The Invisible Cut: CARVE Framework for Content-Aware Image Retargeting Using Dynamic Programming

Muhammad Aufar Rizqi Kusuma - 13524061
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: ksmaachmad@gmail.com, 13524061@std.stei.itb.ac.id

***Abstract*— Content-aware image retargeting seeks to resize images across heterogeneous display dimensions while preserving visually important content. Existing seam carving approaches rely predominantly on gradient-based energy functions, which detect edges well but are insensitive to texture-rich regions that carry significant perceptual importance without producing strong directional gradients. This paper proposes CARVE (Content-Aware Retargeting with Variance-based Energy), a dynamic programming framework that augments the conventional seam carving energy formulation with a local variance component to capture such regions. The composite energy function $E_{\text{CARVE}} = \alpha \cdot e_{\text{grad}} + \beta \cdot e_{\text{var}} + \gamma \cdot e_{\text{fwd}}$ integrates gradient, variance, and forward energy terms into a single unified DP recurrence, with an optional content-protection mask for user-defined regions of interest. Experiments on 15 images from the RetargetMe benchmark demonstrate that all seam-based methods outperform naive cropping on the Energy Preservation Ratio by a mean margin of 0.117, confirming that seam carving preferentially discards low-energy content. Among seam-based methods, CARVE achieves the highest structural similarity (SSIM) on images with rich local texture, validating the discriminative contribution of the variance component.**

Keywords: dynamic programming, seam carving, image retargeting, content-aware resizing, energy function.

I. INTRODUCTION

A. Background & Motivation

The rapid proliferation of digital display devices, ranging from wearable smartwatches and handheld smartphones to high-resolution desktop monitors and large-format projection screens has created an increasingly demanding requirement for visual content to adapt across wildly different aspect ratios and resolutions. A photograph composed for a widescreen display, for instance, carries spatial relationships and focal subjects that may be entirely disrupted when rendered on a portrait-oriented mobile screen. Conventional resizing strategies fail to address this challenge in a satisfying way. Uniform scaling distorts the geometric proportions of every element in the image equally, making circular objects appear elliptical and people appear unnaturally tall or squat. Cropping, while preserving local geometric fidelity, resolves the problem by discarding potentially critical visual content at the image boundary. Neither approach reasons about what is in the image or where the visually important content resides.

Content-aware image retargeting addresses this by incorporating image semantics into the resizing process. Avidan and Shamir [1] proposed seam carving, in which an optimal 8-connected pixel path is iteratively identified and removed using dynamic programming in $O(M \times N)$ time. However, their gradient-based backward energy criterion ignores the new pixel adjacencies introduced after each removal, producing progressive visual artifacts. Rubinstein et al. [2] remedied this with a forward energy criterion that selects the seam whose removal introduces the least new energy, yielding substantially cleaner results. Subsequent work showed that no single operator performs well across all image types, motivating hybrid multi-operator approaches [3] and systematic benchmarking [4].

Yet a persistent limitation remains. Both backward [1] and forward energy [2] rely solely on the image gradient, which detects edges well but is insensitive to regions of rich local texture where those areas carry significant perceptual importance without producing strong directional gradients. This motivates the CARVE (Content-Aware Retargeting with Variance-based Energy) framework, which augments the existing energy formulation with a local variance component and embeds this composite energy directly into the DP recurrence to better preserve both edge-defined and texture-rich content.

B. Problem Statement

Formally, let an input image $\mathbf{I}$ be represented as a matrix of dimensions $M \times N$, where M and N denote the number of rows and columns respectively, and each entry $I(i, j) \in \mathbb{R}^3$ represents the RGB color values of a pixel at row i and column j. The image retargeting problem seeks a transformation T such that

$$T : \mathbf{I}^{M \times N} \to \mathbf{I}'^{M \times N'}$$

where $N' < N$ for width reduction (or $N' > N$ for width expansion), while maximizing the preservation of visually important content. Without loss of generality, the subsequent formulation focuses on vertical seam removal (width reduction, $M' = M$), as horizontal seam removal follows analogously by transposing the image. A vertical seam s is defined as an ordered set of pixels spanning every row exactly once, where

$$s^x = \{s_i\}_{i=1}^{M} = \{(i, x(i))\}_{i=1}^{M}, \quad \text{subject to} \\ |x(i) - x(i-1)| \leq 1$$

where $x(i)$ denotes the column position of the seam at row i, and the connectivity constraint ensures the seam is 8-connected and monotonic. The cost of a seam is determined by a pixel energy function $e : \mathbb{R}^3 \to \mathbb{R}$, which assigns a non-negative scalar importance score to each pixel. The optimal seam s^* is then defined as the seam of minimum cumulative energy

$$s^* = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{M} e(I(i, x(i)))$$

Reducing the image width by k columns therefore requires finding and removing k such optimal seams sequentially, yielding a sequence of transformations $T_1, T_2, \ldots, T_k$ applied to the progressively narrowed image.

