Methods of Images Contrast Enhancement

NLA course project

Snayperskaya Vintovka Dragunova - SVD

1. Problem statement

Have: Brain magnetic resonance images

Problem: Images affected by noise, artefacts, speckles, poor quality, low contarast

To do: Make image more contrasted

Formal problem statement

Given: A - 2-dimensional array with size 256×256. a_{ij} - color value of pixel (i, j), $a_{ij} \in [0, 255]$, where 0 is black and 255 is white.

Output: $\hat{A} = F(A) -$ transformed 2-dimensional array with size 256×256, which minimizes the metric Q:

$$\min_F Q(\hat{A}, A),$$

where $Q \in \{\mathbf{PSNR}, \mathbf{QRCM}, \mathbf{SSIM}, \mathbf{FSIM}, \mathbf{AMBE}, \mathbf{EME}\}$ and F is an enhancement transformation.

Example of brain MRI

In [16]:

```
interact(vary_coordinate,
         coordinate_axial = sld_axial,
         coordinate_sagital = sld_sagital,
         coordinate_coronal = sld_coronal,
         axis = 'axial');
```

2 . Ideas and mathematical description of the algorithms

2.1 Histogram Equalization

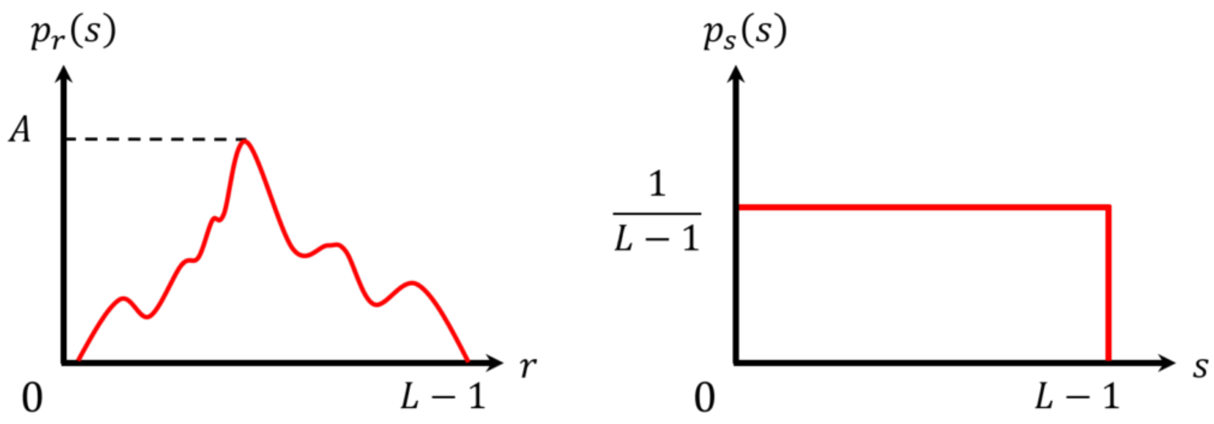
- The technique which includes the transformation $T(i) = 255 \cdot cdf(i)$, where cdf – cumulative sum of all the probabilities.

$$cdf(x) = \sum_{k=0}^{x} p_k$$

$$p_k = \frac{number\ of\ pixels\ with\ intensity\ k}{total\ number\ of\ pixels}, \quad k = 0, 1, \dots, 255$$

We apply the GHE method to low contrast T1-w (**A1**) brain MRI in order to have equalized images referred as **A2**.

Histogram equalisation example



```
In [26]: A1_to_cv = np.uint8(cv2.normalize(A1, None, 0, 255, cv2.NORM_MINMAX))
         A2 = cv2.equalizeHist(A1_to_cv)
```

2.2 Discrete wavelet transform

- Discrete wavelet transform (DWT) - technique for analysis, de-noising and compression of signals and images.
- Advantage - can choose the signal's coefficients with a significant energy and discards the others that have a very low percentage of all energy.
- Daubechie wavelet functions used: $9/7$ tap and 1 tap.
- Obtain four frequency sub-bands **LL**, **LH**, **HL**, **HH** for **A1** and **A2**.

In [49]:
```
titles = ['LL1', 'LH1', 'HL1', 'HH1']

w = pywt.Wavelet('db1')
coeffs = pywt.dwt2(A1, w)

LL1, (LH1, HL1, HH1) = coeffs

fig = plt.figure(figsize=(16, 4))
for i, a in enumerate([LL1, LH1, HL1, HH1]):
    ax = fig.add_subplot(1, 4, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
```

2.3 Singular value decomposition

Singular Value Decomposition (SVD): any matrix $A \in \mathbb{C}^{n \times m}$ can be represented as the product of three matrices:

$$A = U \cdot \Sigma \cdot V^*$$

where U, V are unitary and Σ is diagonal with singular values of A on the diagonal.

