KAIST
EE535 Digital Image Processing
Assignment 2

Honi Selahaddin (20226089)

April, 2022

Noise reduction using spatial filters

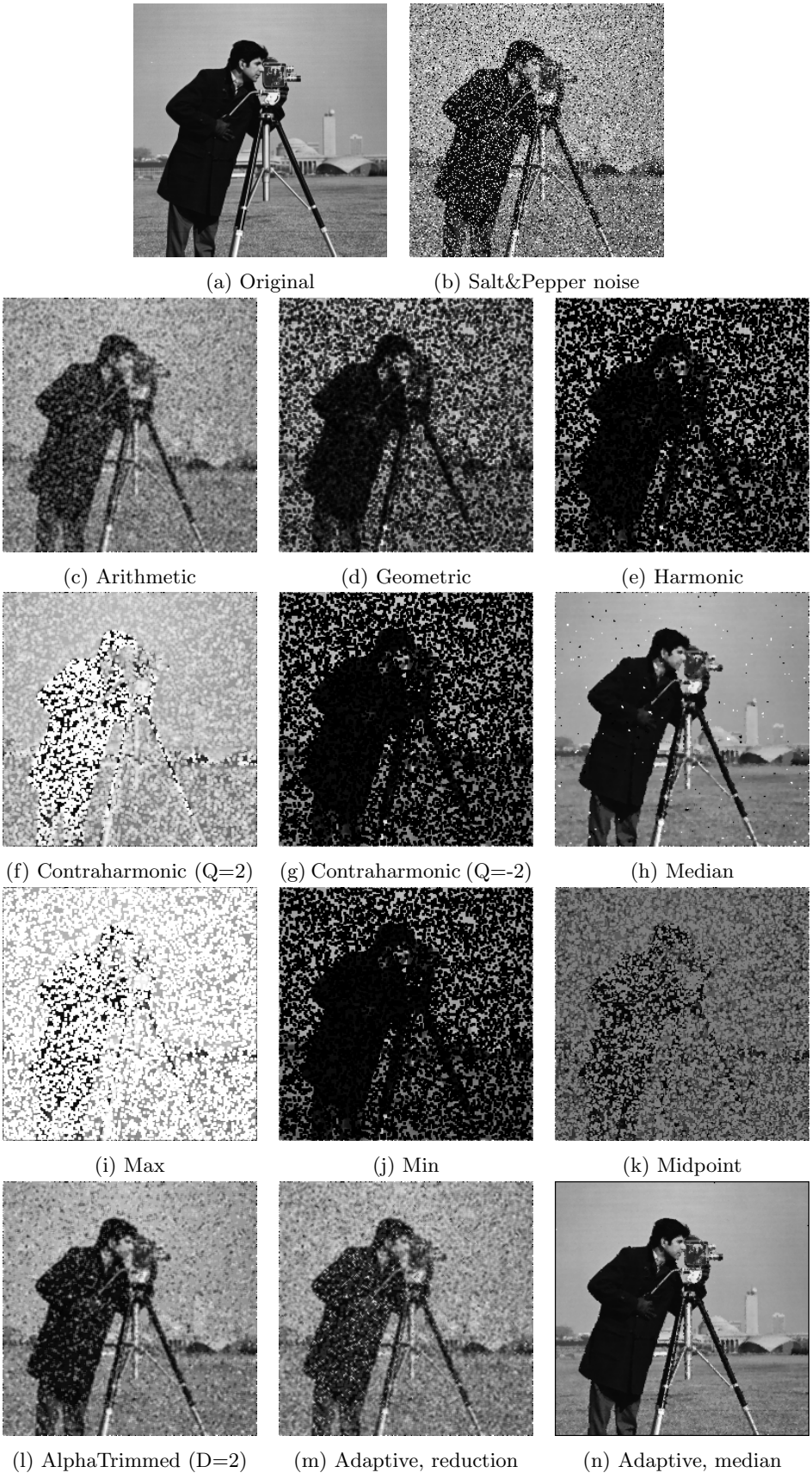
Figures 4-5-6 show the result of mean (c,d,e,f,g), order-statistic (h,i,j,k,l) and adaptive filter (m,n) outputs for given corrupted 'cameraman' image (b) by Salt&Pepper noise ($P_s = P_p = 0.25$), Gaussian noise ($\sigma = 20$) and both Salt&Pepper+Gaussian noises, respectively. Same order follows for 'lena' image in Figures 1-2-3.