C. Research Objectives

This paper aims to address the identified limitations of existing seam carving approaches through the following objectives:

1) To design a multi-component energy function combined with dynamic programming recurrence (CARVE) that enables the energy function to discriminate not only edge-defined regions but also texture-rich areas that are perceptually important yet poorly captured by gradient alone.
2) To implement the CARVE framework as a fully functional image retargeting system capable of reducing and expanding images along both spatial dimensions, with an optional content-protection mask for preserving user-defined regions of interest.
3) To evaluate CARVE experimentally against naive cropping, classical backward-energy seam carving [1], and forward-energy seam carving [2] across diverse image categories, using quantitative and qualitative metrics including the Structural Similarity Index (SSIM) [5] and Energy Preservation Ratio (EPR) to assess content preservation quality.

II. THEORETICAL FRAMEWORK

A. Image Representation

A digital image is formally defined as a two-dimensional discrete function $f : \mathbb{Z}^2 \to \mathbb{R}^C$, mapping spatial coordinates (i, j) to a C-dimensional intensity value, where $i \in \{0, 1, \ldots, M-1\}$ and $j \in \{0, 1, \ldots, N-1\}$ index the rows and columns of a matrix of dimensions $M \times N$ [6]. Each discrete element of this matrix is called a pixel (contraction of picture element), representing the smallest addressable unit of visual information in the image.

Pixels and Quantization. The process of constructing a digital image from a continuous visual scene involves two successive discretization steps [6]. The first, spatial sampling, partitions the continuous spatial domain into a finite grid, determining the image resolution $M \times N$. The second, amplitude quantization, maps each continuous intensity value to the nearest representable discrete level. For a quantization depth of b bits, the intensity at each channel is represented by one of $L = 2^b$ discrete levels, typically spanning the integer range $[0, 2^b - 1]$. The standard choice of $b = 8$ yields $L = 256$ levels per channel, a range sufficient for most natural image content while remaining computationally tractable. Higher quantization ($b = 12$ or $b = 16$ bits) can increase the memory required to store the image, potentially leading to performance degradation in memory-constrained environments.

Types of Image Representation. Digital images are stored and processed under several distinct representational schemes, each suited to different application requirements [6]. *Binary images* represent each pixel as a single bit $f(i, j) \in \{0, 1\}$, where 0 typically denotes black and 1 denotes white. This representation is widely used in document scanning, optical character recognition (OCR), and simple graphics applications where only two intensity values are required. While memory-efficient, binary images lack the tonal range necessary for representing natural scenes and cannot capture intermediate gray levels.

Grayscale Images. Grayscale images extend this representation by assigning each pixel a single intensity value $f(i, j) \in \{0, 1, \ldots, L-1\}$, where L is the total number of discrete levels. For standard 8-bit grayscale images, $L = 256$, allowing for 256 distinct shades of gray ranging from black (0) to white (255).

Color Images. Color images, conversely, represent each pixel by a vector of intensity values, one for each color channel, usually $C = 3$ for the RGB (Red, Green, Blue) model. Each channel is typically quantized to 8 bits, resulting in a total of 24 bits per pixel and $16,777,216$ possible colors, as seen in standard RGB color images. This depth is sufficient to reproduce the vast majority of colors perceived by the human visual system.

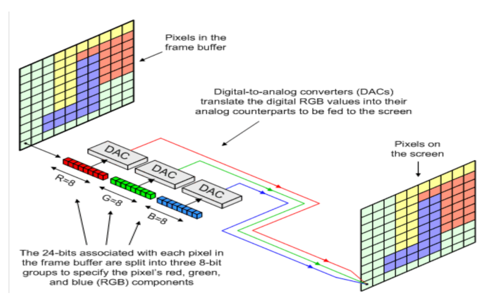

Fig. 1: RGB Color Model. Source: [7]

While standard RGB color images are sufficient to reproduce the vast majority of colors perceived by the human visual system, other color models such as HSV and CMYK are used for specific applications, each offering different trade-offs in terms of memory usage, computational complexity, and compatibility with display devices.

