RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	CPS 843
Course Title	Introduction to Computer Vision
Semester/Year	F2023

Instructor	Dr. Guanghui Richard Wang

Project

Assignment Title	CPS 843 Final Project Report

Submission Date	December 16th, 2023
Due Date	December 16th, 2023

Student Name	Abdulrehman Khan	Hamza Iqbal
Student ID	500968727	500973673
Signature*	A.K	H.I

**By signing above you attest that you have contributed to this written lab report and confirm that all work you have swung the lab contributed to this lab report is your own work.*

Title: Unveiling the Amalgamation of Compressed Image Feature Extraction: A Comprehensive Exploration

Authors: Abdulrehman Khan and Hamza Iqbal

Affiliation: Introduction to Computer Vision (CPS843)

Abstract:
To navigate the vast image data landscape requires a nuanced approach. This report goes into the complexities of extracting features from compressed images. An intersection exists between compression and feature extraction techniques; this is a joint effort to get it done right. This article uses deep learning, machine learning and computer vision applications to explore the relationship between techniques. It offers insight on how these sciences can be combined in reality. The report details how important edge detection is in compressed images with regard to digital image processing and computer vision.

Link to Source Code: https://github.com/ark22oct/Computer-Vision/tree/main/Project

Introduction:
With data on the rise, this has given us a task we cannot avoid--the processing and understanding of vast image datasets. The objective of this report is to provide some information on compressed image feature extraction, which falls within the delicate realms of edge detection. The article is then arranged around several algorithms of edge detection, such as adaptive, canny and that based on the color sum function or an energy-function method, to describe how this technology has become one indicator for measuring digital image components. A foundation stone in the retrieval of information from images, edge detection is therefore an important component with wide applications.

Moreover, the extraction of features from compressed images is a key step in describing image and video content. What drives this exploration is the need to handle large image datasets efficiently, and one asks if low-level feature extraction can be performed directly from these compressed images. This strategic approach aims to expedite the characterization process, facilitating a quicker analysis of the given content [1].

Technical Breakdown:

Image Compression and Feature Extraction:
The collaborative amalgamation of image compression and feature extraction serves as a strategic pillar for managing voluminous data effectively. Image compression, as a means of reducing storage and transmission costs, involves minimizing byte size without sacrificing image quality [2]. Feature extraction, an integral facet of the dimensionality reduction process, entails breaking down raw data into smaller, manageable groups (variables). These variables are meticulously selected and combined into features, presenting a more efficient and accurate description of the original dataset [3].

Traditional methods used specialized algorithms for feature detection. Then, with deep learning (DL), image and video analysis changed dramatically once again. With this novel method there is no need to do feature extraction, making it a landmark development in the field of computer vision applications. Whether via initial deep network layers or traditional extraction techniques, the report shows that effective representation of image features is universally important [3].

Edge Detection:
This report looks into the vital role of edge detection in digital image processing, analysis and recognition. Testing traditional edge detection algorithms (Roberts, Sobel, Prewitt) shows that they cannot handle the subtle details of a real-world image. Also inadequately sensitivity to noise are canny and fuzzy techniques.

The famous Canny operator is introduced by starting with the three standard deviations that make it up--fine S/N, single edge response and optimal location [5]. The report takes Canny operators as examples, overviews recent enhancement plans put forward by academics--an improved version of non-sampling contourlet transforms (ACC), maximum posterior probability between classes and an adaptive algorithm for the detection of edges (ACAE) [5].

Experiments and Analysis:
This section provides a comprehensive insight into the proposed methods, featuring detailed outcomes of the compression process and elucidating the benefits of the integrated approach.

Image Compression:
A representative example demonstrates the performance obtained by compressing a true-color image with set partitioning in hierarchical trees - 3D ('spiht_3D') encoding method [6]. A meticulous code snippet is presented, detailing the compression process and providing quantitative measures of compression ratio and bits per pixel:

```
load mask;
X = imread('wpeppers.jpg');
[cratio,bpp] =
wcompress('c',X,'wpeppers.wtc','spiht','maxloop',12)
cratio = 1.6549
bpp = 0.3972
Xc = wcompress('u','wpeppers.wtc');
delete('wpeppers.wtc')
Display the original and compressed images.
subplot(1,2,1)
image(X)
title('Original image')
axis square
subplot(1,2,2)
image(Xc)
title('Compressed image')
axis square
```

