Universal Pattern Recognition

Project Summary:

Most of our existing optical metrology solutions focus on hard-coded, manual spatial domain methods for identifying and quantifying features within a display surface or illuminated object. This project will explore the possibility of identifying and quantifying abnormalities within the frequency domain to highlight periodic patterns, which may be visible to our customers. Future work includes integration within our Radiant TrueTest sequence so that it may be executed between the time of image capture and the exportation of results to enable its usage during proto and xVT builds. The completion of this project includes explaining a framework for the given objectives:

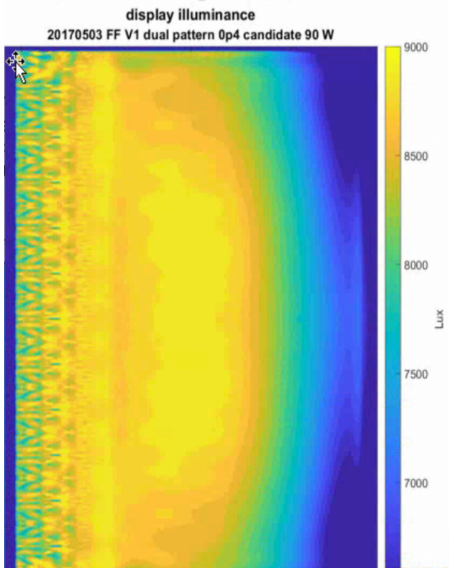
1. machine learning technique so the script recognizes circular light ring displays against squares, rectangles, circles, triangles
2. functioning code that imports an image array (in a scale such as luminance) and exports a singular score ranking the intensity of the detected patterns
3. automated method of detecting where the LED edge is, to enable the algorithm to function properly by determining the frequencies present which are parallel to the LED array
4. how much of the display area that this periodic pattern defect is consuming

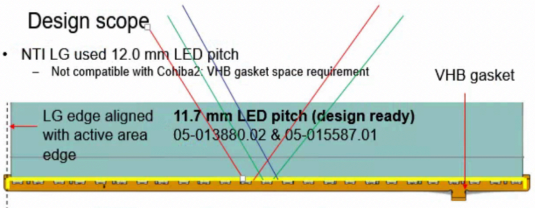
The intent of this algorithm is to pass in a 2D array and return a score. This study utilized units of Nits for each segment of the 2D array, however, an improved correlation with visual perception may be obtained by utilizing L* or contrast. In addition defining the display in terms of visual angle instead of size will help to enable this metric to be more universal.

Tools and Milestone Timelines:

	Phase 1 : Shape Detection	Phase 2: Score	Phase 3: LED location based on	Phase 4: Defect Area relative to display
Round 1	Week 6, (8/3/20) OpenCV functions : 1) arclength 2) approxPolyDP 3) linearPolar	Week 2, (7/6/20) FFT, OpenCV, Scipy Sum of total FFT peaks	Week 3, (7/15/20) FFT, OpenCV, Scipy Crop Section FFT method	Week 4 (7/14/20) Sobel, Scharr operator
Round 2	Week 10, (9/4/20) Pytorch 1) CNN 2) MaxPool2D 3) ReLu	Week 7, (8/10/20) DCT-based Bayesian MLE density function counting black pixels	Week 7, (8/10/20) DCT-based Bayesian MLE DCT model output image	Week 7, (8/10/20) DCT-based Bayesian MLE Segmentation contrast

Background of Pattern Defects :





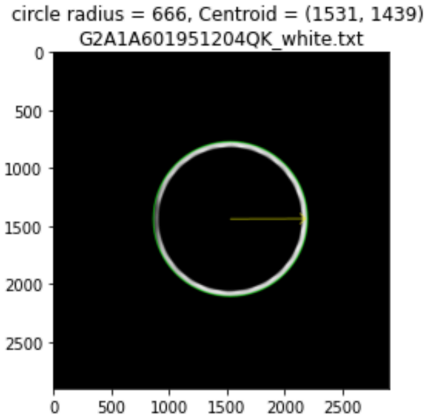
Periodic defects are resultant from the extraction patterns at the bottom of the Light Guide (LG) and microlenses, **as shown in MP product to the leftmost image.** LG and microlenses function similar to optical diffraction gratings, with periodic pitches in spatial variation located nearby the LED edge. By use of Total Internal Reflection, the light is input, either from the left-side **("Long" axis)** or from the bottom of the display **("Short" axis)**, and guided throughout the rest of the display to illuminate in Amber, White or Mixed mode.

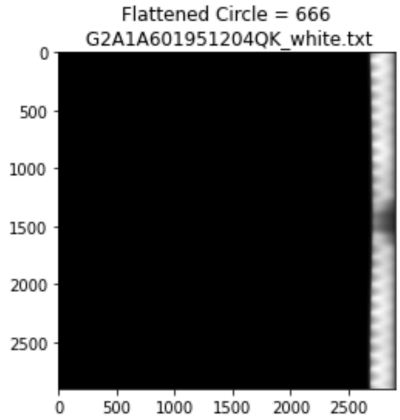
Middle image shows a heat map of display where the left edge exhibits a "Long" periodic defect.

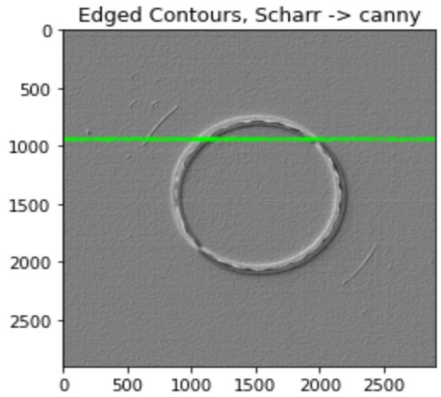
Tenet:

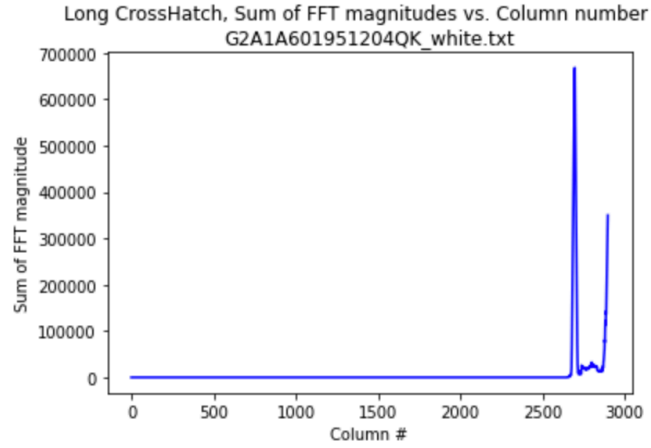
"Short Pattern": Refers to an instance where the LED array is located on the short-side of a rectangular display for Martini/Malbec.
"Long Pattern": Refers to an instance where the LED array is located on the long-side of a rectangular display for Vermouth/Singer.
"Circular Pattern": Refers to an instance where the LED array is distributed along the circumference of the light guide for Laser.

Phase 1) Shape detection: To enable the algorithm to function with any device, without any hardcoding: we first begin by padding the perimeter of the image with 0's for the shape recognition, binarize the image to black and white, and calculate the contours/edges of black vs. white pixels as approximate the polygon. Contours are passed as arrays into OpenCV's arclength and approxPolyDP, which then approximates the edges as the nearest neighboring shape. By counting the number of vertices in that shape, we recognize circular light rings or displays (possibly train with squares, rectangles, circles, and rings) without machine learning for faster processing. Sample arclength and approxPolyDP taken : https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/



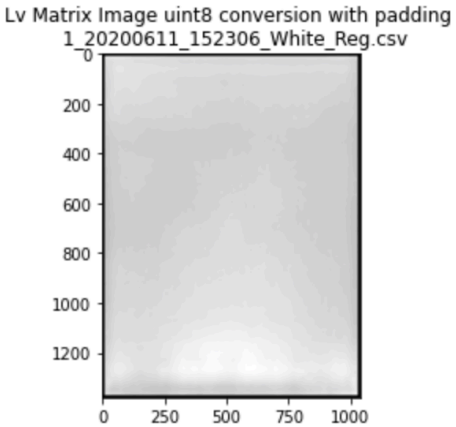


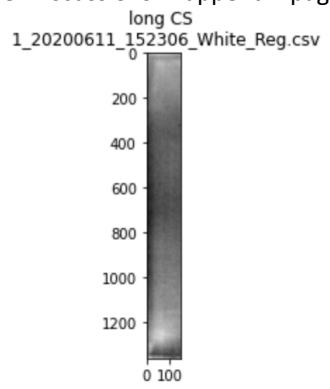




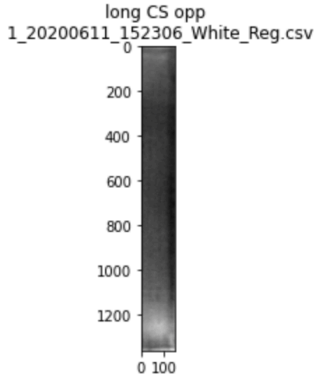
Then OpenCV's minEnclosingCircle is used to find the radius and center of the circle, where the radius + center is used to determine the polar to cartesian conversion of the image.
linearPolar: https://stackoverflow.com/questions/23127769/linearpolar-logpolar-transformation-distoring-an-iris-in-opencv, is used to convert images from Cartesian to Polar **(shown in top middle image)**; it starts from radius=0, theta=0, and unpacks the pixels starting from the center of circle, to outer radius edge. Finally, in the flattened image in middle, shows column #0 corresponding to the center of the original circle and column #3000 corresponds to outer edge of circle. The LongFFT analysis is performed on flattened image to determine where the LED edges are located and what the visual score is. Basically the column # corresponding to the highest FFT sum of amplitudes correspond to LED edge.

Again, the perimeter is padded with 0's for shape recognition. Below on this page is an example of a rectangle that was also shape detected in the same code. Discussions in appendix page, reveal that simply cropping & FFT'ing the areas fail.

