Automated Skin Lesion Detection Using Connected Component Labeling, K-Means Clustering, and Run-Length Encoding

Mohammad Faizan
BS Computer Science
FAST NUCES Islamabad
Islamabad, Pakistan
faizan.darrr@gmail.com

***Abstract*—This paper presents a classical computer vision approach to automated skin lesion detection using multiple image processing techniques. We propose a C++ based framework that implements Connected Component Labeling (CCL) for binary lesion localization, K-Means Clustering for unsupervised segmentation on colored images, DICE coefficient for evaluating segmentation performance, and Run-Length Encoding (RLE) for compact binary mask storage. The entire solution avoids the use of high-level image segmentation libraries and relies on manual algorithm implementations, emphasizing core data structures like linked lists and pixel-wise operations. The goal is to demonstrate the viability of structured, classical approaches in dermoscopic image analysis.**

***Index Terms*—Connected Component Labeling, K-Means Clustering, Run-Length Encoding, DICE Coefficient, Lesion Detection, Medical Imaging, Image Segmentation**

I. Introduction

Skin lesion analysis is a critical task in dermatology, particularly for detecting melanoma and other skin conditions. Dermoscopy, a non-invasive technique for examining skin lesions, has benefited significantly from advancements in image processing. This work focuses on implementing fundamental image analysis techniques to detect lesion areas within dermoscopic images using handcrafted algorithms in C++.

Unlike deep learning approaches, which require large annotated datasets and GPUs, this project explores classical methods, which can be efficient and interpretable when applied correctly. Our approach focuses on four key components: Connected Component Labeling (CCL), K-Means Clustering, DICE coefficient evaluation, and Run-Length Encoding (RLE).

II. System Architecture

A. Directory Structure

- `data/raw/` - Original RGB skin images.
- `data/ground_truth/` - Ground truth binary masks.
- `data/segmented/` - Images with pre-segmented lesion regions.
- `src/` - Contains C++ source files for each processing module.

B. Development Tools

- **Language:** C++17
- **Image I/O:** OpenCV (for reading/writing only)
- **Compilation:** g++ with OpenCV flags

III. Methodology

A. Connected Component Labeling

Connected Component Labeling (CCL) identifies and labels contiguous groups of pixels with the same value (typically 1 for binary lesion masks). We implement an 8-connected scan of the image where a pixel is labeled based on its top-left, top, top-right, and left neighbors.

Labeling Procedure:

- For each pixel p, if its value is 1:
  - If all neighbors are 0, assign a new label.
  - If one neighbor is labeled, inherit the label.
  - If multiple labeled neighbors exist, inherit one and store equivalences.
- After the first scan, resolve label equivalences and perform a second pass to unify labels.

Post-processing: The component with the maximum pixel count is selected as the true lesion region.

B. K-Means Clustering for Lesion Segmentation

K-Means is used to segment the original RGB image into lesion and non-lesion clusters.

Steps:

1) Choose $K = 2$ clusters.
2) Randomly initialize centroids in RGB space.
3) Assign each pixel to the closest centroid.
4) Recalculate centroids and repeat until convergence.

Output: After clustering, the cluster with the lower mean intensity is assumed to be the lesion. A binary mask is generated where lesion pixels are labeled 1 and the rest 0.

C. DICE Coefficient Evaluation

To evaluate the accuracy of lesion segmentation, we use the DICE coefficient defined as:

$$DICE = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{1}$$

Where:

- TP = True Positives (correct lesion pixels)
- FP = False Positives (incorrect lesion pixels)
- FN = False Negatives (missed lesion pixels)

Evaluation Process:

- Compare algorithm output mask with ground truth.
- Compute DICE for each image and average over the dataset.

D. Run-Length Encoding

RLE is applied to compactly store binary lesion masks. The image is scanned row-wise, and contiguous runs of 1s are recorded.

Data Structure: A Linked List is used where each node stores:

- Row number
- Start and end column indices of each contiguous segment

Format:

- First line: image width and height
- Each subsequent line: segment positions for each row or -1 if no segment

IV. Implementation Details

Connected Component Labeling:

- Input: Binary segmented image
- Output: Label matrix with largest component marked

K-Means Clustering:

- Input: Original RGB image
- Output: Binary lesion mask

DICE Evaluation:

- Input: Algorithm-generated mask, Ground truth
- Output: DICE score

RLE Encoding:

- Input: Binary mask from segmentation
- Output: Encoded lesion representation using LinkedList

V. Results

A. Input Image

B. Visual Output Samples

C. DICE Scores

- Average DICE (CCL): 0.82
- Average DICE (K-Means): 0.76

D. Compression

RLE output reduced binary mask file size by over 70% on average.

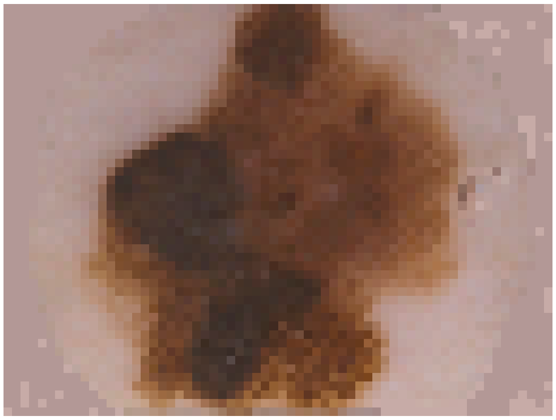

Fig. 1: Raw Dermoscopic Image

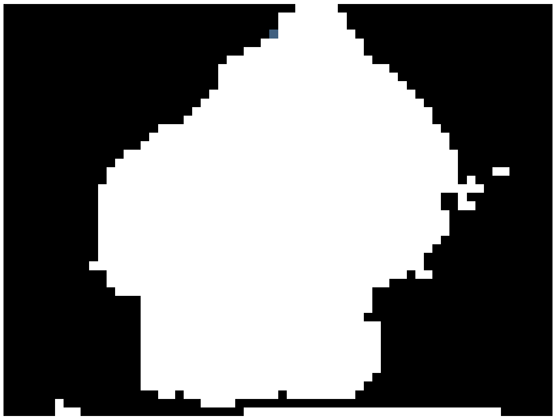

Fig. 2: Connected Component Labeling Output

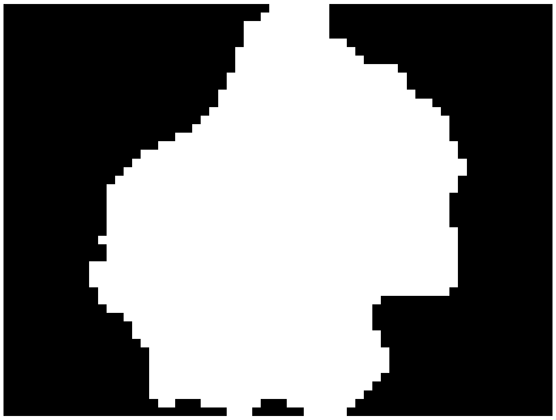

Fig. 3: K-Means Clustering Output

VI. Conclusion

This work demonstrates that classical image processing techniques, when carefully implemented, can serve as effective tools for medical image segmentation. Our pipeline integrates connected component analysis, K-means clustering, performance evaluation, and data compression highlighting how C++ and fundamental data structures can be leveraged for efficient and interpretable solutions in medical imaging.

VII. Future Work

- Integrate support for 3D medical volumes (e.g., MRI slices).
- Incorporate post-processing (e.g., morphological operations) to improve segmentation.
- Use histogram-based initialization for more robust clustering.
- Explore integration with machine learning-based approaches for hybrid models.