Algorithms for Reducing Noise in Synthetic Aperture Radar Images

Troy Peterson Kling
& Jeffrey Kidwell

University of North Carolina Wilmington

April 24, 2015
Introduction

Problem Description
   Natural Language Description
   Formal Description

Why CSC 380?

Image Filtering Algorithms

Local Filters
   Boxcar Filter
   Gaussian Convolution
   J. S. Lee’s Filter

Non-local Filters
   Non-local Means
   Fourier Transform
   Modified ART-2

Conclusion

The Numbers
Results for SAR Images
Future Work
Questions
Images of Earth’s surface taken by synthetic aperture radar (SAR) instruments are inherently noisy. In order to perform any kind of analysis or classification on SAR images, it is desirable to first reduce their granularity. This can be achieved by applying noise reduction filters to the images.
Images of Earth’s surface taken by synthetic aperture radar (SAR) instruments are inherently noisy. In order to perform any kind of analysis or classification on SAR images, it is desirable to first reduce their granularity. This can be achieved by applying noise reduction filters to the images.

**Problem description:** Given a noisy synthetic aperture radar image and a selection of image filters (Boxcar, Gaussian convolution, Lee, non-local means, Fourier transform, and modified ART-2 neural networks), the goal is to select the filter that best reduces noise in the image while also retaining image clarity by preserving edge details.
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution
Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2) \quad (1)$$
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2)$$  

(1)

where $z = Ae^{i\theta}$ is a vector with amplitude $A$ and phase $\theta$, and given a set of noise reduction filters $F = \{f_1, f_2, \ldots, f_k\}$, find $f_p \in F$ such that

$$\text{MSE}(f_p(X)) \leq \text{MSE}(f_q(X)) \quad \forall f_q \in F$$  

(2)

and where $T$ is the ideal, non-noisy image.

$$\text{MSE}(f_q(X)) = \sum_{i=0}^{n} \sum_{j=0}^{m} (f_q(X_{ij}) - T_{ij})^2$$  

(3)
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2) \quad (1)$$

where $z = Ae^{i\theta}$ is a vector with amplitude $A$ and phase $\theta$, and given a set of noise reduction filters $F = \{f_1, f_2, \ldots, f_k\}$, find $f_p \in F$ such that

$$MSE(f_p(X)) \leq MSE(f_q(X)) \quad \forall f_q \in F \quad (2)$$
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2)$$  \hspace{1cm} (1)

where $z = Ae^{i\theta}$ is a vector with amplitude $A$ and phase $\theta$, and given a set of noise reduction filters $F = \{f_1, f_2, \ldots, f_k\}$, find $f_p \in F$ such that

$$MSE(f_p(X)) \leq MSE(f_q(X)) \ \forall \ f_q \in F$$  \hspace{1cm} (2)

where

$$MSE(f(X)) = \sum_{i=0}^{m} \sum_{j=0}^{n} \left( f(X_{ij}) - T_{ij} \right)^2$$
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2)$$ (1)

where $z = Ae^{i\theta}$ is a vector with amplitude $A$ and phase $\theta$, and given a set of noise reduction filters $F = \{f_1, f_2, ..., f_k\}$, find $f_p \in F$ such that

$$MSE(f_p(X)) \leq MSE(f_q(X)) \ \forall \ f_q \in F$$ (2)

where

$$MSE(f_q(X)) = \frac{1}{nm} \sum_{i=0}^{n} \sum_{j=0}^{m} (f_q(X_{ij}) - T_{ij})^2$$ (3)
Formal Description

Given an $n$-by-$m$ synthetic aperture radar image, $X$, with speckle noise modeled as a standard complex Gaussian distribution

$$p(z) = \frac{1}{\pi} \exp(-|z|^2)$$  \hspace{1cm} (1)

where $z = Ae^{i\theta}$ is a vector with amplitude $A$ and phase $\theta$, and given a set of noise reduction filters $F = \{f_1, f_2, \ldots, f_k\}$, find $f_p \in F$ such that

$$MSE(f_p(X)) \leq MSE(f_q(X)) \ \forall \ f_q \in F$$  \hspace{1cm} (2)

where

$$MSE(f_q(X)) = \frac{1}{nm} \sum_{i=0}^{n} \sum_{j=0}^{m} (f_q(X_{ij}) - T_{ij})^2$$  \hspace{1cm} (3)

and where $T$ is the ideal, non-noisy image.
Noise reduction has several real world applications, especially in the remote sensing and medical fields.
Why CSC 380?

