COMP90086 – Computer Vision – Assignment 3

Arya Araban – 1439683 – aaraban@student.unimelb.edu.au

1. Reconstruction analysis - varying masks:

In this section we analyzed the reconstruction error of the two images, namely brick.png (Image 1) and basket.png (Image 2), using the given existing inpainting algorithm. The algorithm was run with five different mask locations for each image, and the root mean square (RMS) error for the reconstruction at each mask location was computed.

The mask locations were chosen to cover various points of the image, ensuring a comprehensive analysis as visualized in Figure 1.

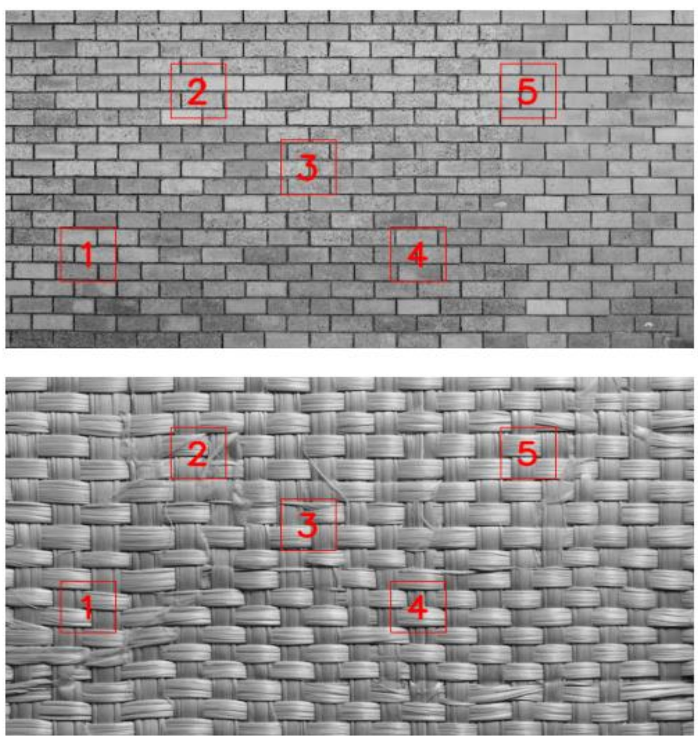


Figure 1 – Mask Locations

Following this, we performed the inpainting algorithm on these mask locations and achieved the results as shown in Figure 2. This chart highlights the differences in RMS errors between the two images and across different mask locations.