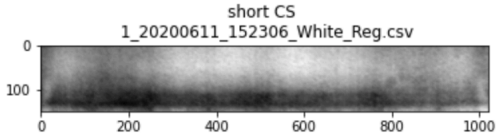




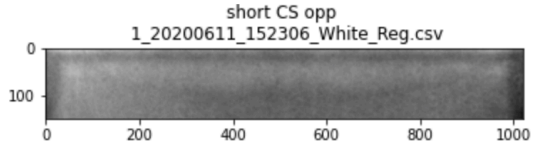
score is 528481.9952524701



score is 170279.76761110834



score is 420870.27015969466



score is 154651.743620164

Phase 2 and 3) FFT SHORT ANALYSIS: To quantify a pattern(s) of defect score and LED location within the illuminated area that may be visible to our customers. First a series of spatial domain plots, with their corresponding frequency domain amplitudes, must be inspected in order to ensure the correct threshold is placed in the findpeaks function.

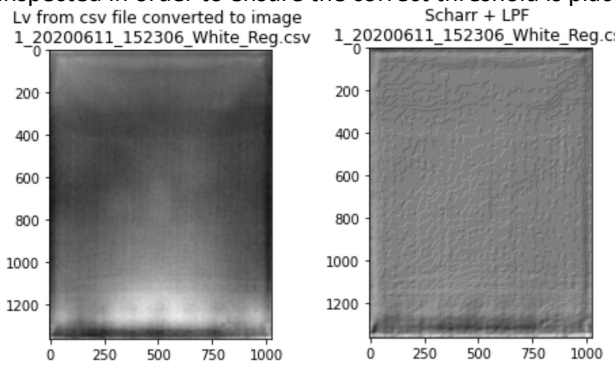


Figure 1: The leftmost image is the original Lv matrix of a "White" Martini short-defect dataset. White displays have 100% of power in white LED temperature, Mixed has 90% Amber and 10% White, and Amber is 100% amber LEDs. **By inspection, it seems that after Row #1200 and closer to #1350, the periodicity is most noticeable.** On average, the short crosshatch pattern is such that White images have the worst/highest score, then amber, then mixed.

From top leftmost images in Figure 1, at row 500, there is no apparent crosshatch defect in the displays.

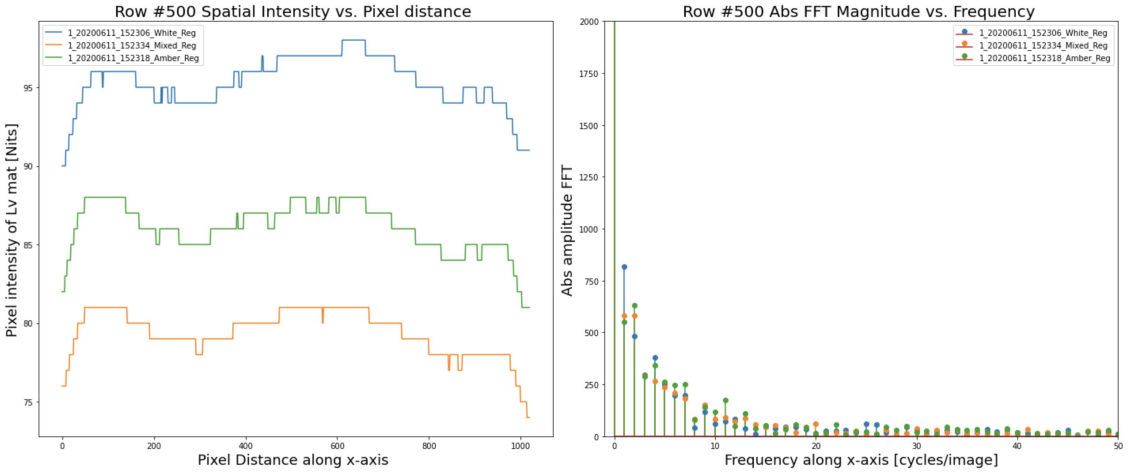


Figure 2a : leftmost plots are the spatial domain plots overlayed—legend shows blue is "White" LED display, orange is "Mixed", and green is "Amber". Sampling frequency is 100 Hz for all FFT long and short analysis.

Therefore, the **top left Fig2a plots looks relatively smooth**. Because of its uniformity, it is unclear where to set the FFT amplitude threshold in findpeaks function also indicated in the top right FFT plot. In addition, the White plot averages about 95 nits, which is more brighter than Amber and Mixed**. The top right plot**s **Fig2a**, shows the overlay of power vs. frequency (cycles/image). In other words, the rightmost image is a construct of power spectral density estimates, where the units are linear similar to [mW/Hz] or [V^2/ Hz]. Therefore, it is customary to take the logarithm of that power, because the log-scale is a variance stabilizing transformation for the power spectral density. In optics, this is akin to working with dBm (log-scale) as opposed to mW (linear). However, for the purposes of visualization, it is best to see the the FFT graphs in linear scale. The scores will be later resolved into log-scale do this later, because they are in the order of millions. Converting to log-scale makes more sense for assigning scores

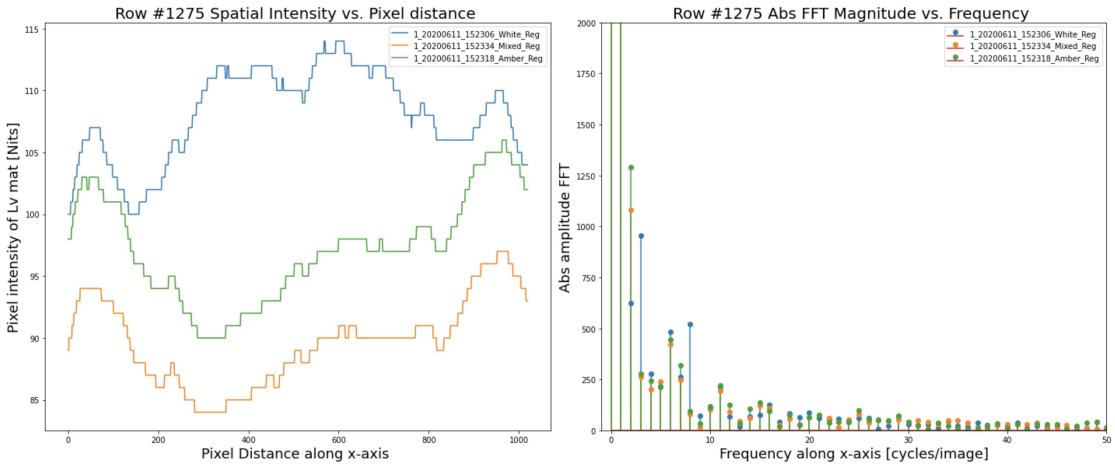
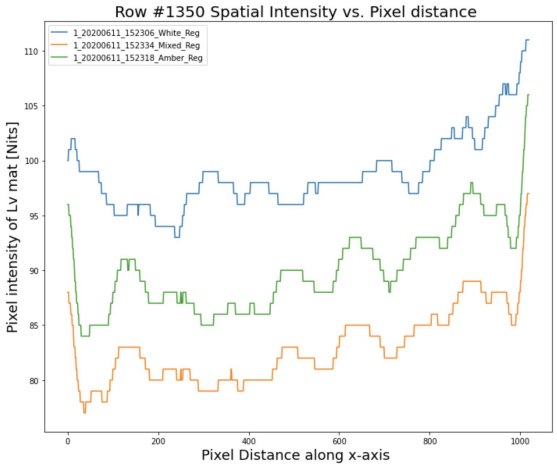


Figure 2b: Leftmost FFT plots, shows how some of the **power is starting to center around 8 cycles/image**. This is for row #1275, when the periodicity starts to increase by inspection from **Figure 1**



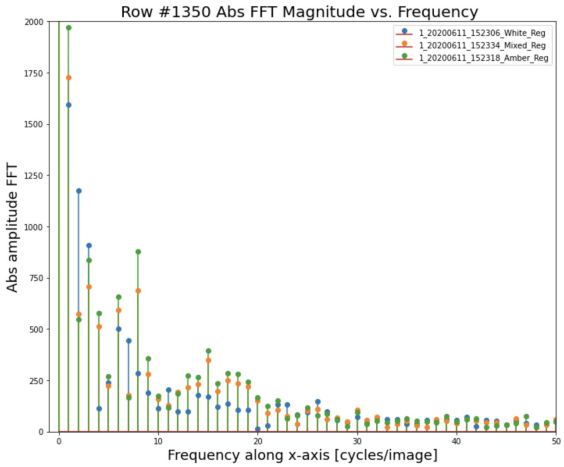


Figure 2c: left plots shows how even more of the power is starting to center around 8 cycles/image—increased magnitude from 500 to nearly 1,000 in magnitude

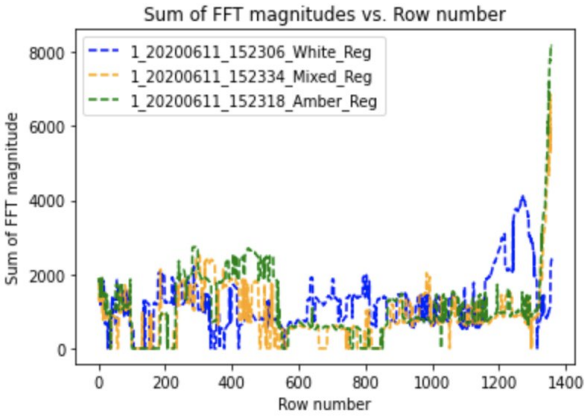


Figure 2d: When setting the threshold FFT height=[375, 2000] which is around 8 cycles/image, all datasets received the same score 1360. Therefore, instead of counting the number of FFT peaks strictly, we must count not only the total number of peaks but moreover the numerical amplitude of FFT peak. **FFT short analysis score is the total number of peaks within the FFT magnitudes height=[375, 2000].** As seen in left plots, since there's a maximum of FFT magnitude at row 1360, which corresponds to highest periodicity in intensity, **this is the LED edge.** Ranked from highest to lowest: White Score is: 1,685,107 ; Amber Score is: 1,572,936 ; Mixed score: 1,252,548