- Noise reduction has several real world applications, especially in the remote sensing and medical fields.
- We researched and implemented six different algorithms for solving a complex problem.
Why CSC 380?

- Noise reduction has several real world applications, especially in the remote sensing and medical fields.
- We researched and implemented six different algorithms for solving a complex problem.
- We analyzed the performance of each algorithm using Big-O notation and run-time tests.
Noise reduction has several real world applications, especially in the remote sensing and medical fields.

We researched and implemented six different algorithms for solving a complex problem.

We analyzed the performance of each algorithm using Big-O notation and run-time tests.

We found the optimal image filter by comparing the root mean squared error (RMSE) and peak signal-to-noise ratio (PSNR) statistics for each filter.
Boxcar Filter

The Boxcar filter is a non-adaptive local filter. It reduces the overall variation present in an image by setting each pixel’s intensity equal to the average of its neighborhood.
Boxcar Filter

The Boxcar filter is a non-adaptive local filter. It reduces the overall variation present in an image by setting each pixel’s intensity equal to the average of its neighborhood.

Let \( X_i \) be a pixel from the unfiltered image, and let \( Y_i \) be the corresponding pixel in the filtered image. Then the Boxcar filter can be written as follows:

\[
Y_i = \text{Box}(X_i) = \frac{1}{|W_i|} \sum_{j \in W_i} X_j
\]
The Boxcar filter is a non-adaptive local filter. It reduces the overall variation present in an image by setting each pixel’s intensity equal to the average of its neighborhood.

Let \( X_i \) be a pixel from the unfiltered image, and let \( Y_i \) be the corresponding pixel in the filtered image. Then the Boxcar filter can be written as follows:

\[
Y_i = Box(X_i) = \frac{1}{|W_i|} \sum_{j \in W_i} X_j
\]

(4)
Boxcar Filter

The Boxcar filter is a non-adaptive local filter. It reduces the overall variation present in an image by setting each pixel’s intensity equal to the average of its neighborhood.

Let $X_i$ be a pixel from the unfiltered image, and let $Y_i$ be the corresponding pixel in the filtered image. Then the Boxcar filter can be written as follows:

$$Y_i = \text{Box}(X_i) = \frac{1}{|W_i|} \sum_{j \in W_i} X_j \quad (4)$$

where $W_i$ denotes the set of all pixels in pixel $i$’s neighborhood.
Boxcar Filter Results

**Figure**: *Left*: Original image. *Center*: Noise added programmatically. *Right*: Image after applying a 5x5 Boxcar filter.

**RMSE**: 10.444  
**PSNR**: 27.753
Gaussian Convolution

A Gaussian convolution filter is a signal processing technique that convolves a signal $f$ with a Gaussian kernel. Mathematically, this amounts to a Weierstrass transform

$$f(x) = \frac{1}{\sqrt{4\pi}} \int_{-\infty}^{\infty} f(y) e^{-\frac{(x-y)^2}{4}} dy$$

(5)

Gaussian convolution can be expressed as a moving weighted average. The process of applying a Gaussian kernel $K_G$ of radius $r$ to a single pixel, $X_{ij}$, can be written as follows.

$$Y_{ij} = \frac{1}{\sum K_G_{i+r} \sum p=i-r \sum q=j-r} K_G(p-r, q-r) \cdot X_{pq}$$

(6)
Gaussian Convolution

A Gaussian convolution filter is a signal processing technique that convolves a signal $f$ with a Gaussian kernel. Mathematically, this amounts to a Weierstrass transform

$$F(x) = \frac{1}{\sqrt{4\pi}} \int_{-\infty}^{\infty} f(y)e^{-\frac{(x-y)^2}{4}} dy$$  \hspace{1cm} (5)
Gaussian Convolution

A Gaussian convolution filter is a signal processing technique that convolves a signal $f$ with a Gaussian kernel. Mathematically, this amounts to a Weierstrass transform

$$F(x) = \frac{1}{\sqrt{4\pi}} \int_{-\infty}^{\infty} f(y) e^{-\frac{(x-y)^2}{4}} dy \quad (5)$$