Image Properties. Beyond the raw pixel matrix, several higher-order spatial properties of images are central to content-aware processing.

Image gradient. The spatial gradient of a grayscale image f at pixel (i, j) is the vector

$$\nabla f(i, j) = \begin{bmatrix} \frac{\partial f}{\partial i} \\ \frac{\partial f}{\partial j} \end{bmatrix}$$

whose magnitude,

$$\|\nabla f(i, j)\| = \sqrt{\left(\frac{\partial f}{\partial i}\right)^2 + \left(\frac{\partial f}{\partial j}\right)^2}$$

quantifies the rate of intensity change. In discrete implementations, partial derivatives are approximated using convolutional kernels; the Sobel operator applies a 3×3 kernel that simultaneously estimates the gradient and suppresses noise. High gradient magnitude reliably identifies edges and boundaries, regions that, if removed during retargeting, would produce perceptible distortions.

Local texture and variance. Texture refers to the spatial arrangement of intensity variations within a local image region, characterized by properties such as regularity, directionality, and coarseness [6].

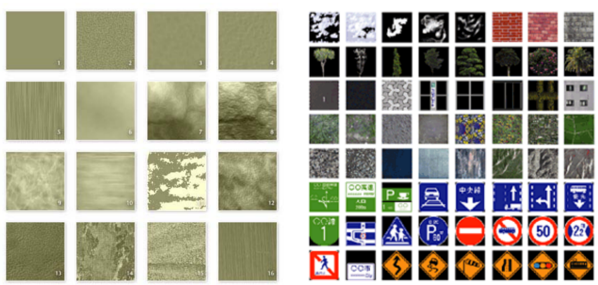

Fig. 2: Texture Example. Source: [7]

A compact numerical descriptor of local texture is the local variance, defined over a neighborhood $\mathcal{W}$ centered at (i, j) as

$$\sigma^2(i, j) = \frac{1}{\|\mathcal{W}\|} \sum_{(u,v) \in \mathcal{W}} (f(u, v) - \mu(i, j))^2$$

Unlike the gradient, which responds strongly only to directional transitions, local variance captures the statistical richness of a pixel's neighborhood regardless of edge orientation. A high variance value indicates a region dense with fine detail or irregular pattern, content that is perceptually salient but may be overlooked by gradient-only energy functions.

B. Dynamic Programming

Dynamic programming (DP) is an algorithmic paradigm for solving complex optimization problems by decomposing them into simpler, interdependent subproblems and storing the solution to each subproblem exactly once to avoid redundant computation [8]. The term was coined by Richard Bellman in the 1950s [9], originally in the context of multistage decision processes in operations research.

Characteristics of DP Problems . A problem is amenable to dynamic programming if and only if it exhibits two structural properties [8].

Optimal substructure exists when the optimal solution to the problem contains within it optimal solutions to its subproblems. More formally, if the optimal solution to a problem P decomposes into optimal solutions for subproblems $P_1, P_2, \ldots, P_k$, then an optimal solution S^* to P must be composed of the optimal solutions $S_1^*, S_2^*, \ldots, S_k^*$ for each of the subproblems $P_1, P_2, \ldots, P_k$. This property legitimizes the bottom-up construction of an optimal global solution from locally optimal sub-solutions. Without it, greedily combining optimal subproblem solutions would not guarantee global optimality. A standard proof technique for establishing optimal substructure is the cut-and-paste argument. Suppose by contradiction that S^* contains a suboptimal solution S_i to subproblem P_i; replacing S_i with the true optimal S_i^* would then yield a strictly better solution to P, contradicting the optimality of S^* [8].

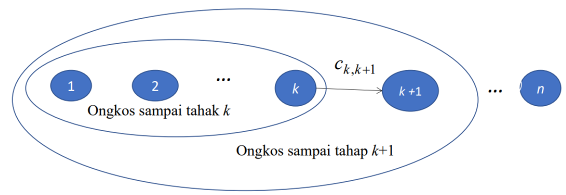

Fig. 3: Optimal Substructure of a Problem. Source: [10]

The second is *Overlapping subproblems* means that the recursive solution explores the same subproblems repeatedly. If a naive recursive algorithm solves each subproblem independently, the computational cost grows exponentially with problem size due to this redundancy. DP exploits this property by storing and reusing solutions to each unique subproblem exactly once, ensuring that each subproblem is computed only once.