- **LL** - intensity information
- **LH**, **HL**, **HH** - edge information
- Hence, apply SVD to **LL** subbands to modify intensity and protect edges

In [32]:
```
U1, S1, V1 = np.linalg.svd(LL1, full_matrices = False)
U2, S2, V2 = np.linalg.svd(LL2, full_matrices = False)
```

Modifying intensity

- Singular value matrix contains intensity information $\Rightarrow$ update it
- Correction coefficient ξ

In 2014, Bhandari

$$\xi = \frac{\max(U_{LL2})}{\max(U_{LL1})} \quad \Rightarrow \quad \Sigma = \xi \cdot \Sigma_{LL2}$$

In 2016, Bhandari

$$\xi = \frac{\max(U_{LL2}) + \max(V_{LL2})}{\max(U_{LL1}) + \max(V_{LL1})} \quad \Rightarrow \quad \Sigma = \xi \cdot \Sigma_{LL2}$$

In 2019, M. Sahnoun

$$\xi = \frac{\max(U_{LL2}) + \max(V_{LL2})}{\max(U_{LL1}) + \max(V_{LL1})} \quad \Rightarrow \quad \Sigma = \mu \cdot \xi \cdot \Sigma_{LL1} + (1-\mu) \cdot \frac{1}{\xi} \cdot \Sigma_{LL2}$$

Reconstracting image

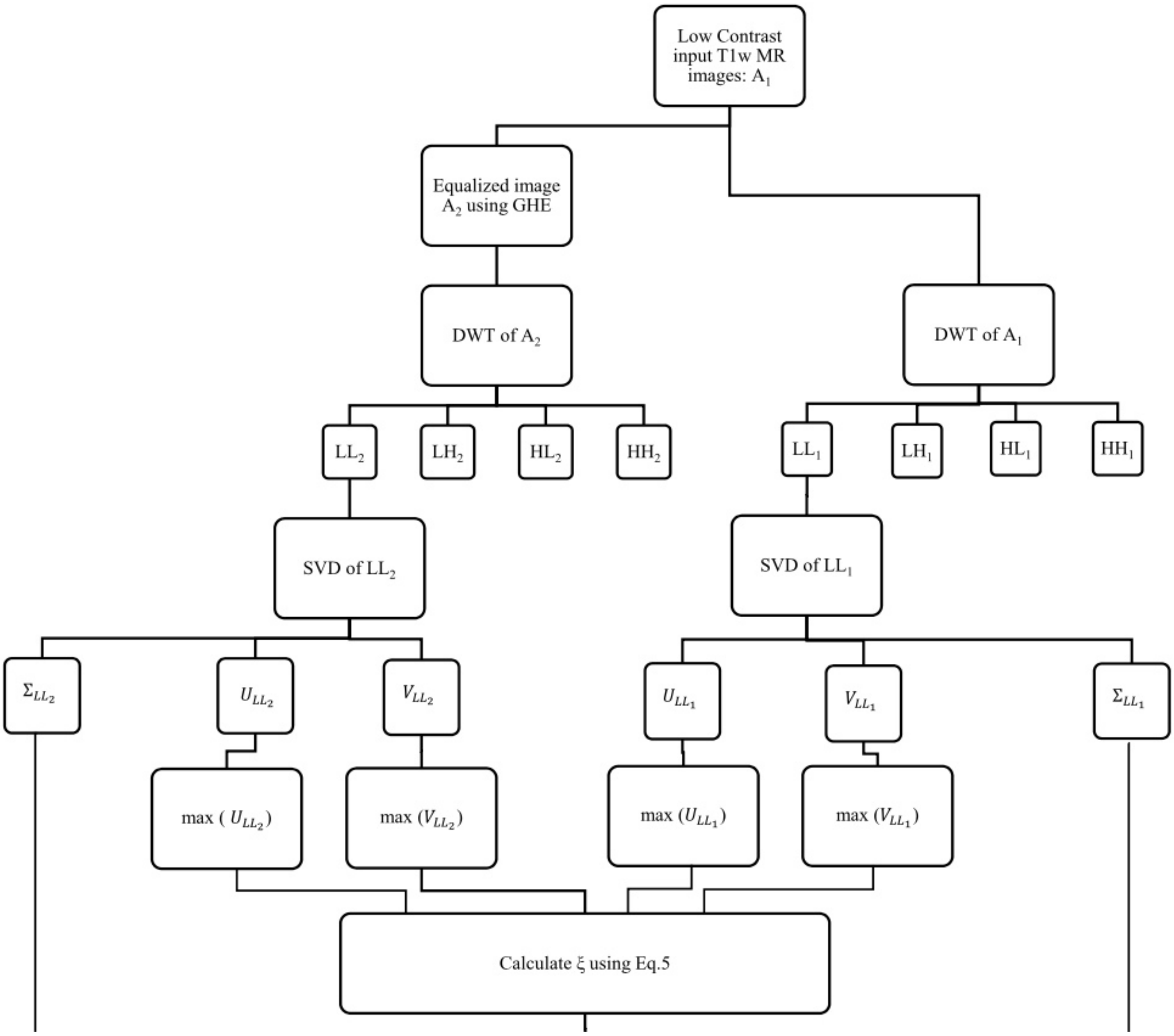
ISVD:

$$LL = U_{LL2} \Sigma V^*_{LL2}$$

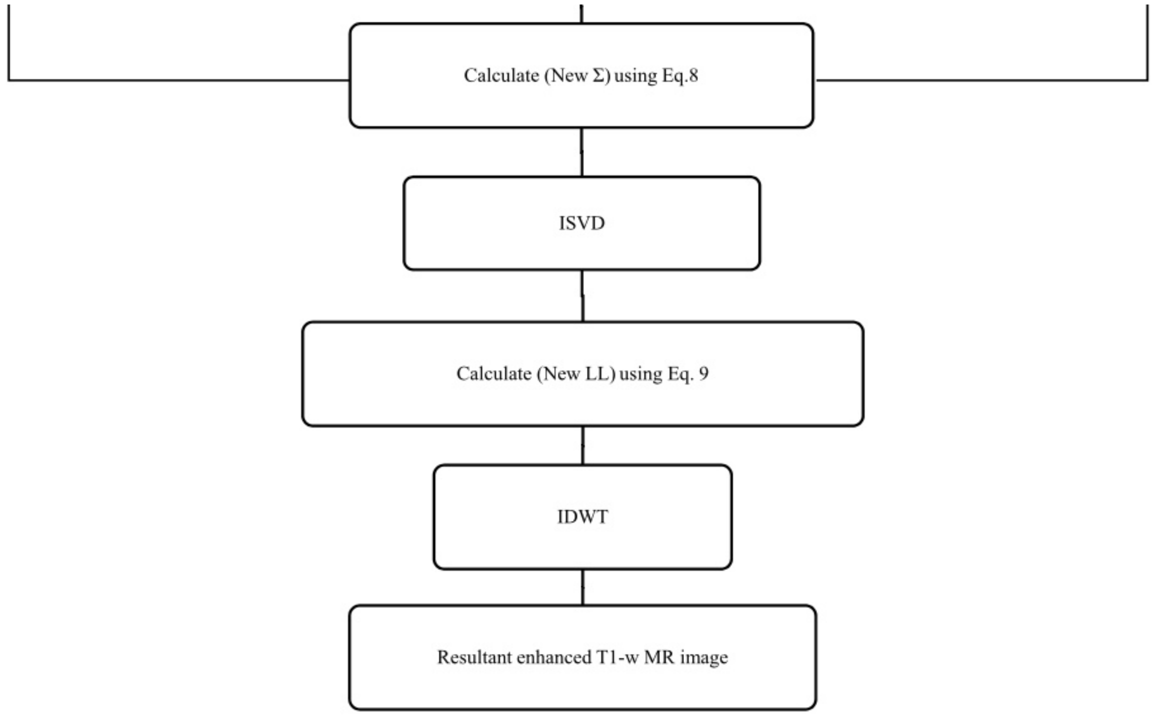
IDWT:

$$\hat{A}2 = \mathbf{IDWT}(LL2, LH2, HL2, HH2)$$

Workflow



[source (https://www.sciencedirect.com/science/article/pii/S1959031819301290)](https://www.sciencedirect.com/science/article/pii/S1959031819301290)



source (https://www.sciencedirect.com/science/article/pii/S1959031819301290)

