



# SPECTRAMATCH

## Technical Datasheet

Textile Quality Control System

Professional Color & Pattern Analysis

*Version: 2.2.3*

**Institution:** Pamukkale University  
Department of Electrical and Electronic Engineering, Denizli, Turkey

**Author:** Abdelbary Algamel

**Supervisor:** Dr. Adem Ükte

# Contents

<b>1 System Overview</b>	<b>3</b>
1.1 Definition . . . . .	3
1.2 Objectives . . . . .	3
1.3 Practical Benefits . . . . .	3
1.4 System Workflow . . . . .	4
1.5 Application Forms . . . . .	5
1.5.1 Web Application . . . . .	5
1.5.2 Desktop Application . . . . .	5
1.5.3 Shared Features . . . . .	6
1.6 Image Input Environment . . . . .	6
1.6.1 Dual Image Mode . . . . .	6
1.6.2 Single Image Mode . . . . .	6
1.6.3 Region Selection . . . . .	6
1.7 Point Selection . . . . .	6
1.8 Advanced Settings . . . . .	7
1.8.1 General . . . . .	7
1.8.2 Color . . . . .	7
1.8.3 Pattern . . . . .	7
1.8.4 Illuminant . . . . .	7
1.8.5 Report Sections . . . . .	7
1.9 Processing Time . . . . .	7
1.10 Report Output . . . . .	8
1.11 Continuous Improvement . . . . .	8
<b>2 System Processing Units</b>	<b>9</b>
2.1 Color Unit . . . . .	9
2.1.1 Cover Page . . . . .	9
2.1.2 Input Images . . . . .	10
2.1.3 RGB Values . . . . .	11
2.1.4 Lab* Values . . . . .	11
2.1.5 XYZ Values . . . . .	13
2.1.6 CMYK Values . . . . .	14
2.1.7 Difference Metrics . . . . .	14
2.1.8 Illuminant Analysis . . . . .	15
2.1.9 Delta Heatmap . . . . .	16
2.1.10 Spectral Proxy . . . . .	17
2.1.11 RGB Histograms . . . . .	18
2.1.12 Color Similarity Index (CSI) . . . . .	18
2.1.13 CSI2000 . . . . .	19
2.1.14 Recommendations . . . . .	20
2.2 Pattern Unit . . . . .	21
2.2.1 Cover Page and Input Images . . . . .	21

2.2.2	Structural SSIM . . . . .	22
2.2.3	Gradient Similarity . . . . .	23
2.2.4	Phase Correlation . . . . .	24
2.2.5	Structural Difference Analysis . . . . .	25
2.2.6	Gradient Boundary Detection . . . . .	27
2.2.7	Phase Boundary Detection . . . . .	28
2.2.8	Fourier Domain Analysis . . . . .	29
2.2.9	GLCM Texture Analysis . . . . .	31
2.2.10	Verdict . . . . .	32
2.2.11	Recommendations and Conclusion . . . . .	33
2.3	Single Image Unit . . . . .	34
2.3.1	Input . . . . .	34
2.3.2	Measurement Data . . . . .	35
2.3.3	Spectral Analysis, Fourier Analysis, and Recommendations . . . . .	36
<b>3</b>	<b>Frequently Asked Questions</b>	<b>37</b>
<b>4</b>	<b>Software Implementation</b>	<b>42</b>
4.1	Programming Tools and Technologies . . . . .	42
4.2	Codebase Language Distribution . . . . .	43
4.3	File and Folder Structure . . . . .	43
4.4	Key Functions and Classes . . . . .	44
4.4.1	app.py — Flask Application . . . . .	44
4.4.2	ColorUnitBackend.py — Color Analysis . . . . .	44
4.4.3	PatternUnitBackend.py — Pattern Analysis . . . . .	45
4.4.4	SingleImageUnitBackend.py — Single Image Analysis . . . . .	45
4.4.5	RecommendationsEngine.py — Findings and Recommendations . . . . .	46
4.4.6	ReportUtils.py — Shared PDF Utilities . . . . .	46
4.4.7	ReportTranslations.py — Bilingual Dictionary . . . . .	46
4.4.8	SettingsReceipt.py — Configuration Receipt . . . . .	46
4.4.9	desktop/app_desktop.py — Desktop Launcher . . . . .	47
4.4.10	Frontend JavaScript . . . . .	47
<b>A</b>	<b>Glossary of Abbreviations</b>	<b>48</b>
<b>B</b>	<b>List of Figures</b>	<b>49</b>
<b>C</b>	<b>Known Limitations (v2.2.3)</b>	<b>50</b>

# Chapter 1

## System Overview

### 1.1 Definition

SpectraMatch is a research-grade image quality control system designed for the textile industry. It provides quantitative color and pattern analysis by comparing a **reference** image against a **sample** image, producing detailed metrics, visual diagnostics, and downloadable PDF reports. The system is developed as part of ongoing research at the Department of Electrical and Electronic Engineering, Pamukkale University (PAU), Denizli, Turkey.

### 1.2 Objectives

- Provide objective, quantitative measurement of color difference between textile samples using established colorimetric standards (CIE  $\Delta E_{2000}$ , CIELAB).
- Evaluate structural similarity and pattern fidelity between reference and production samples using image analysis techniques (SSIM, GLCM, FFT, Phase Correlation).
- Support standalone single-image analysis for color distribution, texture uniformity, and defect indicators without requiring a reference.
- Generate professional, multi-page PDF reports with charts, tables, and threshold-aware recommendations in English and Turkish.
- Serve as a practical tool suitable for both laboratory evaluation and production-floor use.

### 1.3 Practical Benefits

- Replaces subjective visual inspection with repeatable, numeric measurements.
- Configurable pass/conditional/fail thresholds allow adaptation to different quality standards.
- Bilingual interface and reports (English and Turkish) enable use by diverse teams.
- Supports multiple illuminants (D65, D50, A, C, F2, TL84) to simulate different viewing conditions.
- Generates comprehensive PDF documentation suitable for archival, audit, and inter-departmental communication.

## 1.4 System Workflow

Figure 1.1 illustrates the complete processing workflow of SpectraMatch.

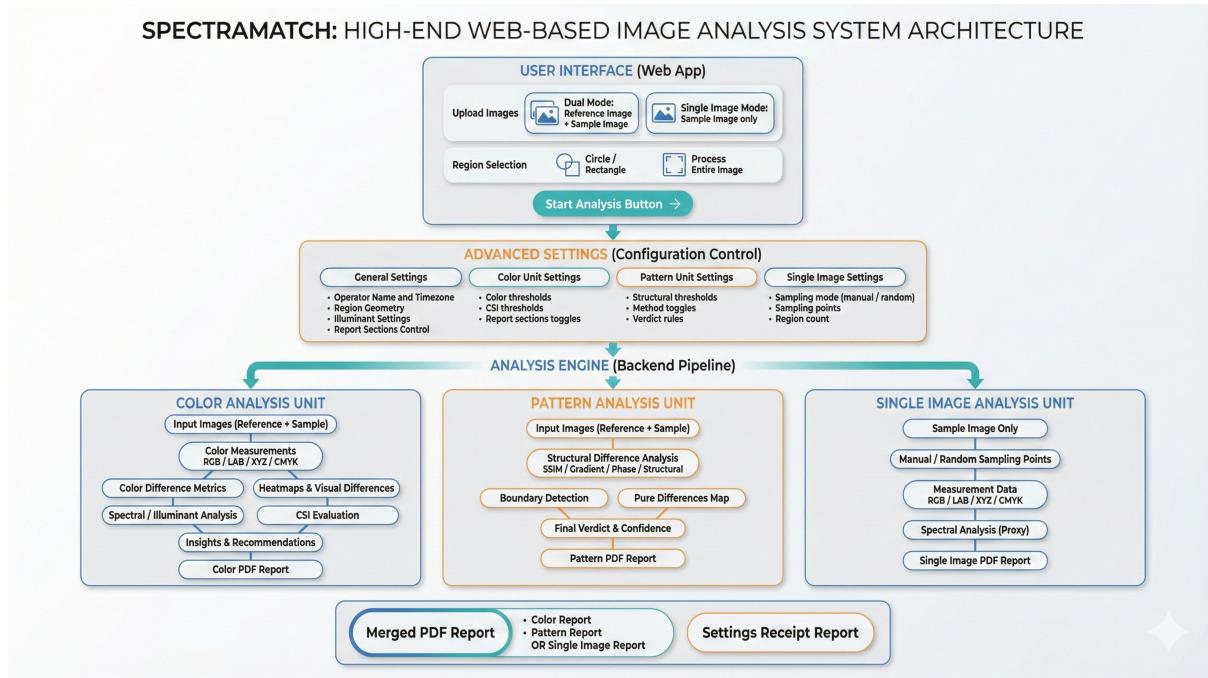


Figure 1.1: SpectraMatch system workflow. The user provides images, selects operating mode, configures advanced settings, and the system processes images through the applicable units. Reports are generated as downloadable PDFs.

The workflow proceeds as follows:

- Image Input:** The user uploads a reference image and a sample image (Dual Image Mode) or a single sample image (Single Image Mode).
- Mode Selection:** The user selects between Dual Image Mode (for comparison) or Single Image Mode (for standalone analysis).
- Advanced Settings:** The user configures analysis parameters through the Settings panel.
- Processing:** The system routes images through the applicable units:
  - Color Unit** — computes color difference metrics ( $\Delta E_{2000}$ , CSI, illuminant analysis).
  - Pattern Unit** — evaluates structural similarity (SSIM, Gradient, Phase, Structural Match, Fourier, GLCM).
  - Single Image Unit** — performs standalone color distribution and texture analysis.
- Report Generation:** The system produces PDF reports:
  - Full Report** — unified report merging a cover page, color pages, and pattern pages.
  - Single Image Report** — standalone analysis results.
  - Settings Report (Receipt)** — a separate PDF documenting all configuration parameters used.

## 1.5 Application Forms

The system is delivered in two forms that share the same backend logic:

### 1.5.1 Web Application

Accessible via a browser at the configured server address. The web interface provides all analysis features and additionally allows the user to download the desktop application as a ready-to-use .exe installer. Figure 1.2 shows the web application interface.

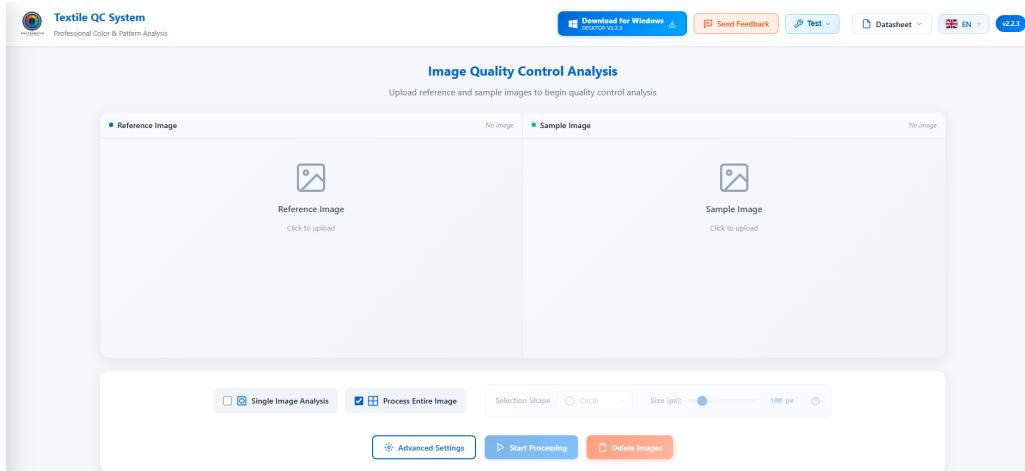


Figure 1.2: SpectraMatch web application interface.

### 1.5.2 Desktop Application

A native Windows application that wraps the same Flask backend inside a desktop window using pywebview. It includes a splash screen during startup, supports dark and light mode, and provides a toolbar with keyboard shortcuts. Figure 1.3 shows the desktop application interface.

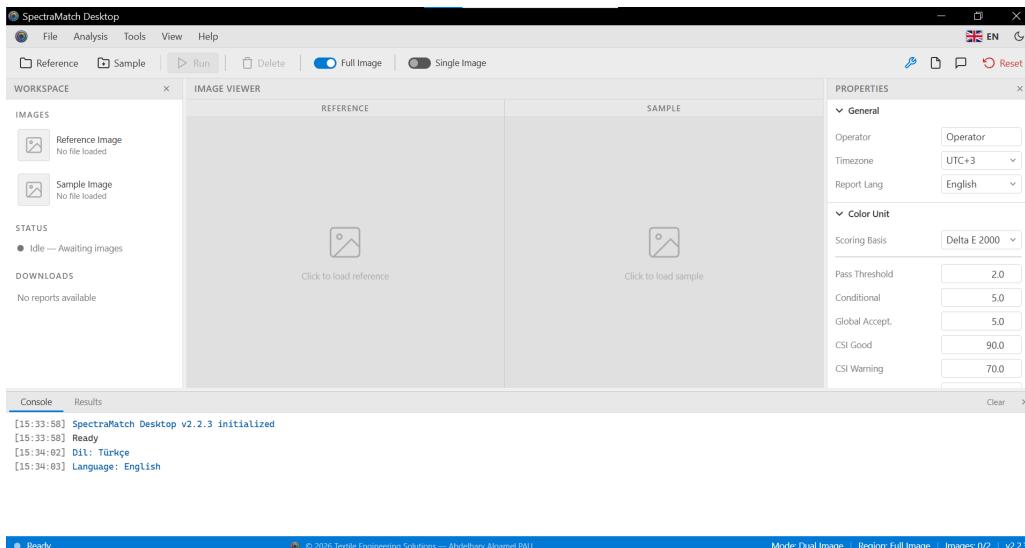


Figure 1.3: SpectraMatch desktop application interface.