Gaussian convolution can be expressed as a moving weighted average. The process of applying a Gaussian kernel $K_G$ of radius $r$ to a single pixel, $X_{ij}$, can be written as follows.
A Gaussian convolution filter is a signal processing technique that convolves a signal $f$ with a Gaussian kernel. Mathematically, this amounts to a Weierstrass transform

$$F(x) = \frac{1}{\sqrt{4\pi}} \int_{-\infty}^{\infty} f(y)e^{-\frac{(x-y)^2}{4}} dy$$  \hspace{1cm} (5)$$

Gaussian convolution can be expressed as a moving weighted average. The process of applying a Gaussian kernel $K_G$ of radius $r$ to a single pixel, $X_{ij}$, can be written as follows.

$$Y_{ij} = \frac{1}{\sum K_G} \sum_{p=i-r}^{i+r} \sum_{q=j-r}^{j+r} K_G(p-r, q-r) \cdot X_{pq}$$  \hspace{1cm} (6)$$
Gaussian Convolution

The following is an example of a 5x5 Gaussian convolution filter used in the image processing domain.
The following is an example of a 5x5 Gaussian convolution filter used in the image processing domain.

\[
Y_{ij} = \frac{1}{159} \begin{bmatrix}
2 & 4 & 5 & 4 & 2 \\
4 & 9 & 12 & 9 & 4 \\
5 & 12 & 15 & 12 & 5 \\
4 & 9 & 12 & 9 & 4 \\
2 & 4 & 5 & 4 & 2
\end{bmatrix} * W \quad (7)
\]
Gaussian Convolution

The following is an example of a 5x5 Gaussian convolution filter used in the image processing domain.

$$Y_{ij} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * W \quad (7)$$

This process is repeated for each pixel in the image.
Gaussian Convolution Results

Figure: **Left:** Original image. **Center:** Noise added programmatically. **Right:** Image after applying a 5x5 Gaussian convolution filter, $\sigma = 1.25$.

**RMSE:** 9.506  
**PSNR:** 28.571
J. S. Lee’s Filter

The Lee filter is an adaptive local filter. Compared to the Boxcar filter, the Lee filter performs better at preserving sharp edges and point scatterers. It can be written as follows.

\[ Y_i = \text{Lee}(X_i) = \text{Box}(X_i) + k(X_i - \text{Box}(X_i)) \] (8)

where \( k \) is an adaptive filtering coefficient defined by

\[ k = \frac{\text{Var}(Y)}{\text{Var}(W)} = \frac{\text{Var}(W)}{1 + \sigma^2} \] (9)

where \( \text{Var}() \) is the variance operator, \( \text{E}() \) is the expectation, and \( \sigma^2 \) is the a priori variance of the speckle noise.
J. S. Lee’s Filter

The Lee filter is an adaptive local filter. Compared to the Boxcar filter, the Lee filter performs better at preserving sharp edges and point scatterers. It can be written as follows.

\[ Y_i = Lee(X_i) = Box(X_i) + k(X_i - Box(X_i)) \] (8)
J. S. Lee’s Filter

The Lee filter is an adaptive local filter. Compared to the Boxcar filter, the Lee filter performs better at preserving sharp edges and point scatterers. It can be written as follows.

\[ Y_i = \text{Lee}(X_i) = \text{Box}(X_i) + k(X_i - \text{Box}(X_i)) \quad (8) \]

where \( k \) is an adaptive filtering coefficient defined by

\[ k = \frac{\text{Var}(Y)}{\text{Var}(W)} = \frac{\text{Var}(W) - \text{E}^2(W)}{\sigma^2 \left( 1 + \sigma^2 \right)} \quad (9) \]

where \( \text{Var}(\cdot) \) is the variance operator, \( \text{E}(\cdot) \) is the expectation, and \( \sigma^2 \) is the \textit{a priori} variance of the speckle noise.
The Lee filter is an adaptive local filter. Compared to the Boxcar filter, the Lee filter performs better at preserving sharp edges and point scatterers. It can be written as follows.

\[ Y_i = \text{Lee}(X_i) = \text{Box}(X_i) + k(X_i - \text{Box}(X_i)) \]  