Phase 2 and 3) FFT LONG ANALYSIS: Quantify patterns of defect score and LED edge location

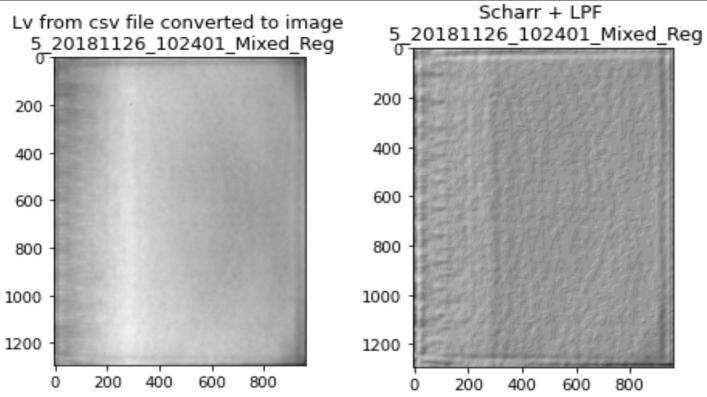
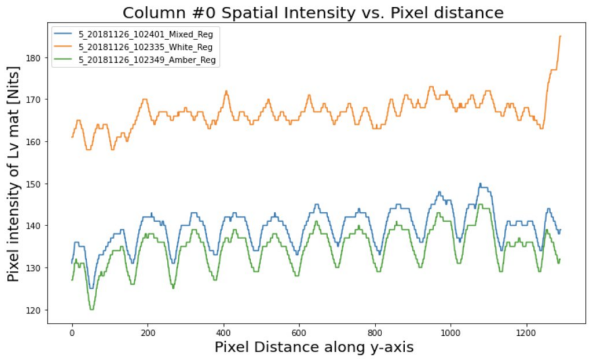
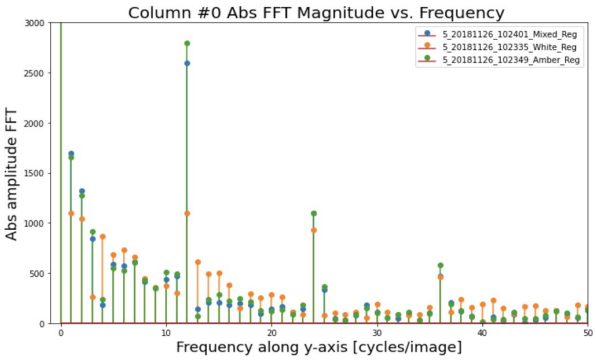


Figure 3: The leftmost image is the original Lv matrix of a "Mixed" Vermouth long periodic-defect dataset. **Notice how the periodicity is dramatic from Column #0 all the way until #75 roughly. We hope to re-create the same plots of maximum FFT magnitudes at Column #0, which indicates LED edge.**
Score is: 3,015,536

Figure 4a: Bottom plots, while ignoring the DC order in the bottom right plot below, most of the energy density is centered and periodic about 12 cycles/image, 24 then 36.





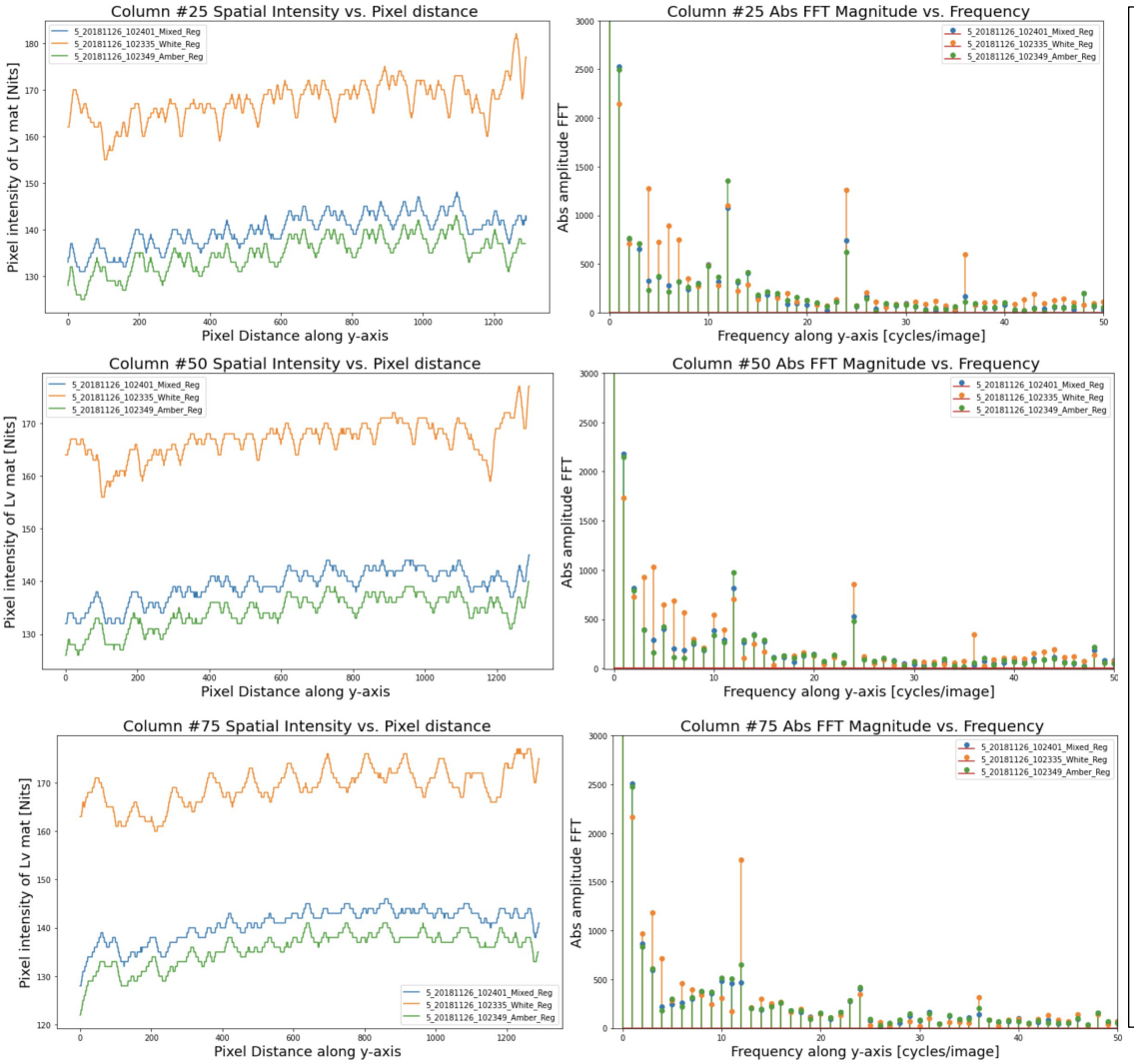


Figure 4b: While ignoring the DC order in the top right plot, most of the energy density is centered and periodic about 12 cycles/image, along every column slice FFT. Then, 24 cycles/image starts to become as dramatic, as its magnitude becomes comparable to 12 cycles/image at around Column #25. This observation will be further elaborated in the upcoming DCT-based Bayesian MLE discussion.

Since the 24 cycles/image correspond to the number of peaks in the left-edge of **Figure 3**, which is correlated to the number of LEDs also found in the left edge, the FFT score threshold is set at FFT height=[250, 3000]

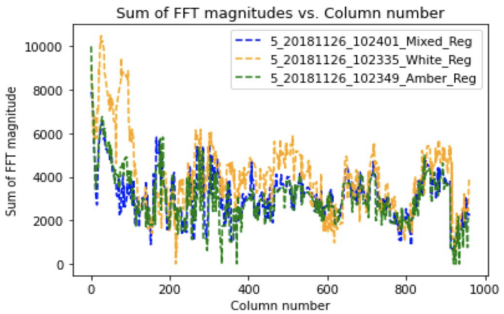
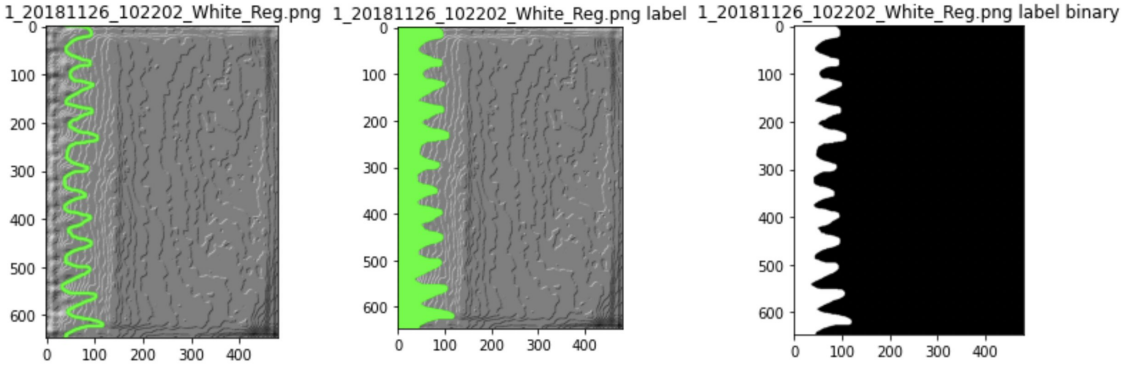


Figure 4c: **FFT Long analysis score is the total number of peaks within the FFT magnitudes height=[250, 3000].** As seen in left plots, since there's a maximum of FFT magnitude at column #0, which corresponds to highest periodicity in intensity, **this is the LED edge.** Ranked from highest to lowest: White Score is: 3,875,370 ; Amber Score is: 2,898,055 ; Mixed score: 3,015,536

Phase 1 and 4) SHAPE DETECTION AND AREA OF DEFECT: Shape detection of a circle vs. rectangle is easy, but after the image is classified as a rectangle, then another shape detection step is required to differentiate the area of peaks as either long or short defect. Using Bayesian Decision Theory (BDT), we present a framework for estimating the maximum likelihood expectation (MLE)



that a crosshatch defect is detected in a given image: given the rectangle display input, can we teach a computer to segment it into a foreground (short crosshatch or long crosshatch) and background (rest of display without crosshatch defect)?