C. Seam Carving Algorithm

Avidan and Shamir [1] define a vertical seam as an 8-connected, monotonic path of pixels traversing the image from the topmost to the bottommost row, containing exactly one pixel per row.

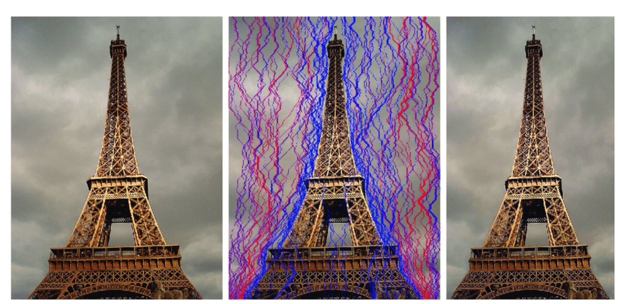

Fig. 4: Illustration of a vertical seam in an image. The seam is a connected path of pixels running from top to bottom, highlighting the region that will be removed or shifted.

Formally, given an image $\mathbf{I}$ of dimensions $M \times N$, a vertical seam is defined as

$$s^x = \{s_i\}_{i=1}^{M} = \{(i, x(i))\}_{i=1}^{M}, \quad x(i) \in \{1, \ldots, N\}, \\ |x(i) - x(i-1)| \leq 1$$

where $x(i)$ denotes the column index of the seam at row i, and the connectivity constraint $|x(i) - x(i-1)| \leq 1$ ensures that consecutive seam pixels are horizontally adjacent or directly above one another. Symmetrically, a horizontal seam s^y spans the image from left to right with exactly one pixel per column as

$$s^y = \{s_j\}_{j=1}^{N} = \{(y(j), j)\}_{j=1}^{N}, \quad y(j) \in \{1, \ldots, M\}, \\ \text{with} \quad |y(j) - y(j-1)| \leq 1$$

The total number of possible vertical seams in an $M \times N$ image grows as $O(3^M)$, since at each of the M rows the seam may branch to one of at most three column positions. Exhaustive enumeration is therefore computationally infeasible for images of any practical size, motivating the use of dynamic programming to identify the optimal seam efficiently.

Energy Function. Central to seam carving is a pixel-level energy function $e : \mathbf{I} \to \mathbb{R}_{\geq 0}$ that assigns a non-negative importance score to every pixel, reflecting its visual significance [1]. A pixel with high energy is considered important and should be preserved; a pixel with low energy is dispensable and is preferentially included in seams for removal. The classical energy function proposed in [1] is the l_1-norm of the image gradient expressed as

$$e_1(i, j) = \left|\frac{\partial \mathbf{I}}{\partial x}\right| + \left|\frac{\partial \mathbf{I}}{\partial y}\right|$$

where the partial derivatives are approximated in practice using the Sobel operator, which is a 3×3 convolutional kernel that estimates the horizontal and vertical intensity gradients while simultaneously suppressing noise. For color images, the gradient is computed independently on each RGB channel and summed, or alternatively computed on the luminance channel after color space conversion. The energy function e_1 assigns high values to edge pixels and low values to smooth, homogeneous regions, thereby directing seams through visually unimportant areas.

Dynamic Programming Formulation. Given the energy map $e(i, j)$, the problem of finding the optimal vertical seam reduces to identifying the path s^* of minimum cumulative energy. This problem exhibits both structural properties required for dynamic programming, that is the problem has an optimal substructure, since the lowest-cost path to row i at column j is determined entirely by the lowest-cost path to row $i-1$ at the optimal predecessor column and overlapping subproblems. The dynamic programming proceeds bottom-up by constructing a cumulative minimum energy map $M(i, j)$, defined recursively as

$$M(i, j) = e(i, j) + \min \begin{cases} M(i-1, j-1) \\ M(i-1, j) \\ M(i-1, j+1) \end{cases}$$

with boundary conditions

$$M(1, j) = e(1, j), \quad \forall j \in \{1, \ldots, N\} \\ M(i-1, 0) = M(i-1, N+1) = \infty, \quad \forall i > 1$$

The value $M(i, j)$ represents the minimum cumulative energy of any vertical seam reaching pixel (i, j) from the top of the image. This recurrence is an instantiation of the Bellman equation, with the state $\sigma = (i, j)$, the immediate cost $c = e(i, j)$, and the value function $V(\sigma) = M(i, j)$.