3. Experiments

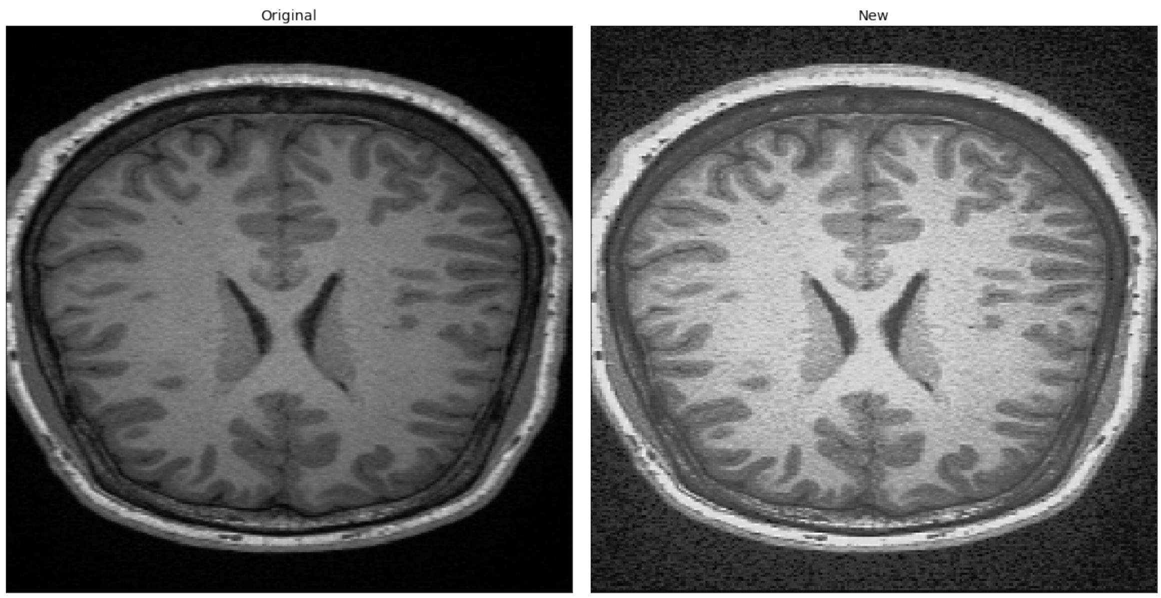
In 2014, Bhandari

$$\xi = \frac{\max\left(U_{LL2}\right)}{\max\left(U_{LL1}\right)} \quad \Rightarrow \quad \Sigma = \xi \cdot \Sigma_{LL2}$$

In [82]:

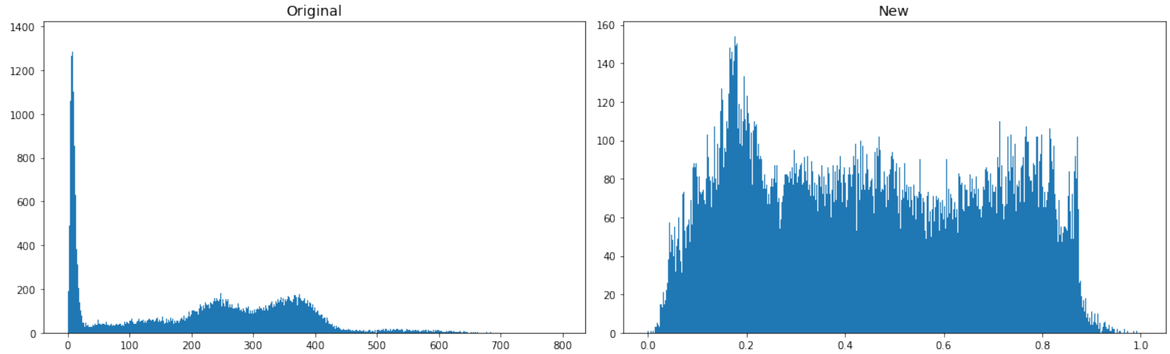
```
new_S = new_sigma1(S2, xi_2)
LL = new_LL(new_S)
brain1 = pywt.idwt2((LL, (LH2, HL2, HH2)), w)

titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 8))
for i, a in enumerate([A1, brain1]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=14)
    ax.set_xticks([])
    ax.set_yticks([])
fig.tight_layout()
plt.show()
```



In [84]:

```
titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 5))
for i, a in enumerate([A1, brain1]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.hist(a.reshape(-1), bins = 1000)
    ax.set_title(titles[i], fontsize=14)
fig.tight_layout()
plt.show()
```