(8)

where \( k \) is an adaptive filtering coefficient defined by

\[ k = \frac{\text{Var}(Y)}{\text{Var}(W)} = \frac{\text{Var}(W) - E^2(W)\sigma^2}{\text{Var}(W)[1 + \sigma^2]} \]  

(9)
J. S. Lee’s Filter

The Lee filter is an adaptive local filter. Compared to the Boxcar filter, the Lee filter performs better at preserving sharp edges and point scatterers. It can be written as follows.

\[ Y_i = Lee(X_i) = Box(X_i) + k(X_i - Box(X_i)) \]  

(8)

where \( k \) is an adaptive filtering coefficient defined by

\[ k = \frac{\text{Var}(Y)}{\text{Var}(W)} = \frac{\text{Var}(W) - E^2(W)\sigma^2}{\text{Var}(W)[1 + \sigma^2]} \]  

(9)

where \( \text{Var}(\cdot) \) is the variance operator, \( E(\cdot) \) is the expectation, and \( \sigma^2 \) is the \textit{a priori} variance of the speckle noise.
J. S. Lee’s Filter Results

Figure: Left: Original image. Center: Noise added programmatically. Right: Image after applying a 5x5 Lee filter.

RMSE: 8.740
PSNR: 29.300
Non-local Means

- Non-local filters consider all pixels in the *entire* image.
Non-local Means

- Non-local filters consider all pixels in the *entire* image.
- Similarity values are determined by comparing small windows.
Non-local Means

- Non-local filters consider all pixels in the *entire* image.
- Similarity values are determined by comparing small windows.
- Using this knowledge about similarity, non-local filters can more accurately estimate true pixel intensities.
Non-local Means

- Non-local filters consider all pixels in the entire image.
- Similarity values are determined by comparing small windows.
- Using this knowledge about similarity, non-local filters can more accurately estimate true pixel intensities.
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = \text{NL}(X_i) = \sum_{j \in W_i} w(i, j) X_j \]

where \( w(i, j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i, j) = \frac{1}{Z(i)} e^{-\frac{d_2(i, j)}{h^2}} \]

where \( h \) is a tuning parameter, \( d_2(i, j) \) is the Euclidean norm, and \( Z(i) = \sum_j e^{-\frac{d_2(i, j)}{h^2}} \).
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = NL(X_i) = \sum_{j \in W_i} w(i, j)X_j \] (10)

where \( w(i, j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i, j) = \frac{1}{Z(i)} e^{-\frac{d^2(i, j)}{2h^2}} \] (11)

where \( h \) is a tuning parameter, \( d^2(i, j) \) is the Euclidean norm, and \( Z(i) = \sum_{j} e^{-\frac{d^2(i, j)}{2h^2}} \) (12)
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = NL(X_i) = \sum_{j \in W_i} w(i, j) X_j \]  \hspace{1cm} (10)

where \( w(i, j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i, j) = \frac{1}{Z(i)} e^{-\frac{d^2(i, j)}{h^2}} \]  \hspace{1cm} (11)

where \( h \) is a tuning parameter, \( d^2(i, j) \) is the Euclidean norm, and \( Z(i) \) is the normalizing constant.

\[ Z(i) = \sum_j e^{-\frac{d^2(i, j)}{h^2}} \]  \hspace{1cm} (12)
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = NL(X_i) = \sum_{j \in W_i} w(i, j)X_j \quad (10) \]

where \( w(i, j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i, j) = \frac{1}{Z(i)} e^{-\frac{d^2(i,j)}{h^2}} \quad (11) \]
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = NL(X_i) = \sum_{j \in W_i} w(i, j)X_j \quad (10) \]

where \( w(i, j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i, j) = \frac{1}{Z(i)} e^{-\frac{d^2(i, j)}{h^2}} \quad (11) \]

where \( h \) is a tuning parameter, \( d^2(i, j) \) is the Euclidean norm, and \( Z(i) \) is the normalizing constant.
Non-local Means

In the non-local means paradigm, each pixel is assigned a new value equal to a weighted sum of all pixels in the image.

\[ Y_i = NL(X_i) = \sum_{j \in W_i} w(i,j)X_j \]  \hspace{1cm} (10)

where \( w(i,j) \) is a weight associated with each pair of pixels, calculated as follows.

