

IE423 - QUALITY ENGINEERING

Project Part 3

A COMPREHENSIVE ANALYSIS REGARDING QUALITY CONTROL ON LINEN IMAGES



Group 2

Esra Ath

Elif Gül

Saime Nur Benliler

Instructor: Mustafa Gökçe Baydoğan

1. Introduction

Linen stands out in the textile business due to its labor-intensive manufacturing technique and specific features. Clothes made of linen, which is prized for its exceptional absorbency and ability to keep you cool and fresh, are particularly useful in hotter areas (Source: Wikipedia). Our objective as quality engineers in a linen production company is to put in place a strong quality control system that is adapted to the specifics of linen production.

For a number of reasons, linen processing needs to be closely watched. To achieve the high standards demanded of this luxury fabric, linen must be manufactured with the precise attention to detail that is necessary in its labor-intensive process. Any departure from these requirements can affect the manufacturer's reputation and consumer satisfaction by compromising the quality of the product. Perfect construction is also essential to linen's reputation for comfort in warm weather. Maintaining these coveted qualities during the manufacturing process requires early flaw detection and rectification.

The motivation behind employing image-based defect identification methods in linen manufacturing stems from the necessity for efficient and reliable quality control. Conventional inspection methods often fall short when dealing with intricate textures like those found in linen. The adoption of automated visual inspection processes presents a sophisticated and efficient solution for identifying defects in materials with unique textures, spanning tiles, textiles, leather, and more (Source: Rimac-Drlje, Snježana, Drago Žagar, and Slavko Rupčić. "Adaptive Image Processing Technique for Quality Control in Ceramic Tile Production." *Strojarstvo* 52.2 (2010): 205-215.). By leveraging advanced image acquisition systems and implementing rapid and effective texture analysis procedures, our project aims to revolutionize quality control in linen manufacturing, ensuring the consistent delivery of flawless products to customers.

2. Background Information

Throughout history, traditional methods of quality control in the textile industry have involved manual inspection and sampling techniques. Such traditional processes mainly included fabric inspection by humans, physically and visually. These techniques, however, can be laborious, subjective, and ineffective for detecting flaws in complex textures, such as those found in linen. As a result, automated process monitoring systems are becoming more and more common in a variety of manufacturing sectors, including the textile industry. Focusing mainly on the process monitoring of linen, many aspects have been examined in the literature, and there continues to be an emphasis on monitoring and improving productivity and efficiency in linen and textile services. For instance, UniFirst, a workwear and textile services company, developed a centralized, web-based application for real-time monitoring of systems performance, energy consumption, equipment utilization, and labor activity in their plants. This system, known as UniFirst Production Monitoring (UPM), delivers extensive data in report and dashboard forms and interfaces to all automated production systems. It

facilitates the rapid diagnosis of problems with process flow, enhances system functionality, and enables overall productivity. (Source: [Tracking, Measuring, Improving Laundry Performance | American Laundry News](#))

3. Approach Explanation and Results

A Baseline Defect Detection Approach from a Statistical Data Analysis Perspective

In the pursuit of enhancing defect detection methodologies within linen image analysis, the first strategy employs a baseline approach rooted in statistical data analysis principles. This approach is designed to unveil potential anomalies in pixel distributions, leveraging statistical insights to identify and address defects systematically. The stepwise process encompasses the analysis of pixel value distributions, parameter estimation based on probability distributions, and subsequent identification of outlier pixels within 0.001 probability limits and trial-and-error probability limits.

Before working on the defect detection part of the project, first the image has been transformed into a grey-scale image instead of R,G,B (red, blue, green) format. Afterwards, the pixel matrix regarding the image is obtained as one can see below.

```
Pixel Matrix:
[[214 204 191 ... 222 205 202]
 [188 210 216 ... 252 194 153]
 [221 192 156 ... 219 228 206]
 ...
 [231 211 246 ... 252 226 173]
 [218 237 188 ... 224 213 253]
 [231 219 203 ... 216 211 202]]
```

Figure 1: Pixel matrix regarding the image for group 2

Two different approaches are proposed to fit a distribution to the pixel matrix of the linen image. In this part, both approaches are tried and one of them is chosen.

First Approach: Fitting a distribution for rows and columns separately by getting row averages and column averages. After obtaining these two distributions, out of control pixels are determined both row-wise and column-wise.

Second Approach: The pixel matrix is flattened and brought to a one dimensional array. Afterwards a distribution is fitted to the whole matrix and out of control situations are determined according to one distribution.

In the first approach, first average values for each row are obtained and a histogram is plotted. Afterwards, it is decided that the normal distribution will fit well to the histogram with mean 205.93 and standard deviation 2.19 by using the .fit() built-in function in scipy.stats library of Python.

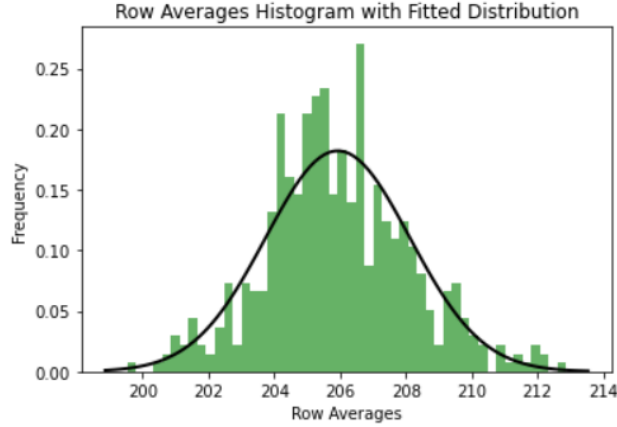


Figure 2: The histogram and the fitted distribution for the row averages

Afterwards, the same procedure is followed for the column averages of the pixel matrix and it is observed from the histogram that a normal distribution which has two modes meaning a mixture of two different normal distributions would fit well by observation. Therefore, a gaussian distribution is fitted as one can see from the figure below.

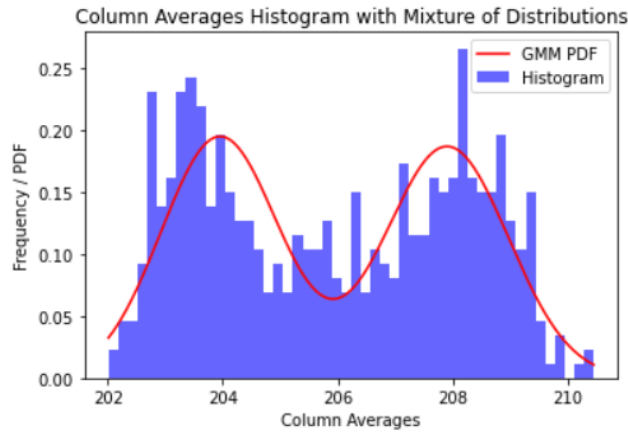


Figure 3: The histogram and the fitted distribution for the column averages

Row averages are fitted to a normal distribution and column averages are fitted to a mixture of normal (Gaussian) distribution. However, in that case, since we have fitted the distributions according to the averages, we don't have enough data for the approximation. Also, since the out of control pixels are determined both row-wise and column-wise, there are more than necessary out of control points which causes Type-2 error, false alarms. Therefore the second approach is taken by flattening the pixel matrix.

In the second approach, to get rid of false alarms the pixel matrix is flattened into a 1-D array and a normal distribution with mean 205.93 and standard deviation 22.05 is fitted almost perfectly since the sample size is large.