Figure 5: (previous page) top leftmost image is the labelled data, using a Stinger image. Initially, 12 peaks were only drawn according to what the long cross hatch roughly looked like. There are 12 prominent large peaks, so in green they were labelled with an online paint program (https://jspaint.app/#local:9e4be4c3f95f3). Previous page rightmost top image is the binarized image, so given that the maximum likelihood of the black pixels are background, (# black pixels / total # of pixels), what percentage is white? The answer is the total probability that this image is long crosshatch. **This binarized image is our ground-truth for analyzing every other long crosshatch image in Round 1 .**

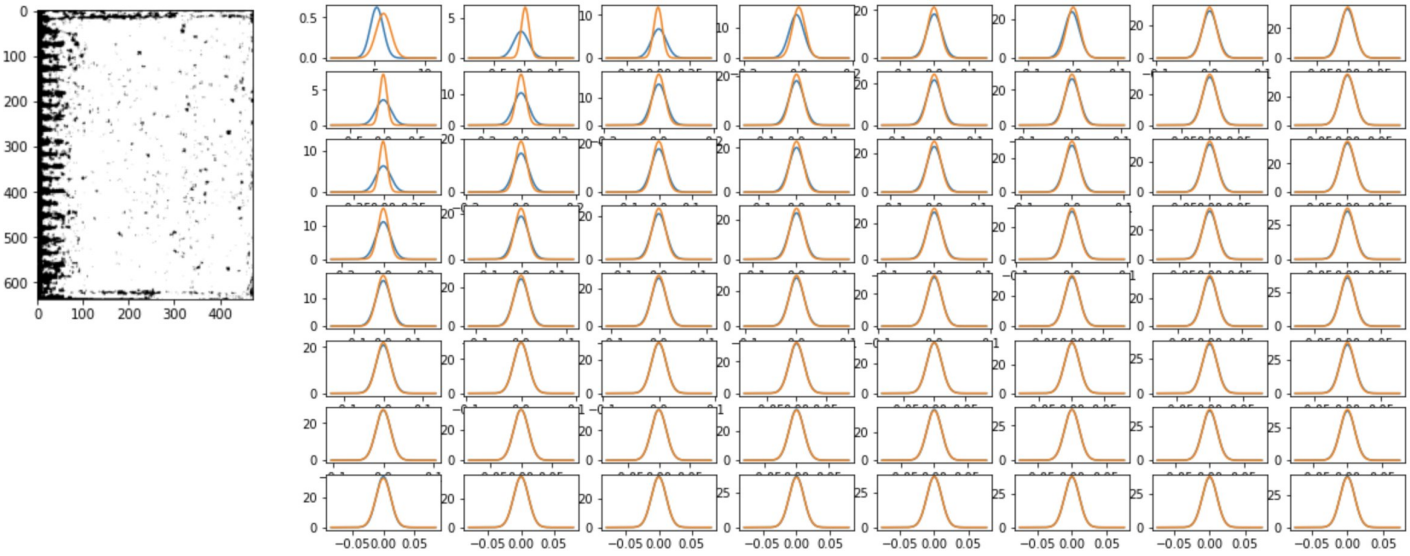
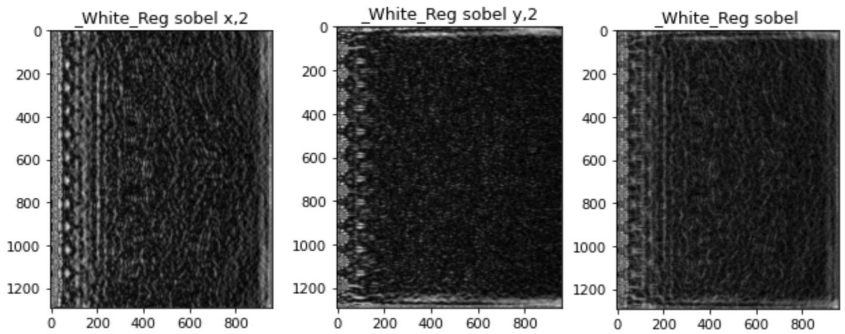


Figure 6: The DCT-based Bayesian MLE model generates a corresponding output image with black pixels filled in—top left image where the black peaks are an estimated likelihood of long periodic-defect pixels. The input image is divided into 8x8 patches and each 8x8 patch's DCT coefficients are computed. Then, each of the input image's DCT coefficients are checked against the labelled ground truth DCT coefficients. Calculating the likelihood that the input images' patch belongs to the Foreground (blue density function), we compute the conditional probability, given the mean and covariance of the labelled truth data, in order to determine what's the likelihood that the current input image's DCT coefficients most closely resembles a Foreground (crosshatch) or Background (not crosshatch). **The number of estimated likelihood crosshatch black peaks totals to 24 as shown in the left image of figure 6 (just count the number of black peaks at the left edge of this image**). **Remarkably, the DCT model was able to locate the other smaller amplitude peaks (ie, recall in Methods section, only 12 of the most prominent peaks were labelled in green and binarized, but the model still found 12 more of the smaller peaks, totaling 24). This Phase 4 result confirms the initial FFT stem plot analysis in Phase 2 and 3 section of FFT Long Analysis.** Therefore, it makes sense that most of the power would be centered at 12 cycles/image harmonics, then 24.



Sobelx2 was calculated with the derivatives in x—its purpose is to dramatize variations along the x-axis.
Sobely2 was calculated with the derivatives in y—its purpose is to dramatize variations along the y-axis.
Sobel is the culmination of the gradients calculated by combining both directional gradients in x and y

Limitations: Ideally, the culmination of these outputs (scharr, bilateral blur, laplacian, sobel) should be used to potentially integrate 2 scores into the TrueTest software. That way we know how visible the pattern is in terms of intensity which can then be converted to JND, and we will know how large it is in terms of the display area. When we sum all the magnitudes in FFT short and long analysis, we convolve the 2 factors, so the current limitation is that people will not be able to relate to what the value is (is it large because the signal is strong or is it large because it extends over a large area). The results must be separated to 2 of these independent random variables. In addition, each image takes about 10 minutes to process every single pixel with this macbook, which is too slow. However, if the goal is to save time, then skipping every other pixel drives down the processing time to 5 minutes per image while sacrificing resolution. If a lower resolution output image is acceptable, then further reduction of pixel comparators can drive down processing time to a few seconds with a GPU, more CPU cores parallelizing, or AWS server.

Appendix [the best part]: Pattern Detection and Shape Recognition

1) Some closing remarks on TrueTest:

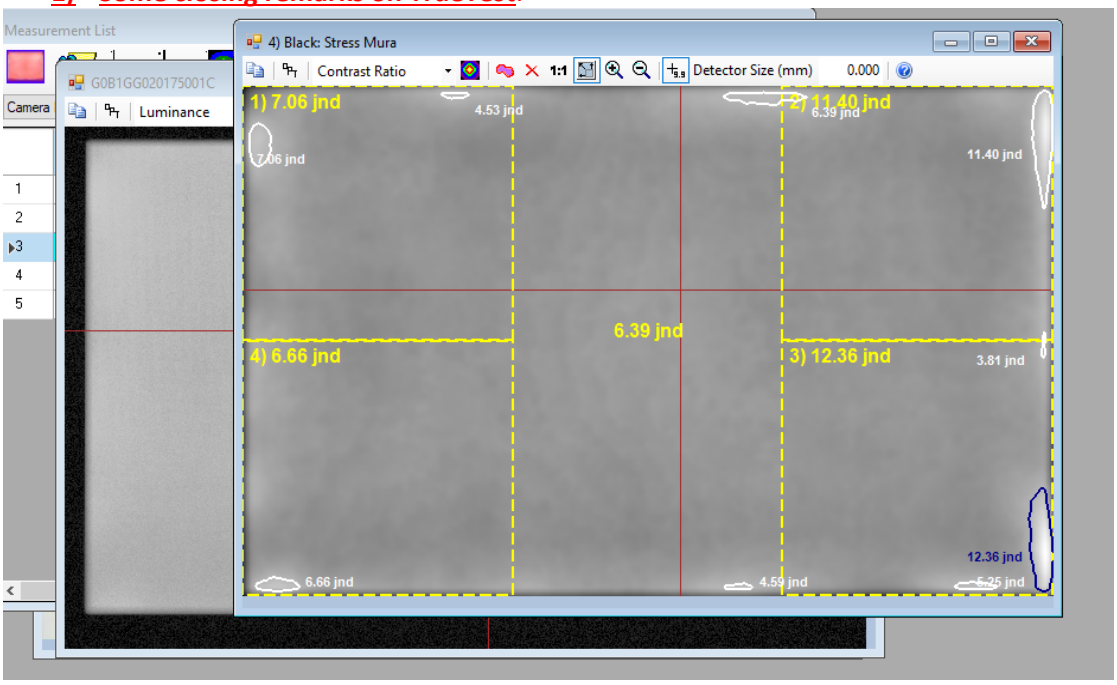
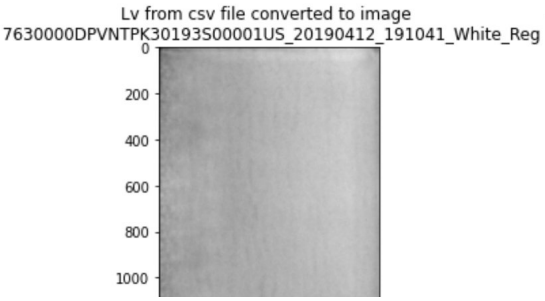


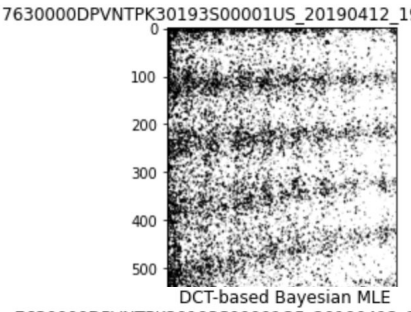
Figure 7: Top image is screenshot of the TrueTest Run analysis on Stress Mura. Ideally, this python script should be loaded into a DLL, so a custom FFT run analysis can be made for each device under test (DUT) to determine the severity of crosshatch defect score in jnd—similar to how stress mura is currently being run. The output image of the DCT-based Bayesian MLE, where the black peaks are labelled as a periodic pattern, should be outputted as one of the run analysis outputs images for seeing area. In addition to the sobel and scharr output image for the area of defect visualization should be done. The intent of this algorithm is to pass in a 2D array in the TruthTest run attached processes to return a score.