\[ w(i,j) = \frac{1}{Z(i)} e^{-\frac{d^2(i,j)}{h^2}} \]  \hspace{1cm} (11)

where \( h \) is a tuning parameter, \( d^2(i,j) \) is the Euclidean norm, and \( Z(i) \) is the normalizing constant

\[ Z(i) = \sum_j e^{-\frac{d^2(i,j)}{h^2}} \]  \hspace{1cm} (12)
Non-Local Means Results

Figure: Left: Original image. Center: Noise added programmatically. Right: Image after applying a non-local means filters, using 5x5 similarity windows, 35x35 search windows, and $h = 13.5$.

**RMSE:** 6.158  
**PSNR:** 32.342
Fourier Transform

Fourier transforms are used to transform a signal from the time domain to the frequency domain.
Fourier transforms are used to transform a signal from the time domain to the frequency domain.

\[
f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi ikx} \, dk\] (13)
Fourier Transforms

Fourier transforms are used to transform a signal from the time domain to the frequency domain.

\[ f(x) = \int_{-\infty}^{\infty} F(k)e^{2\pi ikx} \, dk \]  

(13)

This transformation can be reversed with the inverse Fourier transform.
Fourier Transforms are used to transform a signal from the time domain to the frequency domain.

\[
f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} \, dk
\]  

(13)

This transformation can be reversed with the inverse Fourier transform

\[
F(x) = \int_{-\infty}^{\infty} f(k) e^{-2\pi i k x} \, dk
\]  

(14)
Fourier Transform

Fourier transforms are used to transform a signal from the time domain to the frequency domain.

\[ f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} \, dk \]  

(13)

This transformation can be reverse with the inverse Fourier transform

\[ F(x) = \int_{-\infty}^{\infty} f(k) e^{-2\pi i k x} \, dk \]  

(14)

Noise reduction can be achieved by taking the Fast Fourier Transform (FFT) of an image, filtering certain frequencies, and then performing the inverse FFT.
Fourier Transform Results

**Figure:** *Left:* Original image. *Center:* Noise added programmatically. *Right:* Image after applying a band-pass Fourier transform.

**RMSE:** 10.197  
**PSNR:** 27.961
ART-2

The Adapative Resonance Theory (ART-2) neural network is specifically for pattern recognition.
ART-2

- The Adapative Resonance Theory (ART-2) neural network is specifically for pattern recognition.
- ART-2 is able to categorize continuous input patterns into different groups based on similarity.
ART-2

- The Adapative Resonance Theory (ART-2) neural network is specifically for pattern recognition.
- ART-2 is able to categorize continuous input patterns into different groups based on similarity.
- $F_1$ layer acts as short-term memory, $F_2$ as long-term.
ART-2

- The Adapative Resonance Theory (ART-2) neural network is specifically for pattern recognition.
- ART-2 is able to categorize continuous input patterns into different groups based on similarity.
- $F_1$ layer acts as short-term memory, $F_2$ as long-term.
- Black circles indicate normalization.
ART-2

Short-term memory equations
Short-term memory equations

\[ p_i = u_i + \sum_j g(y_j)z_{ji} \]
\[ u_i = \frac{v_i}{e + ||v_i||} \]
\[ w_i = l_i + au_i \]
\[ q_i = \frac{p_i}{e + ||p_i||} \]
\[ v_i = f(x_i) + bf(q_i) \]
\[ x_i = \frac{w_i}{e + ||w_i||} \]
Short-term memory equations

\[ p_i = u_i + \sum_j g(y_j)z_{ji} \]
\[ u_i = \frac{v_i}{e + \|v_i\|} \]
\[ w_i = l_i + au_i \]

\[ q_i = \frac{p_i}{e + \|p_i\|} \]
\[ v_i = f(x_i) + bf(q_i) \]
\[ x_i = \frac{w_i}{e + \|w_i\|} \]

Filter function

\[ f(x) = \begin{cases} 0 & \text{if } 0 \leq x < \theta \\ x & \text{if } x \geq \theta \end{cases} \]
Short-term memory equations

\[ p_i = u_i + \sum_j g(y_j)z_{ji} \]

\[ u_i = \frac{v_i}{e + \|v_i\|} \]

\[ w_i = l_i + a u_i \]

\[ q_i = \frac{p_i}{e + \|p_i\|} \]

\[ v_i = f(x_i) + bf(q_i) \]

\[ x_i = \frac{w_i}{e + \|w_i\|} \]