### 1.5.3 Shared Features

Both application forms provide the following shared features:

- **Send Feedback:** A button used during the trial phase and for ongoing development to collect user observations and suggestions.
- **Ready-to-Test Images:** A trial-only feature that provides predefined sample image pairs (six built-in test images organized as three pairs) for quick evaluation without requiring user-supplied images.
- **Datasheet:** A button that generates and provides access to this technical document.

## 1.6 Image Input Environment

### 1.6.1 Dual Image Mode

The user uploads two images: a **reference** image (the standard or target) and a **sample** image (the production output to be evaluated). Both images are processed through the Color Unit and Pattern Unit.

### 1.6.2 Single Image Mode

The user uploads a single **sample** image. The system performs standalone analysis of color distribution, texture uniformity, and spectral characteristics without comparison to a reference.

### 1.6.3 Region Selection

Before analysis, the user may define a region of interest on the images using one of those tools:

- **Circle** — circular selection with adjustable diameter.
- **Square** — square selection with adjustable side length.
- **Rectangle** — rectangular selection with independent width and height control.
- **Freehand (Pen)** — draw a custom closed polygon directly on the image.

All tools support click-to-place, drag-to-move, and scroll-to-resize (except Pen). The selected region is visually overlaid on both images simultaneously. If no region is selected, the full image is used.

## 1.7 Point Selection

The Color Unit and Single Image Unit sample colors at discrete points within the selected region. The system supports:

- **Manual placement** — the user clicks directly on the image to place each sampling point.
- **Random generation** — the system automatically generates  $N$  random points uniformly distributed within the selected region using rejection sampling.
- **Hybrid** — the user places some points manually, and the system fills the remainder with random points.

The backend enforces strict validation: every point (manual or random) must lie within the defined region and within the image bounds. Points placed outside the region are rejected.

## 1.8 Advanced Settings

The Advanced Settings panel is organized into five subsections:

### 1.8.1 General

- Selection shape (circle, square, rectangle, pen).
- Report language (different from the application language)
- Operator name (included in reports).
- Timezone offset (UTC, used for report timestamps and ID generation).

### 1.8.2 Color

- Color scoring method ( $\Delta E_{2000}$ , CSI, or CSI2000).
- Number of sampling points ( $N$ ).
- Sampling mode (manual, random, hybrid).
- Pass threshold (Parameter values which a point is classified as PASS).
- Conditional threshold (Parameter values which a point is classified as CONDITIONAL).
- Scores ( Good / Warning ) thresholds.
- Lab\* component thresholds ( $\Delta L^*$ ,  $\Delta a^*$ ,  $\Delta b^*$ ).

### 1.8.3 Pattern

- Global pattern threshold.
- Per-method thresholds (SSIM, Gradient, Phase, Structural Match), each with pass and conditional values.

### 1.8.4 Illuminant

- Primary illuminant selection (D65, D50, A, C, F2, CWF, TL84).
- Test illuminants list (one or more illuminants for multi-illuminant comparison).

### 1.8.5 Report Sections

- Toggles for individual report sections

## 1.9 Processing Time

Processing time depends on image size and the number of enabled report sections. Larger images require more computation for pixel-level operations (heatmaps, SSIM, FFT, structural analysis). Disabling sections such as GLCM or Fourier analysis removes their computation entirely, reducing total processing time.

## 1.10 Report Output

Reports are generated as PDF files in the selected language. Each report is named using an automatically generated analysis ID: **SPEC\_YMMDD\_HHMMSS** where the timestamp uses the configured timezone offset. A language suffix (\_EN or \_TR) is appended. A separate **Settings Report** (configuration receipt) is generated alongside the analysis report, sharing the same base filename with an **\_AYARLAR** suffix, documenting all configuration parameters used. Figure 1.4 shows the cover page of a Settings Report.

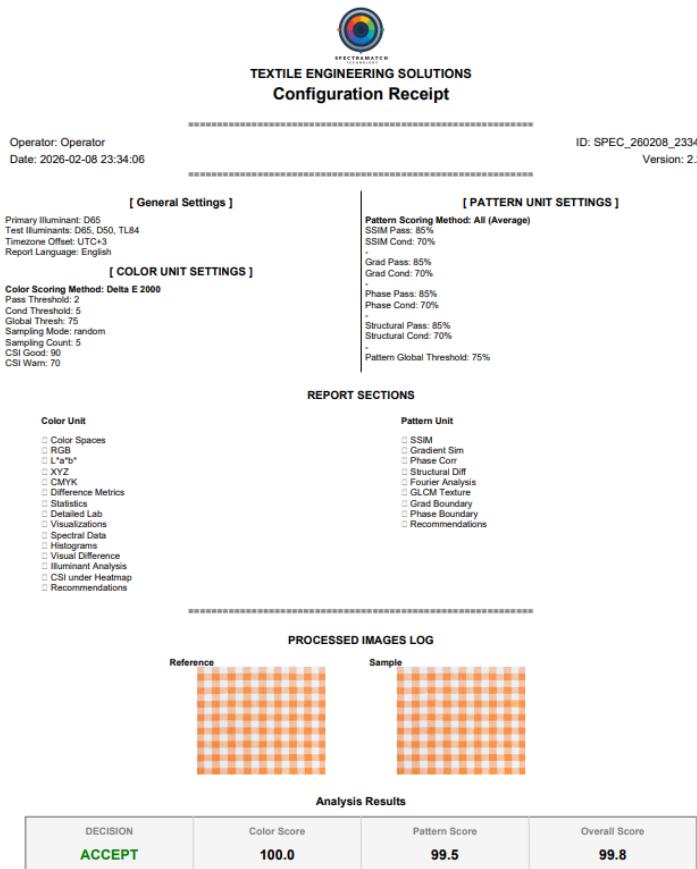


Figure 1.4: Settings Report — Cover page of the configuration receipt PDF. Documents all analysis parameters including thresholds, illuminant selections, enabled sections, and scoring method.

## 1.11 Continuous Improvement

The project is actively refined beyond version 2.2.3. Improvements are driven by real-world usage observations, structured testing, and academic research requirements.

# Chapter 2

## System Processing Units

This chapter describes the backend processing backbone of SpectraMatch. Each processing unit is responsible for a specific category of analysis and produces a dedicated section within the generated PDF report. For every report section, table, and figure produced by the system, this chapter provides: (1) definition and purpose, (2) how it is calculated, (3) how to read the result, and (4) how to evaluate it.

### 2.1 Color Unit

The Color Unit performs quantitative color comparison between a reference image and a sample image. It extracts color values at  $N$  sampling points, converts them through multiple color spaces, computes color difference metrics, and generates threshold-aware verdicts.

#### 2.1.1 Cover Page

##### Definition and Purpose

The cover page is the first page of the Color Report. It presents the report identity, metadata, and an executive summary of the overall color analysis result.

##### How It Is Generated

The cover page includes the company logo, report title, generation timestamp, operator name, primary illuminant, report ID, and software version. An Executive Summary banner displays the overall status (PASS, CONDITIONAL, or FAIL) with color coding (green, orange, red). Below the banner, the primary scoring metric is displayed (depending on the selected scoring method:  $\Delta E_{2000}$ , CSI, or CSI2000), followed by a summary statistics table showing mean, standard deviation, minimum, and maximum values for  $\Delta E_{76}$ ,  $\Delta E_{94}$ , and  $\Delta E_{2000}$ .

##### How to Read the Result

The executive summary banner provides an immediate visual indication of the overall color match quality. The primary score is displayed prominently with its value out of 100. The statistics table provides the distribution characteristics of the color difference across all sampling points.

## How to Evaluate

- **PASS** (green): The sample meets the defined quality thresholds.
- **CONDITIONAL** (orange): The sample is near the quality limits; additional review is recommended.
- **FAIL** (red): The sample does not meet the required quality standards.

### 2.1.2 Input Images

#### Input Images

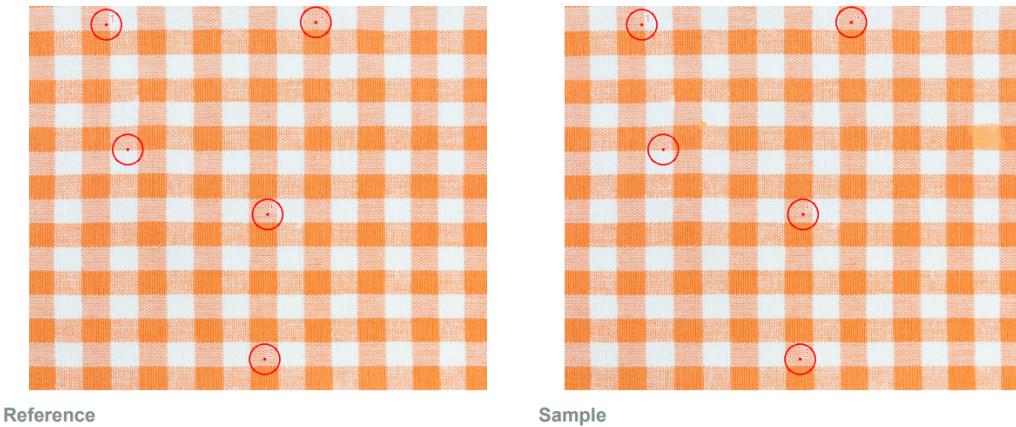


Figure 2.1: Color Unit — Input Images. Reference (left) and sample (right) with sampling point overlays. Green circles indicate manually placed points; orange circles indicate randomly generated points.

## Definition and Purpose

This section displays side-by-side the reference and sample images with the sampling point locations overlaid as numbered circles.

## How It Is Generated

The system draws circles at each sampling point location on both images. Points are color-coded: green for manually placed points and orange for randomly generated points. Each point is numbered sequentially.

## How to Read the Result

The overlay allows visual verification that sampling points are positioned on the intended areas of the fabric. The numbering corresponds to the rows in subsequent measurement tables.

## How to Evaluate

Verify that all sampling points fall within the region of interest and on representative areas of the fabric. Points on edges, folds, or artifacts may produce unreliable measurements.

### 2.1.3 RGB Values

#### RGB Color Values

Region	Position	Ref R	Test R	Ref G	Test G	Ref B	Test B
1	(158, 38)	239	239	198	198	174	174
2	(202, 294)	238	238	210	210	193	193
3	(483, 724)	242	242	186	186	150	150
4	(588, 33)	242	242	189	189	157	157
5	(489, 427)	240	240	191	191	161	161

Figure 2.2: Color Unit — RGB Values table. Per-point Red, Green, and Blue channel values (0–255) for reference and sample.

#### Definition and Purpose

The RGB Values table presents the raw Red, Green, and Blue channel intensities (0–255 scale) extracted at each sampling point for both reference and sample images.

#### How It Is Calculated

At each sampling point, a circular patch of pixels is extracted. Pixels with valid alpha (non-transparent) are averaged to produce the mean RGB value. The patch radius is computed as  $r = \max(12, \lfloor 0.04 \times \min(h, w) \rfloor)$  where  $h$  and  $w$  are the image dimensions.

#### How to Read the Result

Each row corresponds to one sampling point. The table shows Reference R, Sample R, Reference G, Sample G, Reference B, Sample B. Columns are color-coded (red background for R, green for G, blue for B).

#### How to Evaluate

Large differences between reference and sample values in any channel indicate a color shift. RGB values are device-dependent and should be interpreted alongside the Lab\* values for perceptual accuracy.

### 2.1.4 Lab\* Values

Figure below shows the table of mean CIELAB ( $L^*$ ,  $a^*$ ,  $b^*$ ) values for the reference and sample images.

#### LAB\* Color Space Values

Region	Ref L*	Test L*	Ref a*	Test a*	Ref b*	Test b*
1	82.75	82.75	11.00	11.00	17.00	17.00
2	86.27	86.27	7.00	7.00	12.00	12.00
3	80.00	80.00	15.00	15.00	26.00	26.00
4	80.78	80.78	15.00	15.00	23.00	23.00
5	81.18	81.18	13.00	13.00	22.00	22.00

Figure 2.3: Mean CIELAB ( $L^*$ ,  $a^*$ ,  $b^*$ ) values and component-wise differences.