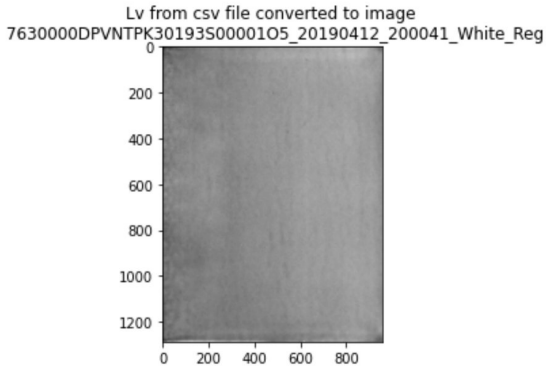
2) Why Phase 3 (LED location detection), of Round 1 Crop Section FFT method kept failing, as stated in bottom page 2:

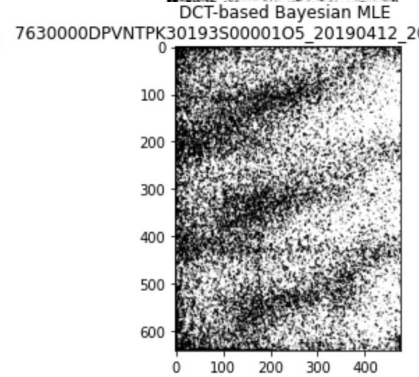
Simply cropping & FFT'ing the expected areas of where the periodic-defect pattern will fail for these reasons. When trying to do a pure FFT analysis on a suspected long periodic-defect, there also seems to be periodic defects in the short axis as shown in the next few pages. I thought the output of the the DCT-based Bayesian was wrong but these indicated in black pixels are truthful artifacts:

DVT Vermouth with likely long crosshatch defects in the left-sided black peaks shown below:



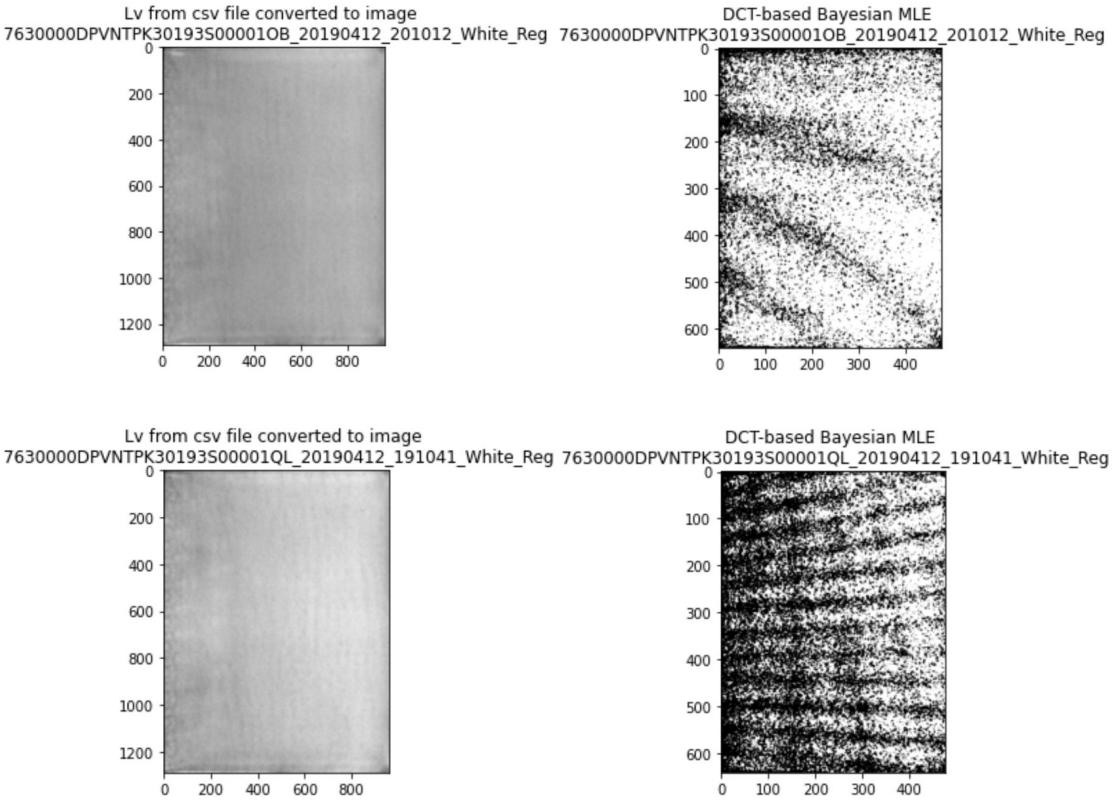






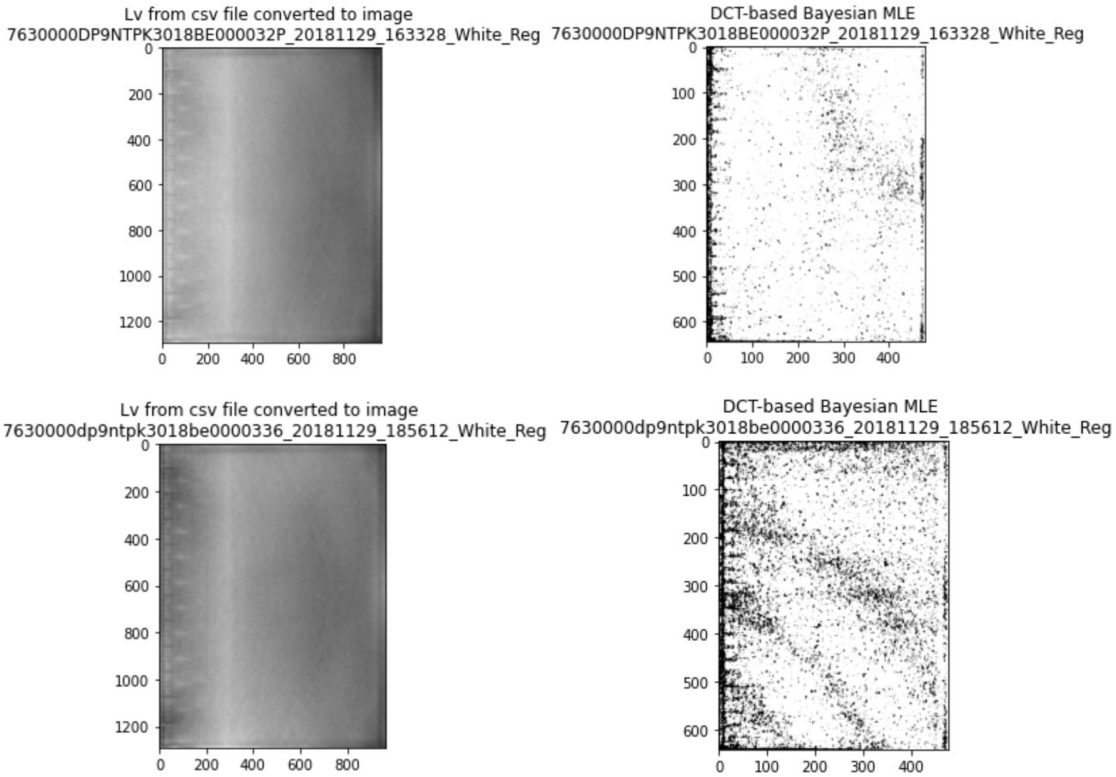
To the left is the original Lv image, to the right is the output of the Bayesian model. Notice the strange diagonal artifacts being detected.

Based on the Vermouth DVT input image, the model estimates that the black pixels had highest likelihood of being the long crosshatch pixels. But it seemed to be wrong, so eventually the labelled truth data was changed.



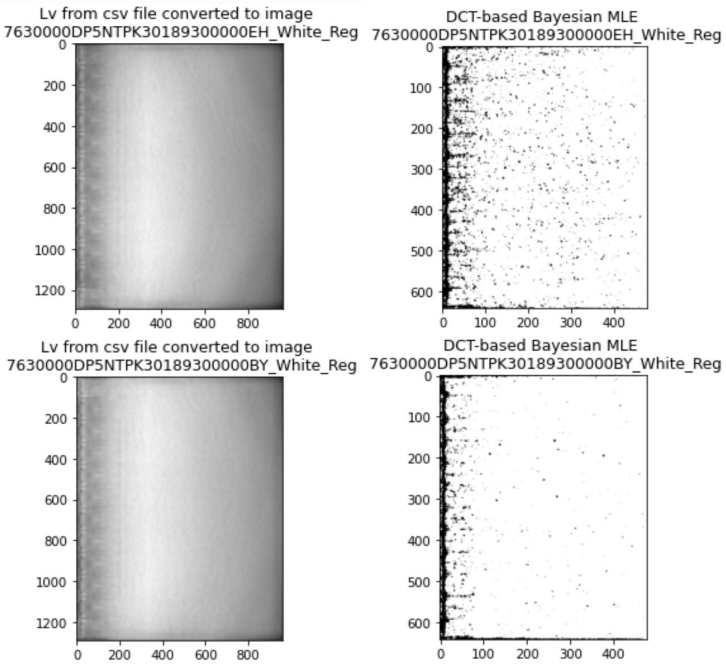
This explains why our first approach kept failing regardless of summing the number of FFT peaks or amplitude of FFTs for cropped cros-sections. Because even in these images, there seems to be both a long and short crosshatch pattern for the DVT Vermouth White dataset—which is supposed to be a pure long crosshatch classifier.