(c) Arithmetic mean filter replaces the center pixel intensity to the average intensity of the neighborhood. Even though its simplicity it manages to remove some part of noise. (d) Geometric filter outperforms the arithmetic mean in images corrupted by Gaussian noise, apparently. (e) Harmonic filter works well for only salt noise; because, images are corrupted by both salt and pepper, output of harmonic filter contains the pepper noise. (f,g) Contraharmonic filter has ability to remove salt or pepper noise for given parameter Q. If Q is negative it eliminates salt noise, again, but for positive Q values it eliminates pepper noise. (h) Median filter is my favorite in this filter application due to its excellent success while still keeping the simplicity. (i,j,k) Max, min and midpoint filters are not convenient for salt and pepper noise; since, extreme values in the intensity range, [0,255], can easily makes the image total bright, dark and gray. (l) Alpha-trimmed mean filter performs well in the combination of Salt&Pepper and Gaussian noise. (m) Adaptive local noise reduction filter estimates the image noise variance and compares each neighborhood instensity variance to preserve the edges while smoothing. Nevertheless, it is not easy to see a superior difference compared to others. (n) Adaptive median filter, the winner of this observation, has an intelligent algorithm to check if replacing the median intensity is really required. In this way, it utilizes the power of median filtering by searching a larger neighborhood depending on the condition of median value must be less than the maximum intensity and greater than the minimum intensity.

Implementation

Python Jupyter notebook file is available in 'src' folder. The program requires OpenCV for image I&O and matrix computations for NumPy. 'Noise Addition' section contains Gaussian and S&P noise addition functions to apply these onto 'lena' and 'cameraman' images. Noisy images are stored in 'A,B,C' lists. 'Mean and Order-statistic Filters' functions are called inside a nested-loop for each neighborhood of each noisy images in 'A,B,C' lists. Parameters (if exist) are determined in function arguments by default. 'Adaptive Filters' have a bit different loop design such that 'Local Noise Reduction Filter' calls 'estimateNoiseVar()' function, firstly, to predict global image noise variance by calculating image histogram. 'AdaptiveMedian()' filter function gets each neighborhood in the image and executes recursively.

Neighborhood size 'k=3' for all filters; however, adaptive median filter can change the neighborhood size if specified conditions are not satisfied. Therefore, maximum allowed neighborhood size is limited by 'Smax=30'.



