Image De-noising with Kernel Principal Component Analysis implementing the Morlet Wavelet Kernel

Final Presentation

Presented By, *Kernal Kings*

Biswajit Rana
MSc Big Data Analytics (B2330026)

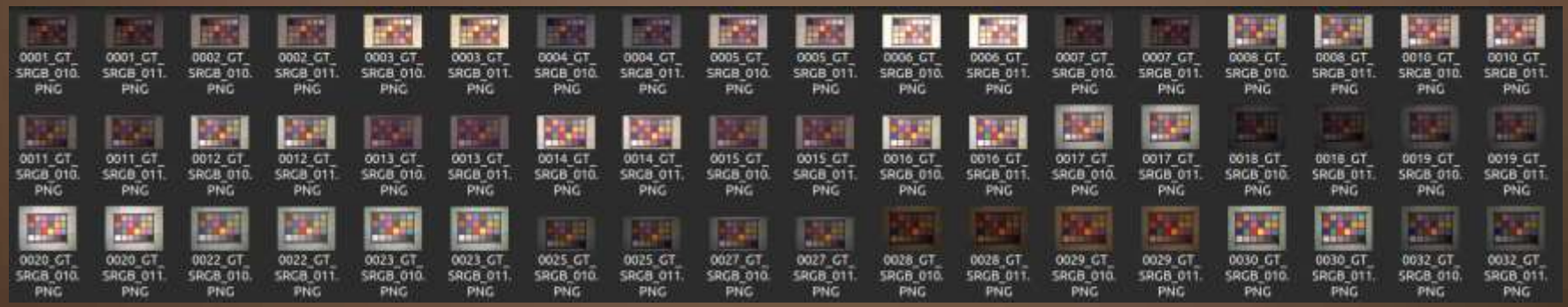
And

Uddalak Mukherjee
MSc Big Data Analytics (B2330042)

WORK PLAN AND TIME LINE

Data Collection:

- We have collected the data from https://abdokamel.github.io/sidd/ (Smartphone Image Denoising Dataset)
- Data contains 320 pairs of Noisy and True images.
- Images are from various scenes with different brightness, Exposure and lighting. Given sample of one scene.



Zoomed Noisy Image

Zoomed True Image

Noisy Images (taken for experiment):

Standard Image (noisy) :

Literature Review :

- The following 3 papers are being studied to get further insights on the use of dimensionality reduction in image processing.

[1] S.Krishnan P.Lio S.Xie, A.T.Lawniczak. "**Wavelet kernel principal component analysis in noisy multiscale data classification**." *ISRN Computational Mathematics*, 259:4–6,2012.

[2] P.Li Y.Tang T.You, Y.Hu. "**An improved imperialist competitive algorithm for global optimization**." *Turkish Journal of Electrical Engineering and Computer Science*, pages 5–6, 2019.

[3]Abdelhamed, Abdelrahman, Stephen Lin, and Michael S. Brown. "**A HighQuality Denoising Dataset for Smartphone Cameras.**" *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1692-1700.

Metrics of Comparison:

- Peak Signal-to-Noise Ratio (PSNR):
  - PSNR $= 10 \log_{10}(\frac{MAX^2}{MSE})$
  - MAX is the maximum possible pixel value of the image (e.g., 255 for an 8-bit image)
  - MSE is the mean squared error between the original and denoised images.
  - A higher PSNR value indicates better denoising performance, with a maximum value of ∞ when the images are identical.
- Structural Similarity Index (SSIM):
  - SSIM(x,y) $= \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x{}^2+\mu_y{}^2+c_1)(\sigma_x{}^2+\sigma_y{}^2+c_2)}$
  - x and y are the original and denoised images, respectively,
  - μ_x and μ_y are the mean intensities of x and y ,
  - σ_x and σ_y are the variances of x and y ,
  - σ_{xy} is the covariance between x and y ,
  - c_1 and c_2 are constants to stabilize the division with weak denominator.

  SSIM values range from -1 to 1, with 1 indicating perfect similarity between the images.

Validating Mathematical Foundation:

- In order to use the proposed function as a Kernel, the Mercer condition needed to be verified.