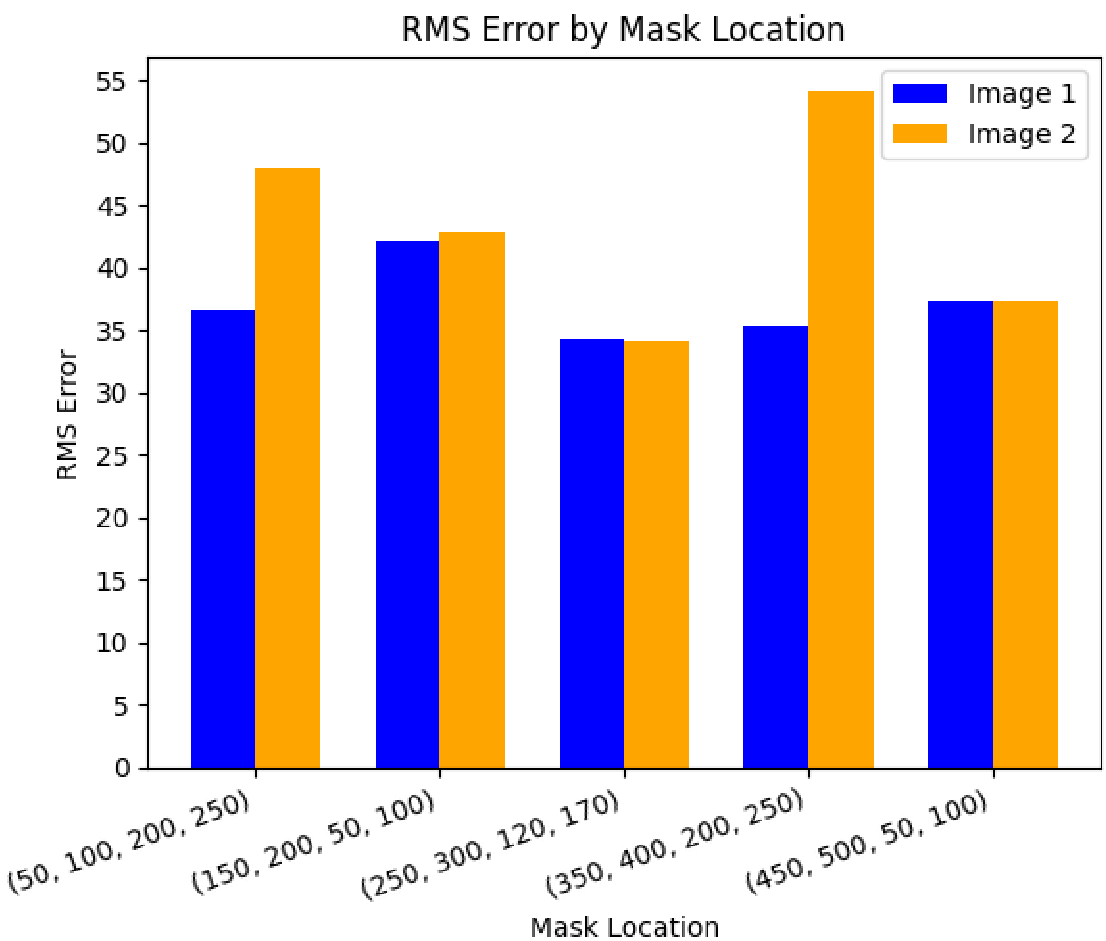


Figure 2 - RMS Error Across Different Mask Locations

For Image 1, the average RMS error was approximately 37.11, indicating a moderate level of reconstruction error. On the other hand, Image 2 had a slightly higher average RMS error of approximately 43.26.

Table 1 displays the outcomes of inpainting the two images across two of the mask locations (specifically masks number 1 and 5).

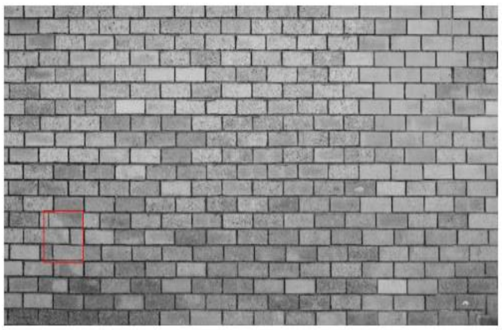
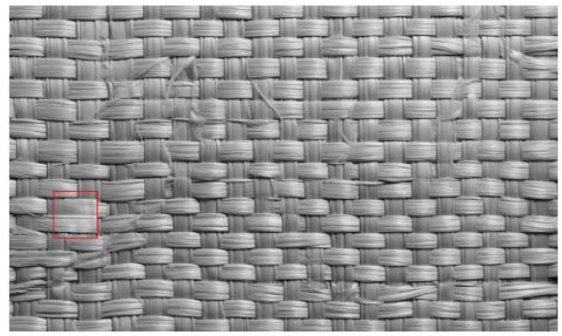
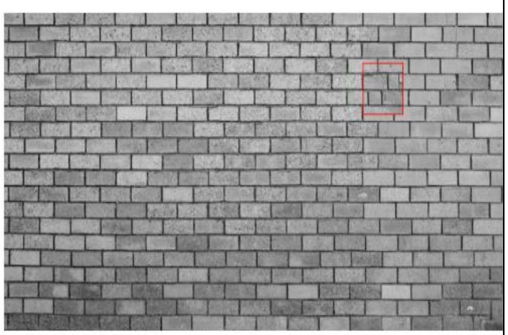
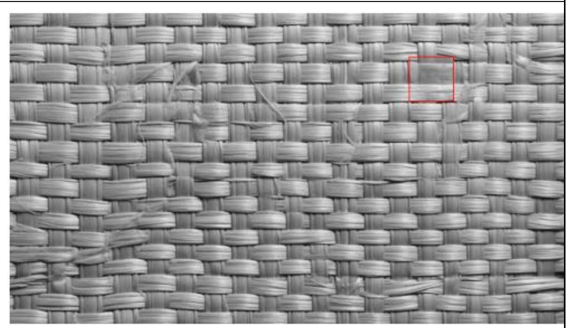
Mask Location	Image 1	Image 2
(50, 100, 200, 250)		
(450, 500, 50, 100)		

Table 1 – Inpainting Visualization Across Mask Locations

Upon inspection, it becomes evident that Image 1 exhibits noticeable artifacts in the inpainted regions, especially around brick edges. Conversely, Image 2, despite having a higher average RMS error, exhibited smoother results with fewer artifacts compared to Image 1.

2. Reconstruction analysis - varying contexts:

In this section we analyze how the shape and size of the context window affect image reconstruction quality and inpainting algorithm runtime. We experimented with various context sizes, and to introduce greater variability into our analysis, we employed distinct mask locations for each context size, aligning the first mask with the first context size and so forth.