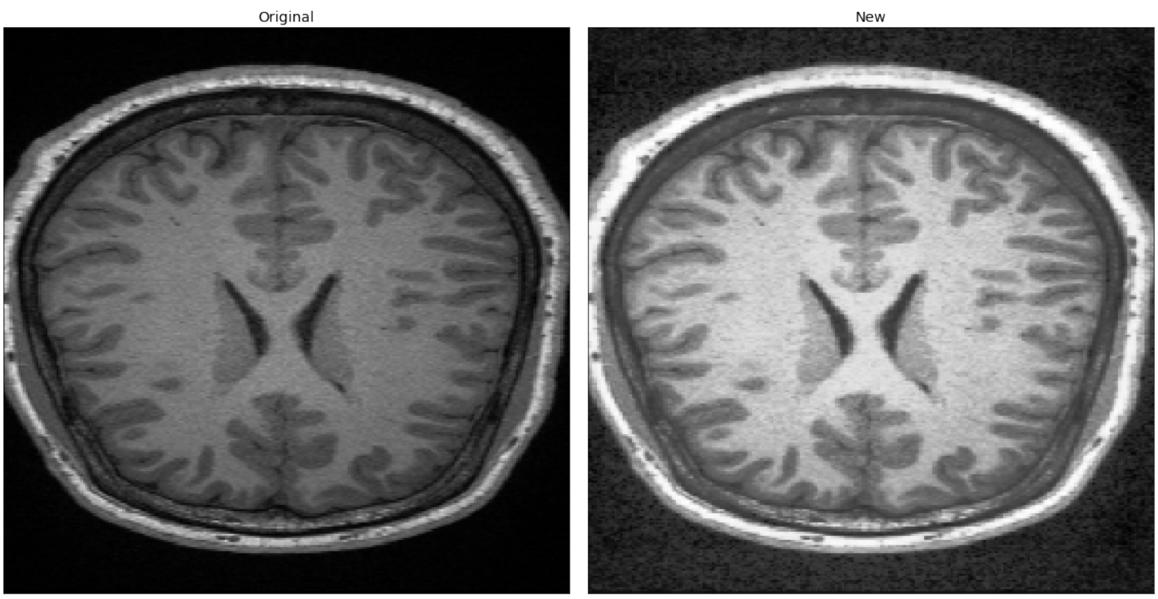
In 2015, Randa Atta

$$\xi = \frac{\max\left(\Sigma_{LL2}\right)}{\max\left(\Sigma_{LL1}\right)} \quad \Rightarrow \quad \Sigma = 0.5 \cdot \left(\xi \cdot \Sigma_{LL1} + \frac{1}{\xi} \cdot \Sigma_{LL2}\right)$$

In [85]:

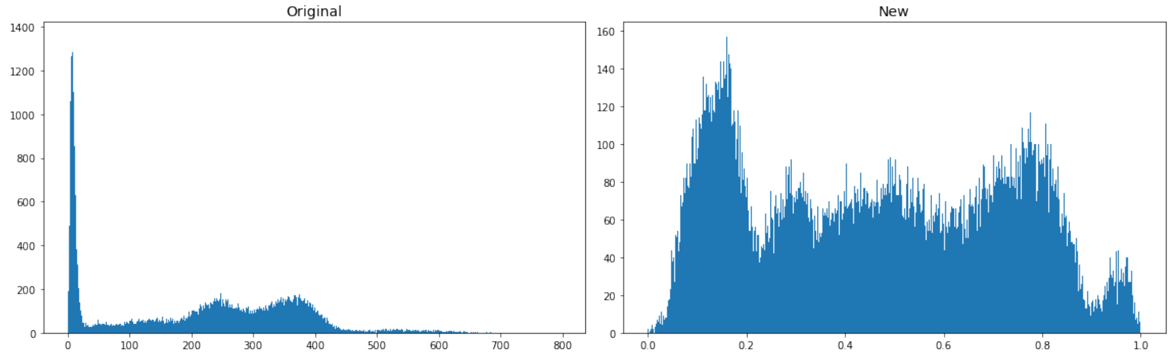
```
new_S = new_sigma2(S1, S2, xi_3)
LL = new_LL(new_S)
brain2 = pywt.idwt2((LL, (LH2, HL2, HH2)), w)

titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 8))
for i, a in enumerate([A1, brain2]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=14)
    ax.set_xticks([])
    ax.set_yticks([])
fig.tight_layout()
plt.show()
```



In [87]:

```
titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 5))
for i, a in enumerate([A1, brain2]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.hist(a.reshape(-1), bins = 1000)
    ax.set_title(titles[i], fontsize=14)
fig.tight_layout()
plt.show()
```