(a) Original (b) Salt&Pepper noise

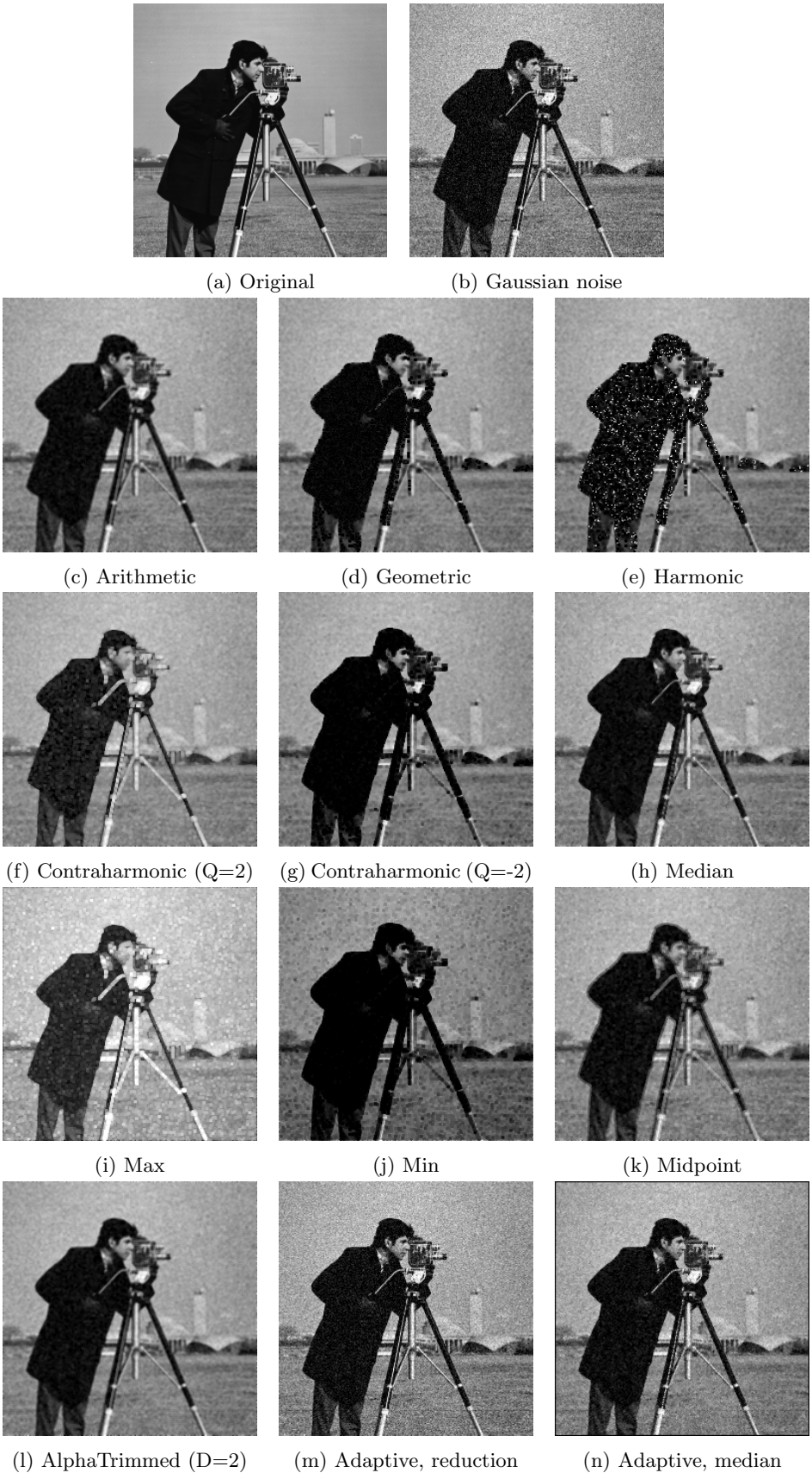
(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 1: Applying mean, order-statistic and adaptive filters to 'cameraman' image corrupted by Salt & Pepper noise.