Condition 1

This is a pretty simple condition. Symmetric condition is given as:

$$K(x, x') = K(x', x)$$

Condition 2

For a given unique set of N data points $x_1, x_2, \ldots, x_N$, the function K is said to be positive semi-definite if:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} K(x_i, x_j) c_i c_j \geq 0$$

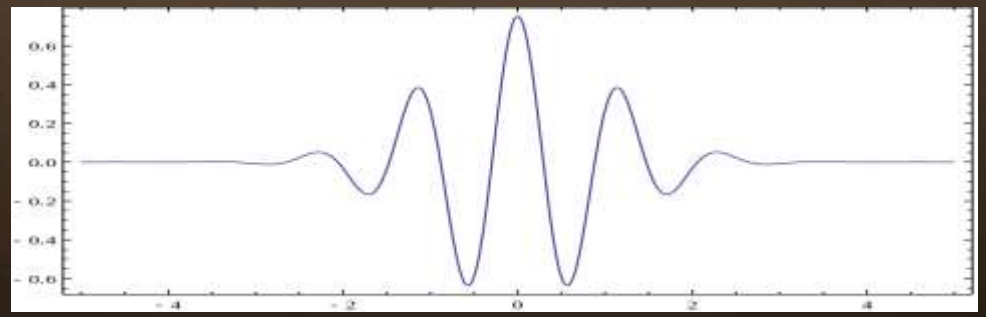
Here, c_i is any real-valued constant number.

- A rigorous proof was penned down to show that the function indeed satisfies the conditions. The fact that is indeed a Kernel was conjectured in [1] and our proof was partly motivated from a proof in [1] which used a similar technique to validate the condition for a different family of wavelet functions.

Implementing Our Model :

- Now we have tried to denoise the image using Wavelet function based kernel.
- We have done the same things with this as we have done in previous case.
- We generated the image one by one .
- Tried to regenerate the denoised image with color in more better way. We got better results when we increased the color intensity of the denoised image.
- The Morlet wavelet kernel based on the Morlet wavelet function $\psi(x) = \cos(5x)\exp(-\frac{x^2}{2})$ as follows:

$$k(x,y) = \prod_{k=1}^{d} \cos\left(\frac{5(x_k - y_k)}{a}\right) \exp\left(-\frac{(x_k - y_k)^2}{2a^2}\right) \quad \text{where } x, y \in \mathbb{R}^d$$



Testing Existing Models :

- We have tried to denoise the image using two existing dimensionality reduction models, PCA and KPCA-RBF.
- As the resolution of the images are very large(5328,3000) , We have used “Patch-Based Image Denoising” process.
- We have read the image in grey-level then denoised it. After that we recolored the image using the color filater of the noisy image.
- We generated the image one by one .
- First we have applied the PCA model which generates the gray level image as following.
- Next we have applied the Kernel PCA model which generates the image as follows.
- Results from existing state of the art models were also taken for the comparison.
  e.g. : CBDNet and CGNet

Comparing Results on Standard Image:

Method	MSE	PSNR	SSIM
PCA	41.957	31.660	0.709
KPCA + RBF	29.1701	33.2396	0.8419
KPCA + Morlet	29.2611	33.2261	0.8415
Mean Filter	29.926	33.128	0.7481
Median Filter	28.421	33.352	0.7544