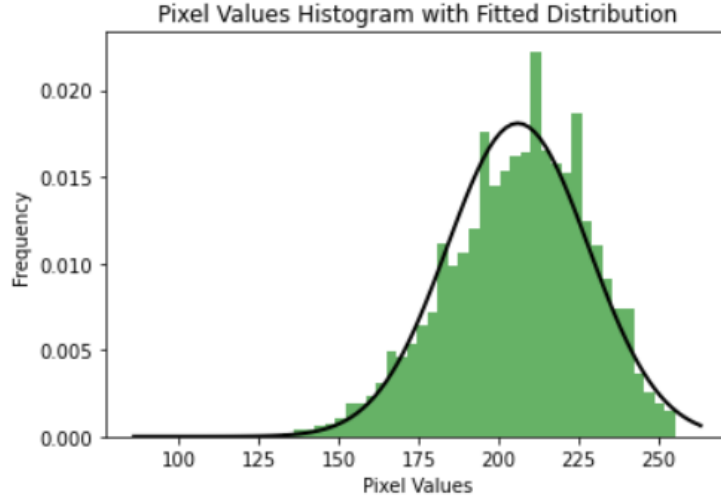


Figure 4: The histogram and the fitted distribution for the 1-D array version of pixel matrix

By utilizing mean and standard deviation values, 0.001 probability limits are easily calculated by taking the inverse of cumulative distribution function of normal distribution.

As a next step, the pixel values falling beyond the probability limits of 0.001 are set 0 which means they will be shown as black in the grayscale image. Figure below shows the original image and modified image together.

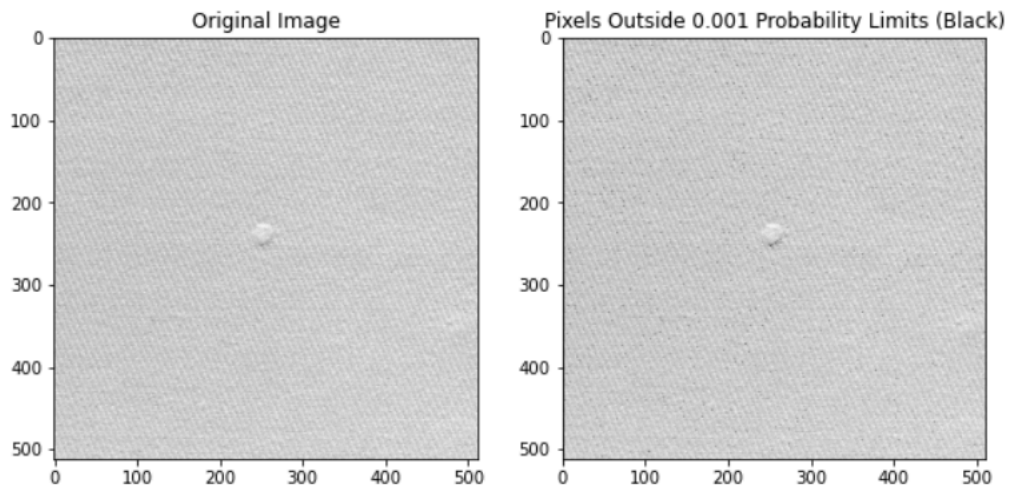


Figure 5: Original image and modified image showing pixel outside of probability limits of 0.001

When the probability limits are set as 0.001, as one can see from the difference between modified image and original image, the defect cannot be detected. However, when the lower bound and upper bound is set as 130 and 210 (by trial and error), the defect can be seen easily as the next figure suggested.

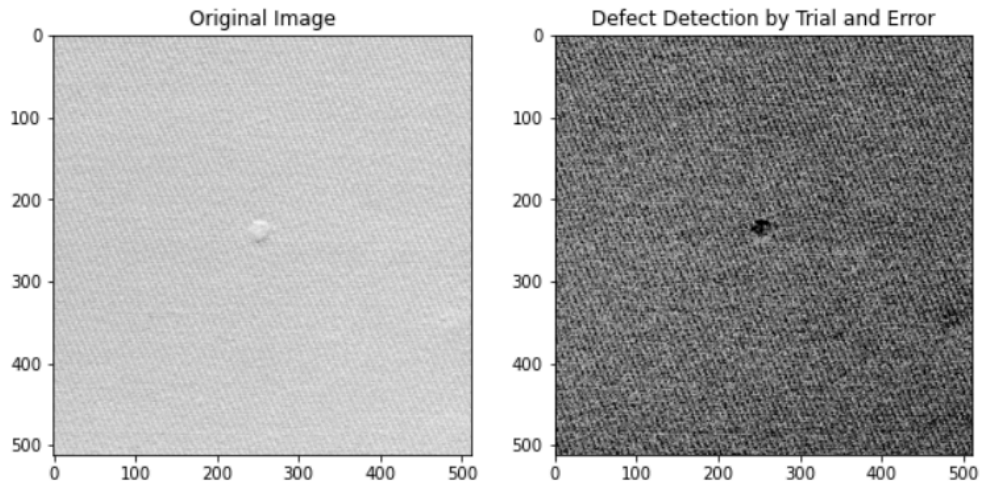


Figure 6: Original image and modified image showing pixel outside of limits found by trial and error

Here one can see that defects can be shown in the middle of the image by correctly choosing lower bound and upper bound.

In the continuous pursuit of enhancing defect detection in linen images, we introduce an advanced strategy focusing on local structures. Utilizing a window size of 51x51 pixels, this patch-based method aims to improve precision by performing image operations on specific patches, capturing nuanced details within the linen material. Repeating the analysis steps from previous tasks within these patches allows for a thorough examination of defects in smaller, more focused areas. The identification and modification of outlier pixels within these patches contribute to a refined defect detection approach. By comparing the modified and original images, our goal is to gain insights into the effectiveness of this localized strategy in identifying and rectifying defects within the linen material.

Each 51x51 patch has different mean and standard deviation parameters, therefore, the limits for each patch will be different. By detecting the pixels outside of the limits we have obtained three sample patches as shown below.

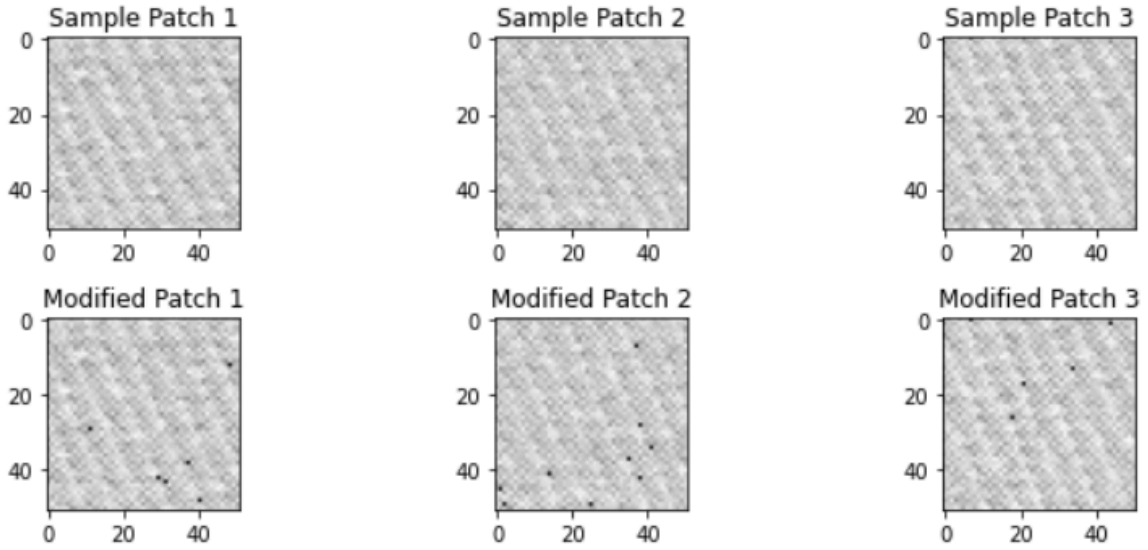


Figure 7: Three sample patches with original and modified versions

To display patch-work, 3 sample patch detection is chosen randomly and shown with their modified versions. Afterwards, patches are combined and the modified version is displayed.

