and their functionalities.
- Visualize feature maps at different layers of a CNN.
- Analyze how CNN filters detect edges, textures, and patterns.
- Investigate the impact of noisy images on CNN feature extraction.
- Craft images with controlled noise and analyze their effect on CNN activations.

Task 1: CNN Layer Anatomy & Filter Visualization

1. Load a Pretrained CNN
   - Use a pretrained model like VGG16, ResNet, or MobileNet from TensorFlow/Keras or PyTorch.
   - Display the architecture of the network and its layer details.
2. Extract and Visualize Filters
   - Select convolutional layers from the model (e.g., conv1, conv2).
   - Extract and visualize the learned filters using Matplotlib.
   - Explain the role of these filters (edge detection, texture recognition, etc.).

Task 2: Feature Map (Activation) Visualization

1. Pass an Input Image Through the CNN
   - Choose a sample image (e.g., a face, an object).
   - Pass it through the model and extract feature maps from different layers.
   - Visualize these feature maps to observe what the network "sees."
2. Compare Shallow vs. Deep Layers
   - Visualize activations from early, mid, and deep layers.
   - Explain how deeper layers focus on abstract patterns instead of edges.

Task 3: Crafting Noise Images & Analyzing Impact on CNN

1. Generate Noisy Images
   - Use the noise techniques from the previous assignment (Gaussian, Salt & Pepper, Poisson).
   - Craft images with increasing levels of noise intensity.
2. Pass Noisy Images Through the CNN
   - Compare feature maps for clean and noisy images.
   - Discuss how noise affects feature extraction at different layers.
3. Analyze Performance Drop
   - Measure classification confidence on clean vs. noisy images.

- Use adversarial-style perturbations (small changes in pixels) to observe CNN robustness.

Grading Criteria:

Criteria	Marks
Pixel Expansion Implementation	2
Noise Addition Implementation	2
Image Quality Analysis	2
Code Efficiency & Documentation	1
Report & Observations	2
Bonus Task (Denoising)	1
CNN Filter Visualization	2
Feature Map Analysis	2
Noisy Image Crafting & Analysis	2
Impact on CNN Performance	2
Report & Observations	2

Expected Submission:

- Python scripts (.py or .ipynb)
- A report explaining the methodology, results, and observations
- Sample images before and after processing

Marking guidelines

Submit notebooks and the answers to the questions above in a separate report (word document). Name the report “ROLLNO_1.doc. Name the archive “ ROLLNO_1.zip” .

Assignment marking is to be done only after the deadline expires, as submission gets blocked after the assignment is marked.

You should keep submitting your incomplete assignment from time to time after making some progress, as you can submit any number of times before the deadline expires.

Warning: Cases of copying from internet sources or fellow students will be dealt with seriously and severely, with recommendation to the Dean to de-register the student from the course.