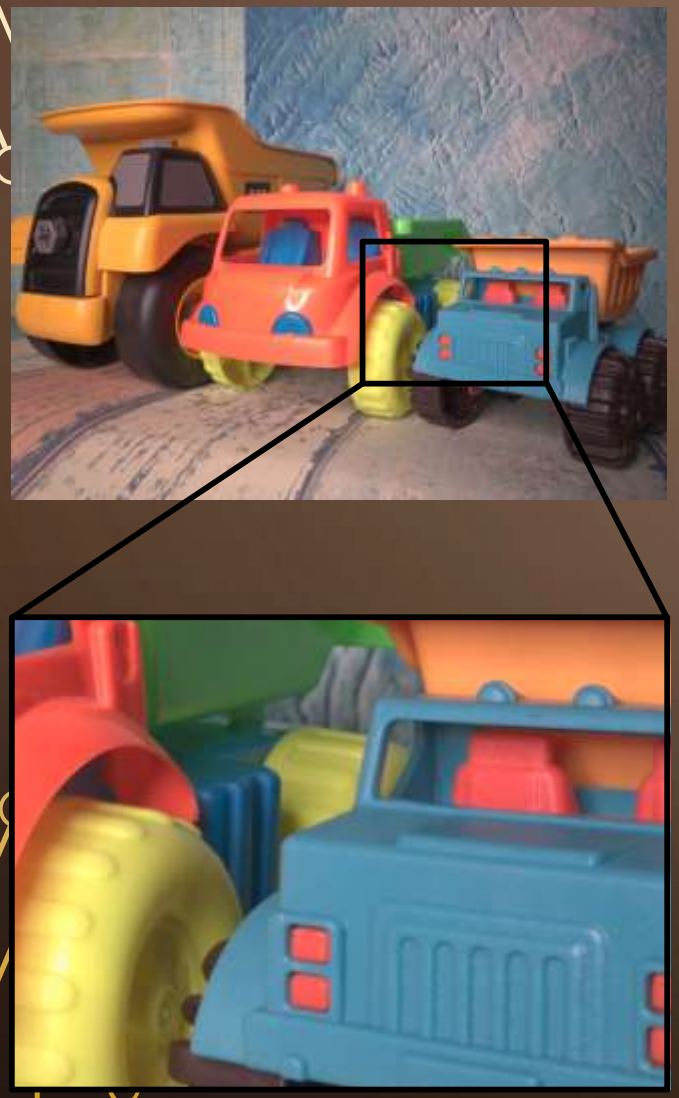
Comparing Results for the 4 selected images:

Methods	Average PSNR	Average SSIM
Morlet KPCA	31.949	0.7306
RBF KPCA	31.567	0.706
Ordinary PCA	30.535	0.6979
Mean Filter	31.092	0.7374
Median Filter	30.909	0.72895
CGNet*	42.431 (40.39)	0.972 (0.964)
CBDNet*	31.74 (30.78)	0.823 (0.801)

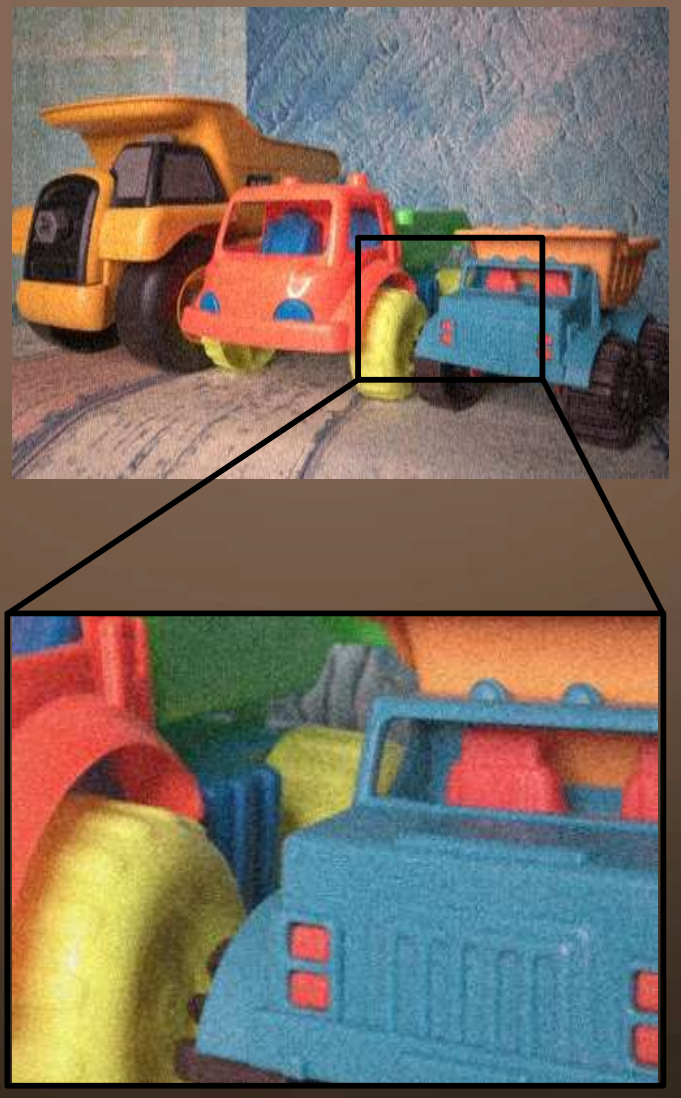
* These methods were implemented on the entire dataset as well as they are state of the art methods. In brackets are there averages over all images in the dataset.