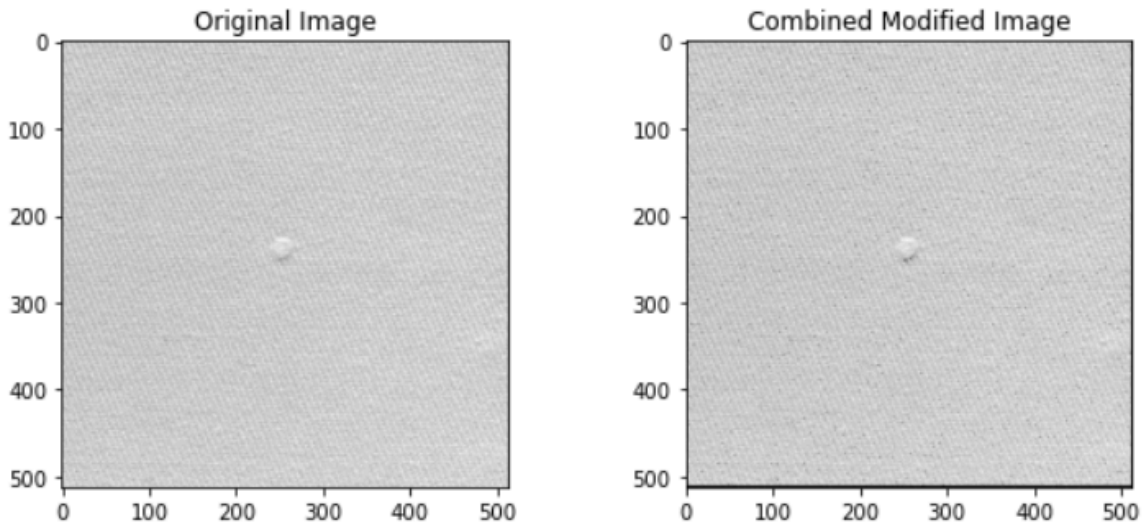


Figure 8: Original image and modified image showing combined patches with out of control pixels

To comment on the detection performance of the patch-work, we have expected it to be true in the first place because it calculated the mean and sigma values of the distribution of each patch and calculated the lower bound and upper bound accordingly. However, it didn't give a better performance as one can observe from the comparison between images above. Because our image doesn't have many different patches, the mean and standard deviation of

different patches didn't vary among each other. That's why patch-work didn't perform as well as expected.

The second reason for the bad performance of the patch-work is that the defect in our image is confined to a specific patch. Therefore the limits of this patch are determined with the specific mean and standard deviation of this patch. So, these limits cannot detect the defect in this image. If the image had a defect which is line-shaped, the patch-work would work better.

The third reason for the bad performance is that our probability limits are really small. In the next figure, the probability limits will be updated as 0.1 and the performance is a little bit higher.

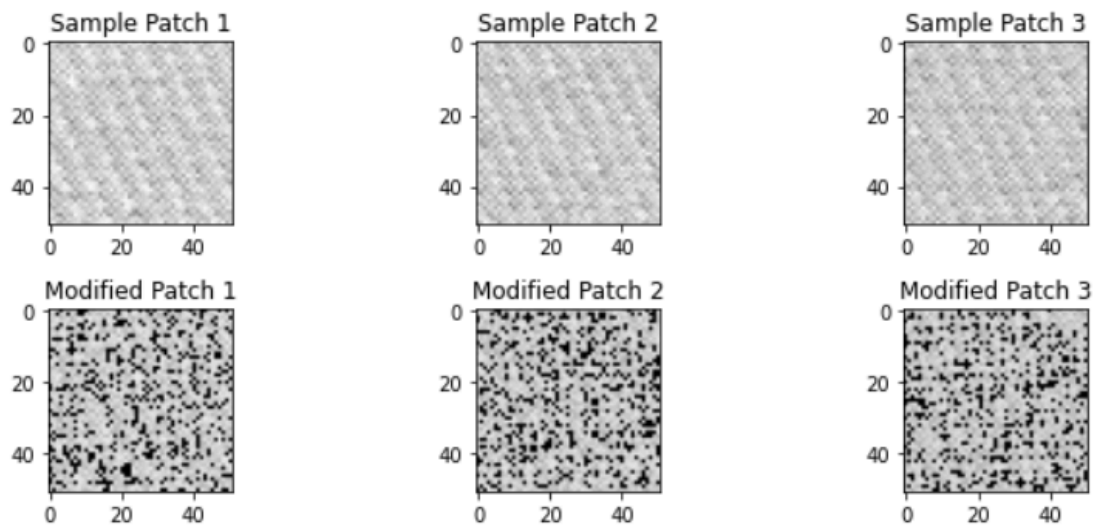


Figure 9: Three sample patches with original and modified versions with 0.1 probability limits

Three sample patches are chosen randomly and shown with their modified versions above. Afterwards, patches are combined and the modified version is displayed as previously discussed.

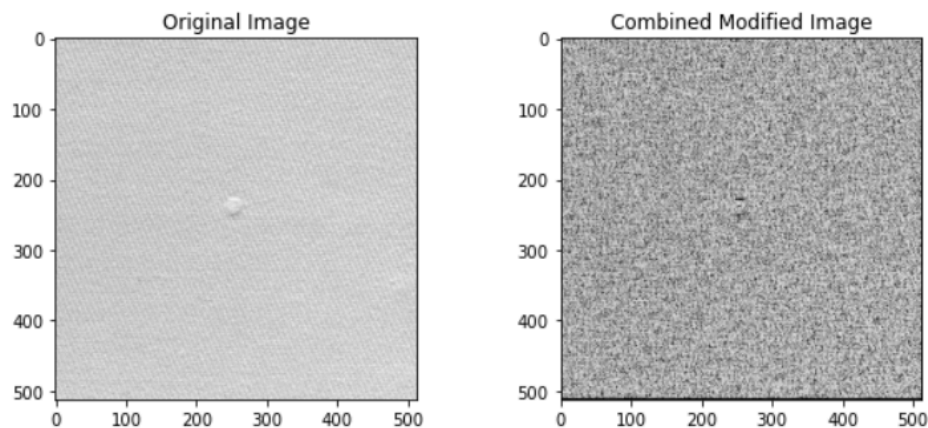


Figure 10: Original image and modified image showing combined patches with out of control pixels with 0.1 probability limits

When we decrease the probability limits, again it didn't detect the defect in the middle with the performance on the general 130 and 210 limits as we have shown above because the defect in the middle is most probably contained in a patch and this patch has a mean and standard deviation accordingly. Therefore, the limits corresponding to the patch containing the defect couldn't detect the defect.