EVT Vermouth images with likely long crosshatch defects in the left-sided black peaks shown below:



Although there's now 24 black peaks to the left side of DCT-Based Bayesian MLE image, the diagonal crosshatches from DVT datasets appear here in EVT, too.

HVT Vermouth images with likely long crosshatch defects in the left-sided black peaks:



At this point, it was unclear if the DCT-based Bayesian MLE model was actually correctly labelling the estimated long crosshatch patterns for reasons:

1) the diagonal artifact in the Vermouth DVT and EVT dataset is inexplainable

2) there are less black pixels in HVT and EVT, as opposed to the DVT—where the DVT has some diagonal crosshatch noise that is corrupting the truthfulness of long periodic-pattern defection. DVT dataset is expected to perform better with a smaller crosshatch score, relative to HVT and EVT. But for some reason it seemed to perform worse.

Therefore, another iteration of labelling with an actual Vermouth image must be done in order to truly capture the defects.

ROUND 2: METHOD FOR DISTINGUISHING BETWEEN LONG AND SHORT CROSSHATCH

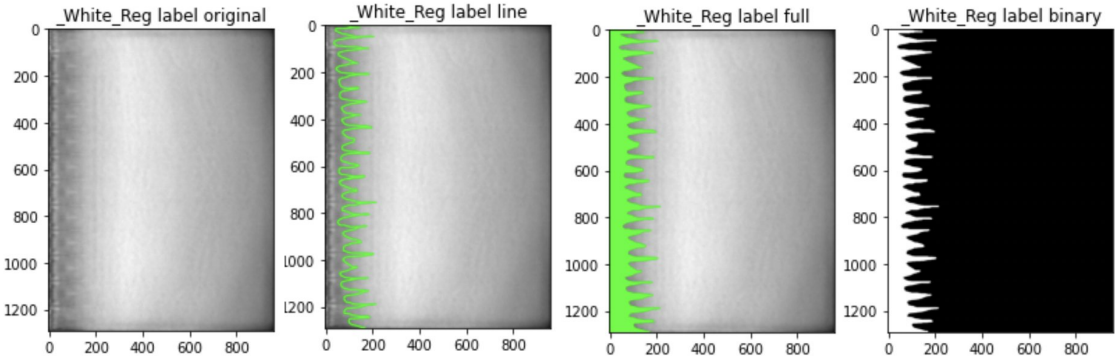


Figure 8: Labelled again, but this time, added the 24 peaks instead of only 12, in order to be as close to the ground truth

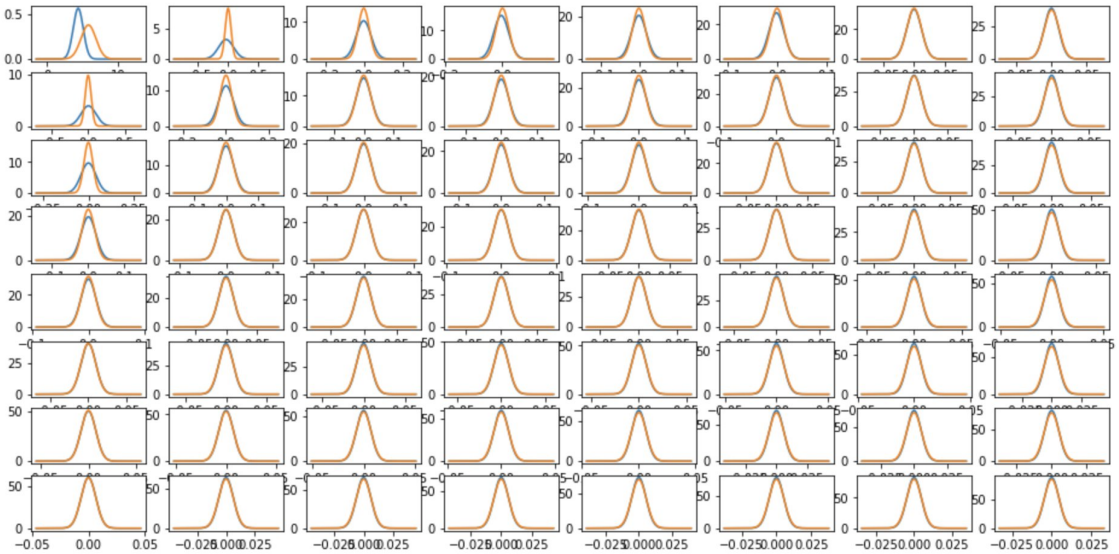
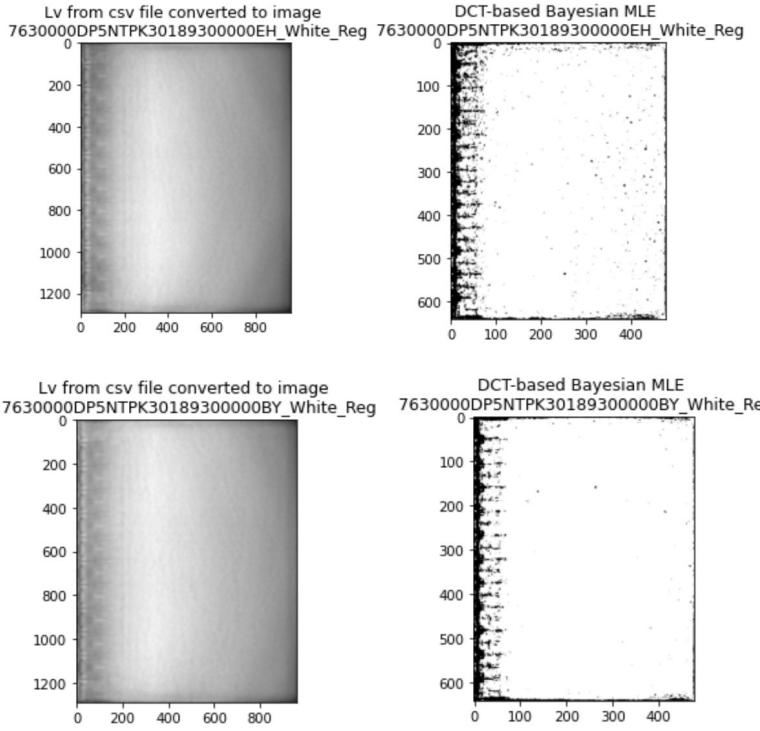


Figure 9: Above is the multivariate Gaussians of 64 dimension distributions of an input image, where blue is the sample-mean, sample-variance and sample-covariance of the Foreground valid crosshatch pixels and orange is of Background invalid crosshatch

pixles. The DC order means are more separated than the first one in page 6, so that's a good indication that the Foreground and Background are well separated and independent density functions.

Vermouth HVT image as our new ground truth yielded more predictable results:

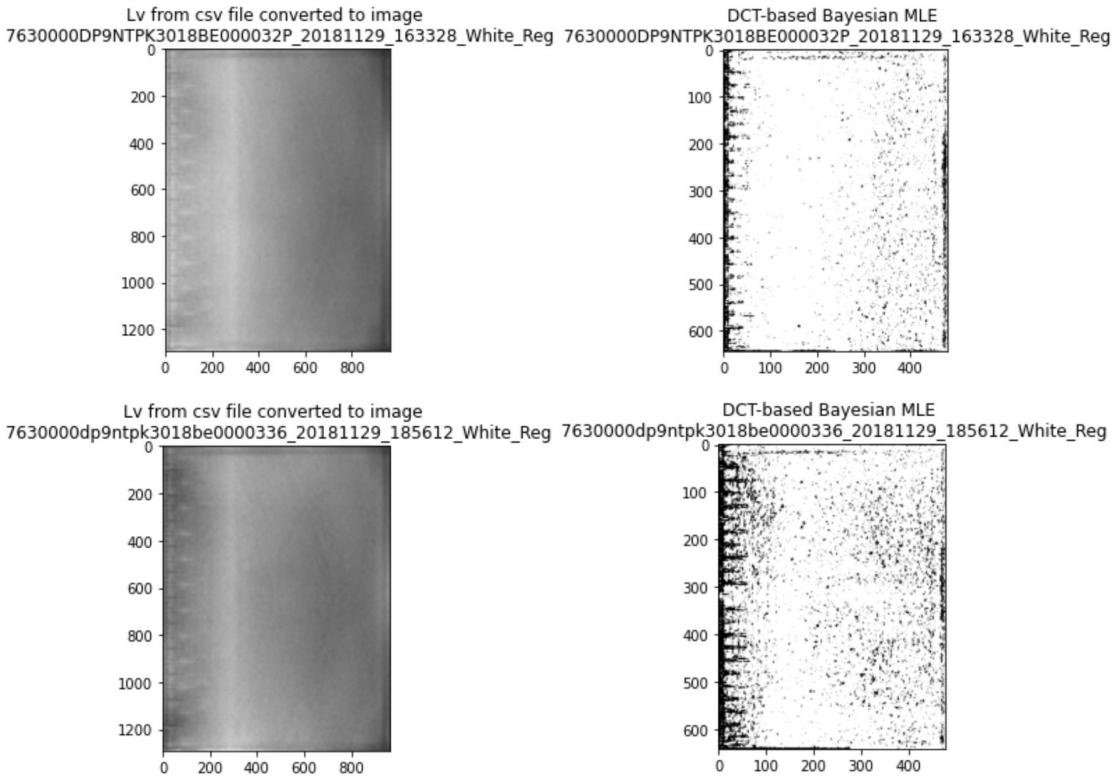


To the left is the original Lv image, to the right is the output of the Bayesian model.