## Lab\* Visualizations

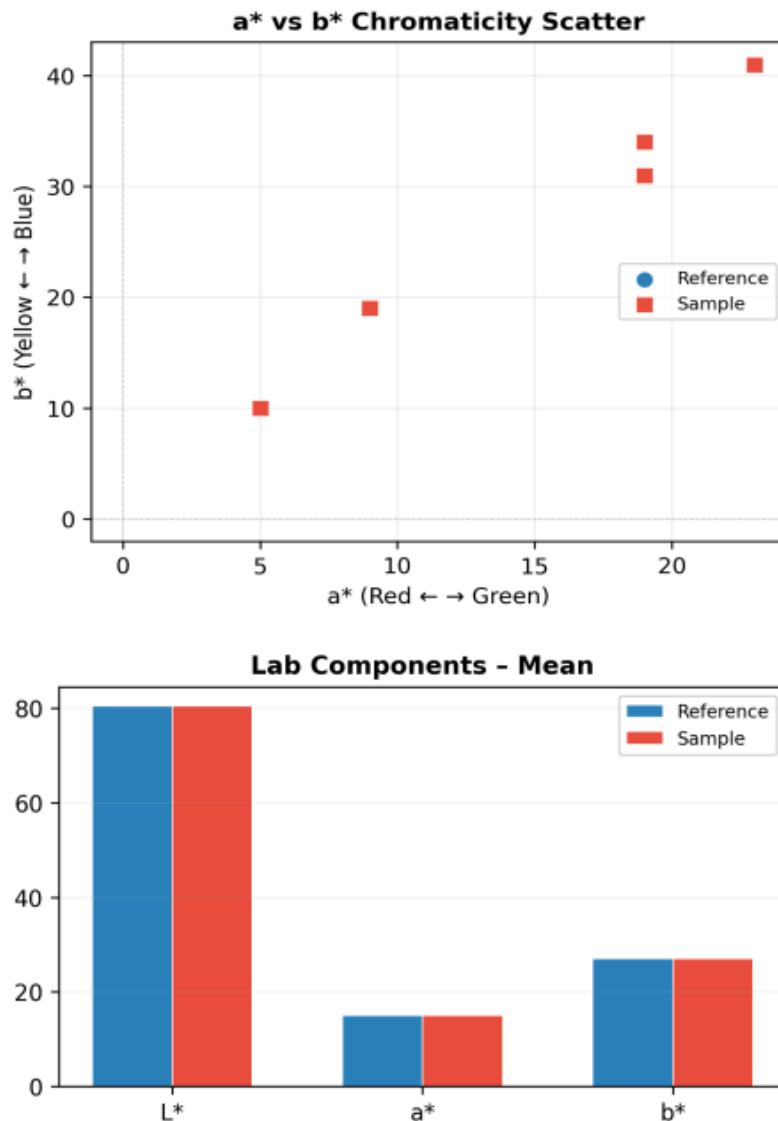


Figure 2.4: Color Unit — Lab\* Visualization. a\*b\* chromaticity scatter plot and grouped bar chart of mean L\*, a\*, b\* for reference and sample.

### Definition and Purpose

The Lab\* section provides a perceptually uniform color space analysis. CIELAB separates lightness ( $L^*$ ) from chromaticity ( $a^*$ : red-green axis;  $b^*$ : yellow-blue axis), making it suitable for quantifying perceived color differences.

### How It Is Calculated

RGB values are converted through the pipeline: sRGB → linear RGB (inverse gamma) → XYZ (using the sRGB-to-XYZ matrix) → chromatic adaptation (Bradford transform to the selected illuminant) → CIELAB (using the illuminant white point). The mean values across all sampling points are computed for each component. Differences  $\Delta L^*$ ,  $\Delta a^*$ ,  $\Delta b^*$  are calculated, and an overall magnitude  $\sqrt{\Delta L^{*2} + \Delta a^{*2} + \Delta b^{*2}}$  is derived.

A quality assessment table compares each component difference against user-defined thresholds and assigns PASS or FAIL per component. Interpretations are provided: negligible ( $< 0.25 \times$  threshold), noticeable ( $<$  threshold), significant ( $< 2 \times$  threshold), or critical ( $\geq 2 \times$  threshold).

An  $a^*$  vs.  $b^*$  chromaticity scatter plot and a grouped bar chart of mean  $L^*$ ,  $a^*$ ,  $b^*$  are generated.

## How to Read the Result

- $\Delta L^* > 0$ : sample is lighter;  $\Delta L^* < 0$ : sample is darker.
- $\Delta a^* > 0$ : sample is more red;  $\Delta a^* < 0$ : sample is more green.
- $\Delta b^* > 0$ : sample is more yellow;  $\Delta b^* < 0$ : sample is more blue.

## How to Evaluate

Each component is compared against its threshold. If all components pass, the report states “within tight tolerance.” If any fail, the failing parameters and their exceedance values are listed.

### 2.1.5 XYZ Values

#### XYZ Tristimulus Values

Region	Ref X	Ref Y	Ref Z	Test X	Test Y	Test Z
1	63.86	62.30	48.80	63.86	62.30	48.80
2	68.32	68.43	60.17	68.32	68.43	60.17
3	60.06	56.40	36.71	60.06	56.40	36.71
4	61.18	58.05	39.88	61.18	58.05	39.88
5	61.05	58.43	41.87	61.05	58.43	41.87

Figure 2.5: Color Unit — XYZ Values table. Per-point CIE XYZ tristimulus values for reference and sample.

## Definition and Purpose

The XYZ table presents the CIE 1931 XYZ tristimulus values at each sampling point. XYZ is the device-independent color space from which CIELAB is derived.

## How It Is Calculated

sRGB values are linearized (inverse sRGB companding), then multiplied by the sRGB-to-XYZ transformation matrix and scaled by 100:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = 100 \times \begin{pmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{pmatrix} \begin{pmatrix} R_{\text{lin}} \\ G_{\text{lin}} \\ B_{\text{lin}} \end{pmatrix}$$

## How to Read the Result

$Y$  corresponds to luminance.  $X$  and  $Z$  encode chromaticity. Values are presented per sampling point for both reference and sample.

## How to Evaluate

XYZ values are intermediate and primarily useful for specialists verifying the color conversion pipeline. For practical quality decisions, Lab\* and  $\Delta E_{2000}$  values are more directly interpretable.

## 2.1.6 CMYK Values

### CMYK Color Values

Region	Ref C	Ref M	Ref Y	Ref K	Test C	Test M	Test Y	Test K
1	0.0	17.0	27.3	6.0	0.0	17.0	27.3	6.0
2	0.0	11.9	19.1	6.3	0.0	11.9	19.1	6.3
3	0.0	23.4	38.1	4.7	0.0	23.4	38.1	4.7
4	0.0	21.8	35.2	4.9	0.0	21.8	35.2	4.9
5	0.0	20.4	32.8	5.9	0.0	20.4	32.8	5.9

Figure 2.6: Color Unit — CMYK Values table. Per-point Cyan, Magenta, Yellow, Key (Black) percentages for reference and sample.

### Definition and Purpose

The CMYK table provides color values in the subtractive color model used in printing and dyeing processes. Values are expressed as percentages (0–100%).

### How It Is Calculated

From the normalized RGB values ( $r, g, b \in [0, 1]$ ):

$$K = 1 - \max(r, g, b), \quad C = \frac{1 - r - K}{1 - K}, \quad M = \frac{1 - g - K}{1 - K}, \quad Y = \frac{1 - b - K}{1 - K}$$

### How to Read the Result

Higher values indicate more ink/dye of that component.  $K = 100\%$  means pure black.

### How to Evaluate

CMYK is informational for users working in dye formulation. Differences between reference and sample CMYK values indicate which dye components need adjustment.

## 2.1.7 Difference Metrics

### Color Difference Metrics

Region	$\Delta E_{76}$	$\Delta E_{94}$	$\Delta E_{2000}$	Status
1	0.00	0.00	0.00	PASS
2	0.00	0.00	0.00	PASS
3	0.00	0.00	0.00	PASS
4	0.00	0.00	0.00	PASS
5	0.00	0.00	0.00	PASS

Figure 2.7: Color Unit — Difference Metrics. Per-point  $\Delta E_{76}$ ,  $\Delta E_{94}$ , and  $\Delta E_{2000}$  values with per-point PASS/CONDITIONAL/FAIL status.

### Definition and Purpose

This section presents the core color difference metrics at each sampling point. Three  $\Delta E$  formulas are computed:  $\Delta E_{76}$  (Euclidean distance in CIELAB),  $\Delta E_{94}$  (improved weighting), and  $\Delta E_{2000}$  (the current CIE standard, most perceptually uniform).

## How It Is Calculated

$\Delta E_{76}$  — Simple Euclidean distance in CIELAB:

$$\Delta E_{76} = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$$

$\Delta E_{94}$  — Adds weighting functions for lightness, chroma, and hue:

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H^*}{k_H S_H}\right)^2}$$

where  $S_L = 1$ ,  $S_C = 1 + 0.045 C_1^*$ ,  $S_H = 1 + 0.015 C_1^*$ .

$\Delta E_{2000}$  — The full CIEDE2000 formula incorporating lightness, chroma, and hue weighting with rotation and interaction terms. The implementation follows the complete specification including the  $G$  chroma adjustment factor,  $T$  hue-dependent weighting,  $R_T$  rotation term, and parametric weighting factors  $k_L$ ,  $k_C$ ,  $k_H$ .

A summary statistics table follows, showing mean, standard deviation, minimum, and maximum for each metric across all sampling points. The  $\Delta E_{2000}$  row carries the overall status.

## How to Read the Result

Each row corresponds to one sampling point. The Status column shows PASS (green), CONDITIONAL (orange), or FAIL (red) based on user-defined thresholds applied to  $\Delta E_{2000}$ .

## How to Evaluate

- $\Delta E_{2000} < 1.0$ : imperceptible difference (instrument noise level).
- $1.0 \leq \Delta E_{2000} < 2.0$ : slight difference, visible only under controlled lighting.
- $2.0 \leq \Delta E_{2000} < 3.5$ : noticeable to trained observers.
- $3.5 \leq \Delta E_{2000} < 5.0$ : clearly visible.
- $\Delta E_{2000} \geq 5.0$ : significant mismatch.

The decision metric is  $\Delta E_{2000}$ . Points with  $\Delta E_{2000} <$  pass threshold are PASS; points  $\leq$  conditional threshold are CONDITIONAL; above that is FAIL.

### 2.1.8 Illuminant Analysis

#### Illuminant Analysis

Illuminant Analysis - Color Similarity Index

Illuminant	Mean $\Delta E_{2000}$	CSI	Status
D65	0.00	99.89	PASS
D50	0.00	100.00	PASS
TL84	0.00	100.00	PASS

Figure 2.8: Color Unit — Illuminant Analysis. Mean  $\Delta E_{2000}$  and CSI computed under each configured illuminant, with per-illuminant status.

## Definition and Purpose

Illuminant Analysis evaluates how the color difference between reference and sample changes under different standard light sources. This detects metamerism — the phenomenon where two samples match under one illuminant but differ under another.

## How It Is Calculated

For the primary illuminant, the canonical  $\Delta E_{2000}$  and CSI values from the main analysis are used. For each additional test illuminant, the XYZ values (originally computed under D65) are chromatically adapted to the target illuminant using the Bradford transform:

$$\mathbf{XYZ}_{\text{target}} = \mathbf{M}_A^{-1} \cdot \text{diag}\left(\frac{\rho_d}{\rho_s}, \frac{\gamma_d}{\gamma_s}, \frac{\beta_d}{\beta_s}\right) \cdot \mathbf{M}_A \cdot \mathbf{XYZ}_{D65}$$

where  $\mathbf{M}_A$  is the Bradford matrix. The adapted XYZ values are converted to CIELAB under the target illuminant's white point, and  $\Delta E_{2000}$  is recomputed. The CSI for each illuminant is derived as  $\text{CSI} = \max(0, \min(100, 100 - \overline{\Delta E} \times 10))$ . The per-illuminant status is determined using fixed thresholds:  $\text{CSI} \geq 90 \rightarrow \text{PASS}$ ,  $\text{CSI} \geq 70 \rightarrow \text{CONDITIONAL}$ , otherwise FAIL.

## How to Read the Result

Each row shows one illuminant with its mean  $\Delta E_{2000}$ , CSI score, and status.

## How to Evaluate

If the status differs across illuminants (e.g., PASS under D65 but FAIL under TL84), metamerism is present. This is critical for textiles viewed under different lighting in retail, warehouse, or outdoor environments.

### 2.1.9 Delta Heatmap

#### Visual Difference Analysis



Figure 2.9: Color Unit —  $\Delta E$  Heatmap. Pixel-level color difference map using the Inferno colormap.

## Definition and Purpose

The  $\Delta E$  Heatmap provides a full-image, pixel-level visualization of color difference between reference and sample.

## How It Is Calculated

Both images are converted to CIELAB. The per-pixel Euclidean distance in Lab space is computed:

$$\Delta E(x, y) = \sqrt{(L_r - L_s)^2 + (a_r - a_s)^2 + (b_r - b_s)^2}$$

The result is rendered as a heatmap using the “inferno” colormap, with a color bar indicating the scale. The maximum display value is clamped to the 99th percentile (minimum 5.0) to avoid outlier distortion.

## How to Read the Result

- **Dark/black regions:** low or zero color difference (good match).
- **Bright/yellow regions:** high color difference (poor match).

## How to Evaluate

Uniform dark appearance indicates consistent color match. Bright spots or bands reveal localized color deviations, which may indicate uneven dyeing, stains, or fabric defects.

### 2.1.10 Spectral Proxy

#### Spectral Analysis (Proxy)

The chart approximates spectral behavior from RGB averages.

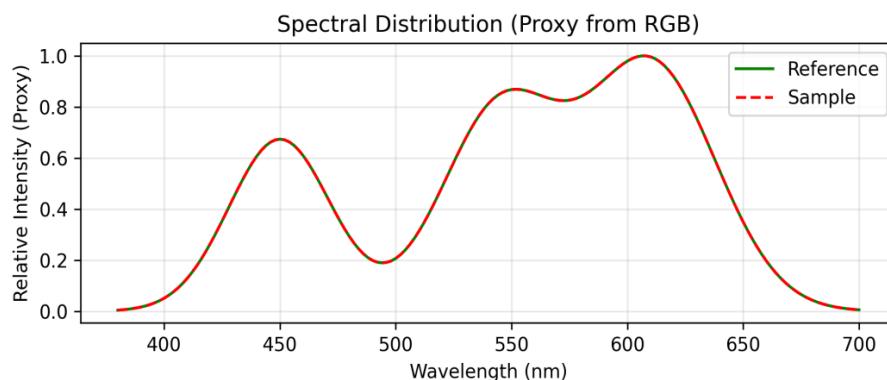


Figure 2.10: Color Unit — Spectral Distribution Proxy. Approximated spectral reflectance curves for reference (green) and sample (red dashed).