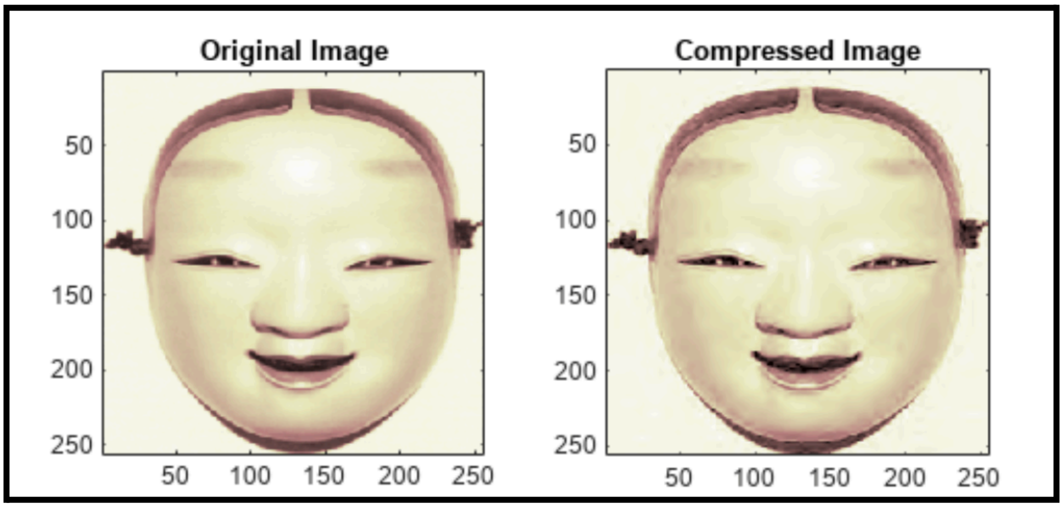

Figure 1: illustrates a comparative analysis between the unaltered original image and its compressed counterpart. This visual representation encapsulates the transformative impact of the applied compression methodology on the image, showcasing the effectiveness of the set partitioning in hierarchical trees - 3D ('spiht_3D') compression method [6].

Feature Extraction:

The report features an exemplary scenario that combines feature detection, extraction, and matching, presenting an object in a cluttered scene [Refer to Figure 2].

Edge Detection:
Edge detection is a basic algorithm in image processing used to find boundaries within an image. This technique is central to many applications in computer vision, such as object recognition and image segmentation [8]. It also serves for feature extraction. Achieving this requires several edge detection algorithms, such as the Sobel operator; Canny edge detector; Prewitt operator and Roberts Cross Operator.

The Sobel operator is a convolution-based approach that employs two 3x3 kernel filters to compute the gradient of an image's intensity in both the horizontal and vertical directions. At each pixel location, the gradient magnitude is determined, and edges are marked when the magnitude exceeds a predetermined threshold [9].

In contrast, the Canny edge detector is a multi-step procedure that first smooths the picture using a Gaussian filter before computing the gradient [5]. Non-maximum suppression, followed by edge following via hysteresis, identifies sharp edges after applying two threshold values [9].

Another straightforward edge detection strategy called the Prewitt operator makes use of 3x3 kernel filters to find the gradient in both the horizontal and vertical directions. The gradients from both axes are subsequently merged to establish the total gradient strength, and edgy areas are defined according to a set threshold [8].

Last but not least, the Roberts Cross operator functions similarly to the Prewitt operator; it utilizes a pair of 2x2 convolution kernels to estimate the gradient in the horizontal and vertical directions. Although it is computationally simpler than the other methods, the Roberts Cross operator might be more susceptible to interference and less precise in particular circumstances [9].

In visual depictions, the results of these algorithms frequently take the form of a grayscale image with prominent edges delineated by bright lines against a darker background [8]. These visualizations enable experts to evaluate the effectiveness of these algorithms and base their decisions on the specific needs of the application at hand.