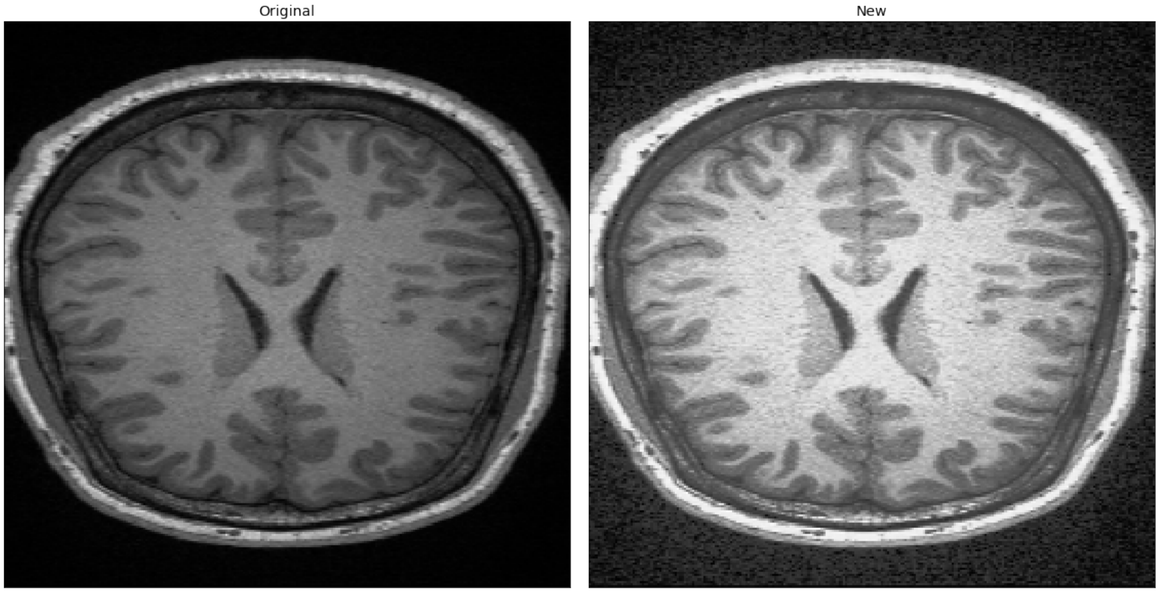


In 2016, Bhandari

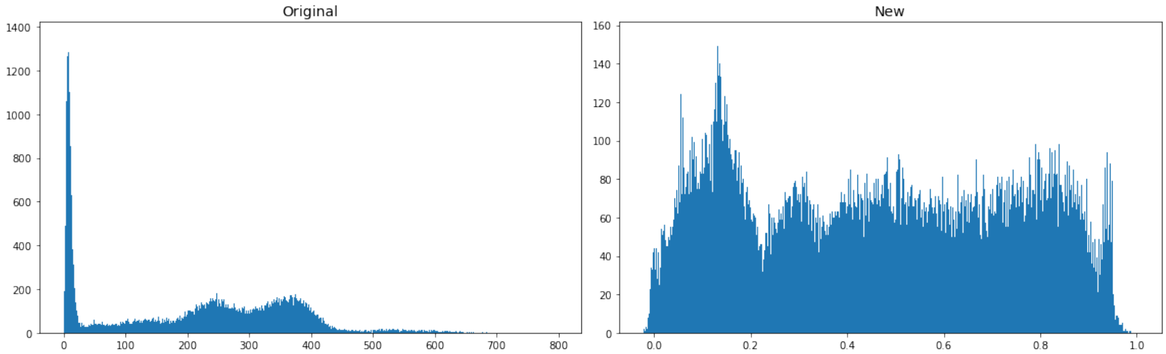
$$\xi = \frac{\max\left(U_{LL2}\right) + \max\left(V_{LL2}\right)}{\max\left(U_{LL1}\right) + \max\left(V_{LL1}\right)} \quad \Rightarrow \quad \Sigma = \xi \cdot \Sigma_{LL2}$$

In [88]:
```
new_S = new_sigma1(S2, xi_3)
LL = new_LL(new_S)
brain3 = pywt.idwt2((LL, (LH2, HL2, HH2)), w)

titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 8))
for i, a in enumerate([A1, brain3]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=14)
    ax.set_xticks([])
    ax.set_yticks([])
fig.tight_layout()
plt.show()
```



In [90]:
```
titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 5))
for i, a in enumerate([A1, brain3]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.hist(a.reshape(-1), bins = 1000)
    ax.set_title(titles[i], fontsize=14)
fig.tight_layout()
plt.show()
```



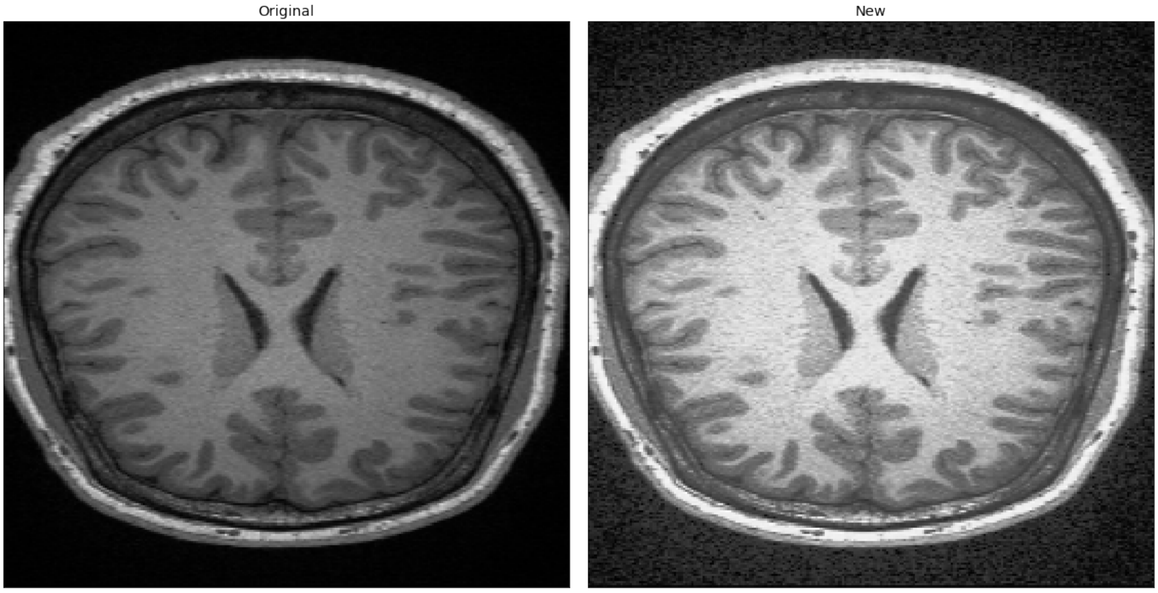
In 2019, M. Sahnoun

$$\xi = \frac{\max\left(U_{LL2}\right) + \max\left(V_{LL2}\right)}{\max\left(U_{LL1}\right) + \max\left(V_{LL1}\right)} \quad \Rightarrow \quad \Sigma = \mu \cdot \xi \cdot \Sigma_{LL1} + (1-\mu) \cdot \frac{1}{\xi} \cdot \Sigma_{LL2}$$