## Definition and Purpose

The Spectral Proxy chart provides an approximate visualization of the spectral reflectance distribution of reference and sample, synthesized from their mean RGB values.

## How It Is Calculated

Three Gaussian basis functions centered at 450 nm (Blue,  $\sigma = 22$ ), 545 nm (Green,  $\sigma = 25$ ), and 610 nm (Red,  $\sigma = 28$ ) are weighted by the mean RGB values (normalized 0–1) of each image. The resulting curves are normalized to peak at 1.0. The wavelength range spans 380–700 nm.

## How to Read the Result

The green curve represents the reference and the red dashed curve represents the sample. Overlap indicates spectral similarity.

## How to Evaluate

This is an approximation (proxy), not a measured spectral reflectance. It provides a qualitative visual comparison. For precise spectral data, a spectrophotometer is required.

## 2.1.11 RGB Histograms

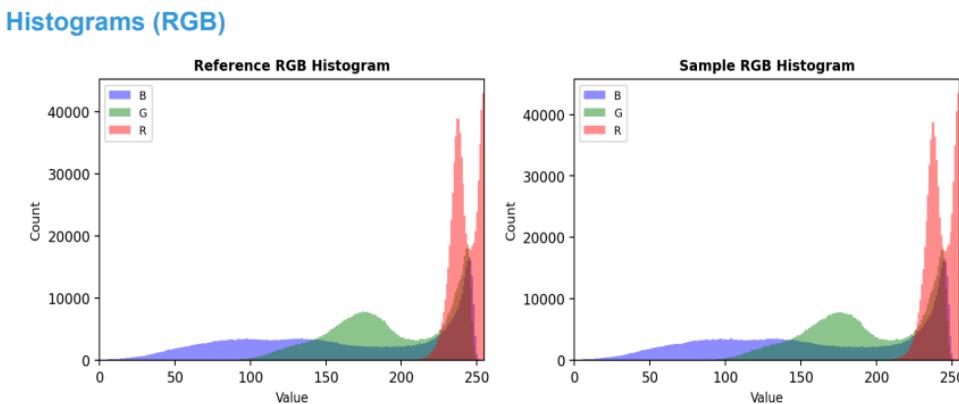


Figure 2.11: Color Unit — RGB Histograms. Side-by-side Red, Green, and Blue channel histograms for reference and sample images.

### Definition and Purpose

Side-by-side RGB histograms show the distribution of pixel intensities in each color channel for reference and sample images.

### How It Is Calculated

For each channel (B, G, R), a 256-bin histogram is computed from the image pixels using OpenCV. The histograms are plotted as overlaid semi-transparent bar charts for each image.

### How to Read the Result

Each histogram shows pixel count (y-axis) vs. intensity value 0–255 (x-axis). Similar histogram shapes between reference and sample indicate consistent color distribution.

### How to Evaluate

Shifts in histogram peaks indicate brightness or color changes. Broadening indicates increased variation. Histograms are informational and complement the numeric  $\Delta E$  metrics.

## 2.1.12 Color Similarity Index (CSI)

# Color Similarity Index (CSI): 99.89 / 100

Figure 2.12: Color Unit — CSI Score display on the cover page, showing the overall color similarity percentage.

### Definition and Purpose

The Color Similarity Index (CSI) provides a single 0–100 score representing overall color similarity between the reference and sample images at the full-image level.

## How It Is Calculated

Both images are downsampled to 25% resolution. Each is converted to OpenCV's Lab representation (L: 0–255, a/b: 0–255 centered at 128). The per-pixel Euclidean distance in this scaled Lab space is computed across all pixels, and the mean is taken:

$$\text{CSI} = \max\left(0, \min\left(100, 100 \times (1 - \overline{\Delta E}_{\text{pixel}}/100)\right)\right)$$

## How to Read the Result

CSI is displayed as a percentage. Higher values indicate better color match.

## How to Evaluate

- $\text{CSI} \geq$  Good threshold (default 90): PASS.
- $\text{CSI} \geq$  Warning threshold (default 70): CONDITIONAL.
- $\text{CSI} <$  Warning threshold: FAIL.

CSI evaluates the *entire image* pixel-by-pixel, whereas  $\Delta E_{2000}$  evaluates only at the  $N$  sampling points.

## 2.1.13 CSI2000

### Definition and Purpose

CSI2000 is a hybrid scoring metric that combines the full-image CSI with the sampling-point-based  $\Delta E_{2000}$  score into a single value.

## How It Is Calculated

A  $\Delta E$  Score is derived as  $\text{Score}_{\Delta E} = \max(0, \min(100, 100 - \overline{\Delta E}_{2000} \times 10))$ . Then:

$$\text{CSI2000} = \frac{\text{CSI} + \text{Score}_{\Delta E}}{2}$$

## How to Read the Result

CSI2000 is displayed as a value out of 100. Higher is better.

## How to Evaluate

CSI2000 balances global color similarity (CSI) with point-specific color accuracy ( $\Delta E_{2000}$  Score). It is evaluated against the same good/warning thresholds as CSI.

## 2.1.14 Recommendations

### Key Findings & Recommendations

Mean $\Delta E_{2000}$	0.00
Imperceptible difference — within instrument noise.	
> No action required. Process is optimal.	
Consistency (Std Dev)	0.00
Excellent uniformity — highly consistent across all regions.	
> Process is stable. Maintain current parameters.	
CSI Score	99.89
Excellent color match — meets or exceeds quality standard.	
> Approved. Proceed with production.	

### Conclusion & Decision

PASS	Sample meets all quality standards within the defined tolerances. Approved for production.
------	--

Figure 2.13: Color Unit — Key Findings & Recommendations. Card-based layout showing metric evaluations and actionable recommendations with a conclusion block.

### Definition and Purpose

The Recommendations section provides threshold-aware, contextual findings and actionable guidance based on the analysis results. It is generated by the Recommendations Engine.

### How It Is Generated

Three metrics are evaluated:

1. **Mean  $\Delta E_{2000}$ :** evaluated relative to the pass and conditional thresholds. Levels: imperceptible, slight, noticeable, significant, severe.
2. **Consistency (Std Dev of  $\Delta E$ ):** evaluated relative to the pass threshold. Levels: excellent ( $< 0.25 \times$  pass), good ( $< 0.5 \times$  pass), moderate ( $<$  pass), high ( $\geq$  pass).
3. **CSI Score:** evaluated relative to the good and warning thresholds.

Each metric produces a finding card with: metric name, value, evaluation text, and a recommendation. A conclusion block summarizes the overall decision with a status color.

### How to Read the Result

Each card is color-coded by severity: green (excellent/good), orange (marginal), red (poor). The conclusion provides the final decision with explanatory text.

### How to Evaluate

Recommendations are calibrated to the user-defined thresholds. They provide textile-specific guidance such as “Review dye concentration, temperature, and timing parameters” or “Reject batch and initiate corrective action procedure.”

## 2.2 Pattern Unit

The Pattern Unit evaluates structural similarity and pattern fidelity between reference and sample images using multiple complementary analysis methods. It produces a composite score from four primary methods, each weighted equally at 25%.

### 2.2.1 Cover Page and Input Images

#### Input Images

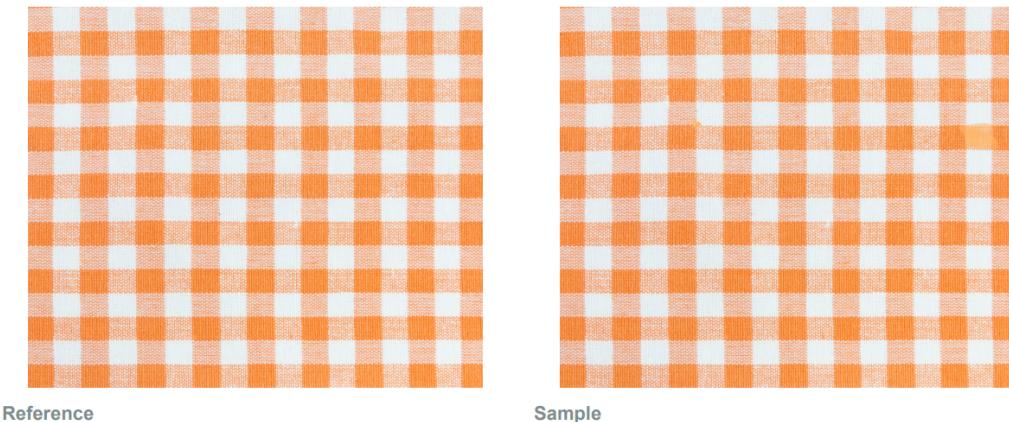


Figure 2.14: Pattern Unit — Input Images. Reference and sample images displayed side by side.

#### Definition and Purpose

The Pattern Unit cover page displays the executive summary with the composite pattern score and per-method scores in a summary table. Input images are shown side by side for visual reference.

#### How It Is Generated

The cover includes the same metadata structure as the Color Unit (logo, title, operator, report ID, version). The executive summary table lists each method (Structural SSIM, Gradient Similarity, Phase Correlation, Structural Match) with its score, weight (25% each), and individual status. The composite score is the weighted sum.

#### How to Read the Result

The composite score (0–100%) provides the overall pattern match quality. Individual method scores identify which aspect of pattern similarity is strongest or weakest.

#### How to Evaluate

- Composite  $\geq$  global threshold: PASS.
- Composite  $\geq$  (global threshold – 15): CONDITIONAL.
- Below that: FAIL.

Note : The user can adjust the threshold values

## 2.2.2 Structural SSIM

### Structural SSIM

**Similarity Score:** 99.93%

PASS

### Difference Visualization



Figure 2.15: Pattern Unit — SSIM difference map. JET colormap where blue indicates high similarity.

#### Definition and Purpose

Structural Similarity Index Measure (SSIM) evaluates perceived structural similarity between two images, considering luminance, contrast, and structure.

#### How It Is Calculated

Both images are preprocessed: composited over black (if alpha channel present), converted to grayscale, and bilateral-filtered (kernel 9,  $\sigma_{color} = 75$ ,  $\sigma_{space} = 75$ ). SSIM is computed using scikit-image with a sliding window, producing a per-pixel similarity map and a scalar score. The score is scaled to 0–100%. The difference map is computed as  $255 - \text{SSIM\_map} \times 255$  and rendered with the JET colormap.

#### How to Read the Result

In the JET-colored difference map: **blue** regions indicate high similarity; **red/yellow** regions indicate low similarity (structural differences).

#### How to Evaluate

Higher SSIM scores indicate better structural preservation. The score is compared against the per-method pass and conditional thresholds defined in settings.

### 2.2.3 Gradient Similarity

## Gradient Similarity

**Similarity Score:** 99.61%

PASS

### Difference Visualization



Figure 2.16: Pattern Unit — Gradient Similarity difference map. HOT colormap where dark indicates similarity and bright indicates edge/textured differences.

#### Definition and Purpose

Gradient Similarity compares the edge and texture structure of reference and sample by analyzing gradient magnitude fields.

#### How It Is Calculated

Sobel operators (horizontal and vertical, kernel size 3) are applied to compute gradient magnitude for each image:  $G = \sqrt{G_x^2 + G_y^2}$ . The gradient magnitudes are normalized to  $[0, 1]$ . SSIM is then applied to the normalized gradient magnitude images to produce a similarity score and difference map. The difference map is rendered with the HOT colormap.

#### How to Read the Result

**Dark** regions indicate matching edge/textured structure; **bright/white** regions indicate gradient differences.

#### How to Evaluate

High scores indicate consistent edge and texture patterns. Low scores may indicate pattern distortion, misalignment, or texture degradation.

## 2.2.4 Phase Correlation

### Phase Correlation

**Similarity Score:** 98.56%

PASS

### Difference Visualization



Figure 2.17: Pattern Unit — Phase Correlation difference map. INFERNO colormap showing translational alignment differences.

#### Definition and Purpose

Phase Correlation measures translational alignment between two images using the Fourier phase relationship.

#### How It Is Calculated

Both grayscale images are converted to float32. OpenCV's `phaseCorrelate` function computes the shift vector and a response value (confidence). The score is  $\text{response} \times 100$ . The absolute pixel difference is computed and rendered with the INFERNO colormap.

#### How to Read the Result

**Dark** regions indicate aligned areas; **bright/yellow** regions indicate misaligned areas.

#### How to Evaluate

High scores indicate good translational alignment. Low scores suggest the sample is shifted, stretched, or distorted relative to the reference.

## 2.2.5 Structural Difference Analysis

### Difference Visualization (Multi-Method)

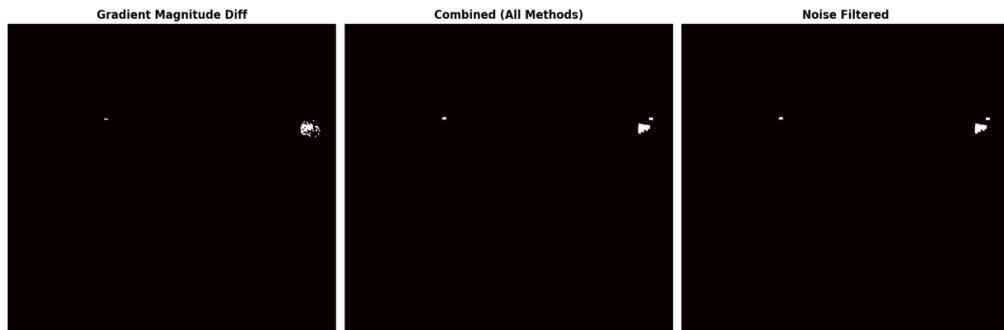


Figure 2.18: Pattern Unit — Structural Difference Analysis. Multi-method difference visualization showing gradient magnitude difference, combined (all methods), and noise-filtered views.