An example of edge detection between Canny and Prewitt methods include:

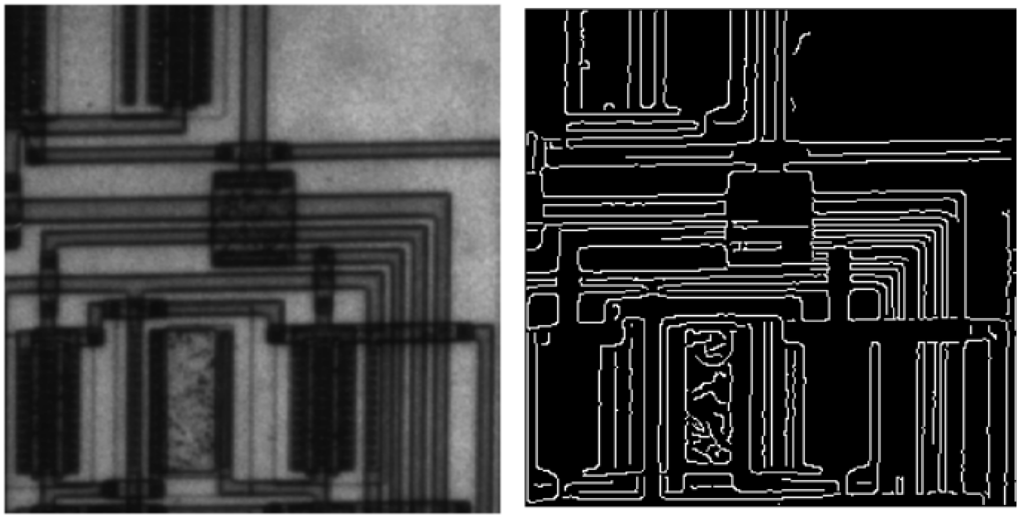

Figure 2.1: Edge detection using Canny [10].

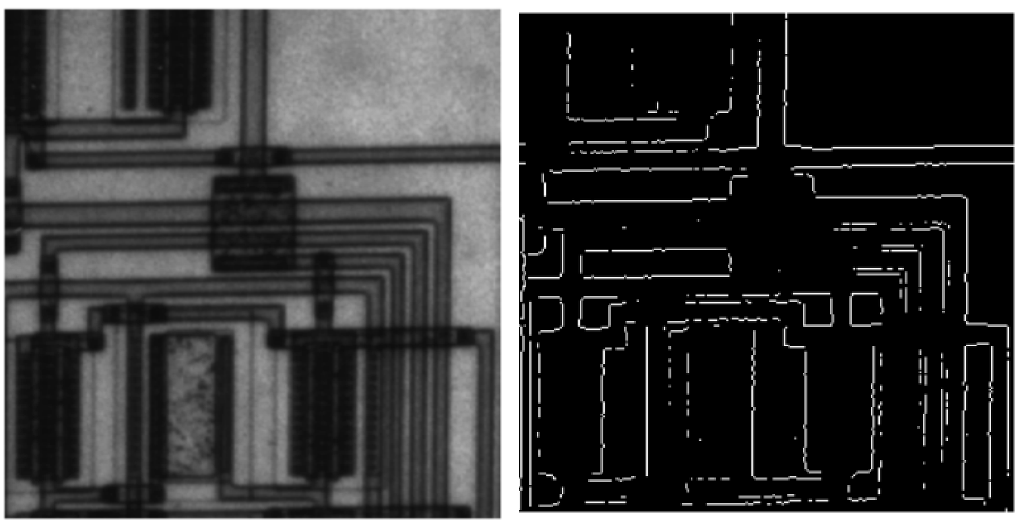

Figure 2.2: Edge detection using Prewitt [10].

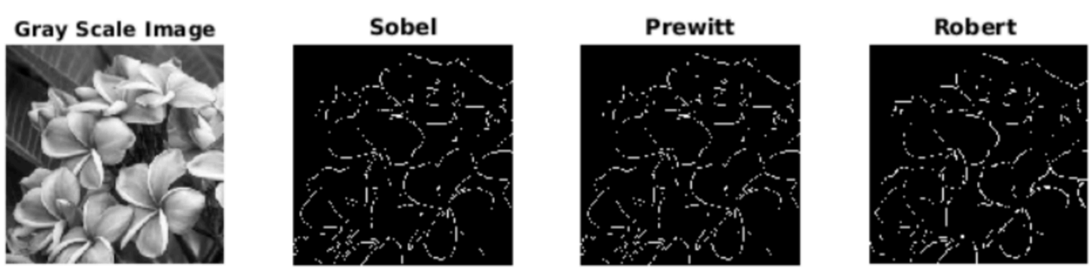

Figure 3.1: Image segmentation applied to a Gray Scale image using Sobel, Prewitt, Robert, Log, Zerocross, and Canny [7].

