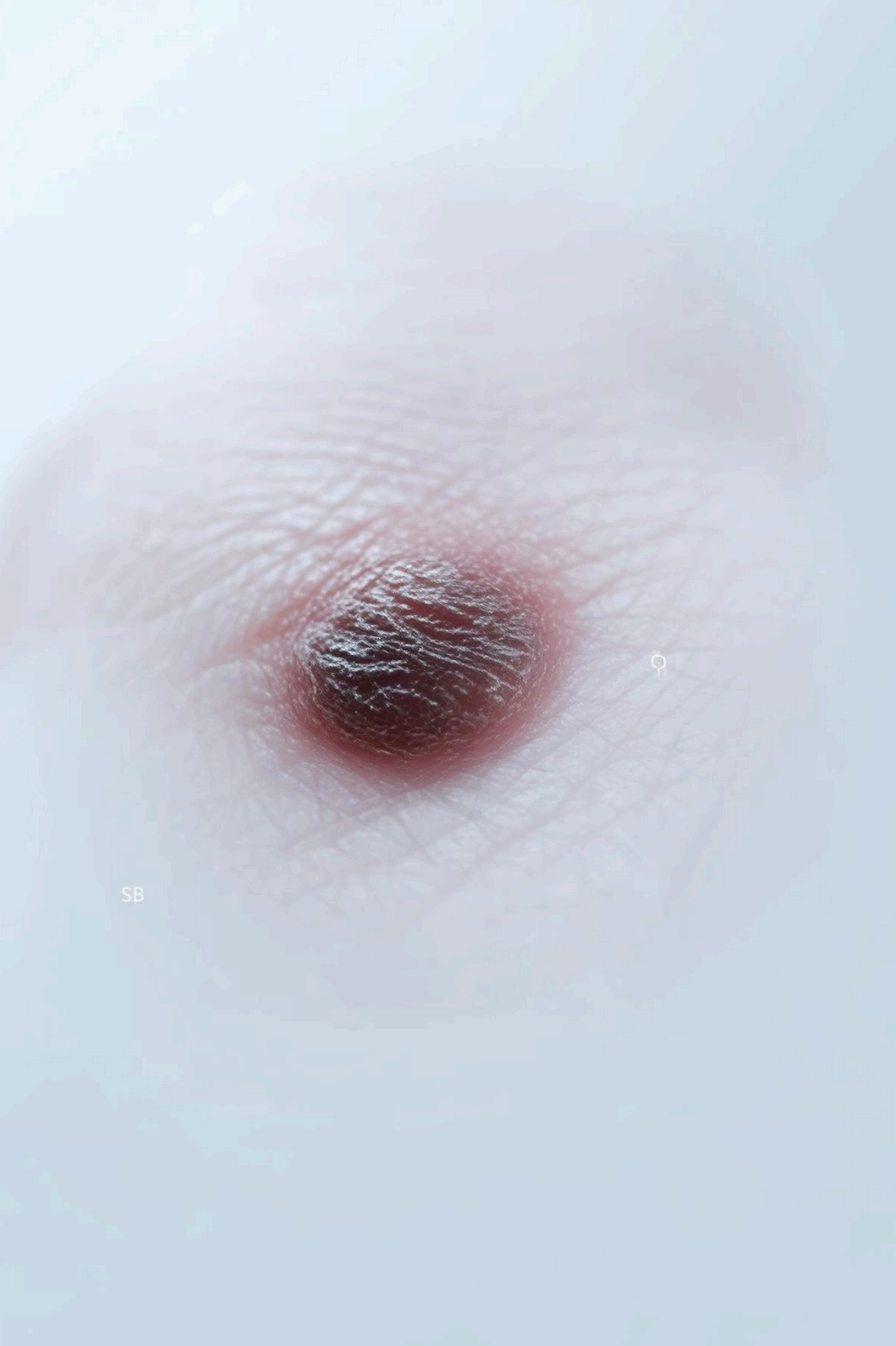
Skin Lesion Analysis

An automated approach to analyzing dermoscopic images for early skin cancer detection, developed by Umut Celik as part of CSC 741 - Digital Image Processing Midterm Project.

This presentation outlines a comprehensive image processing pipeline that segments skin lesions from dermoscopic images and extracts visual features using fundamental digital image processing techniques.

by UMUT ÇELIK



Introduction

The Challenge

Automated analysis of dermoscopic images is crucial for early skin cancer detection, but these images present challenges including varying illumination, hair, diverse skin tones, and subtle lesion characteristics.

Project Goal

To develop an image processing pipeline that segments skin lesions from dermoscopic images and extracts a comprehensive set of visual features from the segmented lesions.

Dataset

The project uses the SIIM-ISIC Melanoma Classification dataset from Kaggle, which contains dermoscopic images in JPEG format along with associated metadata.