The new labelled ground truth yields improvement in visual score. The back pixels are more concentrated and focused at the leftmost edge of the expected long crosshatch. This leads to a strong, centered score where the peaks are countable like **Figure 6, page 6.**

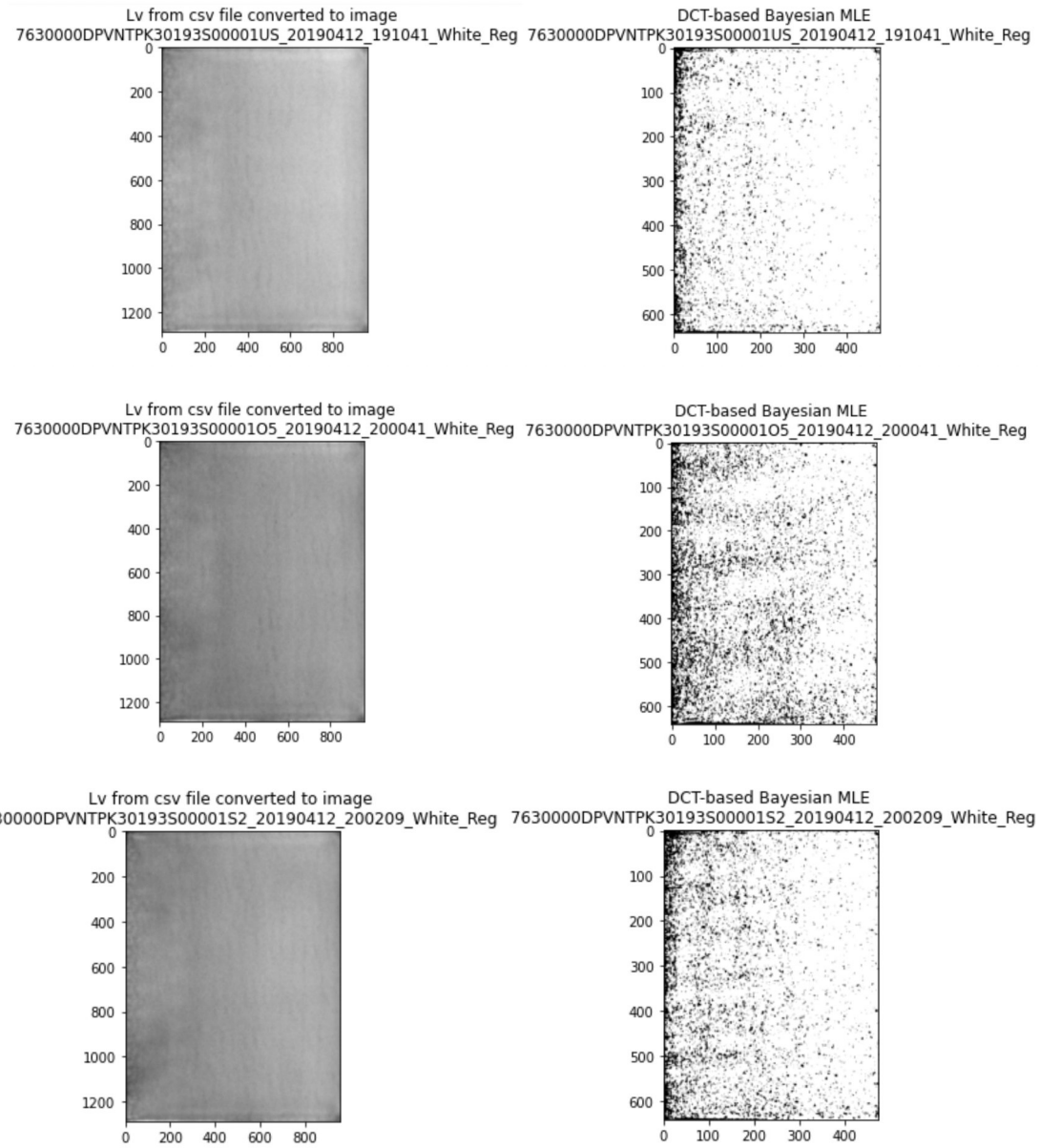
There is still some error to the right of the HVT input images (notice the diffused speckles of black pixels, where the model thinks those areas correspond to the valid periodic patterns).

Vermouth EVT with new label ground truth



However, there appears to be more error in the EVT than HVT dataset, where the black pixels are further to the right and beginning to diffuse more,
as the left-sided crosshatch defect becomes less concentrated and less focused—relative to the HVT images above.

Vermouth DVT with new label ground truth



For some reason, the diagonal artifacts patterns seem to stay in the DVT images, regardless of the new ground truth labelled dataset.

In theory, the more truthful the ground truth is labelled, the better result are shown. However, at this point Matt said those artifacts are a real phenomena, and it's not just a model loss error.

Those diagonal patterns appear to be moiré, which is resultant from refresh rate aliasing from CCD sensor array in the camera and display panel. This is typical in displays where the emission pattern has a regular periodicity from all the pixels and the camera used to image the display also has periodicity in the CMOS sensor array.

Apparently the DCT method is enhancing this issue by removing many other dominant signals. Therefore, more pre-processing was done to further investigate the shape of this noisy diagonal crosshatch pattern that is corrupting our signal left-sided long crosshatch:

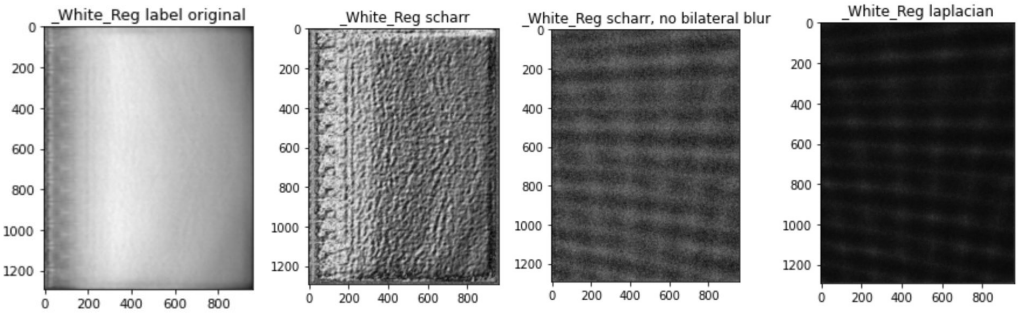


Figure 10: leftmost image on top is the original image. **scharr** was taken with the same kernel = [[0, 1, 0], [1, 0, -1],[0, -1 ,0]] as always. **no bilateral blur** is a normalized output of scharr, where contrast is finally scaled. **laplacian** is the absolute value of the laplacian operated onto the original image. Interestingly, the DCT-based Bayesian MLE model was picking up on contrast as shown

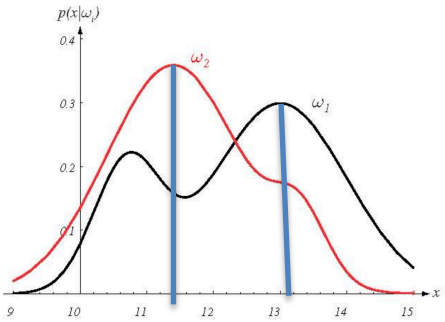
above (since the output of the Scharr filter results in pixel values between [-2, 2], and the final values weren't scaled by 255). Since the model's output was capable of being re-created by the **normalized contrast scale of the Scharr image, it seems like the diagonal noise is emanating from contrast sensitivity.** Further investigation should be done to normalize this contrast sensitivity function to human perception in L*. But we suppose this is a good thing, if the model is capable of detecting artifacts that were not only unintentionally there but also correlated to human perception, and even more artifacts that were unexpected, then that's a win!

3) **A note on the magnitudes of the FFT plot: they can be normalized by dividing by the signal length (# of pixels). But it may also be compounded by other assumptions such as a sampling rate (of 100 cycles/image).**

4) **For the interested reader, Bayesian Decision Theory MLE is super old school machine learning**

[takes deep breath] Here is a high-level explanation of how we're applying it taken from ECE 271A at UCSD. Professor Henrik Christensen, who is basically a genius in pattern detection machine learning
sage professor of lore and wisdom lecture slides : https://www.cc.gatech.edu/~hic/CS7616/pdf/lecture2.pdf
sample code taken from: https://github.com/Neo3333/ECE-271A/blob/master/hw2/ECE%20271A%20HW2.ipynb
input image representation: https://github.com/Neo3333/ECE-271A/blob/master/hw1/hw1intro.pdf
More Tenants [the return!] First we start off by defining some events in probability and Bayes theory. This will be hard to follow if you have not taken a Random Processes or Probability class:

- State of pixel ω (random variable): – e.g., ω1 for Foreground (pixels that are periodic in pattern), ω2 for background of display area
- Probabilities. $P(\omega_1)$ and $P(\omega_2)$ (prior probability):
  – e.g., prior knowledge of how likely image is periodic-defect pixel or background pixel. This prior knowledge is given by our labelled ground truth data and as shown in pages 9-11, the truthfulness of your label matters—the closer to true values you are, the better your outputs will be [refer to code]
- Probability density function. p(x) (evidence):
  – e.g., how frequently we will measure a pattern with periodic-pattern feature value
- Conditional probability density. We want $p(x/\omega_j)$ (likelihood):
  – e.g., how frequently we will measure a pattern with feature value x given that the pattern belongs to class ω_j
- Conditional probability $P(\omega_j/x)$ *(posterior)* :
  – e.g., the probability that the pixel belongs to class ω_j given measurement *x*.