A Simple Defect Detection Approach from a Control Chart Perspective

Another approach in defect detection could be analyzing the rows and columns separately. In this approach, each row is considered to have its own parameters and each column is considered to have its own parameters correspondingly. The motivation behind this approach can be detecting defects that are along a vertical or horizontal line. Such an approach could provide valuable insights regarding defects depending on the tested piece of fabric. In detail, firstly, we have constructed control charts for pixel values of each row with its own mean and variance. We have decided to use 2 sigma limits after comparing with a closer range of control limits. As one might guess, using higher limits would mean that a wider range of data points are considered in-control and thus may result in missing out some of the defects. Additionally, using lower limits may result in too many out-of-control pixels that may not always be considered as defects.

Below, some of the example control charts of 2 rows are displayed with their own parameters.

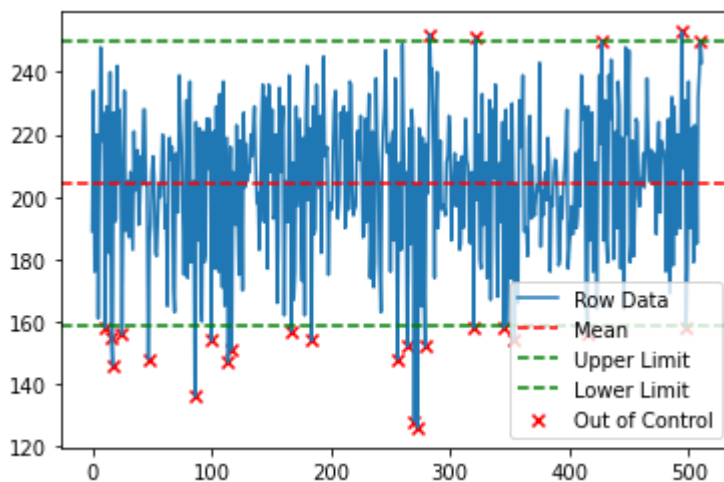


Figure 11: One example of the control chart constructed for the row with mean 204.208984375 and the variance 522.4817161560059

> The mean is 204.208984375 and the variance is 522.4817161560059

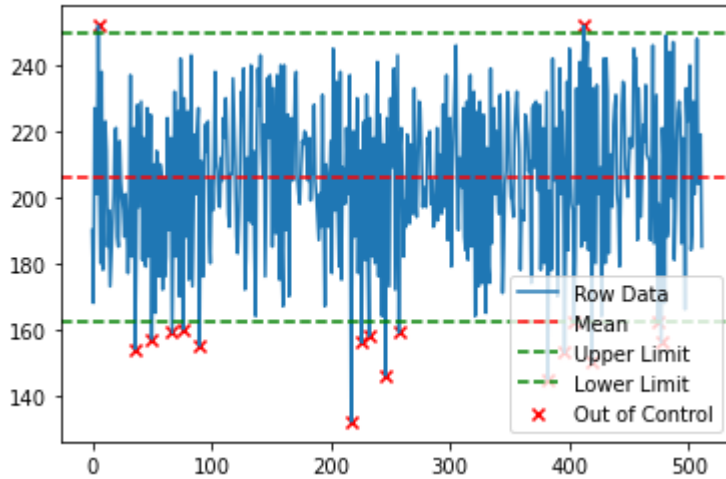


Figure 12: Another example of the control chart constructed for the row with mean 206.025390625 and the variance 481.9114646911621

```
> The mean is 206.025390625 and the variance is 481.9114646911621
```

After constructing control charts, the coordinates of the out-of-control pixels for each row are determined and kept in another matrix as follows:

```
> Coordinates of Out-of-Control Pixels:
[[ 0 61]
 [ 0 99]
 [ 0 119]
 ...
 [511 460]
 [511 468]
 [511 482]]
```

To detect defects visually; by using the coordinates, values of the out-of-control pixels are changed to zero, in other words, to the black color. Then, the modified image and the original one is displayed for comparison. The defect in the middle of the fabric becomes much more visible as we mark the out-of-control pixels with black.

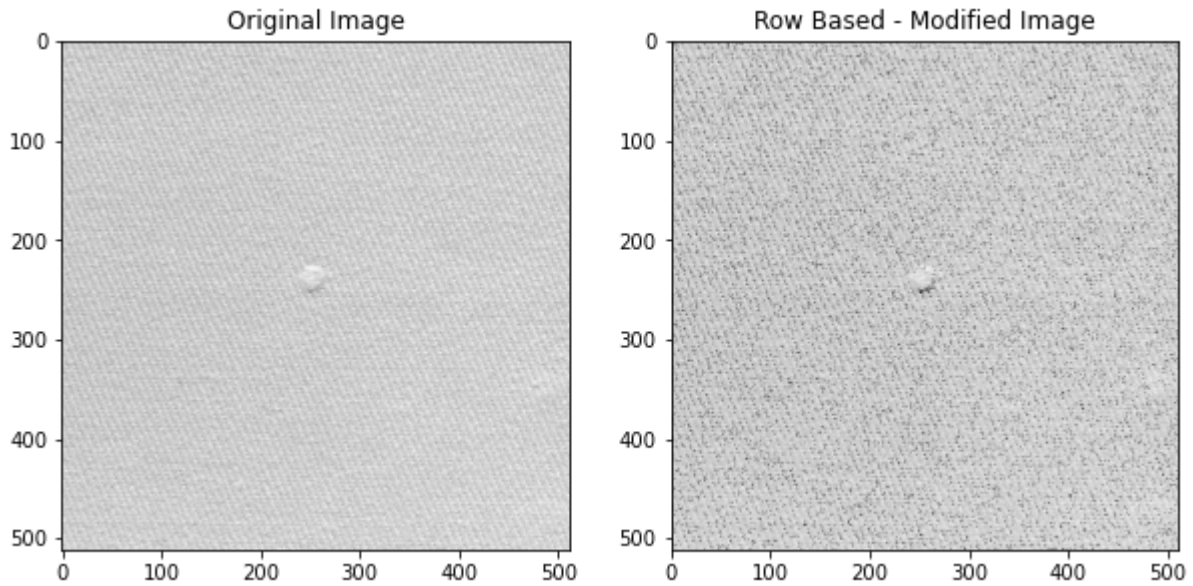


Figure 13: Original image and modified image showing out-of-control pixels with a row-based control chart approach

Following that, the same process is repeated for columns of the pixel matrix, as well. For the column-based approach, the process is very much similar. The only required arrangement is taking the transpose of the pixel matrix and then, the rest of the process is the same.

Again, the modified image and the original one is displayed for comparison. The defect in the middle of the fabric becomes much more visible as we mark the out-of-control pixels with black.