(a) Original (b) Gaussian noise

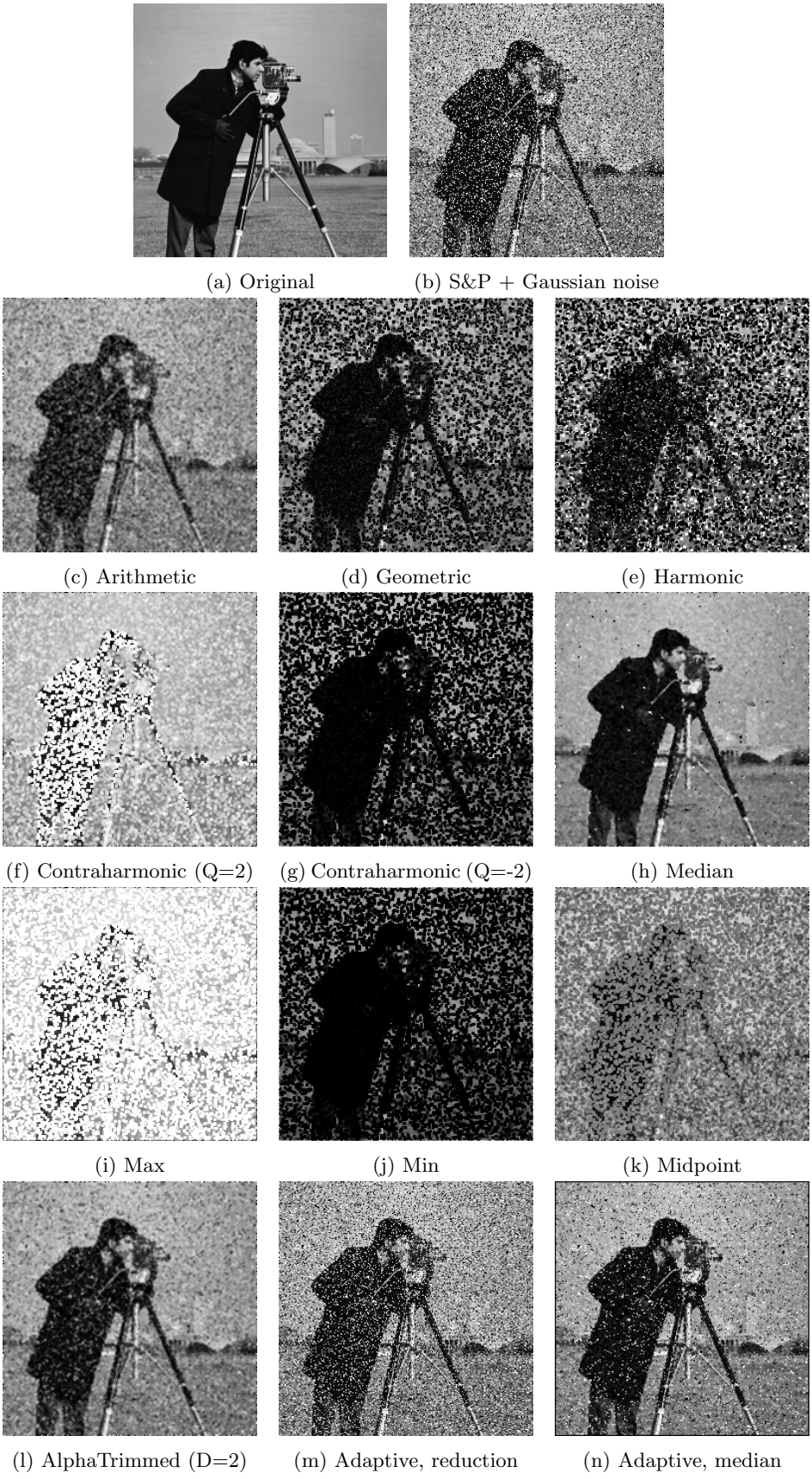
(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 2: Applying mean, order-statistic and adaptive filters to 'cameraman' image corrupted by Gaussian noise.