Image Processing Pipeline

Image Loading & Conversion

Load JPEG images and convert from BGR to RGB

Preprocessing

Convert to grayscale and remove hair artifacts

Color Space Transformation

Convert to HSV color space

Segmentation

Isolate the lesion area

Feature Extraction

Extract meaningful data from the segmented lesion

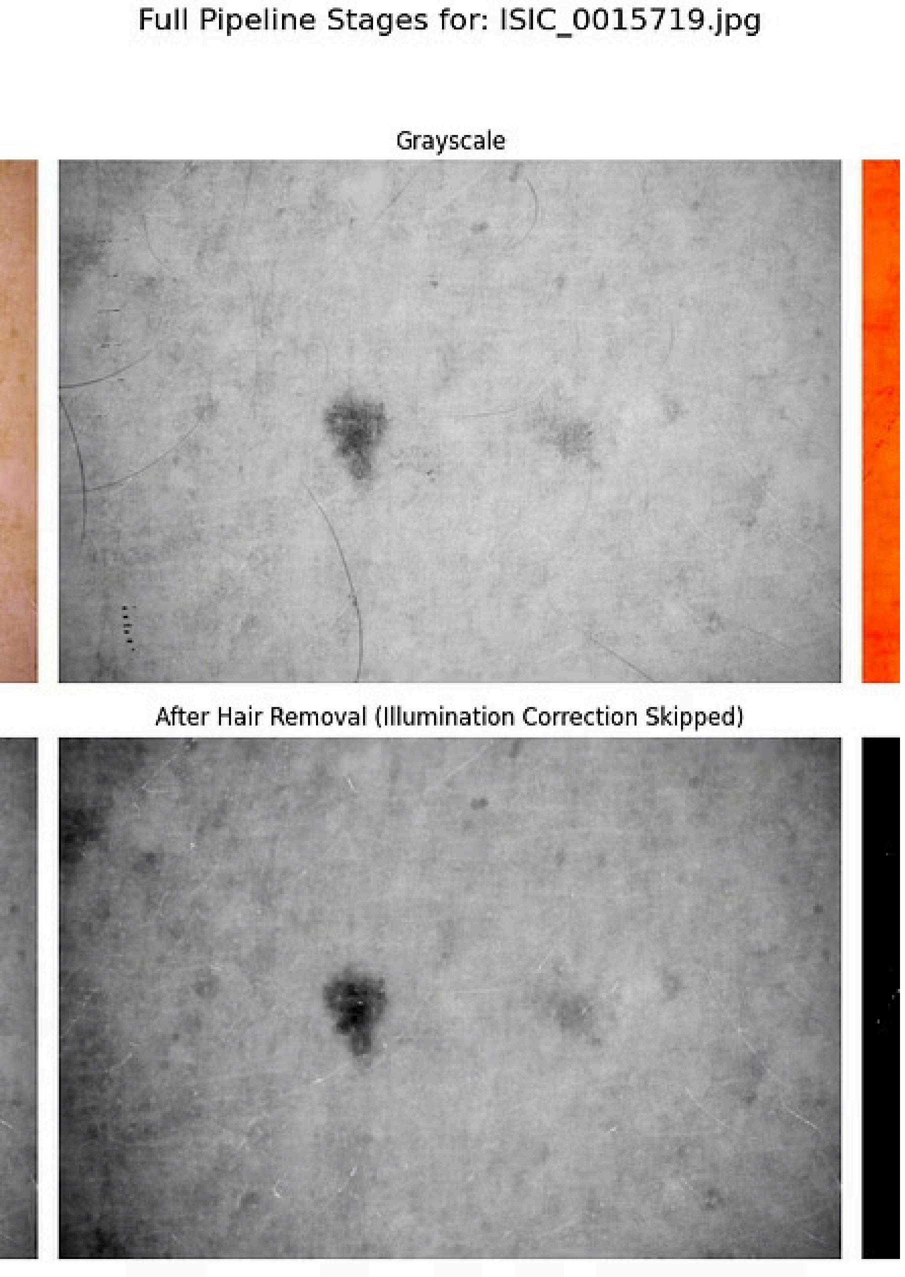
This multi-step approach processes images systematically to extract meaningful data for analysis. Each step builds upon the previous one, creating a comprehensive pipeline for skin lesion analysis.

Image Loading & Initial Conversion

The first step in our pipeline involves loading the dermoscopic images from the JPEG files provided in the dataset. This is a crucial foundation for all subsequent processing steps.

Once loaded, we perform an initial color space conversion from BGR (the OpenCV default format) to RGB. This conversion ensures consistency in processing and display throughout the pipeline.

This step is implemented in the **src/data_loader.py** file, which handles all image and metadata loading operations for the project.



Preprocessing

Grayscale Conversion

Convert RGB images to grayscale for intensity-based operations using the rgb_to_grayscale function in src/color_utils.py. This simplifies subsequent processing steps that rely on intensity values.

Hair Removal

Apply Morphological Black-Hat filtering to identify hair structures, followed by inpainting to remove them. This reduces noise and artifacts caused by hair that could interfere with accurate lesion segmentation.

Illumination Correction

Use morphological opening to estimate and correct non-uniform background illumination. This step is implemented in correct_illumination in src/preprocessing.py, though it may be skipped in some display pipelines.

Color Space Transformation

RGB to HSV Conversion

We convert RGB images to Hue, Saturation, Value (HSV) color space using the rgb_to_hsv function in src/color_utils.py. HSV is more intuitive for color-based feature extraction:

- H (Hue): Represents the dominant color
- S (Saturation): Indicates the purity/intensity of color
- V (Value): Measures brightness

Ida (Darkness) Channel

We calculate the Ida channel defined as max(R,G,B) - min(R,G,B) per pixel using the calculate_ida_channel function in src/color_utils.py. This channel:

- Represents color spread or darkness in the image
- Is used for darkness-based feature extraction (f19-f28)
- Provides additional information not captured by standard color spaces

Lesion Segmentation

Apply Otsu's Thresholding

We apply Otsu's thresholding to the preprocessed (hair-removed) grayscale image using the otsu_threshold function in src/segmentation.py. This method automatically determines an optimal threshold value to separate lesion pixels from the background.

Refine the Mask with Morphological Operations

We improve the quality and contiguity of the mask using morphological operations:

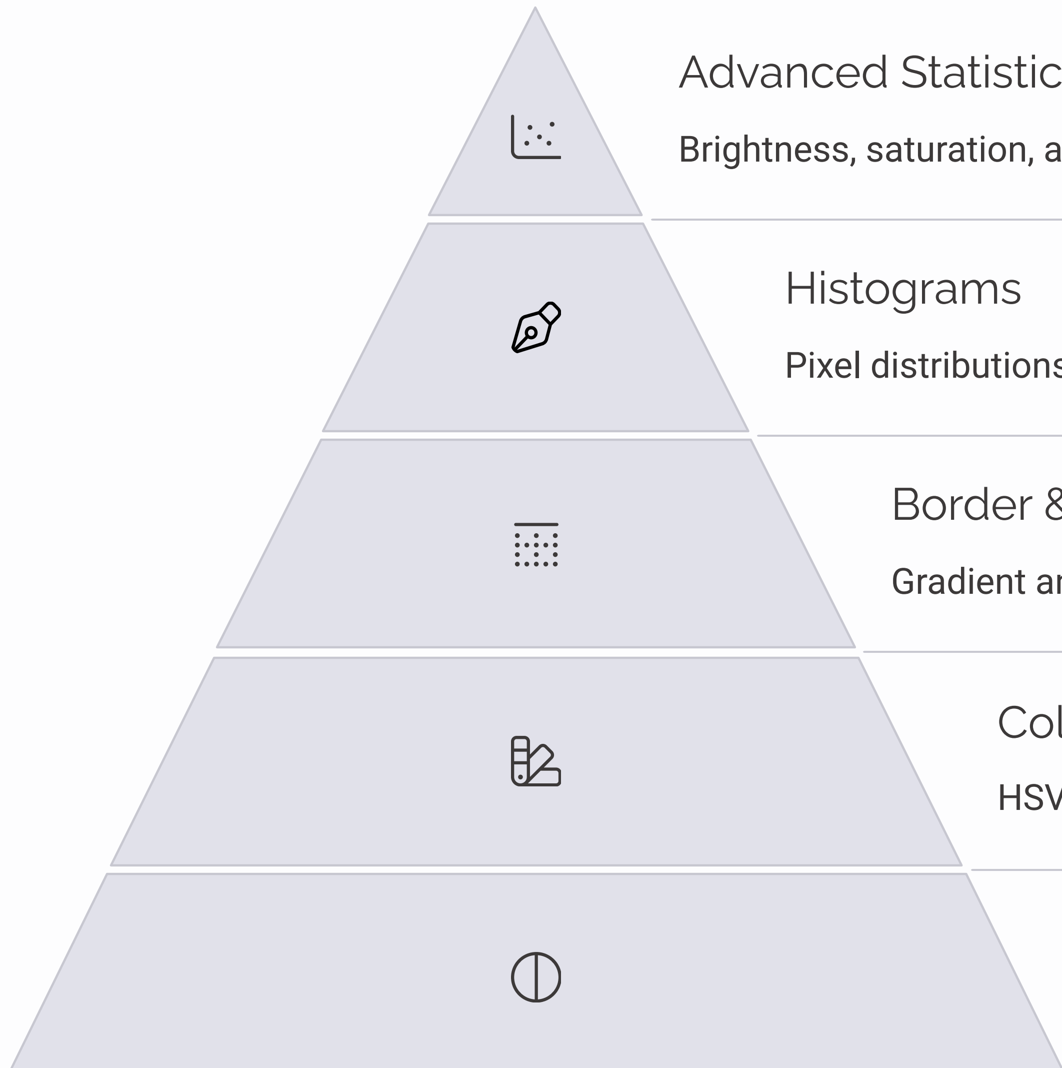
- Opening (erosion followed by dilation) removes small noise/objects
- Closing (dilation followed by erosion) fills small holes within the lesion

These operations are implemented in src/morphology.py and utilized by apply_threshold in src/segmentation.py.

Generate Final Binary Mask

The result is a binary mask that accurately isolates the lesion area from the surrounding skin, providing a foundation for feature extraction in the next step of the pipeline.

Feature Extraction Overview



Advanced Statistical Features

Brightness, saturation, and darkness metrics (f1-f28)

Histograms

Pixel distributions across multiple channels

Border & Texture Features

Gradient and morphological transform metrics

Color Features

HSV channel statistics

Intensity Features

Grayscale image statistics

Feature extraction is performed on the segmented lesion area defined by the refined mask. We extract multiple categories of features, building from basic intensity metrics to advanced statistical features that capture subtle characteristics of the lesion.

Intensity Features

4

Key Metrics

Basic statistical measures extracted from the grayscale image

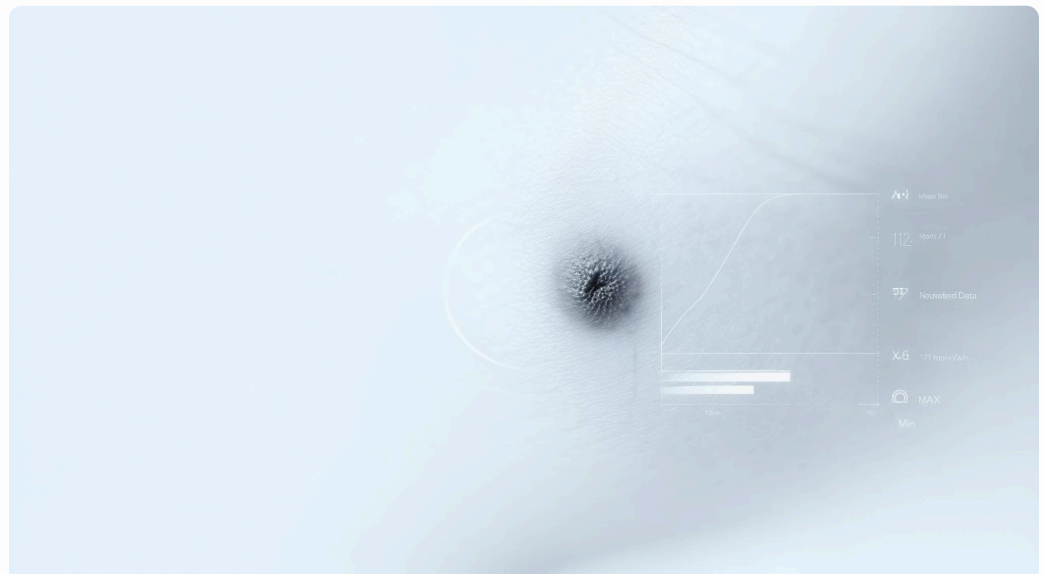
1

Module

Implemented in calculate_intensity_stats in src/feature_extraction.py

Intensity features provide fundamental information about the brightness distribution within the segmented lesion. These features are extracted from the grayscale version of the image and include:

- Mean pixel intensity - average brightness of the lesion
- Standard deviation of pixel intensities - measure of intensity variation
- Minimum pixel intensity - darkest point in the lesion
- Maximum pixel intensity - brightest point in the lesion



Color Features (HSV)

Hue (H)

Mean Hue (using circular mean calculation for angular data) and Standard deviation of Hue. These metrics capture the dominant color and color variation in the lesion.

Saturation (S)

Mean Saturation and Standard deviation of Saturation. These metrics measure the purity or intensity of colors in the lesion, which can be indicative of certain lesion types.