Filter function

\[ f(x) = \begin{cases} 
0 & \text{if } 0 \leq x < \theta \\
x & \text{if } x \geq \theta 
\end{cases} \quad (15) \]
Short-term memory equations

\[ p_i = u_i + \sum_j g(y_j)z_{ji} \]
\[ u_i = \frac{v_i}{e + \|v_i\|} \]
\[ w_i = l_i + au_i \]

\[ q_i = \frac{p_i}{e + \|p_i\|} \]
\[ v_i = f(x_i) + bf(q_i) \]
\[ x_i = \frac{w_i}{e + \|w_i\|} \]

Filter function

\[ f(x) = \begin{cases} 
0 & \text{if } 0 \leq x < \theta \\
x & \text{if } x \geq \theta 
\end{cases} \quad (15) \]

And lots more!
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
- We are in the process of modifying it so that it has similar properties to the non-local means filter.
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
- We are in the process of modifying it so that it has similar properties to the non-local means filter.
- The modified ART-2 neural network takes a sequence of overlapping windows as input patterns.
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
- We are in the process of modifying it so that it has similar properties to the non-local means filter.
- The modified ART-2 neural network takes a sequence of overlapping windows as input patterns.
- Performing categorization on these input patterns tells us which windows are most similar.
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
- We are in the process of modifying it so that it has similar properties to the non-local means filter.
- The modified ART-2 neural network takes a sequence of overlapping windows as input patterns.
- Performing categorization on these input patterns tells us which windows are most similar.
- From there, principles from non-local means can be applied to reduce noise in the image.
Modified ART-2

- ART-2 is capable of performing basic noise reduction.
- We are in the process of modifying it so that it has similar properties to the non-local means filter.
- The modified ART-2 neural network takes a sequence of overlapping windows as input patterns.
- Performing categorization on these input patterns tells us which windows are most similar.
- From there, principles from non-local means can be applied to reduce noise in the image.
- No results yet.
Final Remarks

- Non-local means outperforms all other filters by a sizeable margin.

<table>
<thead>
<tr>
<th>Filter</th>
<th>RMSE</th>
<th>PSNR</th>
<th>Big-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL-M</td>
<td>6.16</td>
<td>32.34</td>
<td>$O(n^2 w^2 k^2)$</td>
</tr>
<tr>
<td>Lee’s</td>
<td>8.74</td>
<td>29.3</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>Conv.</td>
<td>9.51</td>
<td>28.57</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>Fourier</td>
<td>10.20</td>
<td>27.96</td>
<td>$O(n^2 \log(n))$</td>
</tr>
<tr>
<td>Boxcar</td>
<td>10.44</td>
<td>27.75</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>ART-2</td>
<td>N/A</td>
<td>N/A</td>
<td>O(?)</td>
</tr>
</tbody>
</table>

Note: Assuming the image is square.
Final Remarks

- Non-local means outperforms all other filters by a sizeable margin.
- Non-local means is particularly well-suited for SAR images.

<table>
<thead>
<tr>
<th>Filter</th>
<th>RMSE</th>
<th>PSNR</th>
<th>Big-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL-M</td>
<td>6.16</td>
<td>32.34</td>
<td>$O(n^2 w^2 k^2)$</td>
</tr>
<tr>
<td>Lee’s</td>
<td>8.74</td>
<td>29.3</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>Conv.</td>
<td>9.51</td>
<td>28.57</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>Fourier</td>
<td>10.20</td>
<td>27.96</td>
<td>$O(n^2 \log(n))$</td>
</tr>
<tr>
<td>Boxcar</td>
<td>10.44</td>
<td>27.75</td>
<td>$O(n^2 k^2)$</td>
</tr>
<tr>
<td>ART-2</td>
<td>N/A</td>
<td>N/A</td>
<td>$O(?)$</td>
</tr>
</tbody>
</table>

Note: Assuming the image is square.
Final Remarks

- Non-local means outperforms all other filters by a sizeable margin.
- Non-local means is particularly well-suited for SAR images.
- We expect our modified ART-2 algorithm to perform better than Lee’s filter.