(a) Original (b) S&P + Gaussian noise

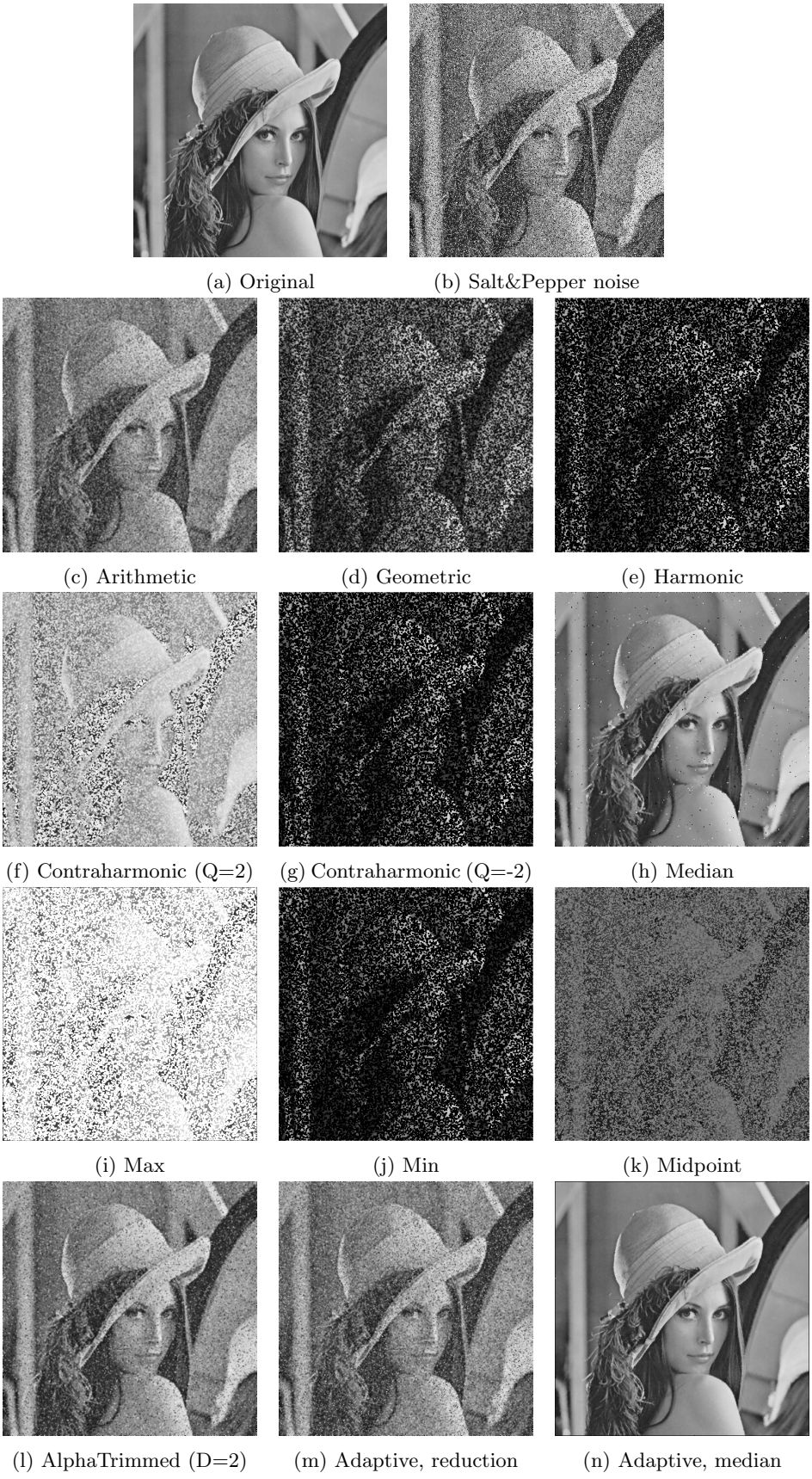
(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 3: Applying mean, order-statistic and adaptive filters to 'cameraman' image corrupted by Salt & Pepper + Gaussian noise.



(a) Original (b) Salt&Pepper noise

(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 4: Applying mean, order-statistic and adaptive filters to 'lena' image corrupted by Salt & Pepper noise.