In [95]:

```
mu = 0.1
new_S = new_sigma3(S1, S2, xi_3, mu)
LL = new_LL(new_S)
brain4 = pywt.idwt2((LL, (LH2, HL2, HH2)), w)

titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 8))
for i, a in enumerate([A1, brain3]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=14)
    ax.set_xticks([])
    ax.set_yticks([])
fig.tight_layout()
plt.show()
```



In [98]:
```
titles = ['Original', 'New']
fig = plt.figure(figsize=(16, 5))
for i, a in enumerate([A1, brain4]):
    ax = fig.add_subplot(1, 2, i + 1)
    ax.hist(a.reshape(-1), bins = 1000)
    ax.set_title(titles[i], fontsize=14)
fig.tight_layout()
plt.show()
```

4. Alternative methods

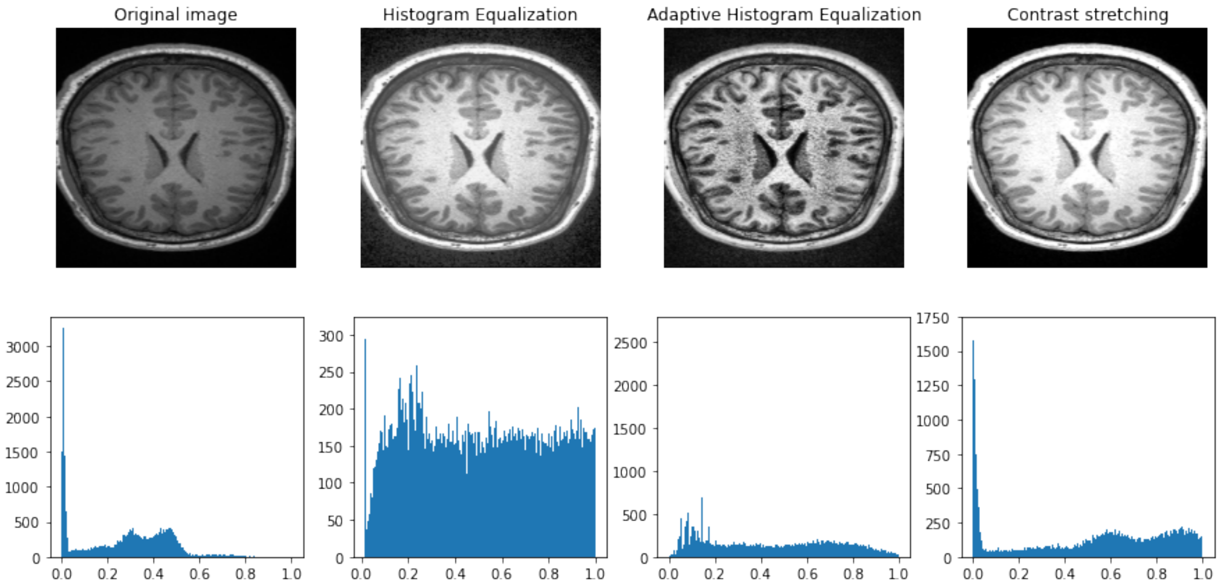
- Histogram Equalization
- Adaptive Histogram Equalization
- Contrast stretching

Simple image enhancement technique which improves the contrast in an image by "stretching" the range of intensity values it contains to span a desired range of values.

Algorithm scans the image to find the lowest (c) and highest (d) pixel values Each pixel color i is scaled using in the following way:

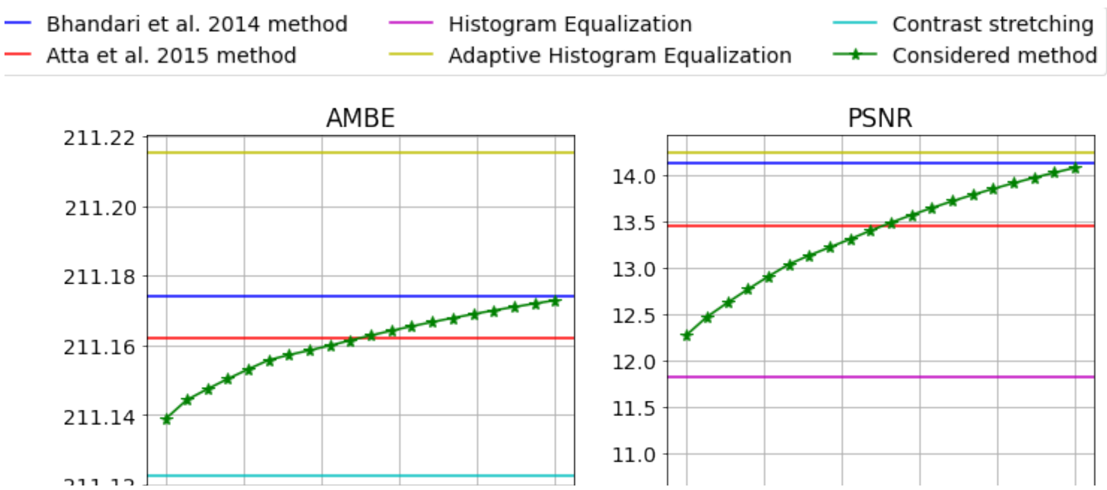
$$T(i) = (i - c) \cdot \frac{b - a}{d - c} + a$$