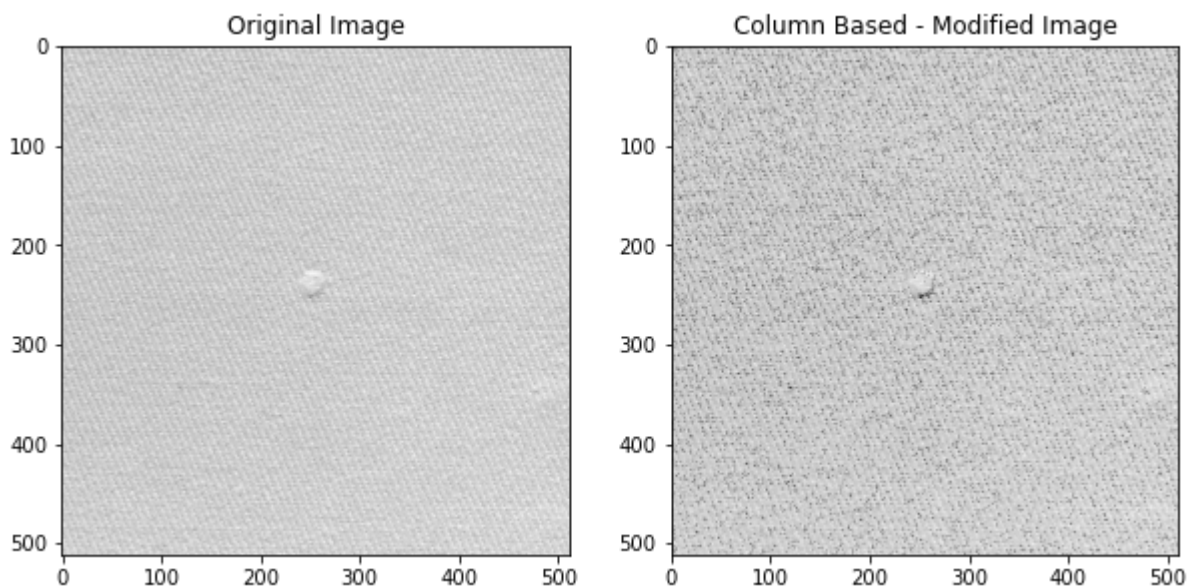


Figure 14: Original image and modified image showing out-of-control pixels with a column-based control chart approach

Finally, we can focus on the similarities and differences between the row-based and column-based approach. To do so, we utilized 2 different methods. The first one is the usual calculation of the correlation between the both image matrices.

```
> 0.8277281361331115
```

The second one utilizes the "Structural Similarity Index" which is particularly used for comparing images.

```
> 0.8272114574699101
```

As one can see from the both methods, the 2 images acquired from the both approaches are very much similar to each other. Yet, there is a little difference that may have arisen from various reasons. The row-based approach focuses on capturing horizontal defects across the fabric. So, it identifies flaws such as horizontal tears, thread breaks or weaving irregularities. However, the row-based approach may fail to detect flaws such as irregularities in the weave pattern, or vertical thread misalignments which are expected to be seen by a column-based approach. Since our linen fabric does not have such a specific defect (not specific to a row or a column), the created images do not differ in a significant manner. Yet, in case of an examination of another piece of fabric, utilizing both the row-based and column-based approaches might be very much helpful in detecting defection.

Our Proposal

In addressing the challenge of applying statistical process control to two-dimensional spatial data inherent in images, particularly those with texture, we adopted an approach that departs from traditional SPC methods. Our methodology centered around the development and application of a 2D autoregressive model, tailored to recognize the spatial dependencies between pixel values in textured images. This was a critical step, as standard control chart techniques, which are effective for independent observations typical in 1D time series data, fall short in the context of image analysis due to the autocorrelation present among pixel values.

The core of our method involved predicting the value of each pixel based on the values of its neighboring pixels, using a convolution-based autoregressive model. This model was implemented using a specially designed kernel, which facilitated the 2D convolution operation over the image. In this operation, the value of each pixel was predicted based on a weighted sum of the values of its immediate neighbors, encompassing both direct (horizontal and vertical) and diagonal neighbors. This approach allowed us to effectively account for the spatial relationships inherent in the image's texture, thereby overcoming the limitations of assuming independence between pixel values.

Upon establishing the autoregressive model, we proceeded to calculate the residuals for each pixel. These residuals represented the deviation of the actual pixel values from their predicted values as per the model. The calculation of these residuals was a pivotal step, as it provided us with a quantifiable measure of the discrepancy between the observed and expected pixel values, essentially highlighting areas where the model's predictions did not align with the actual data.

Furthering our analysis, we established control limits for these residuals based on their statistical distribution. Specifically, we employed two-sigma control limits, which provided a threshold to identify significant deviations. Pixels whose residuals fell beyond these control limits were flagged as potential defects, indicating areas in the image that deviated unusually from the expected pattern.

To enhance the robustness of our defect detection process, we extended our analysis to include both row-wise and column-wise control charts with 2-sigma limits. This entailed treating each row and column of the residual matrix as a separate sequence of data and applying control charts to these sequences. This row-wise and column-wise approach allowed for a more structured examination of the residuals, enabling us to capture anomalies that might be specific to certain rows or columns in the image. As with the global analysis, points that exceeded the control limits in these row-wise and column-wise control charts were marked as potential defects.

Through this comprehensive methodology, we successfully integrated the principles of SPC into the realm of 2D image analysis. The use of both global (based on overall residuals) and local (row-wise and column-wise control charts) approaches allowed for a thorough examination of the images. This novel approach to applying statistical process control in two-dimensional data demonstrates a significant step forward in image analysis, particularly in the context of quality control and defect detection in textured images. By addressing the limitations of traditional SPC methods and incorporating spatial relationships between pixels, this technique provides a robust framework for identifying irregularities and potential defects in complex image datasets.

4. Comparison

To evaluate our proposal on alternative images, we have generated 100 random integers between 2-196, and selected the first 5 images (165, 30, 8, 191, 72) that are available. Global approach, row-wise approach and column-wise approach has been implemented to these five images. Control charts for row-wise and column-wise approaches can be seen in the ipynb file and the modified images can be seen below.

Global Approach Results:

Global approach seems to work better in the cases where the defects are in the form of shade differences or small texture differences across the image. This may be caused by the

fact that our kernel is 3x3, that is really small compared to the overall image and we make our prediction based on very close neighbors of defects.

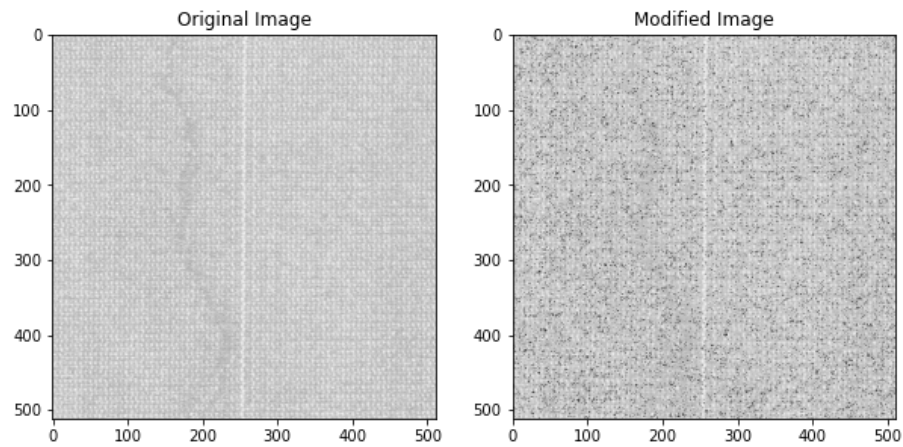


Figure 15: Original image 0165 and modified image showing out-of-control pixels with a image-wise control chart approach

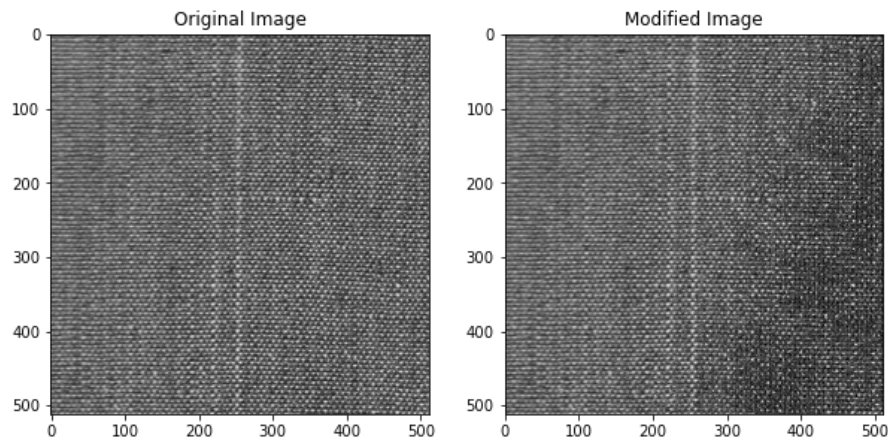


Figure 16: Original image 0030 and modified image showing out-of-control pixels with a image-wise control chart approach