#### Definition and Purpose

Structural Difference Analysis fuses multiple difference detection methods to identify pixel-level structural changes between reference and sample.

#### How It Is Calculated

Both images are converted to grayscale and enhanced with CLAHE (Contrast Limited Adaptive Histogram Equalization). Five methods are computed and fused:

1. **Simple Difference** (25% weight): absolute pixel difference, thresholded at 30, morphologically cleaned.
2. **Edge-based** (20% weight): Canny edge detection on both images, absolute difference of edge maps, dilated.
3. **Gradient** (25% weight): Sobel gradient magnitude difference, thresholded at 40, cleaned.
4. **Frequency** (15% weight): FFT magnitude difference, inverse-transformed to spatial domain, thresholded.
5. **SSIM** (15% weight): inverted SSIM similarity map, thresholded at 200.

The fusion combines these five maps with the listed weights. The result is thresholded, morphologically cleaned, and noise-filtered by removing connected components smaller than 50 pixels.

The similarity score is:  $\text{Score} = \max(0, 100 - \text{change\_percentage})$ .

#### How to Read the Result

The subplot shows three stages: gradient difference, combined (all methods), and noise-filtered. Bright/white regions indicate detected differences; dark/black regions indicate no change.

A **Pure Difference** image is also generated: red pixels overlaid on the grayscale reference show only the detected difference regions (Figure 2.19).

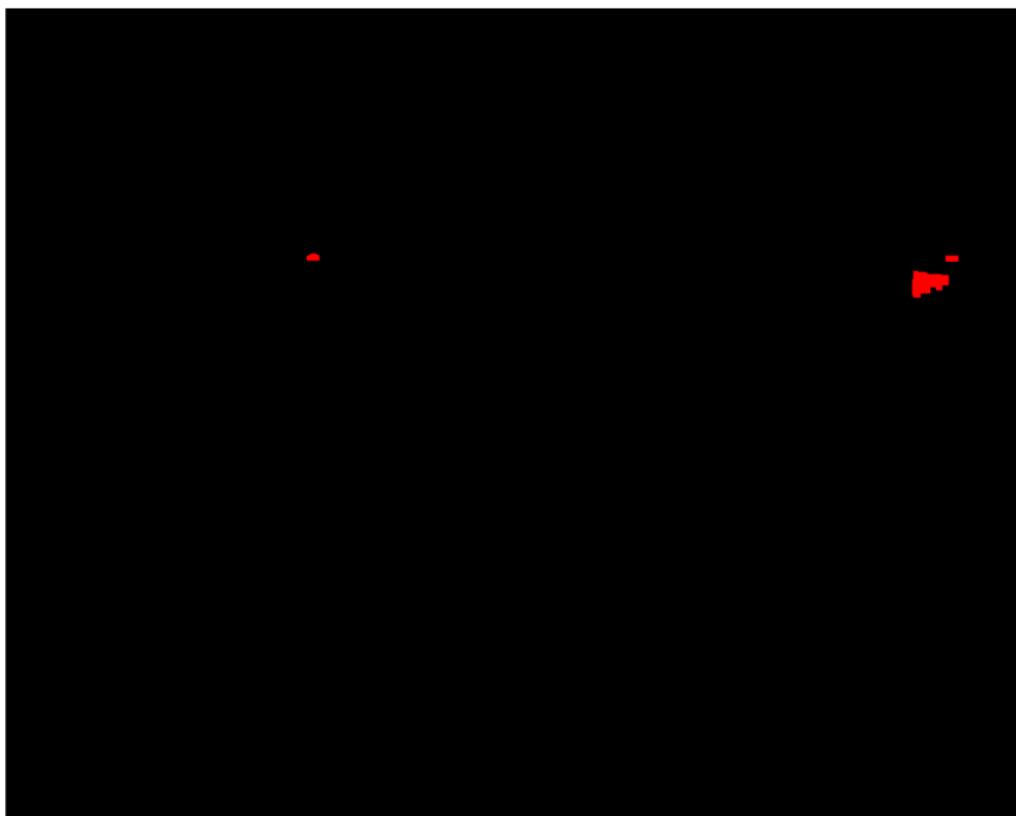


Figure 2.19: Pattern Unit — Pure Difference. Red pixels overlaid on the grayscale reference show only the detected difference regions.

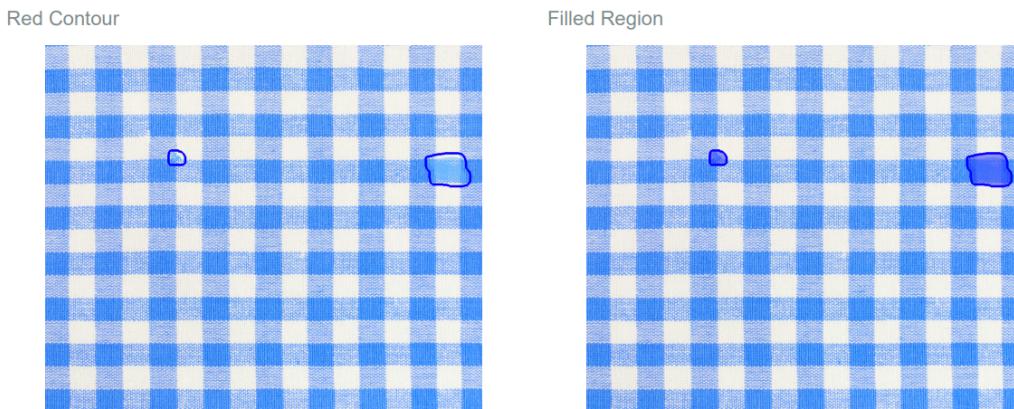
### How to Evaluate

- Similarity  $\geq \text{Threshold}_1$ : IDENTICAL.
- Similarity  $\geq \text{Threshold}_2$ : VERY SIMILAR.
- Similarity  $\geq \text{Threshold}_3$ : SIMILAR.
- Similarity  $< \text{Threshold}_3$ : DIFFERENT.

*Threshold values are defined by the user and can be adjusted according to evaluation requirements.*

## 2.2.6 Gradient Boundary Detection

### Gradient Similarity - Boundary Detection



#### Detection Metrics

Metric	Value
Binary Similarity	99.06%
Weighted Similarity	99.87%

Figure 2.20: Pattern Unit — Gradient Boundary Detection. Left: red ( or blue ) contours outlining detected difference regions. Right: filled overlay highlighting difference areas.

#### Definition and Purpose

Gradient Boundary Detection identifies and outlines specific regions of the sample where gradient-based differences exceed a significance threshold.

#### How It Is Calculated

The gradient difference map is thresholded at the 70th percentile. Morphological closing (kernel  $7 \times 7$ , 2 iterations), opening (1 iteration), and dilation (kernel  $5 \times 5$ , 2 iterations) are applied. Contours are extracted; those with area  $> 100$  pixels are retained. Two visualizations are produced: (1) red contour outlines on the sample image, and (2) red-filled overlay (40% blend) on the sample image.

Two metrics are computed:

- **Binary Similarity:** percentage of non-colored pixels.
- **Weighted Similarity:** same formula but weighted by normalized gradient intensity.

#### How to Read the Result

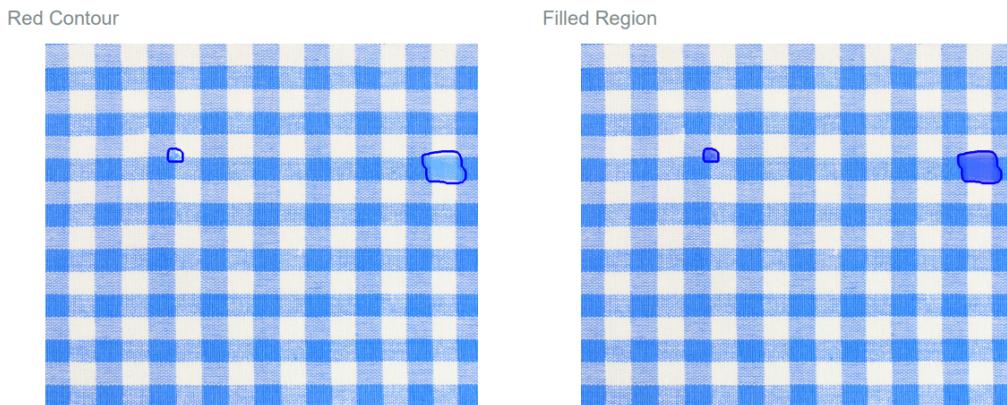
Red ( or blue ) contours/filled regions show where the system detected significant gradient differences. More red area indicates more deviation.

#### How to Evaluate

Fewer and smaller red ( or blue ) regions indicate better pattern match.

## 2.2.7 Phase Boundary Detection

### Phase Correlation - Boundary Detection



#### Detection Metrics

Metric	Value
Binary Similarity	99.13%
Weighted Similarity	99.87%

Figure 2.21: Pattern Unit — Phase Boundary Detection. Same layout as Gradient Boundary but based on phase correlation differences.

#### Definition and Purpose

Phase Boundary Detection identifies regions where phase-based differences exceed the significance threshold, using the same boundary extraction pipeline as Gradient Boundary but applied to the phase difference map.

#### How It Is Calculated

Identical to Gradient Boundary Detection (Section 2.2.6) but using the phase correlation difference map as input.

#### How to Read the Result

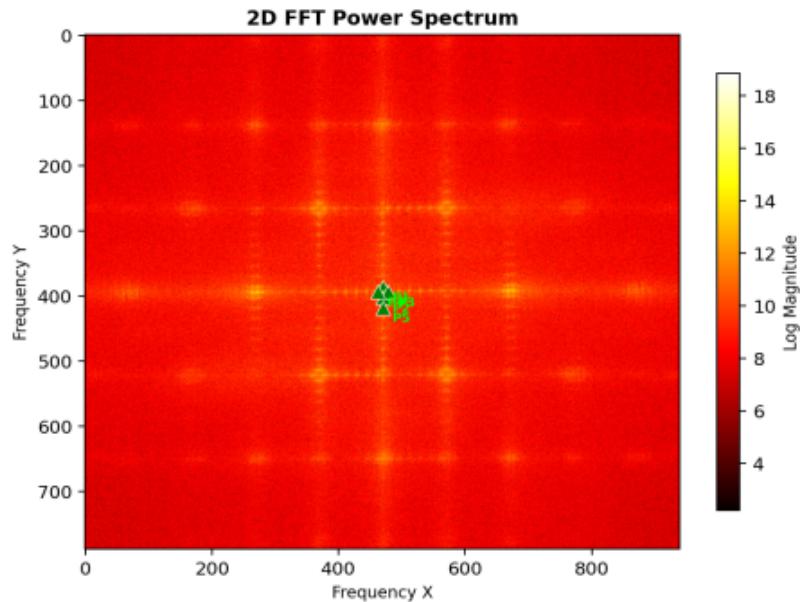
Same interpretation as Gradient Boundary: red ( or blue ) regions indicate detected phase-based differences.

#### How to Evaluate

Phase boundaries may detect different types of distortion (translational shifts) compared to gradient boundaries (edge/texture changes).

## 2.2.8 Fourier Domain Analysis

### 2D FFT Power Spectrum



### Frequency Peaks

Peak	Radius	Angle (°)	Magnitude
P1	8.00	90.00	8126859.89
P2	8.00	-90.00	8126859.89
P3	8.00	0.00	6715481.19
P4	8.00	180.00	6715481.19
P5	24.00	-90.00	2590453.43

### Fourier Metrics

Metric	Reference	Sample
Fundamental Period (px)	117.50	117.50
Dominant Orientation (°)	-90.00	90.00
Anisotropy Ratio	3.00	3.00

Figure 2.22: Pattern Unit — Fourier Domain Analysis. FFT magnitude spectrum heatmap, dominant peaks table, and spectral metrics.

### Definition and Purpose

Fourier Domain Analysis decomposes the image into its frequency components to characterize dominant periodic patterns, textures, and directional structures (e.g., warp/weft alignment in woven fabrics).

## How It Is Calculated

A 2D Fast Fourier Transform (FFT) is applied to the grayscale image. The DC component (zero-frequency) is shifted to the center of the spectrum. The magnitude spectrum is log-transformed for visualization:  $\text{spectrum} = \log(1 + |\text{FFT}|)$ .

**Peak Detection:** The DC component is masked. The top 5 peaks are identified by iterative maximum finding with adaptive neighborhood suppression (radius  $\max(3, \lfloor 0.15 \times r_{\text{peak}} \rfloor)$ ). For each peak, the radius (spatial frequency), angle (orientation in degrees), and magnitude are recorded.

**Metrics:** Three summary metrics are computed:

- **Fundamental Period:**  $\max(h, w)/r_{\text{dominant}}$  — the image dimension divided by the dominant peak's radius (spatial frequency).
- **Dominant Orientation:** angle (degrees) of the strongest peak.
- **Anisotropy Ratio:** ratio of the maximum to minimum peak radii among the detected peaks. A value of 1.0 indicates isotropic (uniform) frequency distribution.