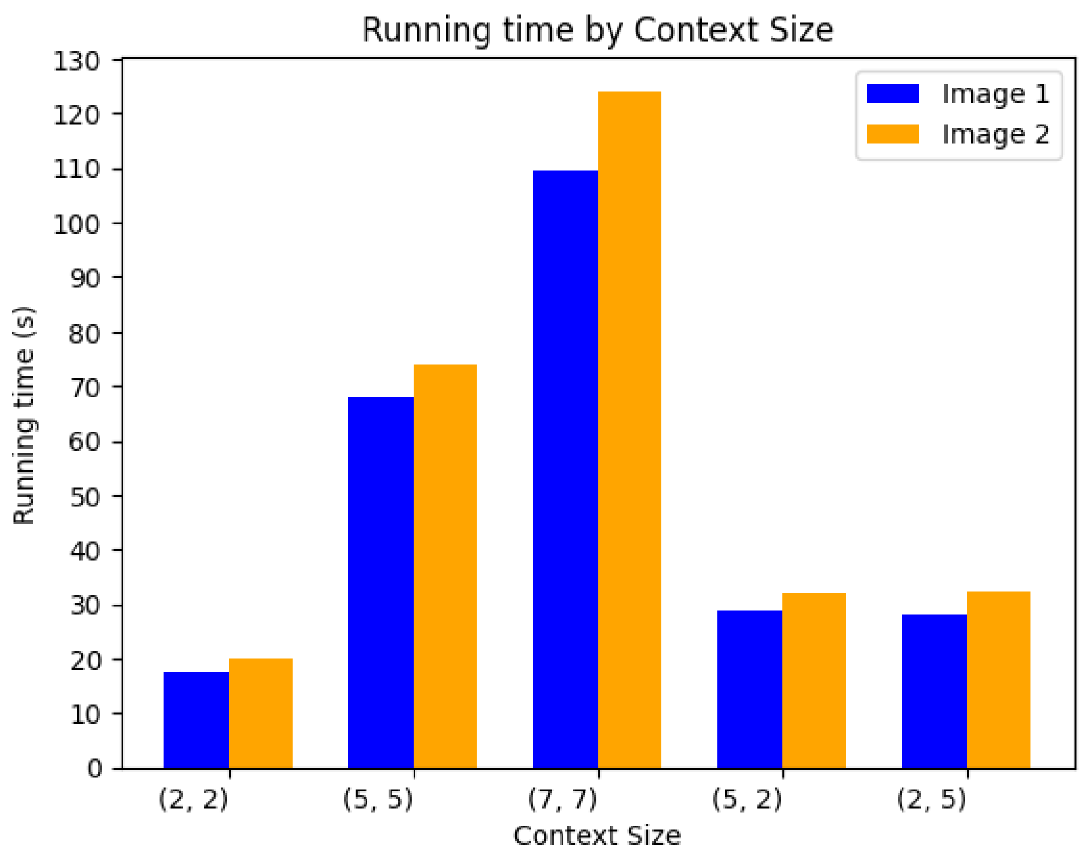


Figure 3 – Running Time by Context Size

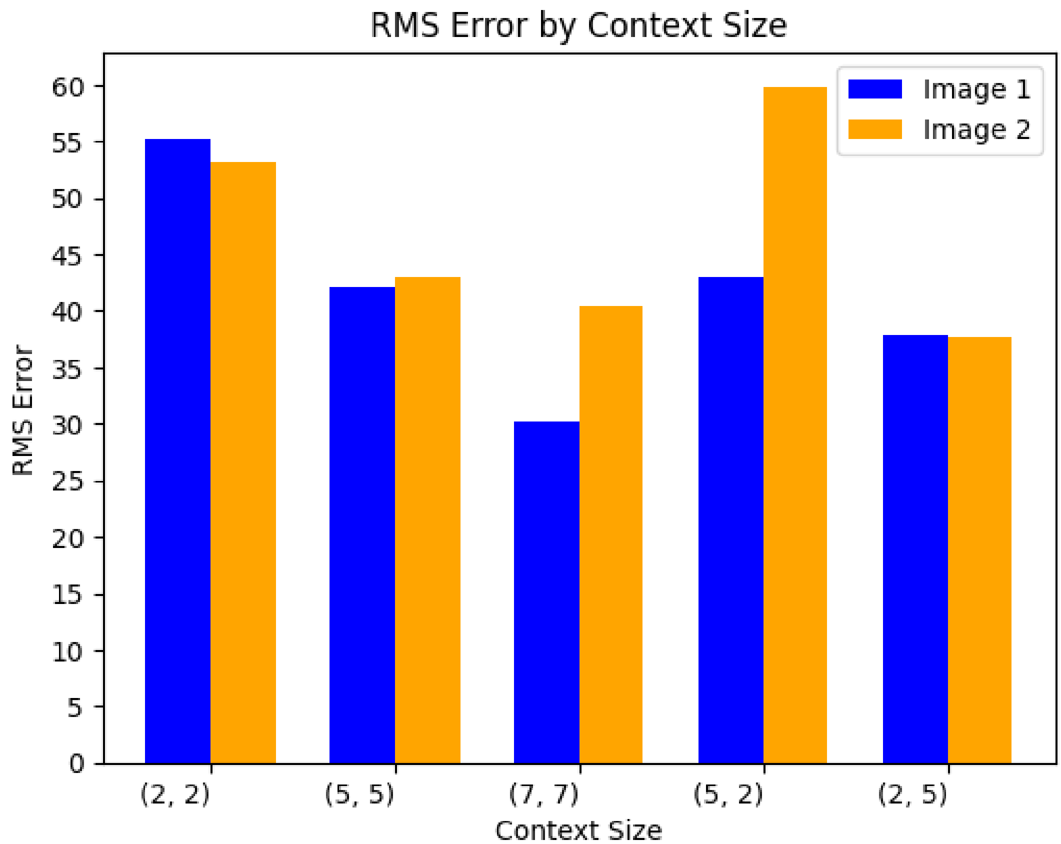


Figure 4 – RMS Error by Context Size

We found that the running time of the algorithm increases almost linearly with the context size. However, the relationship between context size and RMS error is not straightforward. For example, Image 2 shows only slight RMS improvement with a context size of (7,7) compared to (5,5), while for Image 1, this difference is more pronounced. Furthermore, the context size of (5,2) had a high RMS error, but (2,5) performed well in this aspect. This aligns with the findings of question 1, where the effective mask location for (2,5) led to improved outcomes.

For Image 1, the average running time was 80.22 seconds with an average RMS error of 41.6. In comparison, Image 2 had an average running time of 92.38 seconds and an average RMS error of 46.76. These averages illustrate the trade-off between computational efficiency and reconstruction quality in inpainting algorithms.

Table 2 shows the inpainting outcomes for the two images at different context sizes and mask locations.