Seam Identification via Backtracking. Once the cumulative energy table M is fully populated, the optimal seam s^* is identified in two steps [1]. First, the column index of the seam's terminus at the bottom row is located as

$$x(M) = \operatorname*{argmin}_{j} M(M, j)$$

Second, the seam is reconstructed upward through backtracking, that is, for each row i from $M-1$ down to 1, the predecessor column is selected as

$$x(i) = \operatorname*{argmin}_{j \in \{x(i+1)-1, x(i+1), x(i+1)+1\}} M(i, j)$$

This backtracking step recovers the optimal path in $O(M)$ time by following the chain of locally optimal predecessors stored implicitly in the DP table.

Seam Removal and Insertion. Once the optimal seam s^* is identified, it is removed from the image by shifting all pixels to the right of each seam pixel one position leftward, reducing the image width by one column [1]. To reduce the image by k columns, this procedure is repeated k times sequentially on the progressively narrowed image. For image expansion by k columns, naively re-inserting the same minimum-energy seam repeatedly would produce a single duplicated stripe. Instead, Avidan and Shamir [1] propose first computing and storing the k lowest-energy seams simultaneously, then duplicating all k seams in the order they were found, each time averaging the pixel values of the inserted seam with its neighbors to produce a smooth transition.

D. Energy Function

The quality of any seam carving result is determined fundamentally by the expressiveness of the pixel energy function $e : \mathbf{I} \to \mathbb{R}_{\geq 0}$. An ideal energy function should assign values that faithfully reflect the perceptual importance of each pixel; high energy to regions whose removal would be noticed by a human observer, and low energy to regions that are visually dispensable. Formally, the energy function partitions the pixel domain into a saliency ordering over which the DP seam search operates; its design therefore constitutes the single most consequential algorithmic decision in a retargeting pipeline [1]. Multiple energy formulations have been proposed in the literature, each capturing a different aspect of visual importance, and they differ substantially in their sensitivity to distinct image features.

Gradient-Based Energy (Backward Energy). The original energy function proposed by Avidan and Shamir [1] measures the ℓ_1-norm of the spatial image gradient at each pixel

$$e_1(i,j) = \left|\frac{\partial \mathbf{I}}{\partial x}(i,j)\right| + \left|\frac{\partial \mathbf{I}}{\partial y}(i,j)\right|$$

In discrete implementation, the partial derivatives are approximated using the Sobel operator, which is a pair of 3×3 convolutional kernels $\mathbf{G}_x$ and $\mathbf{G}_y$ defined as

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

This energy function is termed backward energy in subsequent literature [2], reflecting the fact that it evaluates only the intrinsic energy of pixels being removed, without regard for the consequences of their removal on the surrounding image structure. While computationally simple and effective for images with well-defined object boundaries, backward energy degrades progressively as seams are removed. Each removal creates new pixel adjacencies that may introduce visible discontinuities, yet these artifacts are invisible to the energy function at the time of the seam selection decision.

Forward Energy. To address the artifact-generating tendency of backward energy, Rubinstein et al. [2] proposed a fundamentally different energy criterion that measures the cost introduced into the image by removing a seam, rather than the cost of the pixels being removed. This forward energy formulation evaluates, for each candidate seam pixel, the magnitude of the new intensity gradient that will be created between the pixels that become neighbors after the seam is deleted.

For a vertical seam at column $x(i)$ in row i, the three possible pixel transitions $(i-1, x(i)-1), (i-1, x(i)), (i-1, x(i)+1)$ introduce the following forward costs [2]

$$C_L(i,j) = |I(i,j-1) - I(i,j+1)| + |I(i,j-2) - I(i,j-1)|$$
$$C_U(i,j) = |I(i-1,j) - I(i+1,j)| + |I(i-2,j) - I(i-1,j)|$$
$$C_R(i,j) = |I(i,j-1) - I(i,j+1)| + |I(i,j+1) - I(i,j+2)|$$

where C_L, C_U, and C_R correspond to the seam arriving from the upper-left, directly above, and upper-right, respectively. The forward energy DP recurrence is then reformulated as

$$M(i,j) = e(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) & \text{if } j > 1 \\ M(i-1,j) + C_U(i,j) & \text{if } j > 1 \\ M(i-1,j+1) + C_R(i,j) & \text{if } j > 1 \end{cases}$$

By embedding the forward cost terms directly into the DP recurrence, the algorithm selects seams that minimize the visual disruption caused by pixel adjacency changes, yielding substantially fewer stripe-like artifacts compared to the backward energy formulation, particularly in smooth or gradually varying image regions [2].