## How to Read the Result

The FFT spectrum image shows bright spots at dominant spatial frequencies. The peaks table lists the top 5 peaks with their spatial frequency (radius), orientation (angle), and strength (magnitude).

## How to Evaluate

High anisotropy ratio indicates directional texture (e.g., strong warp/weft alignment). Short fundamental period indicates fine texture; long period indicates coarse pattern. Comparing reference and sample metrics reveals whether the sample preserves the original texture characteristics.

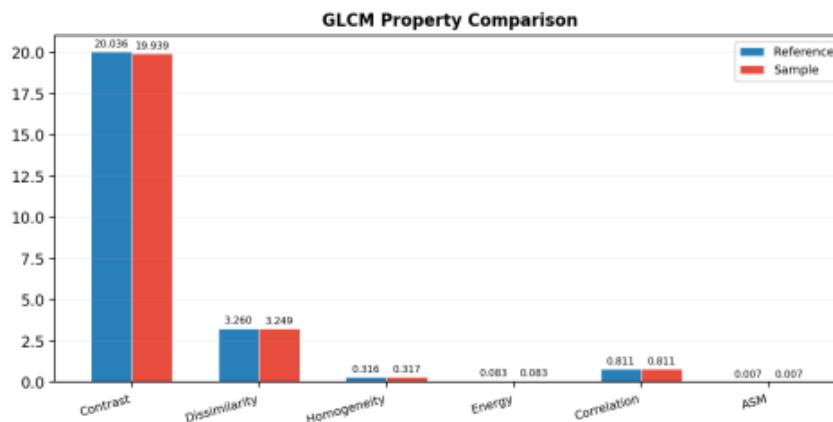
## 2.2.9 GLCM Texture Analysis

### GLCM Texture Analysis

Gray-Level Co-occurrence Matrix quantifies texture properties by analyzing spatial relationships between pixel intensities.

#### GLCM Property Comparison

Property	Reference	Sample
Contrast	20.0359	19.9390
Dissimilarity	3.2605	3.2491
Homogeneity	0.3162	0.3175
Energy	0.0833	0.0832
Correlation	0.8109	0.8113
ASM	0.0070	0.0070



#### GLCM Matrix Visualization

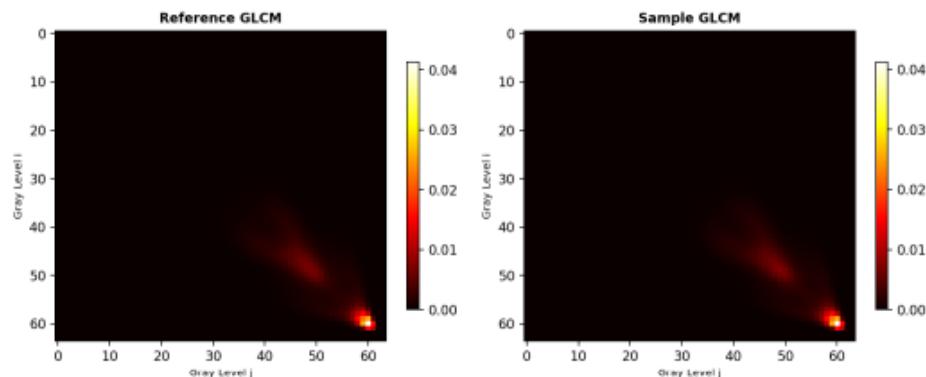


Figure 2.23: Pattern Unit — GLCM Texture Analysis. Property comparison bar chart and co-occurrence matrix heatmaps for reference and sample.

#### Definition and Purpose

Gray-Level Co-occurrence Matrix (GLCM) analysis quantifies texture properties by measuring the statistical distribution of pixel intensity pairs at a specified spatial relationship.

## How It Is Calculated

The grayscale image is quantized to 64 levels. The GLCM is computed using scikit-image's `graycomatrix` at distance 1 with angles  $[0, \pi/4, \pi/2, 3\pi/4]$ , then averaged across angles. Six properties are extracted using `graycoprops`:

- **Contrast:** intensity difference between neighboring pixels.
- **Dissimilarity:** average absolute difference between neighboring pixels.
- **Homogeneity:** closeness of element distribution to the GLCM diagonal.
- **Energy:** sum of squared elements (uniformity measure).
- **Correlation:** linear dependency of gray levels on neighboring pixels.
- **ASM (Angular Second Moment):** square of energy, measures orderliness.

A bar chart compares reference and sample properties. GLCM heatmaps visualize the co-occurrence matrices side by side.

## How to Read the Result

The property comparison table shows reference and sample values for each property. The bar chart provides a visual comparison. The heatmaps show the raw co-occurrence patterns.

## How to Evaluate

Similar values between reference and sample indicate consistent texture. Large differences in contrast or dissimilarity suggest texture degradation. High energy/ASM indicates uniform texture; low values indicate complex or irregular texture.

### 2.2.10 Verdict

#### Analysis Verdict

Metric	Value
Changed Pixels	745
Structural Match	99.90%
VERDICT	≈ VERY SIMILAR

Figure 2.24: Pattern Unit — Structural Difference Verdict. Changed pixels count, similarity score, and verdict (IDENTICAL / VERY SIMILAR / SIMILAR / DIFFERENT).

## Definition and Purpose

The Verdict table summarizes the Structural Difference Analysis result with a single classification.

## How It Is Generated

Changed pixels are counted from the noise-filtered mask. Similarity score is 100–change\_percentage. The verdict is: IDENTICAL ( $\geq 99.9\%$ ), VERY SIMILAR ( $\geq 99.0\%$ ), SIMILAR ( $\geq 95.0\%$ ), or DIFFERENT ( $< 95.0\%$ ).

## How to Read the Result

The verdict is displayed with an icon: ✓ for IDENTICAL, ≈ for SIMILAR/VERY SIMILAR, × for DIFFERENT.

## How to Evaluate

IDENTICAL or VERY SIMILAR indicates production consistency. DIFFERENT requires investigation of the manufacturing process.

### 2.2.11 Recommendations and Conclusion

#### Key Findings & Recommendations

Composite Score	99.5%
Excellent structural alignment — pattern integrity preserved. -> Process is optimal. No corrective action needed.	
Structural Match	99.90%
Structurally identical — no geometric distortion detected. -> No layout issues. Fabric handling is consistent.	

#### Conclusion & Decision

PASS	Sample meets all quality standards within the defined tolerances. Approved for production.
------	--

Figure 2.25: Pattern Unit — Key Findings & Recommendations. Card-based findings for composite score, structural match, and weakest metric.

## Definition and Purpose

Pattern recommendations provide threshold-aware findings and actionable guidance for the pattern analysis results.

## How It Is Generated

Three metrics are evaluated:

1. **Composite Score:** evaluated against the global pattern threshold. Levels: excellent, good, marginal, poor.
2. **Structural Match:** evaluated against fixed thresholds ( $\geq 99.5\%$ : identical,  $\geq 97\%$ : minor deviation,  $\geq 90\%$ : moderate distortion,  $< 90\%$ : severe distortion).
3. **Weakest Metric:** the individual method with the lowest score, flagged if below its pass threshold.

# Conclusion & Decision

ACCEPT

Sample matches reference within acceptable tolerances.

Figure 2.26: Pattern Unit — Conclusion block with overall status and decision text.

## How to Read the Result

Same card-based layout as the Color Unit recommendations. The conclusion block provides the final decision with a color-coded status badge: green (PASS), orange (CONDITIONAL), red (FAIL).

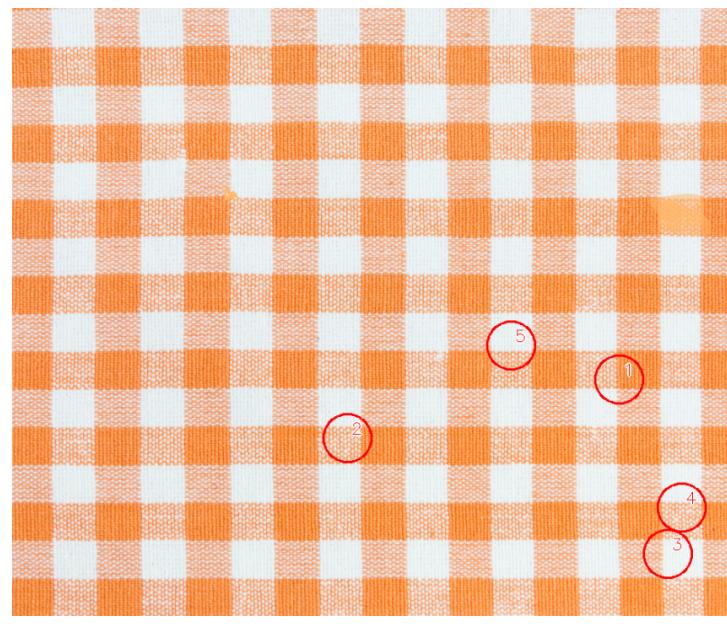
## How to Evaluate

Recommendations include textile-specific guidance such as “Calibrate alignment sensors and verify fabric feed tension” or “Stop production. Perform full machine calibration and inspection.”

## 2.3 Single Image Unit

The Single Image Unit is an extension of the Color Unit adapted for standalone analysis of a single sample image without a reference. It evaluates color distribution, texture uniformity, and spectral characteristics.

### 2.3.1 Input



Sample Image with Analysis Points

Figure 2.27: Single Image Unit — Input. Sample image with analysis point overlays.

## Definition and Purpose

Displays the sample image with numbered sampling point overlays, identical in style to the Color Unit.

## How It Is Generated

Same point placement and visualization as the Color Unit. Green circles for manual points; orange for random.

## How to Read the Result

Verify that sampling points cover representative areas of the fabric.

## How to Evaluate

Points should be distributed across the fabric to capture any spatial variation in color or texture.

### 2.3.2 Measurement Data

#### Measurement Data

##### RGB Values

Pt	Position	R	G	B
1	(786, 481)	246	182	138
2	(434, 557)	241	196	164
3	(849, 707)	242	222	207
4	(867, 647)	244	212	191
5	(646, 437)	239	221	208

##### Lab\* Values (D65)

Pt	L*	a*	b*
1	425.57	82.24	146.52
2	441.20	54.42	102.66
3	475.72	20.23	45.19
4	463.43	34.86	68.38
5	473.50	17.15	40.12

##### CMYK Values

Pt	C	M	Y	K
1	0.00	0.26	0.44	0.03
2	0.00	0.19	0.32	0.05
3	0.00	0.08	0.14	0.05
4	0.00	0.13	0.22	0.04
5	0.00	0.07	0.13	0.06

##### XYZ Values (D65)

Pt	X	Y	Z
1	59.52	55.16	31.63
2	63.15	61.23	43.86
3	74.49	76.17	70.37
4	70.56	70.60	59.33
5	73.18	75.14	70.70

Figure 2.28: Single Image Unit — Measurement Data. RGB, Lab\*, XYZ, and CMYK tables; Lab\* statistics; per-illuminant Lab\* values; and RGB histogram.

## Definition and Purpose

The Measurements section provides per-point color values in multiple color spaces (RGB, Lab\*, XYZ, CMYK), Lab\* statistical summary (mean, std dev, min, max per component) and per-illuminant Lab\* tables.

## How It Is Calculated

At each sampling point, the same circular patch extraction and color conversion pipeline as the Color Unit is applied (sRGB → XYZ → chromatic adaptation → CIELAB; sRGB → CMYK).

For multi-illuminant analysis, each illuminant produces a separate Lab\* table with per-point and mean values. The RGB histogram is computed from the full sample image.

## How to Read the Result

Tables provide numeric values per point. The Lab\* statistics table shows the spread across sampling points. Multiple illuminant tables allow comparison of how the color appears under different lighting.

## How to Evaluate

High standard deviation in  $L^*$  across sampling points indicates poor color uniformity. Wide spread in  $a^*$  or  $b^*$  indicates chromaticity variation. The histogram shape reveals whether the color distribution is tight (uniform) or broad (variable).

### 2.3.3 Spectral Analysis, Fourier Analysis, and Recommendations

These sections are not newly defined here, they correspond to analysis components that already exist within the **Color Unit** and have been previously introduced and described in Section 2.1. In this context, they are reused to provide single-image analysis results without repeating their methodological description.

# Chapter 3

## Frequently Asked Questions

### Q1: What is the difference between RGB and CIELAB?

RGB is a device-dependent color model: the same RGB values may appear differently on different displays. CIELAB (CIE  $L^*a^*b^*$ ) is a perceptually uniform, device-independent color space derived from CIE XYZ. In CIELAB, equal numeric differences correspond to approximately equal perceived color differences. For quality control, CIELAB and  $\Delta E_{2000}$  (derived from CIELAB) are preferred over raw RGB differences because they align with human color perception.

### Q2: Why does the system compute three Delta E formulas?

$\Delta E_{76}$ ,  $\Delta E_{94}$ , and  $\Delta E_{2000}$  represent successive improvements in modeling human color perception:

- $\Delta E_{76}$ : Euclidean distance in CIELAB. Simple but treats all color directions equally, which does not match human perception.
- $\Delta E_{94}$ : Adds weighting to account for the fact that the eye is more sensitive to lightness changes than chroma or hue changes, and sensitivity varies with chroma level.
- $\Delta E_{2000}$ : The current CIE standard. Adds hue rotation terms, improved chroma weighting, and an interaction term between chroma and hue differences. It is the most perceptually accurate and is the decision metric used by SpectraMatch.

All three are reported for completeness.  $\Delta E_{76}$  and  $\Delta E_{94}$  are labeled “informational” in the summary table; only  $\Delta E_{2000}$  determines the pass/fail status.