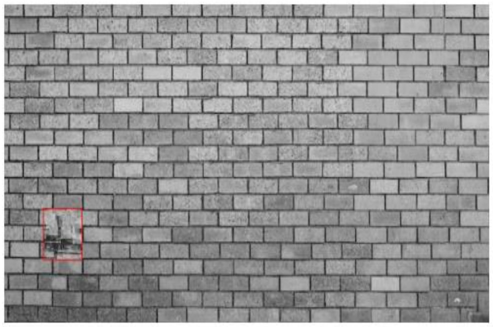
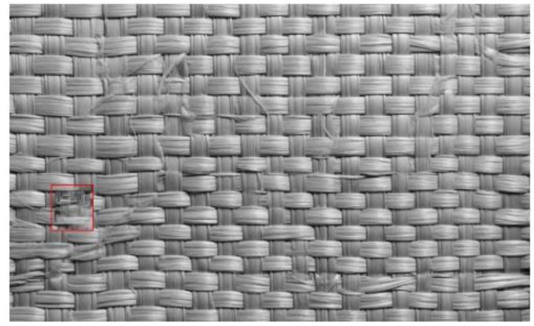
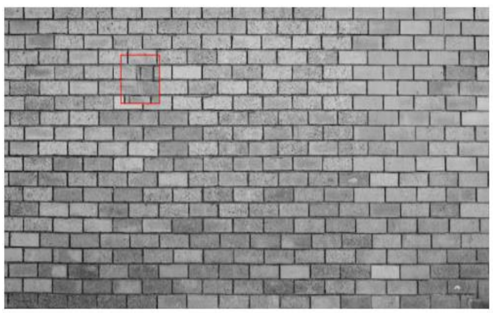
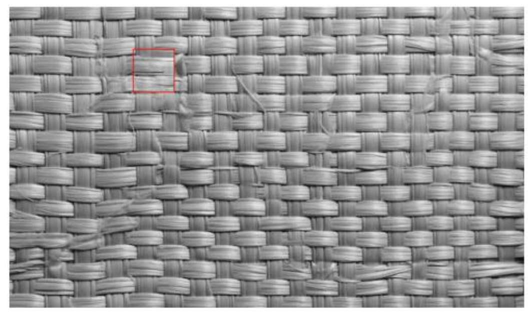
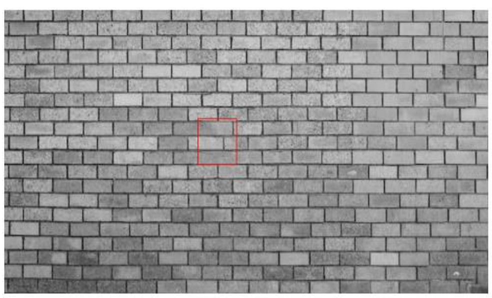
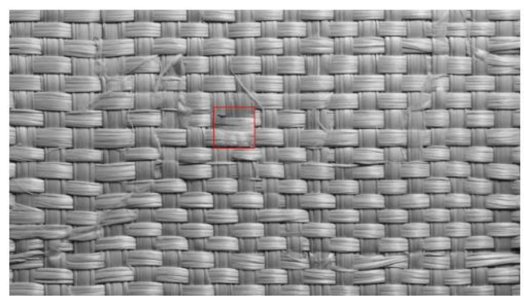
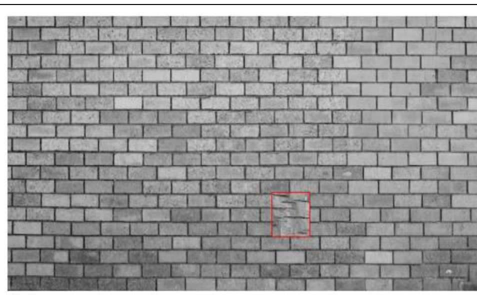
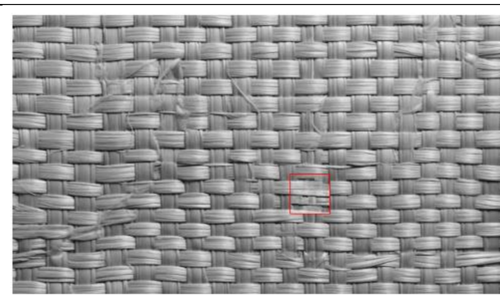
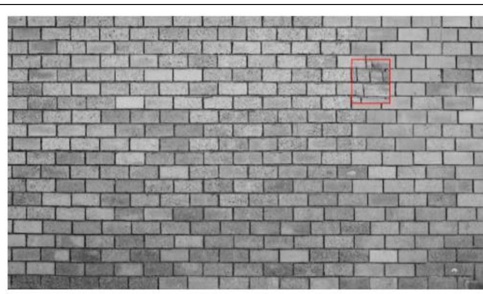
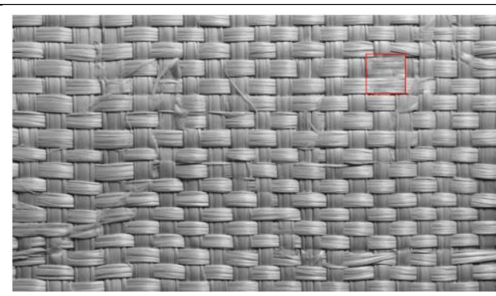
context size	Image 1	Image 2
(2, 2)		
(5, 5)		
(7, 7)		
(5, 2)		
(2, 5)		

Table 2 - Inpainting Visualization Across Context size

While for Image 1 increasing the context size generally reduced the RMS errors, a visual inspection revealed more noticeable artifacts in the reconstructed image compared to Image 2, especially with smaller context sizes like (2,2). This suggests that although a larger context window can improve quantitative measures of reconstruction quality, it does not always result in outcomes of higher visual quality. On the other hand, Image 2 consistently produced smoother reconstructions across all context sizes except (2,2), despite having higher RMS errors compared to Image 1.

Interestingly, while both context sizes (5,2) and (2,5) didn't achieve exceptional visual results, the context size (2,5) produced rather plausible results for both Image 1 and Image 2. This observation underscores the complex interplay between context window shape and reconstruction quality in inpainting algorithms.

3. Reconstruction design:

To improve our inpainting algorithm, we explored two ideas: using absolute distance as a distance metric to select the best pattern and implementing a weighted selection approach. The weighted selection involved considering the top five candidate pixels and assigning them weights based on their distances from the best match.

In this section, we will analyze and compare the impact of these ideas individually by evaluating their effects on the results. For easier comparisons, we will focus on comparing RMS errors for context sizes (2,2) and (5,5), but also consider the average across all of the given context sizes.