BEFORE
AFTER
Frequency
Frequency


Original image
Histogram Equalization
Adaptive Histogram Equalization
Contrast stretching

5. Quality evaluation

- Measure of Peak Signal-to-Noise Ratio (PSNR)
- Measure of Quality-aware Relative Contrast Measure (QRCM)
- Structure similarity index measurement (SSIM)
- Feature similarity index measurement (FSIM)
- Absolute Mean Brightness Error (AMBE)
- Measure of enhancement by entropy (EME)



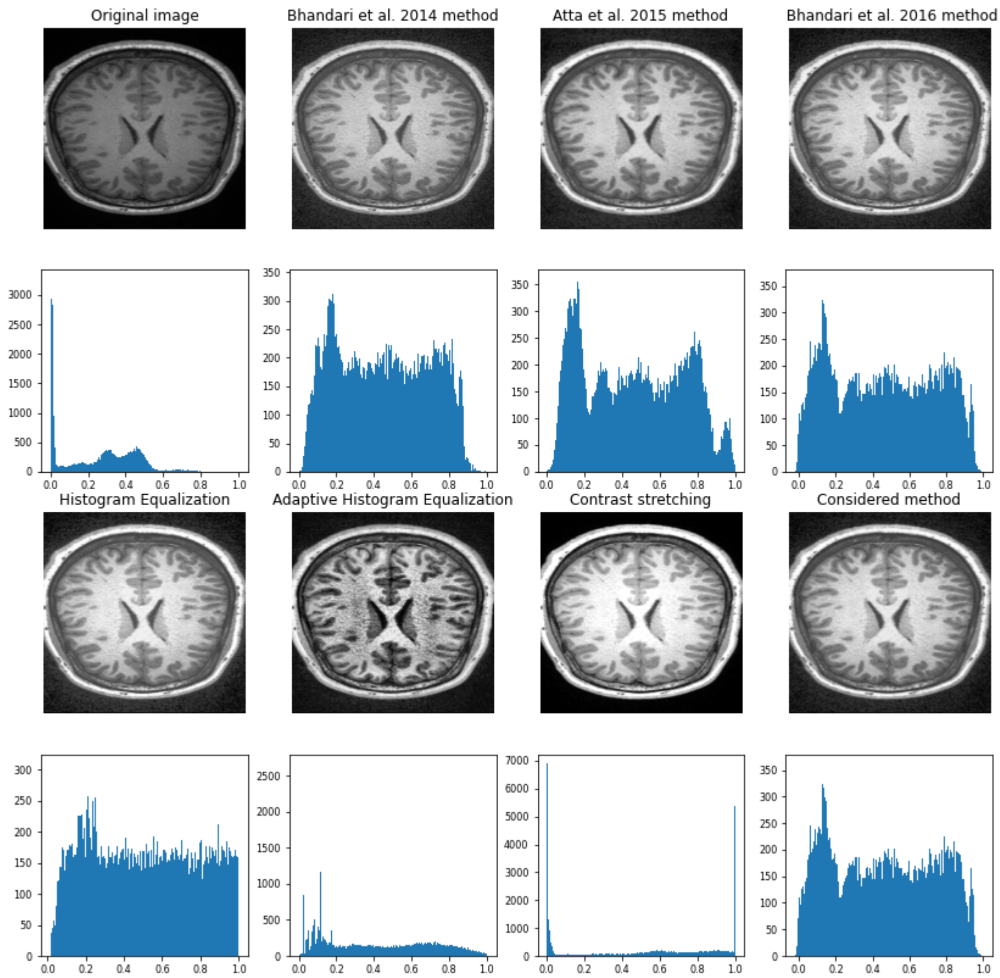
6. Time measurement

- **HE** based methods, contrast stretching - $O(n^2)$
- Methods with SVD decomposition - $O(n^3)$
- But numerical experiments show - 32 fps $\Rightarrow$ method suitable for real-time enhancement

Conclusion

- SVD based methods are more universal
- There exists such parametr μ for which the considered method shows the best performance in terms of quality
- Considered method is enough fast for online image enhancement

Results



In [ ]:

```

```

In [ ]:

```

```