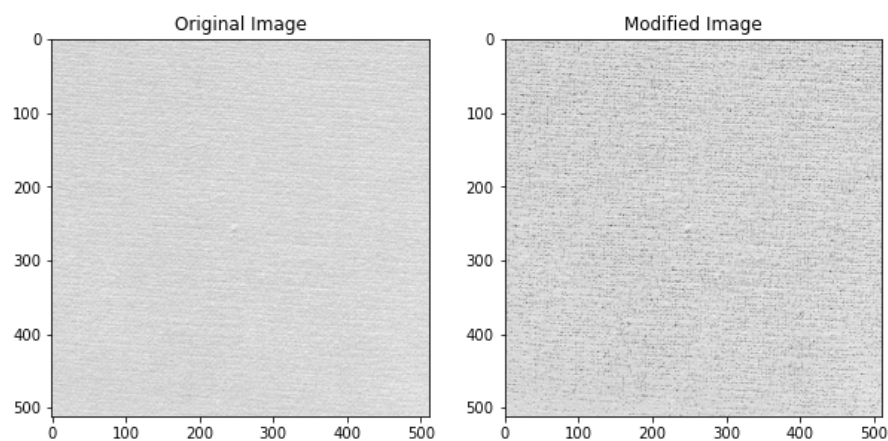


Figure 17: Original image 0008 and modified image showing out-of-control pixels with a image-wise control chart approach

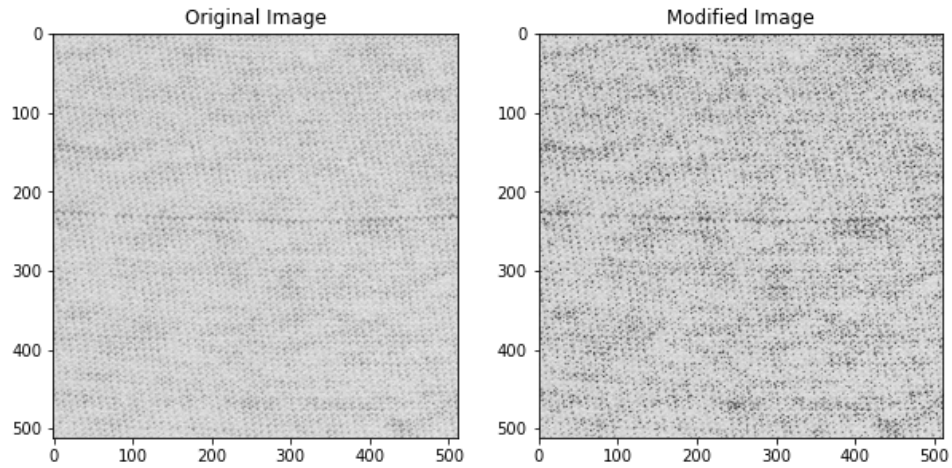


Figure 18: Original image 0191 and modified image showing out-of-control pixels with a image-wise control chart approach

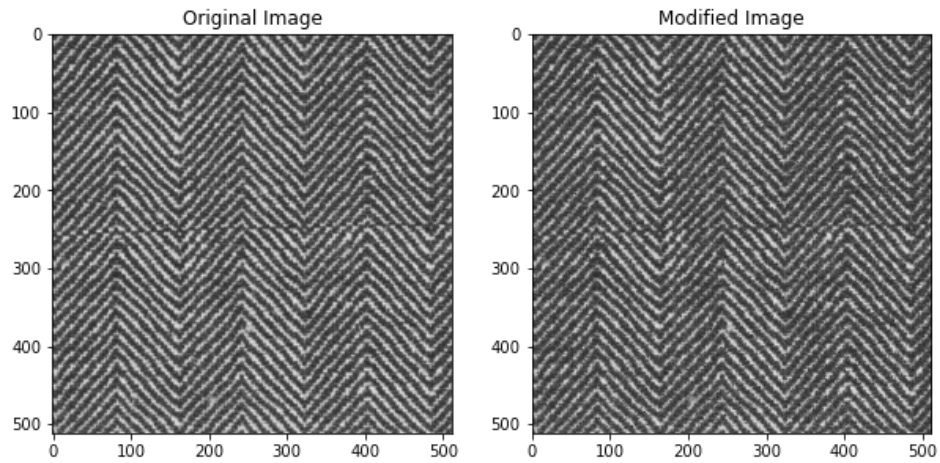


Figure 19: Original image 0072 and modified image showing out-of-control pixels with a image-wise control chart approach

Column-wise Approach Results:

In the column-wise approach, the best defect detection occurred in the image 191. This is because the defect in image 191 is horizontal and the column-wise approach helps better at horizontal defects.

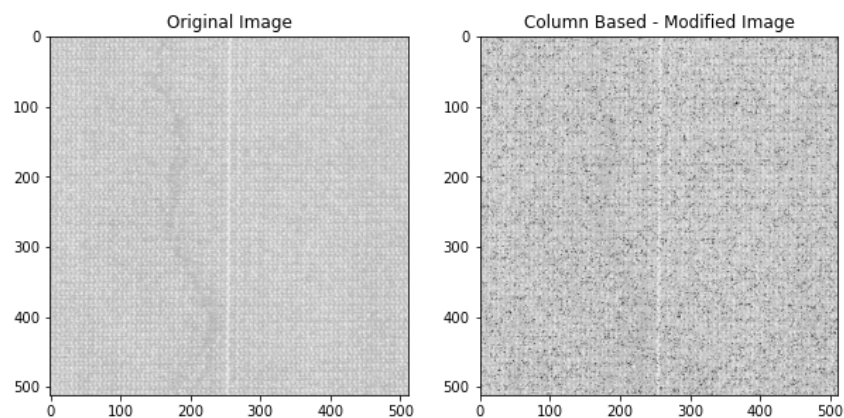


Figure 20: Original image 0165 and modified image showing out-of-control pixels with a column-wise control chart approach

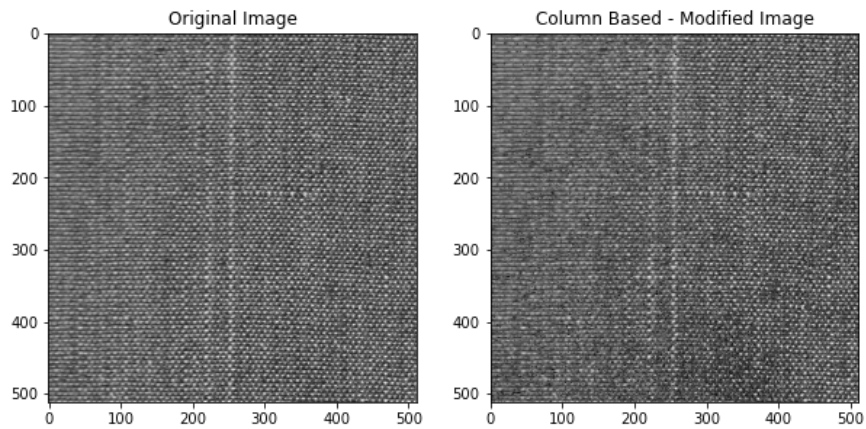


Figure 21: Original image 0030 and modified image showing out-of-control pixels with a column-wise control chart approach