3.1. Absolute Distance:

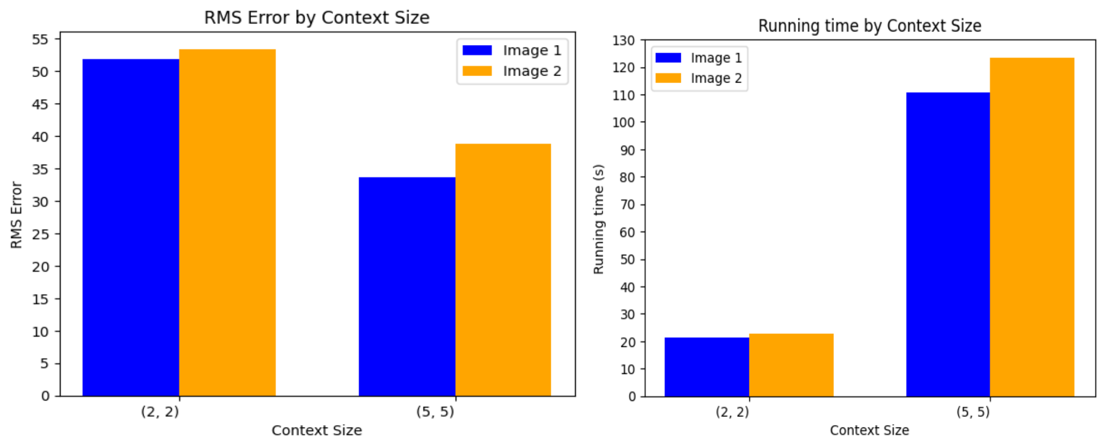


Figure 5

Absolute distance improves reconstruction quality for larger context sizes. Image 1's RMS error decreased from 55.13 to 51.94 for (2,2) and improved from 42.15 to 33.79 for (5,5). Image 2 remained stable around 53.4 for (2,2) and decreased from 42.91 to 38.90 for (5,5).

Image 1's average RMS error improved to 40.2, while Image 2's decreased to 43.6. However, average running times also slightly increased: Image 1 took 85 seconds, and Image 2 