III. PROPOSED SCHEME

A. Framework Overview

The CARVE (Content-Aware Retargeting with Variance-based Energy) framework is a dynamic programming-based image retargeting system designed to resize images along both spatial dimensions while maximizing the preservation of visually important content. CARVE extends the classical seam carving paradigm along two principal axes. First, it replaces the single-component gradient energy with a multi-component energy function that jointly captures edge structure, local texture richness, and forward-looking artifact cost within a single scalar importance score per pixel. Second, it embeds this composite energy directly into a modified DP recurrence that inherits the forward energy formulation of Rubinstein et al. [2], ensuring that seam selection minimizes not only the energy of pixels removed but also the visual disruption introduced by their removal. Together, these two extensions address the two core limitations of prior work, that is, the insensitivity of gradient-only energy to texture-rich regions, and the artifact-generating tendency of backward energy accumulation.

Processing Pipeline. The CARVE framework operates as a sequential five-stage pipeline, illustrated in Fig. 5. Given an input image $\mathbf{I}^{M \times N}$ and a target size $M' \times N'$, the framework proceeds as follows.

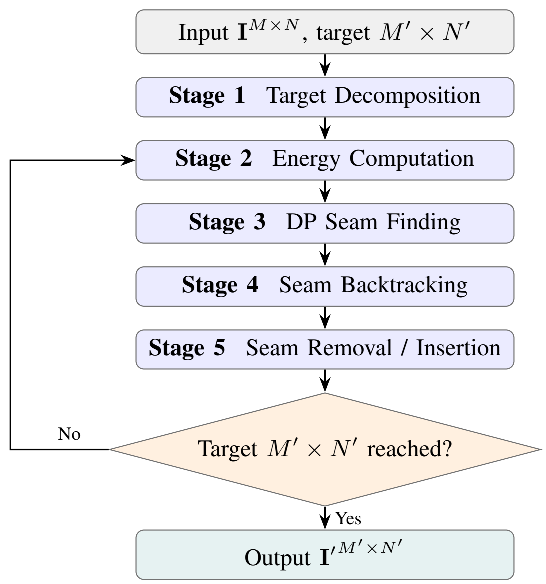

Fig. 5: CARVE framework processing pipeline.

Stage 1 : Target Decomposition. The required dimensional changes $\Delta N = |N - N'|$ and $\Delta M = |M - M'|$ are computed. The framework determines whether each dimension requires reduction (seam removal) or expansion (seam insertion), and computes a seam ordering schedule based on the energy ratio between the two dimensions to minimize compounding artifacts when both dimensions must change simultaneously.

Stage 2 : Multi-Component Energy Computation. For the current image state $\mathbf{I}$, the composite energy map $E_{\text{CARVE}}(i,j)$ is computed by evaluating and combining the gradient, variance, and forward energy components according to the weighted formulation detailed in Section III.B. This stage produces a scalar importance map of the same dimensions as the current image, with high values marking perceptually critical pixels.

Stage 3 : DP Seam Finding. The modified Bellman recurrence is applied bottom-up over the energy map to construct a cumulative minimum energy table $M(i,j)$. The recurrence incorporates forward transition costs alongside the composite energy values, ensuring that the optimal seam minimizes both removed energy and introduced discontinuities.

Stage 4 : Seam Backtracking. The optimal seam s^* is recovered by locating the minimum-energy terminus at the final row (or column, for horizontal seams) and tracing the optimal predecessor chain upward through the DP table in $O(M)$ time.

Stage 5 : Seam Removal or Insertion. For reduction, the identified seam is removed by shifting pixels inward, reducing the image by one pixel in the target dimension. For expansion, a batch of k optimal seams is pre-computed prior to any insertion, and all k seams are duplicated in the order they were found to prevent repeated insertion of the same seam. The pipeline returns to Stage 2 and iterates until the target dimensions $M' \times N'$ are reached.