### Q3: What is CSI and how does it differ from Delta E?

The Color Similarity Index (CSI) is a full-image, pixel-level metric computed by downsampling both images to 25% resolution, converting to CIELAB, and computing the mean per-pixel  $\Delta E_{76}$ . It is then mapped to a 0–100 scale. In contrast,  $\Delta E_{2000}$  is computed only at the  $N$  discrete sampling points. CSI captures global color distribution across the entire image, while  $\Delta E_{2000}$  captures point-specific color accuracy. They are complementary: CSI may detect overall shifts that sampling points miss, while  $\Delta E_{2000}$  provides more precise local measurements.

### Q4: What is CSI2000?

CSI2000 is a hybrid metric that averages the CSI and a  $\Delta E_{2000}$ -derived score:

$$\text{CSI2000} = \frac{\text{CSI} + \text{Score}_{\Delta E}}{2}, \quad \text{Score}_{\Delta E} = \max(0, \min(100, 100 - \overline{\Delta E}_{2000} \times 10))$$

## Q5: What is metamerism and why does the illuminant analysis matter?

Metamerism occurs when two color samples appear identical under one light source but different under another. This is a common problem in textiles: a fabric may match the reference under daylight (D65) but appear different under store lighting (F2, TL84). The illuminant analysis section tests this by re-computing  $\Delta E_{2000}$  under each selected illuminant using the Bradford chromatic adaptation transform. If results differ significantly across illuminants, the dye formulation may need adjustment.

## Q6: What does the spectral proxy chart actually show?

The spectral proxy is an *approximation* of spectral reflectance, synthesized from RGB values using three Gaussian basis functions (centered at 450, 545, and 610 nm). It is *not* a measured spectral reflectance curve. It provides a qualitative visual indication of how the color energy is distributed across wavelengths. For precise spectral data, a spectrophotometer is required.

## Q7: How does the composite pattern score work?

The composite pattern score is a weighted average of four analysis methods, each contributing 25%:

- Structural SSIM (25%)
- Gradient Similarity (25%)
- Phase Correlation (25%)
- Structural Match (25%)

Each method produces a score from 0–100%. The composite is evaluated against the global pattern threshold:  $\geq$  threshold → PASS,  $\geq$  (threshold – 15) → CONDITIONAL, below → FAIL.

The user can, from the advanced settings, configure the report to rely on only one selected metric instead of all of them. The user can also adjust the threshold values according to what they consider appropriate.

## Q8: What is the structural difference analysis?

It is a multi-method fusion approach that combines five different pixel-level difference detection techniques (simple difference, edge-based, gradient, frequency, and SSIM) into a single noise-filtered binary mask. This mask identifies which pixels have structurally changed between reference and sample. The similarity score is the percentage of unchanged pixels. The analysis is independent of the four primary pattern methods and provides an additional structural assessment.

## Q9: What is GLCM and when should it be enabled?

The Gray-Level Co-occurrence Matrix (GLCM) quantifies texture by analyzing how often pairs of pixel values occur at specific spatial relationships. It produces six properties (contrast, dissimilarity, homogeneity, energy, correlation, ASM). GLCM is enabled by default but can be disabled in the Report Sections settings if texture analysis is not required for a given use case.

## **Q10: How does the system handle region selection?**

The user selects a region of interest using one of four tools (circle, square, rectangle, freehand polygon). The selected region is sent to the backend as geometry data (center/radius for circles, bounding box for rectangles). The backend crops both images to the region bounding box and applies a mask (e.g., circular alpha mask for circles). All sampling points must lie within the region; the backend validates each point using geometric containment checks and rejects invalid points.

## **Q11: What happens if I do not select a region?**

If no region is selected, the full image is used for analysis. All sampling points are generated within the full image bounds.

## **Q12: What is the Settings Report?**

The Settings Report (configuration receipt) is a separate PDF generated alongside the analysis report. It documents every configuration parameter used for the analysis, including thresholds, illuminant selections, enabled sections, and scoring method. It enables reproducibility and audit trail verification.

## **Q13: Can I change the report language?**

Yes. The report language (English or Turkish) is selected in the Report Sections tab of Advanced Settings. The interface language and the report language are independent: you can use the English interface but generate Turkish reports, or vice versa.

## **Q14: What are the system requirements?**

SpectraMatch requires Python 3.9 or later with the following packages: Flask, OpenCV (headless), NumPy, PyPDF, Pillow, matplotlib, scikit-image, SciPy, and ReportLab. The desktop application additionally requires pywebview. Maximum upload size is 100 MB per image.

## **Q15: How are temporary files managed?**

Temporary files (cropped images, visualization plots, PDF intermediate files) are stored in the system temporary directory. A background cleanup thread runs every hour and removes temporary files older than 24 hours.

---

## Comparative Discussion — Why Similar Methods Coexist

---

The following questions address a recurring observation: several analysis methods in SpectraMatch appear to measure similar aspects of the image, yet each is retained in the report. The answers below explain *why* apparent redundancy is deliberate and what unique information each method contributes.

### Q16: Why are Gradient Similarity and Phase Correlation used even though they often produce similar results?

At first glance, both methods evaluate “structural closeness” between reference and sample, and their scores frequently land in a comparable range. However, they capture fundamentally different aspects of the image:

- **Gradient Similarity** operates in the *spatial domain*. It computes Sobel gradient magnitudes for each image, then applies SSIM to the gradient fields. It is sensitive to edge sharpness, texture density, and local contrast changes. A blurred or slightly out-of-focus sample will show a gradient drop even if the pattern is positionally correct.
- **Phase Correlation** operates in the *frequency domain*. It uses the Fourier phase relationship to detect translational (shift) misalignment. It is largely insensitive to uniform brightness or contrast changes but highly sensitive to geometric displacement.

#### When they diverge:

- A sample that is *correctly positioned* but has *softer edges* (e.g., due to different weave tension) will score high on Phase Correlation but lower on Gradient Similarity.
- A sample that is *slightly shifted* during capture but otherwise identical in texture will score high on Gradient Similarity but lower on Phase Correlation.

Retaining both methods ensures that the composite pattern score reflects *both* positional accuracy and edge/texture fidelity. Relying on only one would leave a blind spot for the failure mode the other detects.

### Q17: Why are both CSI and $\Delta E_{2000}$ used when they both measure color difference?

CSI and  $\Delta E_{2000}$  appear to answer the same question (“how different are the colors?”) but they differ in *scope*, *spatial resolution*, and *sensitivity*:

Aspect	$\Delta E_{2000}$	CSI
Sampling	$N$ discrete points placed by the user or randomly	Every pixel (at 25% resolution)
Formula	Full CIEDE2000 with hue rotation and interaction terms	Euclidean $\Delta E_{76}$ in scaled Lab
Strength	High perceptual accuracy at each point	Captures global distribution shifts across the full image
Weakness	May miss defects between sampling points	Uses a simpler formula, less perceptually accurate per pixel

### When they diverge:

- A localized stain or streak that falls *between* sampling points will not affect  $\Delta E_{2000}$  but will reduce CSI.
- A subtle, perceptually significant hue shift that is uniform across the fabric will be detected more precisely by  $\Delta E_{2000}$  (which uses the advanced CIEDE2000 formula) than by CSI (which uses the simpler  $\Delta E_{76}$ ).

The hybrid metric **CSI2000** exists precisely because neither alone is sufficient. Together, they provide both breadth (full-image coverage) and depth (perceptual precision at critical points).

### Q18: Why do Gradient Boundary Detection and Phase Boundary Detection both exist if they use the same pipeline?

Both boundary detection methods share identical morphological post-processing (thresholding, closing, opening, dilation, contour extraction). The difference lies entirely in their *input map*:

- **Gradient Boundary** uses the Sobel gradient difference map. It highlights regions where *edge intensity or texture sharpness* differs.
- **Phase Boundary** uses the phase correlation absolute difference map. It highlights regions where *pixel-level translational alignment* differs.

Because the same contour-extraction pipeline is applied to two different source maps, the resulting boundary regions can differ substantially:

- Gradient boundaries tend to appear at *pattern edges* where weave structure or print definition has changed.
- Phase boundaries tend to appear in *broad areas* where the sample is spatially displaced relative to the reference.

In practice, when both boundary maps overlap, it provides strong evidence of a genuine defect. When only one fires, it narrows the root cause (texture degradation vs. positional shift).

### Q19: SSIM is used in the Pattern Unit score and also inside the Structural Difference Analysis — is that double counting?

No. Although the abbreviation “SSIM” appears in both contexts, the two uses serve different purposes and operate on different inputs:

- **Structural SSIM (Pattern Method 1)** applies SSIM to the full grayscale images after bilateral filtering. It produces a 0–100% score and contributes 25% to the composite pattern score. Its role is to provide a *single perceptual similarity number*.
- **SSIM inside Structural Difference Analysis** uses the inverted SSIM map (thresholded at 200) as one of *five* binary difference masks. It contributes only 15% of the fused mask. Its role is to help *localize changed pixels* for the structural verdict (IDENTICAL / VERY SIMILAR / SIMILAR / DIFFERENT).

The first use produces a *score*; the second produces a *spatial mask*. They answer different questions: “How similar overall?” vs. “Where exactly are the differences?” The Structural Difference Analysis deliberately includes SSIM as one of five complementary detectors because each detector has different false-positive and false-negative characteristics. The fusion reduces the weaknesses of any individual method.

# Chapter 4

## Software Implementation

### 4.1 Programming Tools and Technologies

SpectraMatch is built using the following technologies:

Table 4.1: Technology stack used in SpectraMatch v2.2.3.

Category	Technology	Role
Web Framework	Flask	Backend HTTP server, API routing, template rendering
Image Processing	OpenCV (headless)	Image I/O, color space conversion, filtering, morphology, FFT, phase correlation
Numerical Computing	NumPy	Array operations, linear algebra, statistical computations
Scientific Computing	SciPy	Signal processing utilities
Image Analysis	scikit-image	SSIM computation, GLCM texture analysis
Plotting	matplotlib	Chart generation (histograms, scatter plots, bar charts, heatmaps, spectral curves)
PDF Generation	ReportLab	Multi-page PDF report construction with tables, images, styled text, and custom flowables
PDF Merging	PyPDF	Merging cover page, color report, and pattern report PDFs into a single file
Image Handling	Pillow (PIL)	Image format conversion, NumPy-to-PIL bridging for PDF embedding
Desktop Shell	pywebview	Native window wrapper for the Flask backend, JavaScript–Python bridge
Frontend	Vanilla JavaScript	UI controller, region selection, point selection, internationalization
Styling	CSS	Interface styling, responsive layout, dark/light mode

## 4.2 Codebase Language Distribution

The following is an approximate distribution of the codebase by programming language, estimated from file sizes:

Table 4.2: Approximate codebase language distribution.

Language	Approximate Size	Percentage
Python (backend modules)	~240 KB	~30%
JavaScript (frontend)	~365 KB	~45%
HTML (templates)	~196 KB	~20%
CSS / Config / Other	~40 KB	~5%

## 4.3 File and Folder Structure

```

SPECTRAMATCH_PROJECT/
    app.py                      # Flask application entry point
    wsgi.py                     # WSGI deployment entry point
    requirements.txt            # Python dependencies
    README.md                   # Project documentation
    modules/
        ColorUnitBackend.py    # Color analysis and PDF generation
        PatternUnitBackend.py  # Pattern analysis and PDF generation
        SingleImageUnitBackend.py # Single image analysis and PDF
        RecommendationsEngine.py # Threshold-aware recommendations
        ReportUtils.py          # Shared PDF utilities and styles
        ReportTranslations.py   # EN/TR translation dictionary
        SettingsReceipt.py      # Settings receipt PDF generator
    desktop/
        app_desktop.py          # pywebview desktop launcher
    templates/
        index.html              # Web application template
        desktop.html             # Desktop application template
    static/
        js/
            app.js                # Main UI controller
            region-selector.js    # Region selection tools
            region-validator.js   # Point-in-region validation
            point-selector-enhanced.js # Enhanced point selection
            i18n.js                 # Internationalization (EN/TR)
            development-modal.js   # Development modal
        css/                      # Stylesheets
        images/                  # Logo and static images
    desktop/
READYTOTEST/                         # Built-in test image pairs

```

## 4.4 Key Functions and Classes

The following tables list the primary functions and classes in each module with their location and role.

### 4.4.1 app.py — Flask Application

Function / Route	Role
<code>crop_image()</code>	Crops uploaded images to the selected region; applies circular alpha masks for circle selections.
<code>@app.route("/")</code>	Serves the web application template.
<code>@app.route("/desktop")</code>	Serves the desktop application template.
<code>/api/analyze</code>	Main analysis endpoint. Accepts images, settings, region data. Orchestrates Color Unit, Pattern Unit, or Single Image Unit. Generates PDFs, saves visualization images, constructs JSON response.
<code>/api/download_report/</code>	Serves generated PDF reports (merged, color, pattern).
<code>/api/download_receipt/</code>	Serves the settings receipt PDF.
<code>/api/report_image/</code>	Serves visualization images for frontend display.
<code>cleanup_old_files()</code>	Background thread that removes temp files older than 24 hours.
<code>_sanitize_for_json()</code>	Converts NumPy types to Python-native types for JSON serialization.