took 87 seconds. This suggests a trade-off between accuracy and computational efficiency.

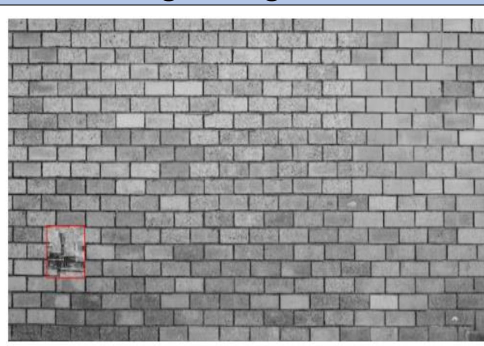
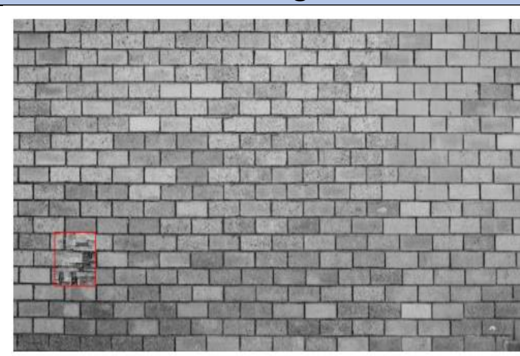
	Original Algorithm	Modified Algorithm
Image1 / (2,2)		

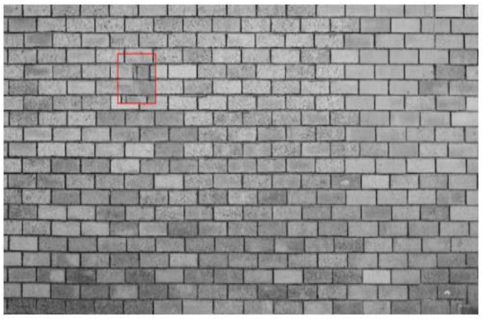
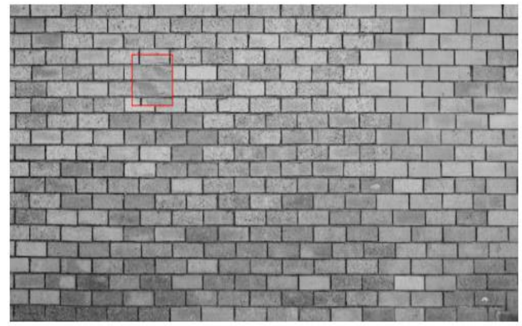
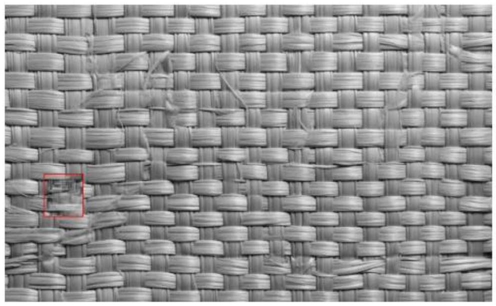
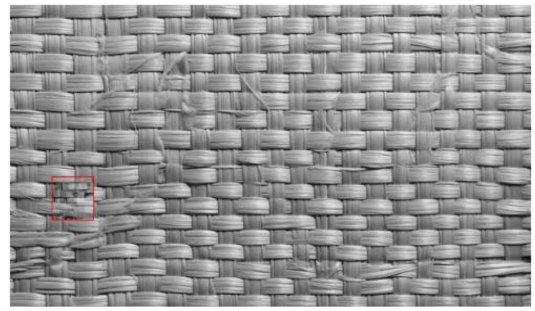
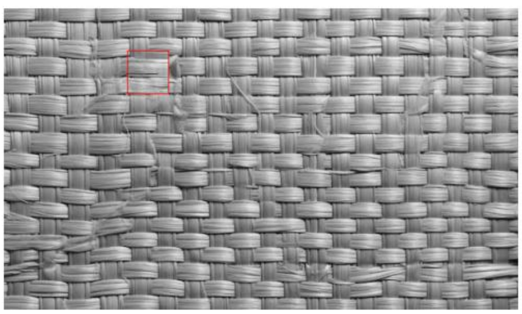
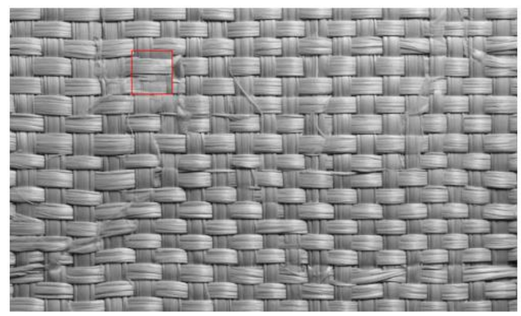
Image1 / (5,5)		
Image2 / (2,2)		
Image2 / (5,5)		