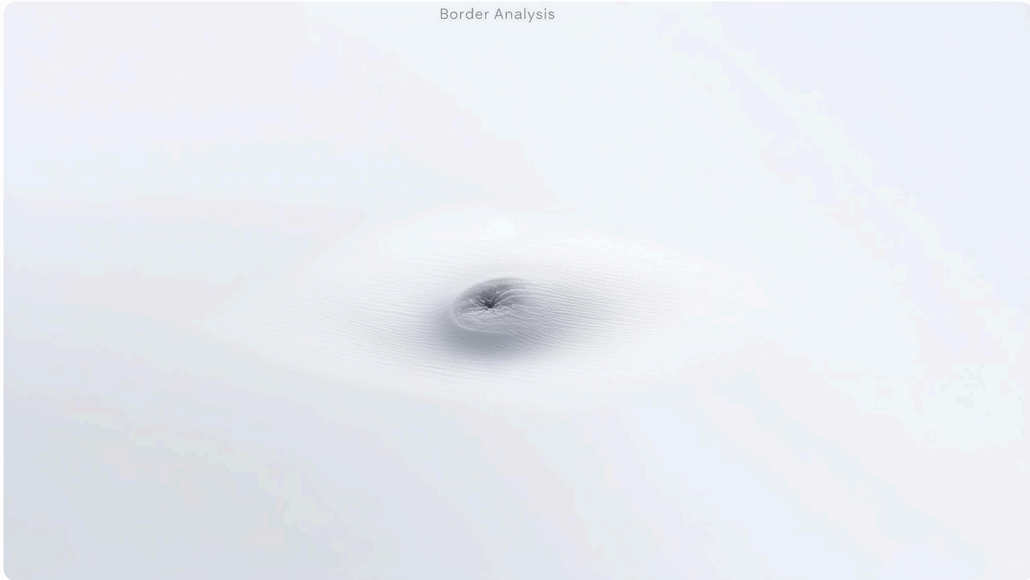
Value (V)

Mean Value (Brightness) and Standard deviation of Value. These metrics quantify the overall brightness and brightness variation across the lesion.

All HSV color features are calculated using the calculate_hsv_stats function in src/feature_extraction.py. These features provide important information about the color characteristics of the lesion, which can be crucial for distinguishing between different types of skin conditions.

Border Features

Border features are calculated from the grayscale image and mask using morphological gradient operations at the lesion border. These features are implemented in the calculate_gradient_features function in src/border_texture_utils.py.



The extracted border features include:

- Mean of border gradient values - average rate of intensity change at the border
- Standard Deviation of border gradient values - variation in border transition sharpness
- Maximum border gradient value - sharpest transition point
- Border Irregularity - ratio of border pixels to mask area, indicating complexity of the lesion shape

Border features are particularly important as irregular borders can be indicative of malignant lesions. The morphological gradient helps quantify how abruptly the lesion transitions to surrounding skin.

Texture Features

Top-Hat Transform

Highlights bright details within the lesion. We calculate the mean and standard deviation of Top-Hat transform values to quantify bright textural elements.

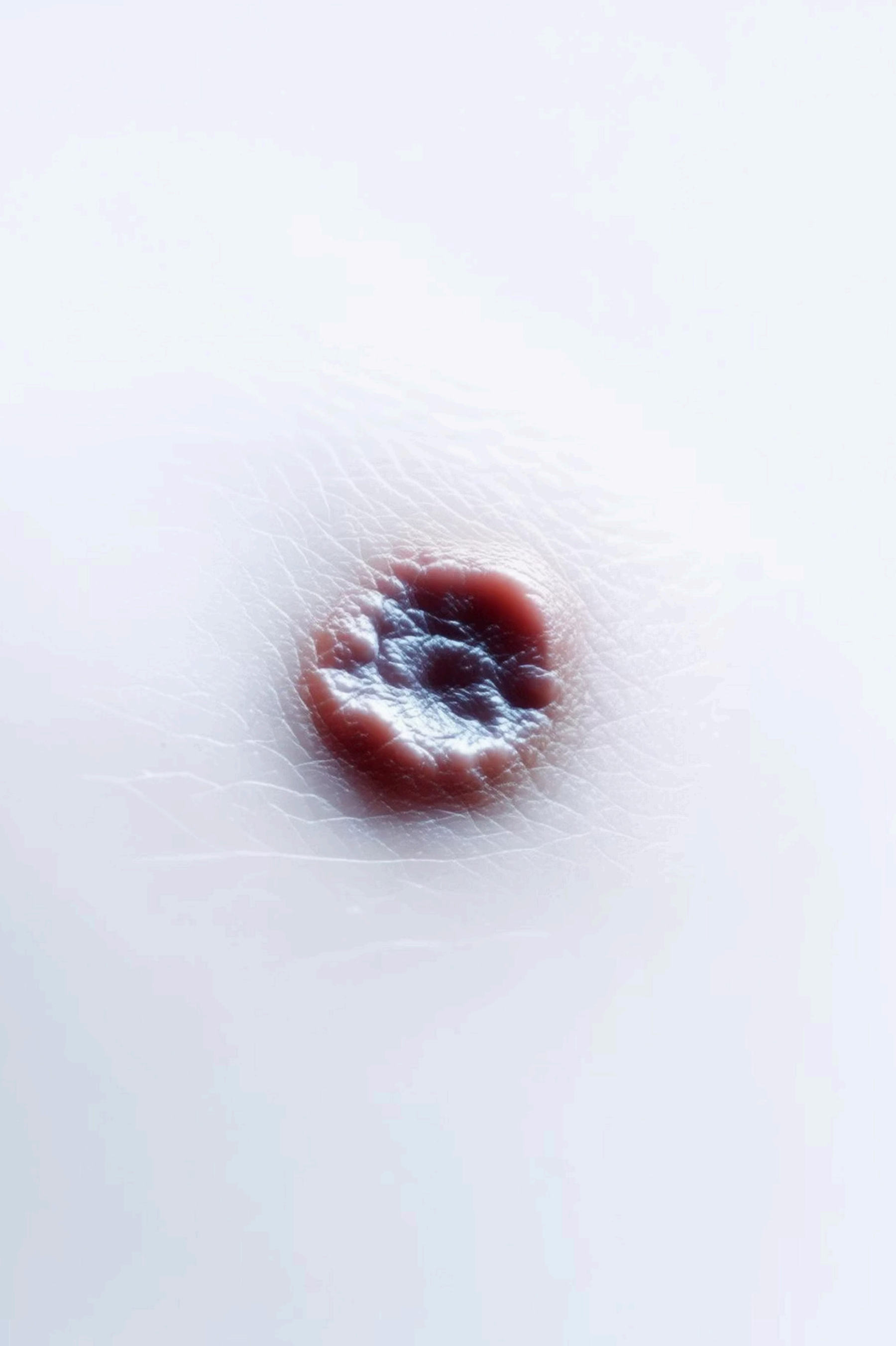
Bottom-Hat Transform

Highlights dark details within the lesion. We calculate the mean and standard deviation of Bottom-Hat transform values to quantify dark textural elements.