### 4.4.2 ColorUnitBackend.py — Color Analysis

Function	Role
<code>srgb_to_xyz()</code>	Converts normalized sRGB to CIE XYZ.
<code>adapt_to_illuminant()</code>	Chromatically adapts XYZ from D65 to a target illuminant using the Bradford transform.
<code>xyz_to_lab()</code>	Converts XYZ to CIELAB.
<code>deltaE76()</code>	Computes $\Delta E_{76}$ (Euclidean distance in CIELAB).
<code>deltaE94()</code>	Computes $\Delta E_{94}$ with lightness/chroma/hue weighting.
<code>deltaE2000()</code>	Computes CIEDE2000 color difference.
<code>region_stats()</code>	Extracts mean RGB, XYZ, Lab, CMYK from a circular patch at a point.
<code>is_point_valid()</code>	Validates whether a point lies within the defined region.
<code>make_points_strict()</code>	Generates $N$ random points within a region using rejection sampling.
<code>analyze_color()</code>	Main analysis function: extracts colors at points, computes all metrics, CSI.
<code>generate_pdf_headless()</code>	Generates the Color Unit PDF report with all sections.
<code>analyze_and_generate()</code>	Entry point: runs analysis then generates PDF.
<code>plot_spectral_proxy()</code>	Generates spectral proxy chart.
<code>plot_rgb_...</code>	Generates side-by-side RGB histograms.
<code>histograms_dual()</code>	
<code>plot_heatmap()</code>	Generates $\Delta E$ heatmap visualization.
<code>plot_lab_scatter()</code>	Generates $a^*$ vs $b^*$ chromaticity scatter plot.
<code>plot_lab_bars()</code>	Generates grouped bar chart of mean $L^*$ , $a^*$ , $b^*$ .

#### 4.4.3 PatternUnitBackend.py — Pattern Analysis

Function	Role
method1_structural_ssim()	Computes SSIM with bilateral filtering and JET-colored difference map.
method3_gradient_similarity()	Computes Sobel gradient magnitude comparison with SSIM scoring.
method6_phase_correlation()	Computes phase correlation alignment score using OpenCV.
structural_difference_... analysis()	Multi-method fusion: simple diff, edge, gradient, frequency, SSIM with noise filtering.
fourier_domain_analysis()	2D FFT, peak detection, fundamental period, orientation, anisotropy.
glcm_texture_analysis()	GLCM computation and property extraction (contrast, homogeneity, etc.).
create_gradient_... red_boundaries()	Generates gradient-based boundary detection visualizations.
create_phase_... red_boundaries()	Generates phase-based boundary detection visualizations.
determine_status()	Generic pass/conditional/fail logic based on thresholds.
generate_pdf_headless()	Generates the Pattern Unit PDF report with all sections.
analyze_and_generate()	Entry point: runs all analyses, computes composite score, generates PDF.
plot_fft_spectrum()	Generates FFT magnitude spectrum heatmap.
plot_glcm_comparison()	Generates GLCM property comparison bar chart.
plot_glcm_heatmaps()	Generates side-by-side GLCM heatmaps.

#### 4.4.4 SingleImageUnitBackend.py — Single Image Analysis

Function	Role
analyze_and_generate()	Entry point: sampling, color extraction, spectral proxy, PDF generation.
_generate_pdf()	Internal PDF generation with cover, visuals, measurements, Fourier, recommendations.
plot_single_... spectral_proxy()	Generates single-image spectral proxy chart.
is_point_valid()	Point validation (circle/rectangle containment).
make_points_strict()	Random point generation within region bounds.

#### 4.4.5 RecommendationsEngine.py — Findings and Recommendations

Function	Role
<code>generate_color_...</code>	Produces findings for Mean $\Delta E$ , Consistency, and CSI with threshold-relative evaluations.
<code>recommendations()</code>	Produces findings for Composite Score, Structural Match, and Weakest Metric.
<code>generate_pattern_...</code>	Produces findings for Luminance, Chroma Spread, Color Uniformity, and Dominant Channel.
<code>recommendations()</code>	Converts findings into styled ReportLab flowables (card-based layout with conclusion block).
<code>render_findings_...</code>	Maps a numeric value to a qualitative level (excellent/good/marginal/poor) relative to thresholds.
<code>to_flowables()</code>	
<code>_relative_position()</code>	

#### 4.4.6 ReportUtils.py — Shared PDF Utilities

Function / Constant	Role
<code>setup_fonts()</code>	Registers TrueType fonts with Turkish character support (Arial → Segoe UI → Tahoma → Helvetica fallback chain).
<code>numpy_to_rl()</code>	Converts a NumPy image array to a ReportLab Image object, handling BGRA alpha compositing.
<code>make_table()</code>	Creates consistently styled tables with alternating row backgrounds.
<code>badge()</code>	Creates a colored rounded-rectangle badge flowable.
<code>make_header_footer()</code>	Returns a callback function that draws page headers (company name, subtitle) and footers (page number, timestamp, logo).
<code>generate_unified_cover()</code>	Generates the unified cover page PDF for merged reports with dual score cards.
<code>pick_logo()</code>	Locates the company logo file from primary and fallback paths.
<code>STATUS_COLORS</code>	Dictionary mapping status strings (PASS, CONDITIONAL, FAIL) to ReportLab colors.

#### 4.4.7 ReportTranslations.py — Bilingual Dictionary

Contains a comprehensive TRANSLATIONS dictionary with English and Turkish entries for all report elements: section titles, table headers, metric labels, status strings, evaluation texts, and interpretive descriptions. Provides `get_translator(lang)` which returns a lookup function for the selected language, and `translate_status(status, lang)` for status string translation.

#### 4.4.8 SettingsReceipt.py — Configuration Receipt

Generates a one-page PDF documenting all analysis configuration parameters: general settings, color thresholds, pattern thresholds, illuminant selections, enabled report sections, and scoring method. The receipt shares the same report ID and timestamp as the corresponding analysis report.

#### 4.4.9 desktop/app\_desktop.py — Desktop Launcher

Component	Role
<code>find_free_port()</code>	Scans for an available TCP port for the Flask backend.
<code>start_flask()</code>	Starts the Flask server in a background thread.
Api class	JavaScript–Python bridge providing <code>save_report()</code> for native file save dialogs.
Splash screen	Displays a loading window while the Flask backend initializes.
<code>webview.create_window()</code>	Creates the native desktop window pointing to the Flask URL.

#### 4.4.10 Frontend JavaScript

File	Role
<code>app.js</code>	Main UI controller: image upload handling, settings management, analysis initiation, result display, report download, feedback submission.
<code>region-selector.js</code>	Interactive canvas-based region selection tools: circle, square, rectangle, freehand polygon with drag, resize, and synchronized dual-image overlay.
<code>region-validator.js</code>	Client-side point-in-region validation for sampling point placement.
<code>pointselectorenhanced.js</code>	Enhanced point selection UI: click-to-place manual points, visual feedback, point count management.
<code>i18n.js</code>	Full English/Turkish internationalization dictionary and language switching logic.
<code>development-modal.js</code>	Development information modal.

# Appendix A

## Glossary of Abbreviations

Abbreviation	Definition
ASM	Angular Second Moment
BGR	Blue–Green–Red
CIE	International Commission on Illumination
CIELAB	CIE L*a*b* color space
CIEDE2000	CIE Delta E 2000
CLAHE	Contrast Limited Adaptive Histogram Equalization
CMYK	Cyan–Magenta–Yellow–Black
CSI	Color Similarity Index
CSI2000	Color Similarity Index 2000
CSS	Cascading Style Sheets
DC	Direct Current
DPI	Dots Per Inch
$\Delta E$	Color Difference
$\Delta E_{76}$	CIE76 Color Difference
$\Delta E_{94}$	CIE94 Color Difference
$\Delta E_{2000}$	CIEDE2000 Color Difference
EN/TR	English / Turkish
FFT	Fast Fourier Transform
GLCM	Gray-Level Co-occurrence Matrix
HTML	HyperText Markup Language
ISO	International Organization for Standardization
JS	JavaScript
Lab	CIELAB
PDF	Portable Document Format
RGB	Red–Green–Blue
sRGB	Standard RGB
SSIM	Structural Similarity Index
UTC	Coordinated Universal Time
WSGI	Web Server Gateway Interface
XYZ	CIE XYZ Color Space

# Appendix B

## List of Figures

Figure	Title
1	System Workflow
2	Web Application Interface
3	Desktop Application Interface
4	Settings Report Cover Page
<i>Color Unit</i>	
5	Input Images with Sampling Points
6	RGB Values Table
7	Mean CIELAB Values and Component-wise Differences
8	Lab* Visualization
9	XYZ Values Table
10	CMYK Values Table
11	Difference Metrics ( $\Delta E$ )
12	Illuminant Analysis
13	$\Delta E$ Heatmap
14	Spectral Distribution Proxy
15	RGB Histograms
16	Color Similarity Index (CSI)
17	Color Findings & Recommendations
<i>Pattern Unit</i>	
18	Pattern Input Images
19	SSIM Difference Map
20	Gradient Similarity Map
21	Phase Correlation Map
22	Structural Difference Analysis
23	Pure Difference Overlay
24	Gradient Boundary Detection
25	Phase Boundary Detection
26	Fourier Domain Analysis
27	GLCM Texture Analysis
28	Structural Difference Verdict
29	Pattern Findings & Recommendations
30	Pattern Conclusion
<i>Single Image Unit</i>	
31	Single Image Input
32	Measurement Data

# Appendix C

## Known Limitations (v2.2.3)

### ‘Planned for v2.3.0 — Not Yet Implemented’

The items listed below are known limitations of SpectraMatch v2.2.3.

They are documented here for transparency and are scheduled to be addressed in the next release.

Each item is marked with a status indicator: ■ **Not implemented** — feature does not exist in v2.2.3.

### ■ L-1 | Single Image Report — Organizational Restructuring NOT IMPLEMENTED

**Current State:** The Single Image Report reuses section definitions from the Color Unit (see Section 2.3). Its internal page flow and section ordering have not been independently optimized for the single-image use case.

**Planned Change:** Organizational-only restructuring of the Single Image Report. No new analysis methods will be added; the existing sections will be reordered and reformatted to improve readability and logical flow when no reference image is present. This includes adjusting section titles, page breaks, and the recommendations layout to better suit standalone analysis.

**Impact:** Low — visual/organizational only. No effect on analysis accuracy or metrics.

### ■ L-2 | Per-Image Illuminant Settings

NOT IMPLEMENTED

**Current State:** The illuminant setting is global: the same primary illuminant and the same set of test illuminants are applied to *both* the reference and the sample image. This works well when both images are captured under identical lighting conditions.

**Limitation:** In practice, the reference and sample images may have been captured under different lighting environments (e.g., the reference under controlled D65 daylight and the sample under warehouse fluorescent TL84). Applying the same illuminant to both introduces a systematic bias in the chromatic adaptation step, which can inflate or deflate  $\Delta E$  values.

**Planned Change:** Allow independent illuminant selection for the reference image and the sample image. The settings panel will provide two separate illuminant selectors. The Color Unit will apply each image’s designated illuminant independently during the sRGB → XYZ → CIELAB conversion pipeline.

**Impact:** Medium — improves accuracy when images are captured under different lighting.  
Requires UI and backend changes.

**L-3 | Image Registration (Major Planned Addition)***NOT IMPLEMENTED*

**Current State:** SpectraMatch v2.2.3 compares the reference and sample images **pixel by pixel** after cropping to the selected region. There is no geometric alignment step. This means the system assumes that both images are already spatially registered — captured from the same viewing angle, at the same scale, and with the same framing.

**Limitation:** In real-world use, the reference and sample images are rarely captured under identical geometric conditions. Common differences include:

- **Viewing angle** — the camera may be tilted or rotated between captures.
- **Scale differences** — the camera-to-fabric distance may vary, producing different magnifications.
- **Image size variations** — different camera resolutions or crop settings.
- **Translation/shift** — the fabric may be positioned differently in the frame.

These geometric mismatches propagate into *every* downstream metric:  $\Delta E$  values are inflated because misaligned pixels are compared, SSIM drops because structures are shifted, and the structural difference analysis reports false positives at every misaligned boundary.

**Planned Change:** Integrate an **Image Registration** preprocessing stage before any analysis. The registration pipeline will employ multiple complementary techniques:

1. **Feature-based alignment** — detect keypoints (e.g., ORB, SIFT, or AKAZE), match descriptors between reference and sample, compute a homography matrix, and warp the sample to align with the reference.
2. **Intensity-based alignment** — use Enhanced Correlation Coefficient (ECC) optimization to refine the geometric transformation by maximizing pixel-level correlation.
3. **Phase-based alignment** — leverage Fourier phase correlation to estimate and correct translational offsets.

The registered (aligned) sample will then be passed to the Color Unit and Pattern Unit for pixel-level comparison, significantly reducing errors caused by image capture conditions.

**Impact:** **High** — this is the single most impactful improvement planned for v2.3.0. It addresses the fundamental assumption of pixel-level correspondence and will substantially improve measurement accuracy for real-world image pairs.

*This appendix is provided for transparency. The limitations listed above do not affect the validity of results when the system is used within its documented operating assumptions. Users are encouraged to ensure consistent image capture conditions until Image Registration is available in v2.3.0.*



# SPECTRAMATCH

## Technical Datasheet

*Version 2.2.3*

---

### Author

**Abdelbary Algamel**

[aalgamel23@posta.pau.edu.tr](mailto:aalgamel23@posta.pau.edu.tr)

### Supervisor

**Dr. Adem Ükte**

Department of Electrical and Electronic Engineering  
Pamukkale University, Denizli, Turkey

---

This document is part of the SpectraMatch v2.2.3 distribution.  
All content is derived from the project source code and reflects  
the actual capabilities of the system at the time of publication.