Table 3

Image 1 and Image 2 performed better than the original method for (2,2) context size. Image 2 had smoother reconstruction with fewer artifacts at (5,5) context size. However, Image 1 showed some "smudges" at (5,5) context size, indicating further optimization may give better visual results.

3.2. Weighted Selection:

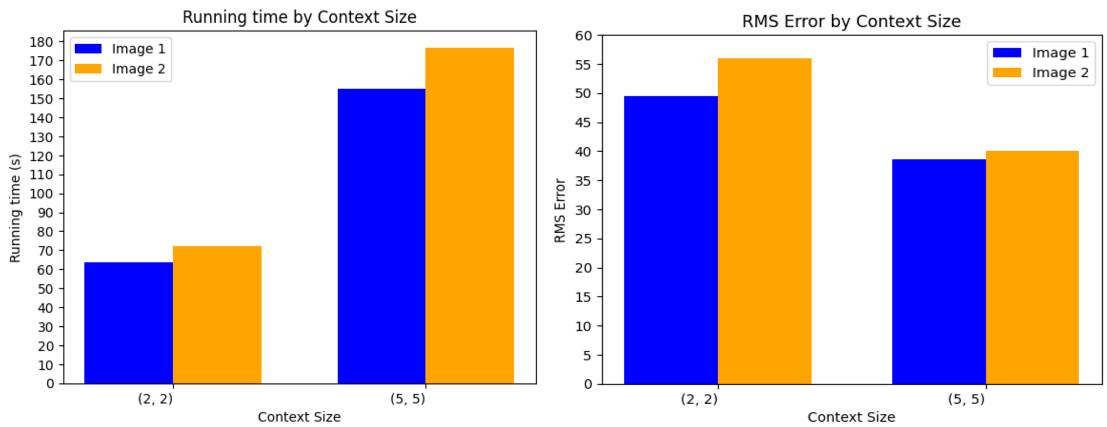


Figure 6

The weighted top 5 pick method slightly improves reconstruction quality for larger context sizes. For (2,2) context size, Image 1's RMS error decreased from 55.13 to 49.57, while Image 2's error increased slightly from 53.18 to 55.92. With (5,5) context size, Image 1's RMS error decreased significantly from 42.15 to 38.66, and Image 2's error decreased from 42.91 to 40.02.

Across all contexts, the weighted top 5 pick method slightly improved average RMS errors to 41.3 for Image 1 and 44.5 for Image 2, but highly increased average running times to 125.4 seconds for Image 1 and 140.8 seconds for Image 2.