Denoised by Wavelet :

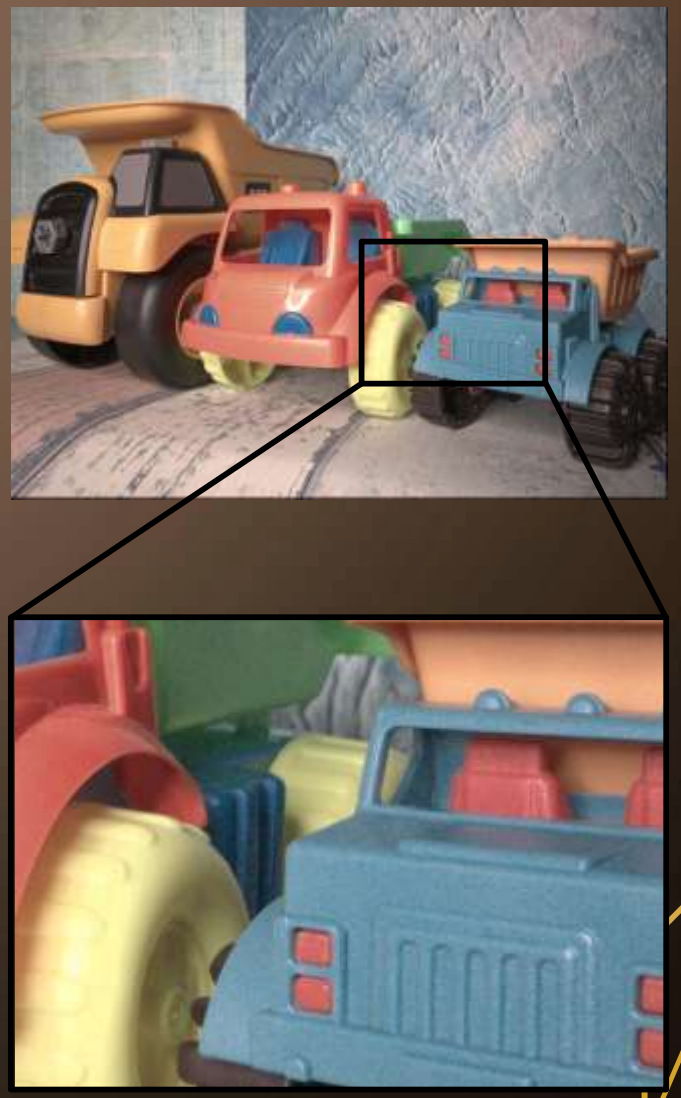
True



Noisy



Denoised



Interpretation:

- Less competitive in time efficiency compared to Gaussian RBF kernel.
- KPCA with Gaussian RBF kernel: 3 minutes per image denoising.
- KPCA with Morlet wavelet kernel: 8 minutes per image denoising on average.
- Other techniques, except CGNet and CBDNet, denoise under 2 minutes.
- Future optimization aims to reduce Morlet kernel's time complexity.
- KPCA with Morlet kernel achieves superior denoising over traditional filters.
- Despite complexity, Morlet kernel outperforms PCA and KPCA with RBF.
- Work is uploaded on *Github* .

Future Scope

- Due to multiplicative nature of the Morlet Wavelet Kernel , we had some issues in implementation, which could be an area of improvement.
- Algorithms like CGNet and CBDNet utilize different pre-processing and implementation techniques alongside kernels to get their results. Implementing those techniques together with the Morlet Kernel may give better results and create a new state of the art method altogether as for base case implementation the Morlet Kernel outperforms other kernels.

Thank You