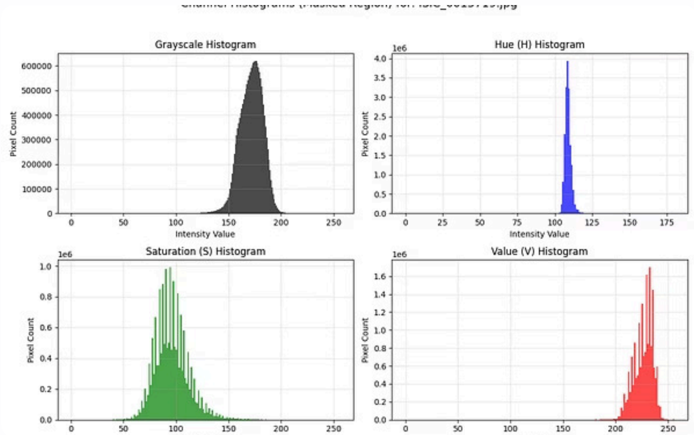
Texture Contrast Index

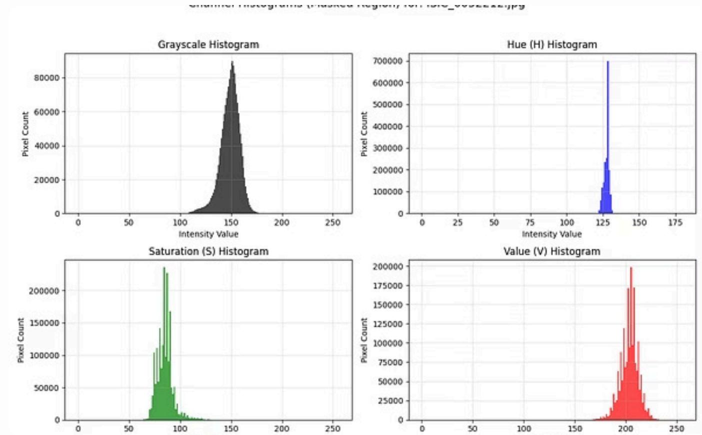
Calculated as the sum of Top-Hat and Bottom-Hat means, this provides an overall measure of textural contrast within the lesion.

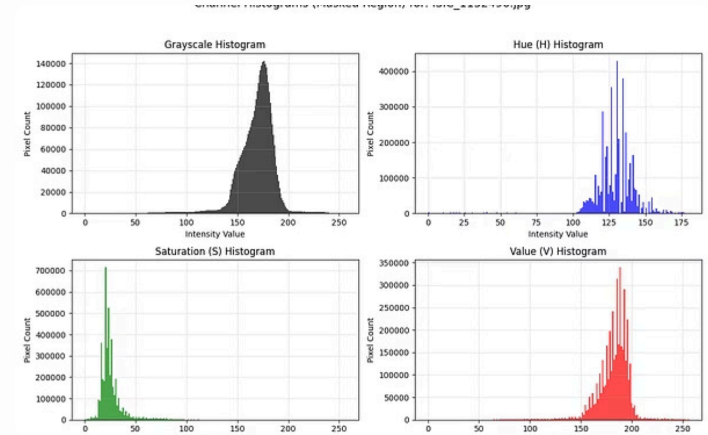
All texture features are calculated using the calculate_texture_features function in src/border_texture_utils.py. These features capture important information about the internal structure and heterogeneity of the lesion, which can be indicative of certain skin conditions.



Histogram Features







Histograms represent the pixel intensity distributions within the masked lesion area for multiple channels. These are calculated using the calc_all_histograms function in src/histogram_utils.py and include:

- Grayscale channel histogram - overall intensity distribution
- Hue (H) channel histogram - color distribution
- Saturation (S) channel histogram - color purity distribution
- Value (V) channel histogram - brightness distribution
- Ida (Darkness) channel histogram - color spread distribution

These histograms provide valuable insights into the distribution patterns of various visual properties within the lesion.

Advanced Statistical Features (f1-f28)

Brightness Features (f1-f9)

Extracted from the Value (V) channel of the HSV color space:

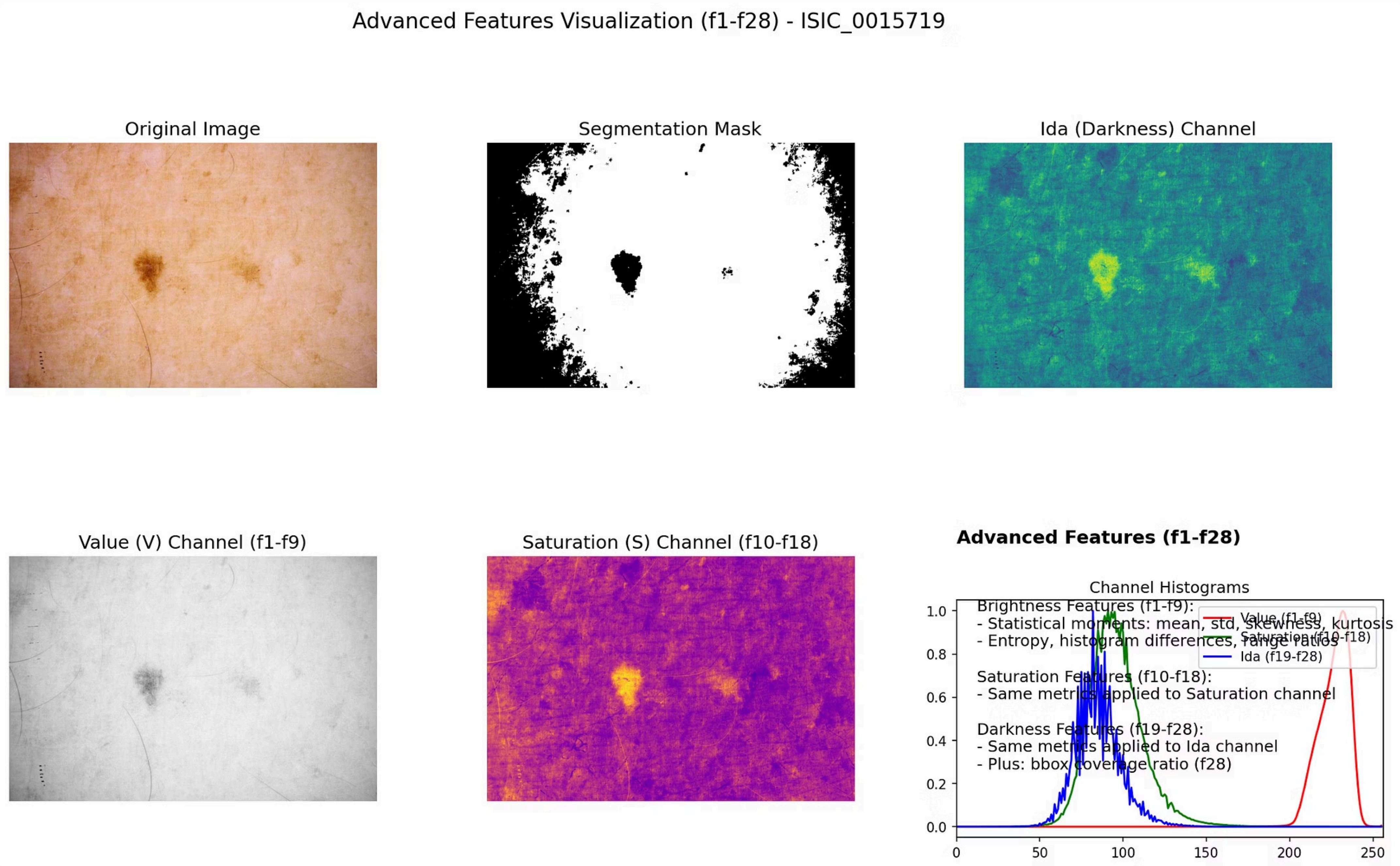
- Basic statistics: Mean (f1), Standard Deviation (f2)
- Higher-order moments: Skewness (f3), Kurtosis (f4)
- Information theory: Entropy (f5)
- Histogram analysis: Average differences (f6), Sum of 10 largest differences (f7)
- Range ratios: High-to-mid range (f8), Mid-to-low range (f9)

Saturation Features (f10-f18)

Similar statistical metrics applied to the Saturation (S) channel:

- Basic statistics: Mean (f10), Standard Deviation (f11)
- Higher-order moments: Skewness (f12), Kurtosis (f13)
- Information theory: Entropy (f14)
- Histogram analysis: Average differences (f15), Sum of 10 largest differences (f16)
- Range ratios: High-to-mid range (f17), Mid-to-low range (f18)

These advanced features are implemented in calculate_brightness_features and calculate_saturation_features functions in src/feature_extraction.py.

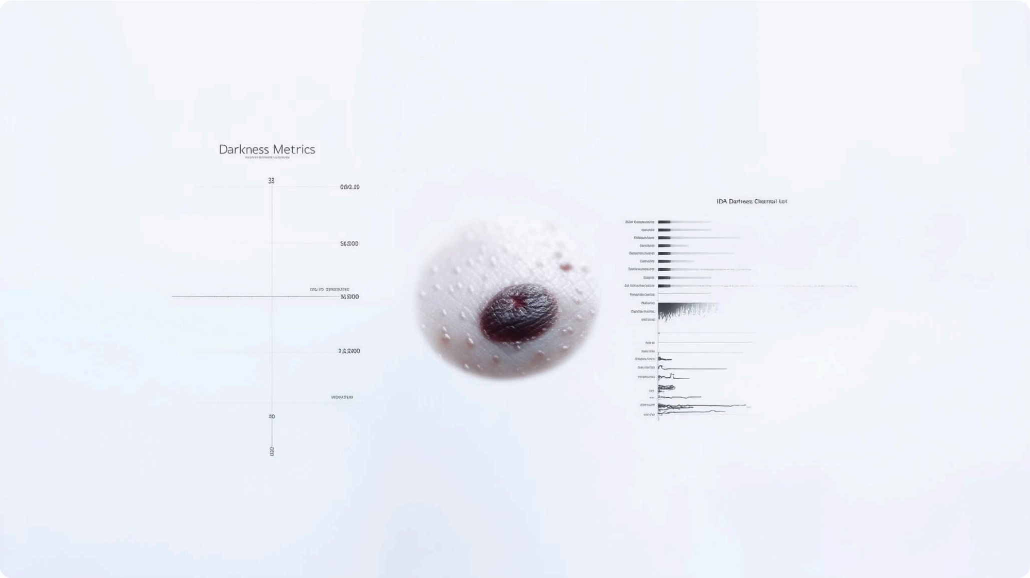


Darkness Features (f19-f28)

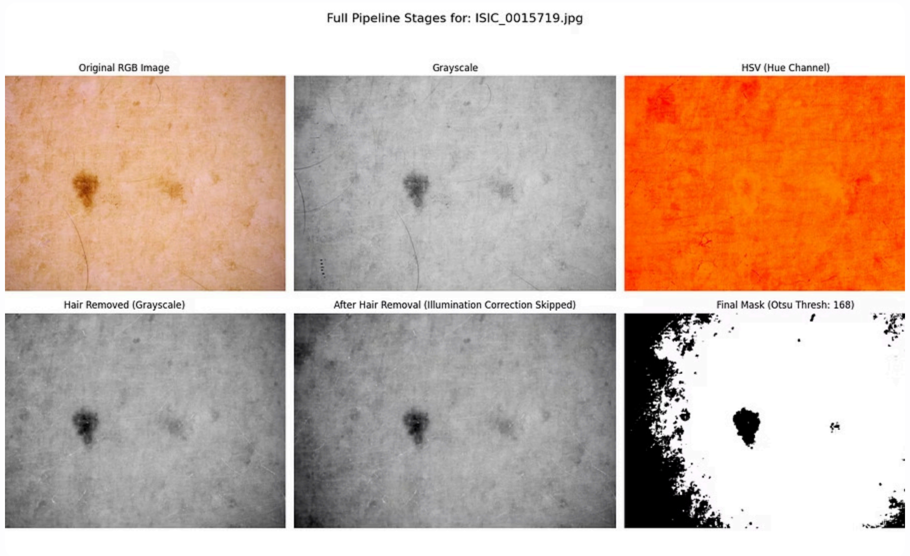
Darkness features are extracted from the Ida channel, which is calculated as max(R,G,B) - min(R,G,B) per pixel. These features include:

- Basic statistics: Mean (f19), Standard Deviation (f20)
- Higher-order moments: Skewness (f21), Kurtosis (f22)
- Information theory: Entropy (f23)
- Histogram analysis: Average differences (f24), Sum of 10 largest differences (f25)
- Range ratios: High-to-mid range (f26), Mid-to-low range (f27)
- Additional feature: Bounding box coverage (f28)

These features are implemented in the calculate_darkness_features function in src/feature_extraction.py.