Complexity. Each full pipeline iteration requires $O(M \times N)$ for energy computation, $O(M \times N)$ for DP table construction, and $O(M)$ for backtracking and seam removal yielding $O(M \times N)$ per seam. For a total of k_v vertical and k_h horizontal seams, the overall complexity is $O((k_v + k_h) \times M \times N)$, identical in asymptotic class to classical seam carving, confirming that the additional energy components introduce no change in algorithmic complexity.

B. Multi-Component Energy Function

The central algorithmic contribution of CARVE is the replacement of the single-component gradient energy with a unified multi-component formulation that jointly captures three distinct aspects of pixel-level visual importance. The composite energy at pixel (i,j) is defined as

$$E_{\text{CARVE}}(i,j) = \alpha \cdot e_{\text{grad}}(i,j) + \beta \cdot e_{\text{var}}(i,j) + \gamma \cdot e_{\text{fwd}}(i,j) + \delta \cdot \mathcal{M}(i,j)$$

where $\alpha, \beta, \gamma, \delta \geq 0$ are non-negative weighting coefficients subject to $\alpha + \beta + \gamma = 1$ for the active energy terms, and $\mathcal{M}(i,j)$ is an optional binary protection mask. Each component is described in detail below.

Gradient Energy e_{grad}. The first component retains the classical Sobel-based gradient magnitude as the edge-detection signal

$$e_{\text{grad}}(i,j) = \sqrt{(\mathbf{G}_x * \mathbf{I})^2_{(i,j)} + (\mathbf{G}_y * \mathbf{I})^2_{(i,j)}}$$

where $\mathbf{G}_x$ and $\mathbf{G}_y$ are the horizontal and vertical Sobel kernels respectively [1]. For color images, this is computed on the luminance channel $Y = 0.299R + 0.587G + 0.114B$ prior to differentiation. This component assigns high energy to pixels lying on or near structural edges and object boundaries, which represent the most perceptually disruptive pixels to remove. The gradient component is normalized to [0,1] by dividing by its per-image maximum before combination.

Variance Energy e_{var}. The second component is the novel contribution of CARVE, introduced to address the systematic insensitivity of gradient-only energy to texture-rich regions. The local variance energy at pixel (i,j) is defined over a square neighborhood window $\mathcal{W}_r(i,j)$ of radius r

$$e_{\text{var}}(i,j) = \frac{1}{|\mathcal{W}_r| - 1} \sum_{(x,y) \in \mathcal{W}_r(i,j)} (I(x,y) - \mu_r(i,j))^2$$

where $\mu_r(i,j)$ is the mean luminance within the window $\mathcal{W}_r(i,j)$, $|\mathcal{W}_r|$ is the number of pixels in the window, and $|\mathcal{W}_r| = (2r+1)^2$. The variance energy captures the statistical richness of a pixel's local context, where a smooth background region yields $e_{\text{var}} \approx 0$ regardless of its absolute intensity, while a pixel embedded in a fine-grained texture (such as foliage, fabric, or hair) yields a high e_{var} value even when no strong directional gradient is present. This property is complementary to e_{grad}. The two components respond to different visual structures and their combination covers a strictly larger class of perceptually important content than either alone. Like e_{grad}, the variance map is normalized to $[0,1]$ per image before combination.

Forward Energy Cost e_{fwd}. The third component integrates the forward energy transition costs of Rubinstein et al. [2] as an explicit energy term. CARVE treats them as a pre-computed additive energy signal that penalizes pixels whose removal would create large intensity discontinuities between their future neighbors. Concretely, the forward cost at pixel (i,j) is defined as the minimum transition cost over the three possible predecessor directions

$$e_{\text{fwd}} = \min(C_L(i,j), C_V(i,j), C_R(i,j))$$

where C_L, C_V, C_R are the transition cost functions for the left, vertical, and right predecessor directions. By including e_{fwd} as an additive term in the composite energy rather than only in the recurrence, pixels that would introduce large new adjacency costs are penalized at the energy-map level, making the DP less likely to route seams through them regardless of the direction of approach.