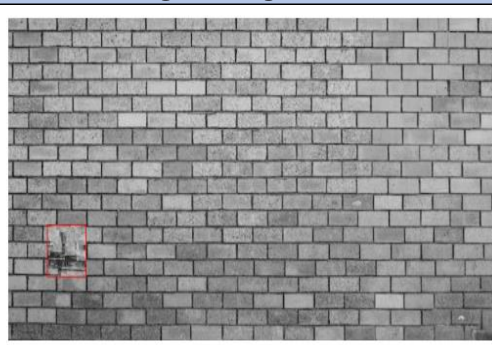
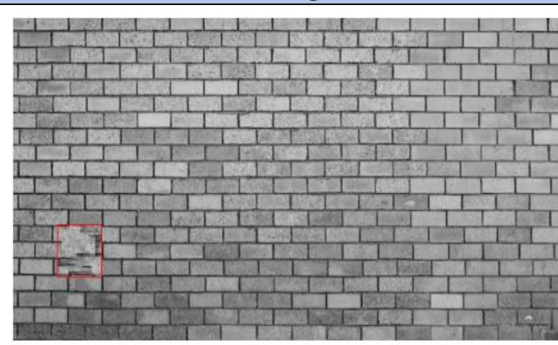
	Original Algorithm	Modified Algortihm
Image1 / (2,2)		

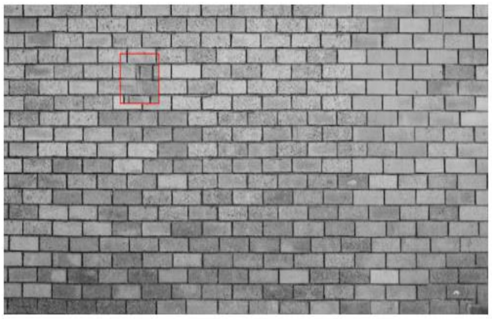
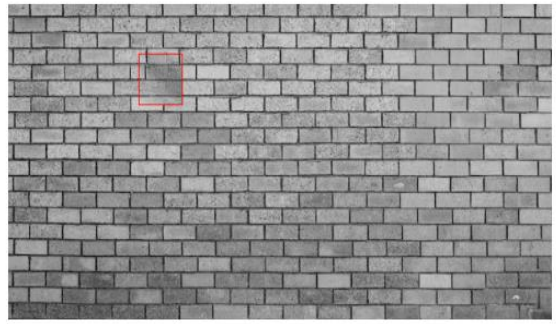
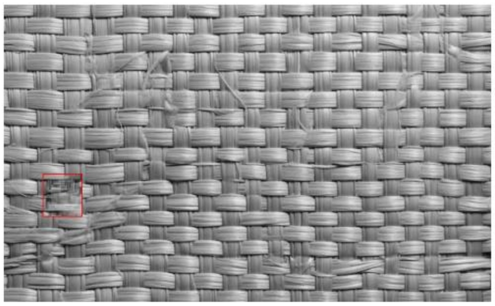
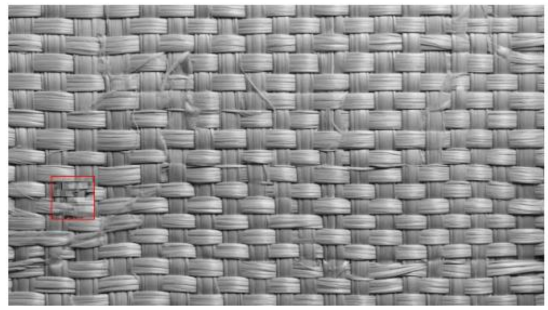
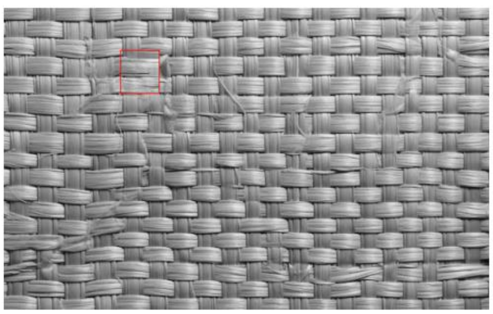
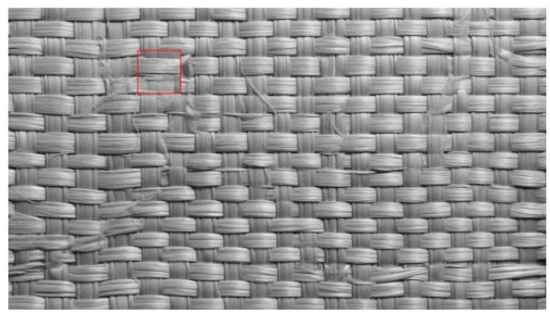
Image1 / (5,5)		
Image2 / (2,2)		
Image2 / (5,5)		

Table 4

The weighted top 5 pick method had limited impact on the quality of reconstructed images for (2,2) context size. For (5,5) context size, some improvement was observed in Image 1, but with misplaced brick lines. The effectiveness of the method varies depending on context size and image characteristics.

3.3. Conclusion:

To Conclude, the method of changing the distance metric to absolute proved more effective, showing significant improvements in both qualitative and quantitative results without a substantial increase in running time. The weighted selection approach, despite some minor improvements, was much less efficient due to a significant increase in running time. Therefore, we recommend the first method for enhancing the performance of the inpainting algorithm.