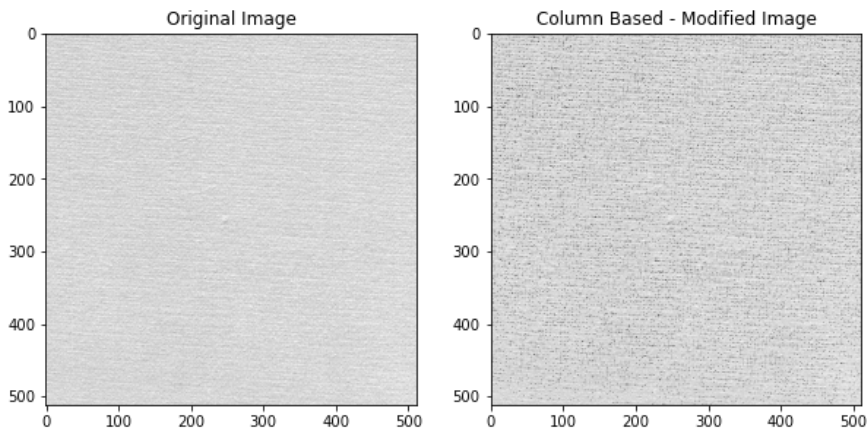


Figure 22: Original image 0008 and modified image showing out-of-control pixels with a column-wise control chart approach

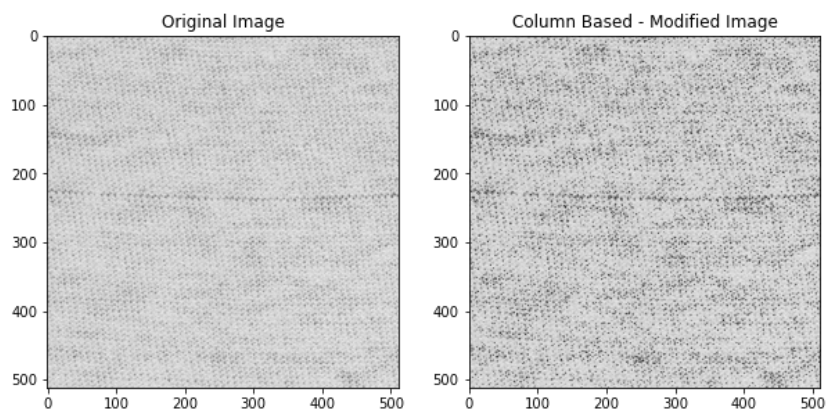


Figure 23: Original image 0191 and modified image showing out-of-control pixels with a column-wise control chart approach

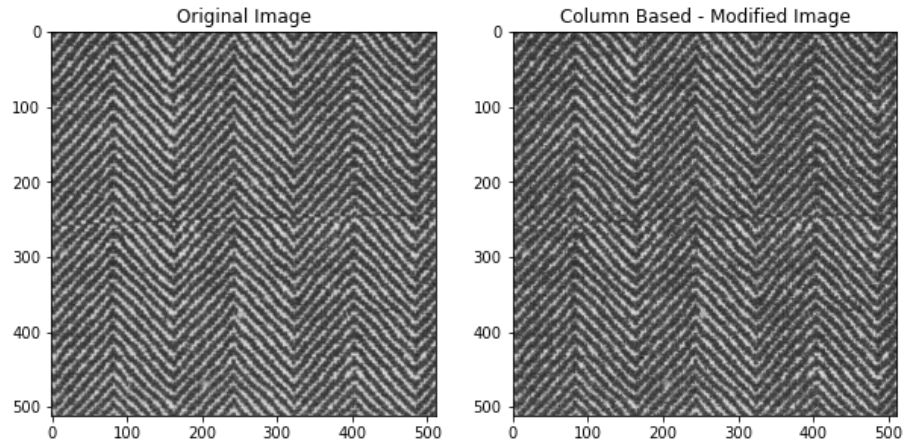


Figure 24: Original image 0072 and modified image showing out-of-control pixels with a column-wise control chart approach

Row-wise Approach Results:

Row-wise approach seems to work well with vertical defects as can be seen in the image 0165.

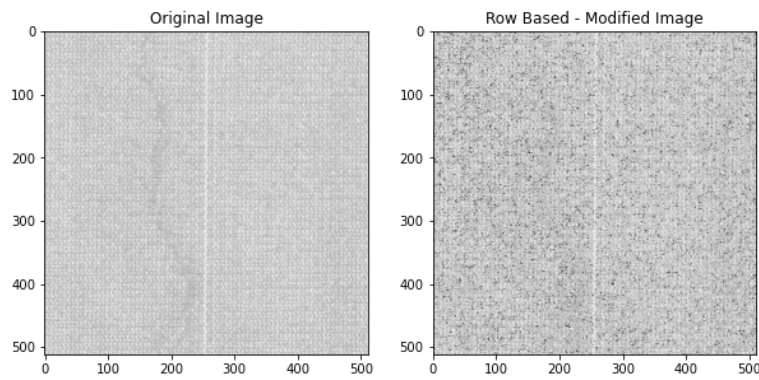


Figure 25: Original image 0165 and modified image showing out-of-control pixels with a row-wise control chart approach

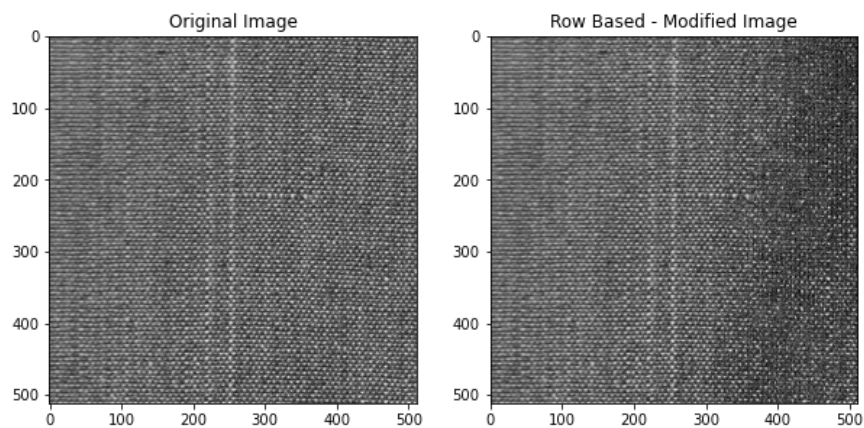


Figure 26: Original image 0030 and modified image showing out-of-control pixels with a row-wise control chart approach

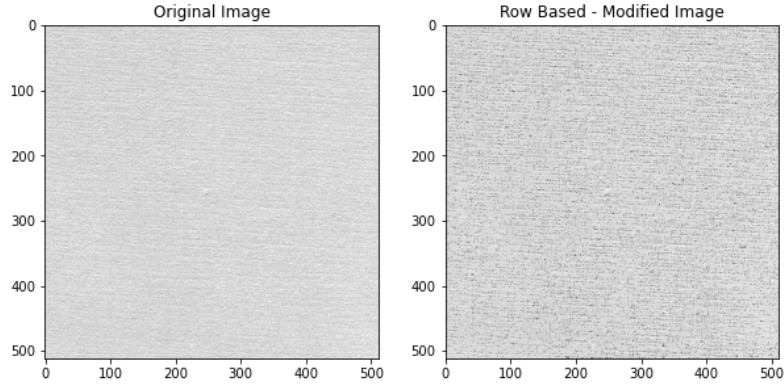


Figure 27: Original image 0008 and modified image showing out-of-control pixels with a row-wise control chart approach