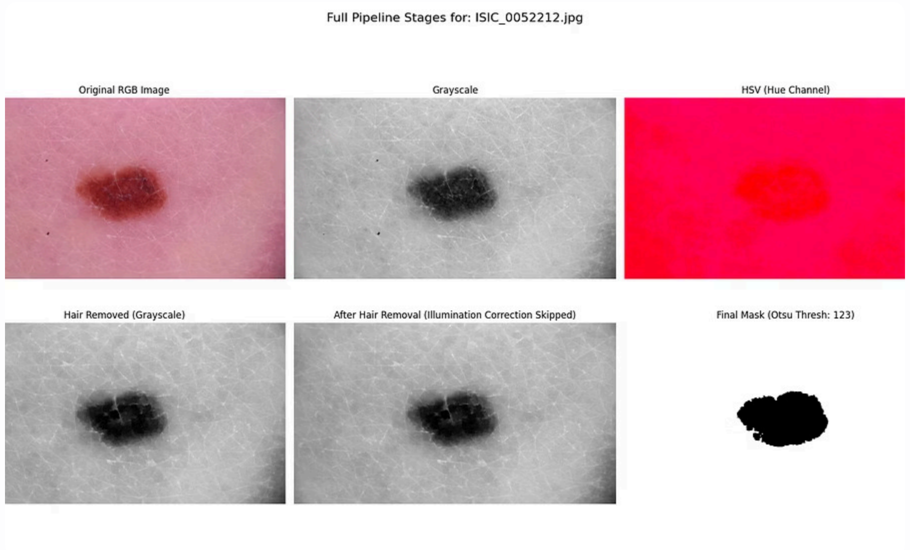


Results & Visualizations: Pipeline Stages



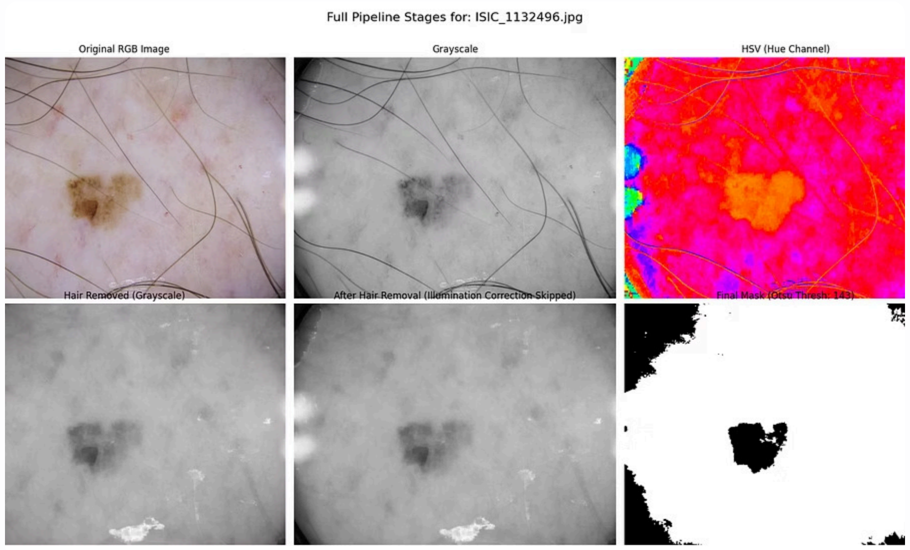
Pipeline Stages

This visualization illustrates key steps in processing an image (ISIC_0015719.jpg), showing the original RGB image, grayscale conversion, HSV (Hue Channel), hair removal process, and the final binary mask from Otsu's thresholding.



Additional Example

Another example (ISIC_0052212.jpg) showing the same pipeline stages. Note how the segmentation adapts to the different lesion characteristics in this image compared to the previous example.



Challenging Case

A more challenging example (ISIC_1132496.jpg) demonstrating how the pipeline handles a lesion with different visual properties, including color and border characteristics.

Results & Visualizations: Feature Summaries

```
Intensity Features for ISIC_0015719.jpg
========================================

Grayscale Intensity:
  Mean: 186.8315
  Std: 10.0121

HSV Channels:
  Hue (H):
    Mean: 0.0000
    Std: 0.0000
  Saturation (S):
    Mean: 0.0000
    Std: 0.0000
  Value (V):
    Mean: 0.0000
    Std: 0.0000
```

```
Intensity Features for ISIC_0052212.jpg
========================================

Grayscale Intensity:
  Mean: 158.0345
  Std: 9.1445

HSV Channels:
  Hue (H):
    Mean: 0.0000
    Std: 0.0000
  Saturation (S):
    Mean: 0.0000
    Std: 0.0000
  Value (V):
    Mean: 0.0000
    Std: 0.0000
```

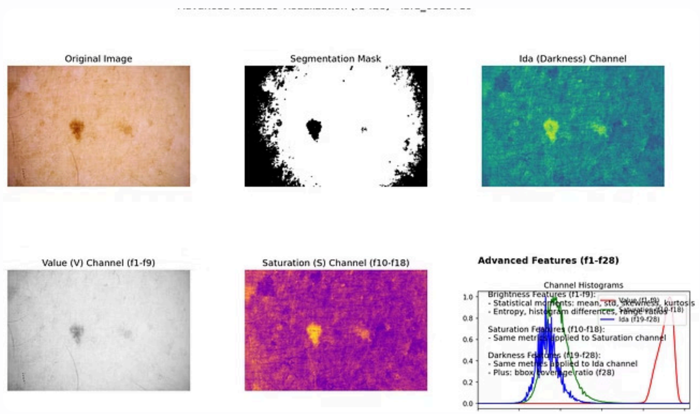
```
Intensity Features for ISIC_1132496.jpg
========================================

Grayscale Intensity:
  Mean: 170.5495
  Std: 17.2539

HSV Channels:
  Hue (H):
    Mean: 0.0000
    Std: 0.0000
  Saturation (S):
    Mean: 0.0000
    Std: 0.0000
  Value (V):
    Mean: 0.0000
    Std: 0.0000
```