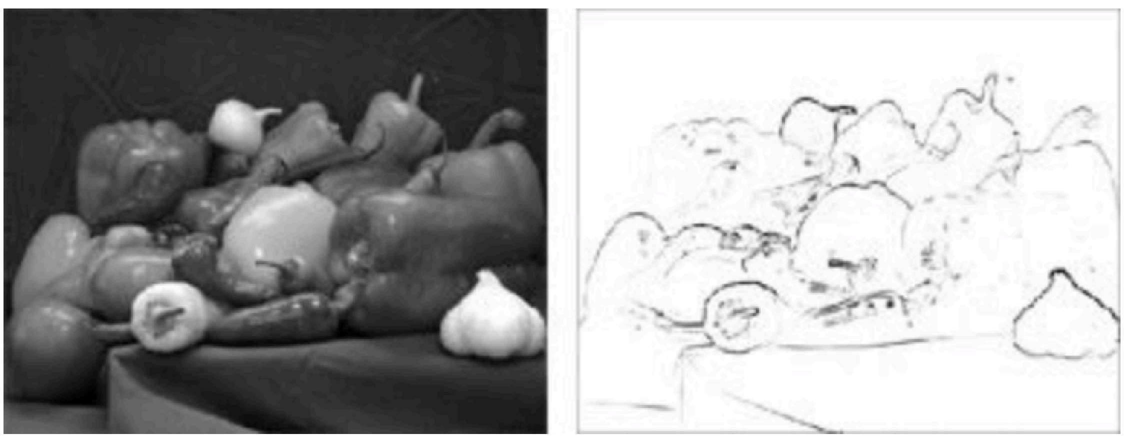

Figure 3.2: Image segmentation being applied via using a Fuzzy Logic Method [4].

Conclusion:
Therefore, this report disentangles the mysteries behind compressed image feature extraction, leaving readers with a nuanced understanding of how it works and its far-reaching applicability. With edge detection and enhancement as its starting points, the study lays down basic requirements for successful computer vision. In the process of exploration both parties participate, and jointly contribute to a comprehensive understanding.

References:

[1] B. Shen and I. K. Sethi, "," SPIE Proceedings, 1996.

[2] "Image Compression," True compression of images using wavelets - MATLAB. [Online]. Available: https://www.mathworks.com/help/wavelet/ug/wavelet-compression-for-images.html.

[3] S. Chatterjee, "What is feature extraction? feature extraction in image processing," Great Learning Blog: Free Resources what Matters to shape your Career!, 26-Jul-2022. [Online]. Available: https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/#:~:text=Feature%20extraction%20is%20a%20part,a%20large%20number%20of%20variables.

[4] "Edge detection," MATLAB & Simulink. [Online]. Available: https://www.mathworks.com/help/images/edge-detection.html.

[5] Ž. Bugarinović, L. Pajewski, A. Ristić, M. Vrtunski, M. Govedarica, and M. Borisov, "On the introduction of canny operator in an advanced imaging algorithm for real-time detection of hyperbolas in ground-penetrating radar data," MDPI, 24-Mar-2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/541.

[6] "Image Compression," True compression of images using wavelets - MATLAB. [Online]. Available: https://www.mathworks.com/help/wavelet/ug/wavelet-compression-for-images.html.

[7] GeeksforGeeks. (2023, 12). Edge Detection using In-built Function in MATLAB. GeeksforGeeks. https://www.geeksforgeeks.org/edge-detection-using-in-built-function-in-matlab/

[8] University of South Florida . (n.d.). Edge Detection. Machine learning. https://cse.usf.edu/~r1k/MachineVisionBook/MachineVision.pdf

[9] Vikram, M. (2022, August 9). Comprehensive guide to edge detection algorithms. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/08/comprehensive-guide-to-edge-detection-algorithms/

[10] MathWorks. (n.d.). Edge detection. Retrieved from https://www.mathworks.com/discovery/edge-detection.html