Similar to Page 6 and page 9, we want probability distributions between foreground and background pixels

FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i. If x represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons,

Bayes decision rule using prior probabilityes: Decide ω1 if P(ω1 = foreground) > P(ω2 = background) ; otherwise decide ω2. In other words, assign a black pixel to the output image location if P(ω1 = foreground) > P(ω2 = background)
Using Bayes' rule, the posterior probability of category ω_j given measurement *x* is given by:

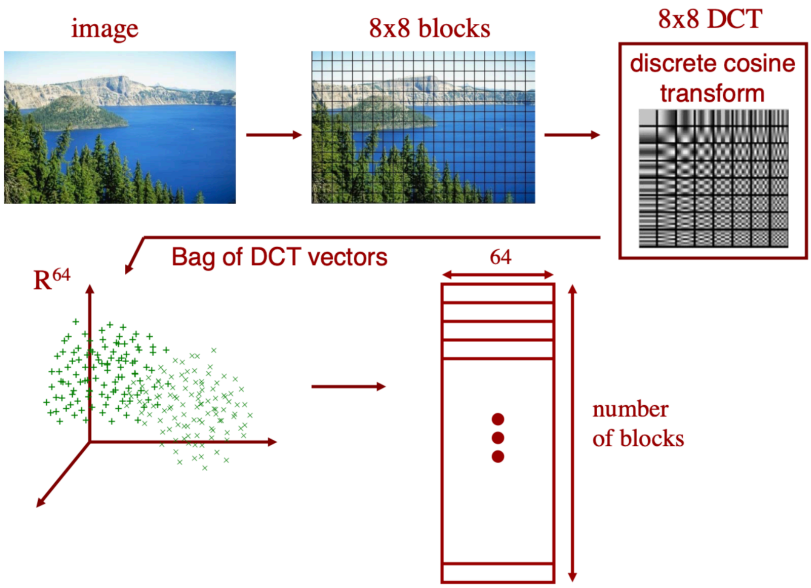
$$P(\omega_j / x) = \frac{p(x/\omega_j)P(\omega_j)}{p(x)} = \frac{likelihood \times prior}{evidence}$$

where $p(x) = \sum_{j=1}^{2} p(x/\omega_j)P(\omega_j)$ (i.e., scale factor – sum of probs = 1)

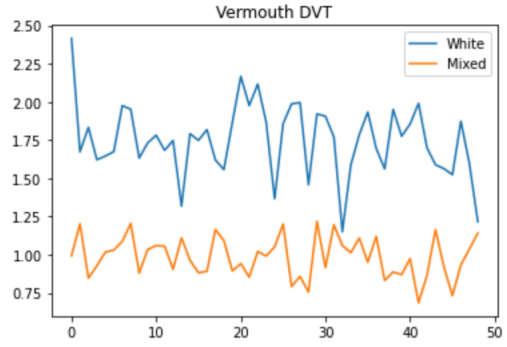
Decide ω_1 if $P(\omega_1/x) > P(\omega_2/x)$; otherwise **decide** ω_2
or
Decide ω_1 if $p(x/\omega_1)P(\omega_1) > p(x/\omega_2)P(\omega_2)$ otherwise **decide** ω_2

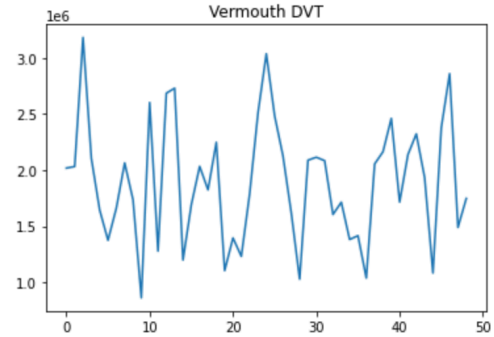
Proof: https://www.statlect.com/fundamentals-of-statistics/multivariate-normal-distribution-maximum-likelihood

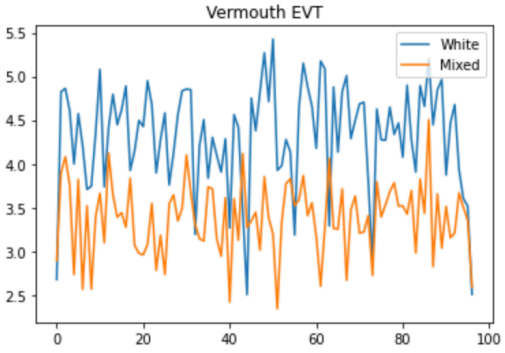
Here is the image representation of how we're patching our input images. Just go through the code at this point and have an informal brief code review:

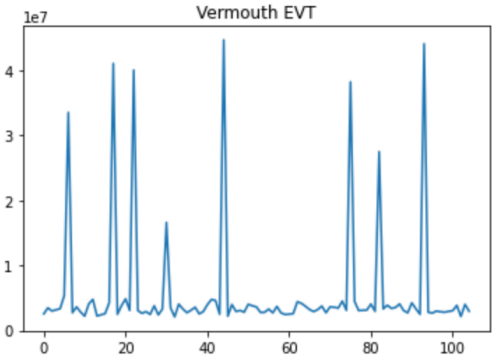


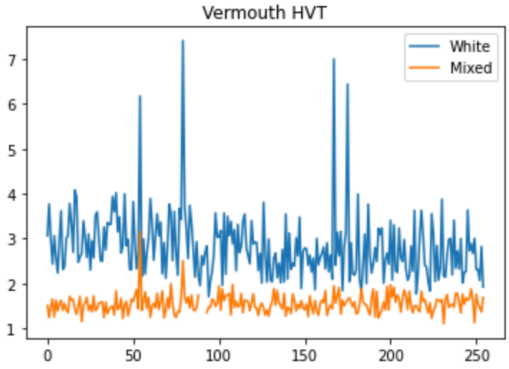
5) Comparison against Dhivya and Matt current Crosshatch metric (left plots) and this new FFT method (right plots)

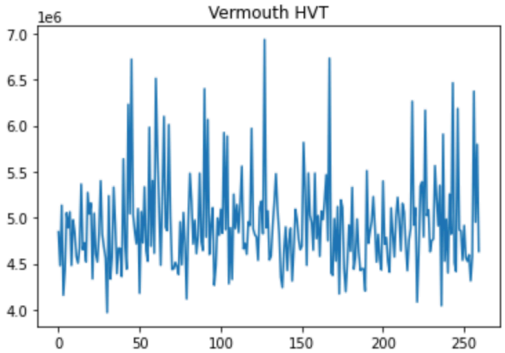






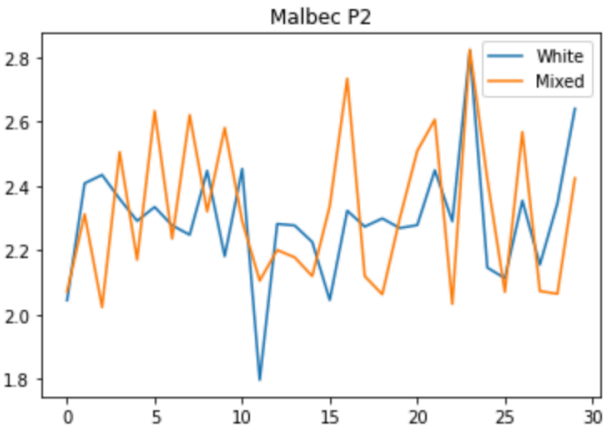


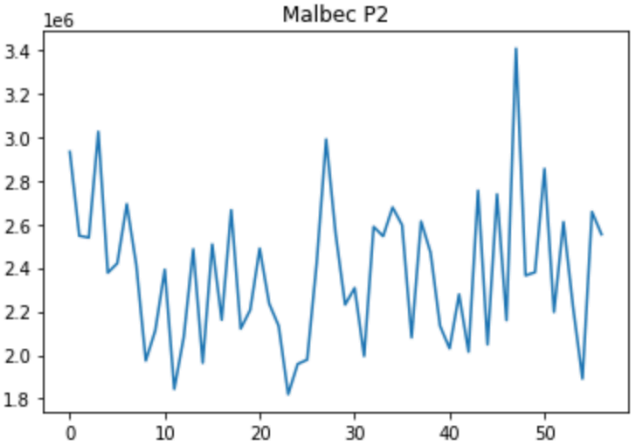


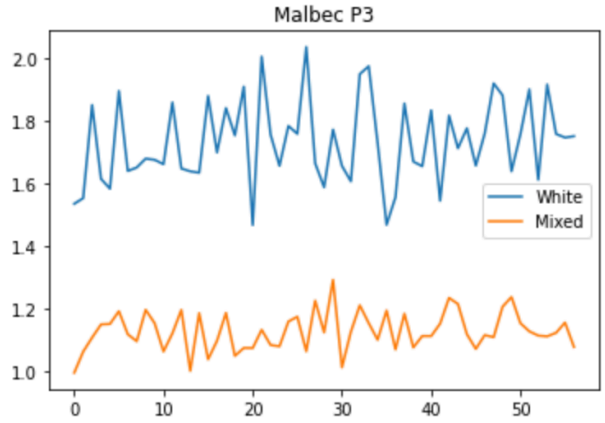


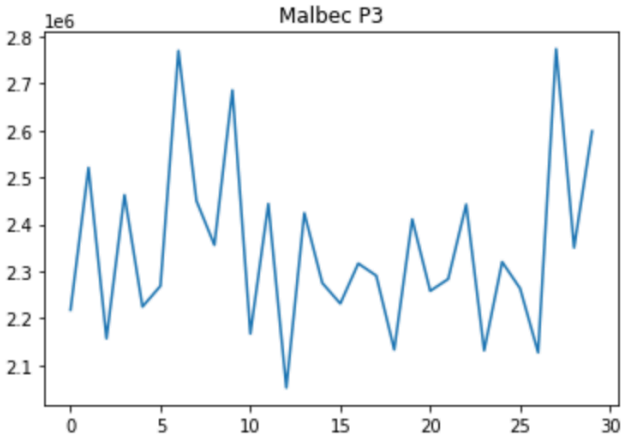
We can normalized the min and max values to be between [0, +1]. However, notice the approximate mean value of the scores improve and decrease from
HVT -> EVT -> DVT .

There are some outliers associated with bad image capture


Malbec P2
White
Mixed
2.8
2.6
2.4
2.2
2.0
1.8
0
5
10
15
20
25
30


Malbec P2
1e6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
0
10
20
30
40
50


Malbec P3
White
Mixed
2.0
1.8
1.6
1.4
1.2
1.0
0
10
20
30
40
50


Malbec P3
1e6
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
0
5
10
15
20
25
30