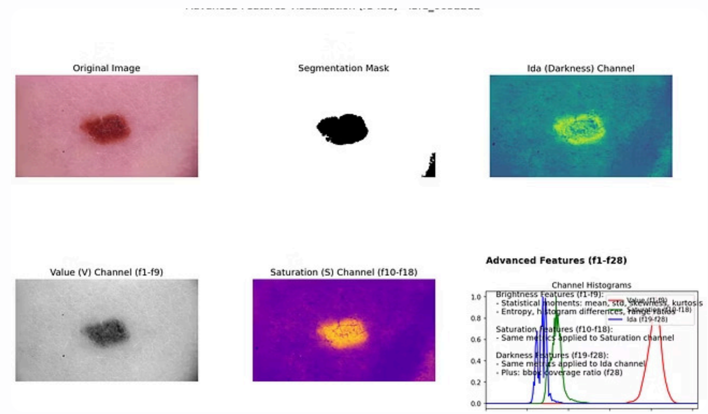
These visualizations provide textual and visual representations of extracted intensity and color features for three different lesion images. Each summary includes:

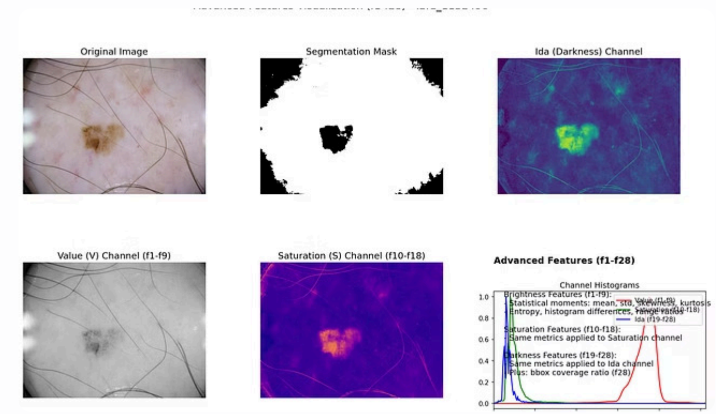
- Intensity statistics from the grayscale image (mean, standard deviation, min, max)
- HSV color statistics (mean and standard deviation for each channel)
- Visual representation of the original image and segmentation mask

These feature summaries allow for quick comparison between different lesions and highlight the variability in visual characteristics that the pipeline can capture and quantify.

Results & Visualizations: Advanced Features







These visualizations showcase the advanced features (f1-f28) implementation, including the Ida channel analysis. Each visualization displays:

- Value (V), Saturation (S), and Ida channels of the original image
- Segmentation mask applied to each channel
- Corresponding histograms showing the distribution of values within the lesion

These advanced feature visualizations provide deeper insights into the color and intensity characteristics of the lesions, which can be crucial for distinguishing between different types of skin conditions.

Code Structure & Key Modules

Core Processing Modules

- color_utils.py: Color space conversions, Ida channel calculation
- preprocessing.py: Hair removal, illumination correction
- segmentation.py: Thresholding methods, mask application
- morphology.py: Morphological operations (opening, closing)
- histogram_utils.py: Masked histogram calculations, advanced statistical functions

Feature Extraction & Display

- feature_extraction.py: Core logic for calculating features (intensity, color, advanced f1-f28)
- border_texture_utils.py: Calculates border and texture specific features
- data_loader.py: Handles image and metadata loading
- batch_processor.py: Enables processing of multiple images
- Various display scripts: Visualize different pipeline stages and extracted features

The project is organized into several Python modules within the src/ directory, with a focus on modularity and reusability. This structure allows for easy testing and development of individual components while ensuring they integrate seamlessly into the complete pipeline.

Challenges & Learnings

Segmentation Accuracy

Achieving perfect lesion segmentation proved challenging due to image variability including hair, bubbles, skin lines, and low contrast. Otsu's method provided a good baseline, and morphological cleanup helped significantly improve results.

Feature Relevance

A key learning was understanding how different features (intensity, color, texture, border) capture various aspects of lesion appearance. This knowledge is crucial for developing effective diagnostic tools.

Pipeline Integration

Ensuring each step correctly feeds into the next and maintaining consistent data formats throughout the pipeline required careful planning and implementation.

Modularity

Structuring the code into reusable functions and modules (e.g., for color conversion, preprocessing, feature calculation) was key for development and testing.

Achievements

End-to-End Pipeline

Successfully implemented a complete image processing pipeline for skin lesion analysis, from loading images to extracting comprehensive features.

DIP Module Development

Developed robust modules for key Digital Image Processing tasks: preprocessing, segmentation, and feature extraction.

Comprehensive Feature Set

Extracted a wide range of features including intensity, color, border, texture, and advanced statistical features (f1-f28).

Visualization Tools

Created comprehensive visualization tools to inspect pipeline stages and results, facilitating analysis and presentation.

Batch Processing

Implemented capabilities for analyzing multiple images, enabling efficient processing of larger datasets.

Future Work

Advanced Segmentation

Explore more robust techniques like active contours, watershed, and ML-based segmentation

Feature Analysis

Evaluate the discriminative power of each feature for melanoma detection

Machine Learning Integration

Use extracted features to train classification models for lesion malignancy prediction

Quantitative Evaluation

Perform evaluation of segmentation accuracy with ground truth masks

5

Expanded Dataset Processing

Analyze larger subsets to generate comprehensive feature statistics

While we've successfully implemented the PRD Feature Set (f1-f28) including the "Ida (Darkness)" channel and advanced statistical features, there are several directions for future development to enhance the system's capabilities and clinical utility.

Thank You & Questions?

Thank you for your attention to this presentation on Skin Lesion Analysis. The project demonstrates how fundamental digital image processing techniques can be applied to address real-world medical imaging challenges.

The implemented pipeline successfully segments skin lesions from dermoscopic images and extracts a comprehensive set of visual features that could potentially aid in early skin cancer detection.

For more information or to discuss this project further, please contact:

Umut Celik
Email: **umut.celik@cix.csi.cuny.edu**