(a) Original (b) Gaussian noise

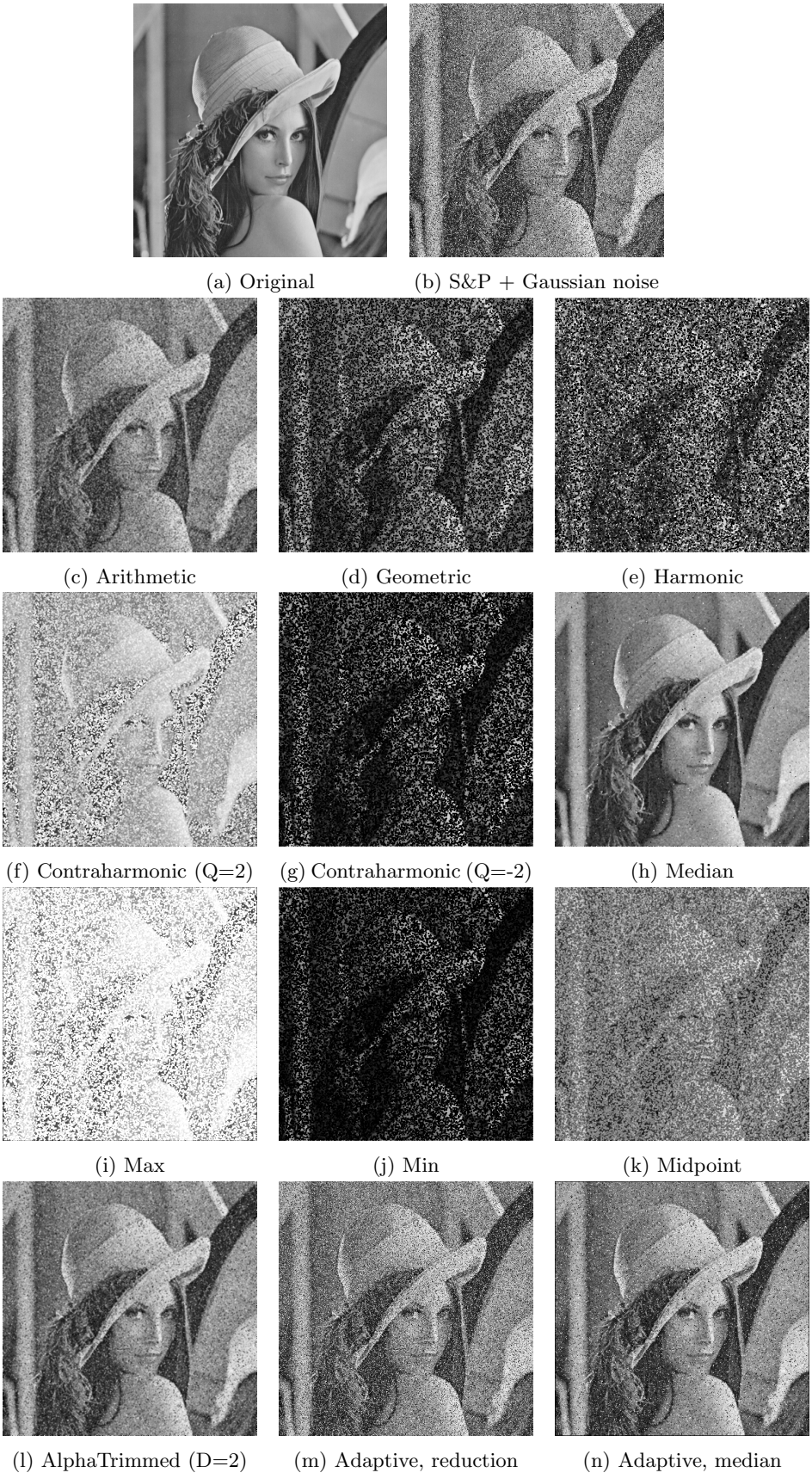
(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 5: Applying mean, order-statistic and adaptive filters to 'lena' image corrupted by Gaussian noise.



(a) Original (b) S&P + Gaussian noise

(c) Arithmetic (d) Geometric (e) Harmonic

(f) Contraharmonic (Q=2) (g) Contraharmonic (Q=-2) (h) Median

(i) Max (j) Min (k) Midpoint

(l) AlphaTrimmed (D=2) (m) Adaptive, reduction (n) Adaptive, median

Figure 6: Applying mean, order-statistic and adaptive filters to 'lena' image corrupted by Salt & Pepper + Gaussian noise.