cs512 f24 - assignment 3 (fitting and recognition)

Due by: 10/19/2024

General Instructions:

In this assignment, you will implement and explore several key concepts in computer vision and deep learning. The tasks involve robust line estimation, image classification, and the use of convolutional neural networks (CNNs) for various image datasets. By working through these exercises, you will gain experience in generating and processing noisy data, estimating model parameters, and building and evaluating CNN architectures.

For the robust estimation task, you will simulate noisy data and investigate methods to estimate the underlying geometric parameters. The goal is to understand how noise and outliers impact estimation accuracy and explore the use of robust techniques to mitigate these effects.

In the image classification tasks, you will train and evaluate CNN models using two widely-used datasets: Kaggle's Cats and Dogs dataset and the CIFAR-10 dataset. You will implement and test different network architectures, data augmentation techniques, and pre-trained models to compare their performance.

Submission instructions:

- Follow the submission instructions of assignment 0.
- Submit a fully executed Python notebook with no gaps in execution. To receive full credit, please ensure the following:
- The notebook includes all cell outputs and contains no error messages.
- It is easy to match each problem with its corresponding code solution using clear markdown or comments (e.g., “Problem 2”).
- Re-run your notebook before submitting to ensure that cells are numbered sequentially, starting at [1].

Questions:

1. Robust Estimation

1. Generate points that belong to a line segment with given parameters (angle and distance from the origin). Test your line generation by creating points for lines with angles between 0 and 90 degrees, and plot the results.
2. Add Gaussian noise with specified mean and standard deviation to the points of a specific line segment, and plot the noisy points. Ensure that the noise is visible.

3. Given the noisy points from the previous step, estimate the parameters of the original line and compute the error compared to the known parameters.
4. Plot a graph showing the error as a function of the noise level.
5. Introduce a percentage of outliers to the point set and re-estimate the line parameters. Plot the error as a function of the percentage of outliers.
6. Use the `cv2.fitLine` function for robust line estimation, utilizing the `CV_DIST_HUBER` distance. Plot the error as a function of the percentage of outliers.
7. Evaluate the algorithm and report results.

2. Image Classification

1. Download the Kaggle Cats and Dogs dataset and select a subset of 2000 dog images and 2000 cat images for training, validation, and testing. The reduced dataset simulates the condition of limited data.
2. Build a convolutional neural network (CNN) with several convolution, pooling, and normalization layers, followed by one or more dense layers. Flatten the output between the convolution and dense layers. Evaluate the model's performance on the validation set.
3. Modify the data generator to include data augmentation, and evaluate the performance on the validation set again.
4. Replace the convolution layers with a pre-trained VGG16 convolutional base, and evaluate the model's performance on the validation set.
5. Evaluate the algorithm and report results.

3. Image Classification 2

1. Download the CIFAR-10 dataset and load the pickled data into your program.
2. Build a basic CNN with several convolution blocks, where each block includes convolution, pooling, and normalization layers. Flatten the output and pass it through a dense layer that uses softmax activation. Evaluate the model on the validation set.
3. Replace the convolution blocks with Inception blocks and test the model's performance.
4. Replace the Inception blocks with residual blocks and test the model's performance.
5. Evaluate the algorithm and report results.