<table>
<thead>
<tr>
<th>Filter</th>
<th>RMSE</th>
<th>PSNR</th>
<th>Big-O</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL-M</td>
<td>6.16</td>
<td>32.34</td>
<td>$O(n^2w^2k^2)$</td>
</tr>
<tr>
<td>Lee’s</td>
<td>8.74</td>
<td>29.3</td>
<td>$O(n^2k^2)$</td>
</tr>
<tr>
<td>Conv.</td>
<td>9.51</td>
<td>28.57</td>
<td>$O(n^2k^2)$</td>
</tr>
<tr>
<td>Fourier</td>
<td>10.20</td>
<td>27.96</td>
<td>$O(n^2 \log(n))$</td>
</tr>
<tr>
<td>Boxcar</td>
<td>10.44</td>
<td>27.75</td>
<td>$O(n^2k^2)$</td>
</tr>
<tr>
<td>ART-2</td>
<td>N/A</td>
<td>N/A</td>
<td>O(?)</td>
</tr>
</tbody>
</table>

Note: Assuming the image is square.
Results for SAR Images

- So far we have seen image filters applied to standard photographs.
Results for SAR Images

- So far we have seen image filters applied to standard photographs.
- How well do these same filters work on synthetic aperture radar images?
Results for SAR Images

- So far we have seen image filters applied to standard photographs.
- How well do these same filters work on synthetic aperture radar images?
- Since we don’t know what the “true” de-noised image looks like, we have to judge it by eye.
Results for SAR Images

Figure: *Top row*: Original SAR image; Boxcar; Gaussian convolution. *Bottom row*: Lee’s filter; non-local means; FFT filter.
Results for SAR Images

Figure: Top row: Original SAR image; Boxcar; Gaussian convolution. Bottom row: Lee’s filter; non-local means; FFT filter.
Future Work

- Finish implementing ART-2.
- Try wavelet transforms instead of Fourier transforms.
- Make filters work for polarimetric and/or interferometric SAR images.
Future Work

- Finish implementing ART-2.
- Try wavelet transforms instead of Fourier transforms.
Future Work

- Finish implementing ART-2.
- Try wavelet transforms instead of Fourier transforms.
- Make filters work for polarimetric and/or interferometric SAR images.
1. What is the difference between local and non-local filters, and why do non-locals filters perform better in most situations?
Question #1

1. What is the difference between local and non-local filters, and why do non-locals filters perform better in most situations?

**Answer:** Local filters estimate each pixel’s true intensity by looking at the pixels immediately surrounding it. Non-local filters estimate each pixel’s true intensity by looking at every pixel in the *entire* image and performing a weighted average based on the degree of similarity between windows.
2. The non-local means filter gave the lowest RMSE out of all the filters we tried, but it's not the best in every respect. What is one major drawback of the non-local means algorithm? (Hint: Think Big-O notation)
2. The non-local means filter gave the lowest RMSE out of all the filters we tried, but it's not the best in every respect. What is one major drawback of the non-local means algorithm? (*Hint: Think Big-O notation*)

**Answer:** The non-local means algorithm has complexity $O(n^2 \cdot w^2 \cdot k^2)$, where $n$ is the width of the image (assuming it is square), $w$ is the width of the search window, and $k$ is the width of similarity window. For even modestly large values of $n$, $w$, and $k$, non-local means takes a very long time to finish running.
3. For the Lee filter, how does the adaptive filtering coefficient, $k$, affect the filtering procedure?
3. For the Lee filter, how does the adaptive filtering coefficient, \( k \), affect the filtering procedure?

**Answer:** Homogeneous regions of the image produce \( k \) values close to zero, which causes the Lee filter to act more like a Boxcar filter. Heterogeneous regions of the image that contain a lot of variation produce \( k \) values close to one, which sets the filtered pixel value equal to the unfiltered pixel value.
4. What was the ART-2 system originally designed for, and how can it be applied to noise reduction?
4. What was the ART-2 system originally designed for, and how can it be applied to noise reduction?

**Answer:** The ART-2 neural network was originally designed for categorization/classification of signal patterns. ART-2 can be applied to noise reduction in several ways - one of which is by categorizing overlapping windows in the image and then applying a filter function for each category.