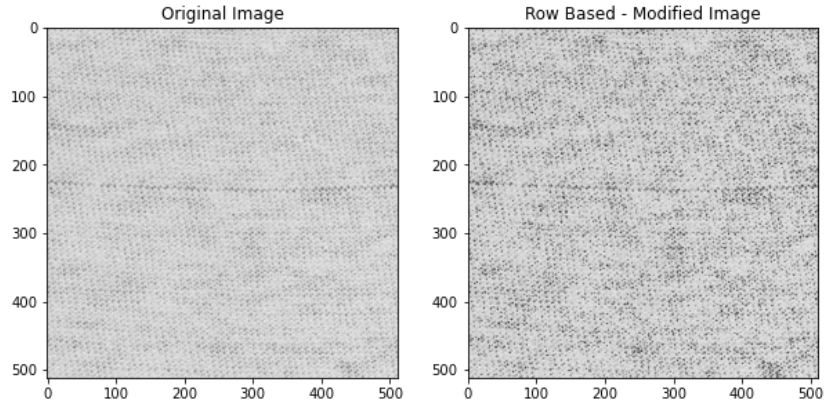


Figure 28: Original image 0191 and modified image showing out-of-control pixels with a row-wise control chart approach

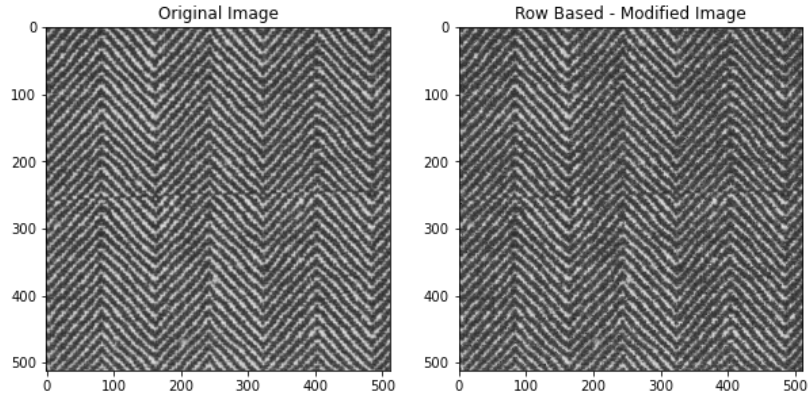


Figure 29: Original image 0072 and modified image showing out-of-control pixels with a row-wise control chart approach

5. Conclusions and Future Work

Our approach using a 2D autoregressive model for defect detection in images achieved promising results. This method effectively identified spatial correlations among pixel values by taking the weighted sum of neighboring pixel values, a significant advancement over traditional SPC methods that treat observations as independent. By calculating residuals between actual and predicted pixel values and applying control limits,

we highlighted areas of potential defects. The addition of row-wise and column-wise control charts further refined our analysis, allowing us to detect anomalies specific to certain image regions like vertical and horizontal.

The current model could be enhanced by incorporating more complex spatial relationships. For instance, using larger kernels or more sophisticated weights within the kernel could capture a broader range of spatial dependencies and improve the model's predictive accuracy. Choosing sigma limits with a more sophisticated approach would also be an improvement to construct our control limits and control charts.

Moreover, dimensionality reduction approach may also be helpful in analyzing spatial data. It can enable us to simplify the 2-dimensional data into a single dimension without losing significant information, making it more manageable and interpretable.

Chat GPT Prompts

> how can i compare the similarity of 2 matrices

To compare the similarity of two matrices, you can use various metrics depending on your specific requirements and the nature of the data. Here are a few common methods for matrix comparison:

Mean Squared Error (MSE):

- Calculate the element-wise squared difference between the two matrices.
- Take the mean of the squared differences.
- Lower MSE indicates higher similarity.

```
mse = np.mean((matrix1 - matrix2)**2)
```

Structural Similarity Index (SSI):

- Particularly useful for comparing images.
- Takes into account luminance, contrast, and structure.
- Higher SSI values indicate higher similarity.

```
from skimage.metrics import structural_similarity as ssim
similarity_index, _ = ssim(matrix1, matrix2, full=True)
```

Cosine Similarity:

- Treat each row or column of the matrix as a vector and calculate the cosine similarity between these vectors.
- Values range from -1 (completely dissimilar) to 1 (completely similar).

```
from sklearn.metrics.pairwise import cosine_similarity
cosine_sim = cosine_similarity(matrix1.reshape(1, -1), matrix2.reshape(1, -1))[0, 0]
```

Correlation Coefficient:

- Measures the linear relationship between the elements of the matrices.
- Values range from -1 (perfect negative correlation) to 1 (perfect positive correlation).

```
correlation_coefficient = np.corrcoef(matrix1.flatten(),
matrix2.flatten())[0, 1]
```

Choose the method that best suits your application and the characteristics of your matrices. Keep in mind that these methods provide different perspectives on similarity, and the choice depends on your specific use case.

> i have 2 images with defects marked with black color. One of them uses a row based approach and the other one uses a column based approach. the similarity of the both images are around 0.82. Can you help me how can i provide some commentary or comparison regarding the both approaches

Certainly! When comparing two images with defects marked in black using a row-based approach for one image and a column-based approach for the other, you can provide commentary on the

differences and similarities in the defect detection strategies. Here's an example of how you might structure your commentary:

Introduction:

Introduce the comparison by stating that you have two images, each with defects marked in black, and that one image utilizes a row-based approach for defect detection, while the other employs a column-based approach. Mention that the similarity between the two images is approximately 0.82.

Row-Based Approach:

Strengths:

- Discuss the advantages of the row-based approach.
- Comment on how this method is effective in capturing defects that span horizontally across the image.
- Highlight any specific characteristics of defects that are well-captured using this approach.

Limitations:

- Acknowledge any limitations of the row-based approach.
- Note instances where vertical defects may be less effectively detected compared to horizontal ones.
- Consider any challenges or blind spots associated with this approach.

Column-Based Approach:

Strengths:

- Discuss the advantages of the column-based approach.
- Highlight how this method excels in capturing defects that span vertically in the image.
- Comment on specific characteristics of defects that are well-detected using this approach.

Limitations:

- Acknowledge any limitations of the column-based approach.
- Note instances where horizontal defects may be less effectively detected compared to vertical ones.
- Consider any challenges or blind spots associated with this approach.

Overall Comparison:

Similarities:

- Comment on the overall similarity of the defect detection results between the two approaches.
- Note areas where both approaches perform well and identify defects accurately.

Differences:

- Highlight key differences in defect detection strategies between the row-based and column-based approaches.
- Discuss scenarios where one approach outperforms the other or where they complement each other.

Complementary Nature:

- Consider how combining both row-based and column-based approaches could enhance defect detection overall.

- Discuss potential synergies and the possibility of integrating both strategies for a more comprehensive defect detection system.

Conclusion:

Summarize the key points from the comparison, emphasizing the strengths and limitations of each approach. Conclude with a recommendation or insight into how a hybrid approach could be beneficial in improving overall defect detection accuracy.

Remember to provide visual aids, such as side-by-side comparisons of defect detection results or annotated images, to support your commentary.