Protection Mask $\mathcal{M}$. The optional protection mask $\mathcal{M} : \mathbb{Z}^2 \to \{0, \Omega\}$ allows designated pixels to be effectively excluded from seam traversal by assigning them an arbitrarily large energy penalty $\Omega >> 1$

$$\mathcal{M}(i,j) = \begin{cases} \Omega, & \text{if } (i,j) \text{ is protected} \\ 0, & \text{otherwise} \end{cases}$$

In this work, $\Omega = 10^5$ is used, a value large enough to dominate the composite energy of any non-protected region. The protection mask can be populated manually by the user to mark regions of interest, or automatically by any binary saliency detector such as a face detector or foreground segmentation method. When no mask is provided, $\delta = 0$ and the term vanishes.

Normalization and Combination. The full composite energy map is computed in a single forward pass over the current image state. Each of the three active components is first computed independently over all $M \times N$ pixels, then normalized to $[0,1]$ by its per-image maximum, and finally combined via the weighted sum. The resulting E_{CARVE}

map, together with the per-pixel forward transition costs C_L, C_U, C_R retained separately, is passed to the DP recurrence in Stage 3.

C. Dynamic Programming Formulation in CARVE

Having defined the composite energy $E_{\text{CARVE}}(i, j)$ in Section III.B, the seam finding problem reduces to identifying the minimum-cost monotonic path through this energy map, solved efficiently via the DP recurrence established in Section II.B. CARVE modifies the classical recurrence of Avidan and Shamir [1] by embedding both the composite energy and the forward transition costs simultaneously, yielding a unified formulation that minimizes removed content importance and introduced visual discontinuity in a single pass.

Modified Recurrence. The cumulative minimum energy table $M_{\text{cum}}(i, j)$ is initialized and computed for $i = 1 \dots M$ using

$$M(i,j) = E_{\text{CARVE}}(i,j) + \min \begin{cases} M(i-1, j-1) + C_L(i,j) \\ M(i-1, j) + C_U(i,j) \\ M(i-1, j+1) + C_R(i,j) \end{cases}$$

where C_L, C_U, and C_R are the forward transition costs defined in Section II.D, and E_{CARVE} replaces the single-component gradient energy of prior formulations. The value $M(i, j)$ thus represents the minimum cumulative cost of any valid seam path from the top row to pixel (i, j), penalizing both the multi-component importance of removed pixels and the new intensity discontinuities their removal would introduce.

Optimal Seam Recovery. Once M is fully populated, the optimal seam $\mathbf{s}^*$ is recovered by first locating the terminus at the bottom row

$$x(M) = \arg\min_j M(M, j)$$

and then backtracking upward through the table

$$x(i) = \arg\min_{j \in \{x(i+1)-1,\ x(i+1),\ x(i+1)+1\}} M(i, j)$$

with $i = M-1, \dots, 1$ yielding the complete seam $\mathbf{s}^* = \{(i, x(i))\}_{i=1}^{M}$ in $O(M)$ additional time. For horizontal seams, the identical formulation applies with rows and columns transposed.

D. Seam Selection Strategy

When retargeting requires changes to both dimensions ($\Delta N > 0$ and $\Delta M > 0$), the order in which vertical and horizontal seams are interleaved affects output quality. CARVE adopts an energy-ratio scheduling strategy. At each iteration, the dimension whose minimum seam energy is lower relative to its total image energy is prioritized, as removing lower-energy seams first reduces the risk of compounding artifacts in subsequent iterations. When only one dimension requires change, the corresponding seam type is applied exclusively.

E. Seam Insertion (Upscaling)

For image expansion by k pixels along a given dimension, naively re-inserting the same minimum-energy seam repeatedly produces an artificial stripe artifact. CARVE instead pre-computes the k optimal seams in a single batch, storing each seam's pixel coordinates before any insertion is performed, and then duplicates all k seams in the order they were originally found. Each inserted seam is populated by averaging the color values of its immediate left and right (or upper and lower) neighbors, producing a smooth transition. This batch strategy ensures that no seam is selected twice, which would otherwise cause a single low-energy path to dominate the expansion result.

IV. IMPLEMENTATION

A. Development Environment

The CARVE framework is implemented in Python 3.10+ and executed on an HP Victus laptop equipped with an AMD Ryzen 7 5800H processor and 16 GB of RAM. The core libraries are NumPy for vectorized matrix operations, OpenCV for image I/O and Sobel gradient computation, SciPy (`uniform_filter`) for efficient local variance calculation, and Scikit-image for SSIM evaluation.

B. Implementation of CARVE

The CARVE framework is organized into six specialized modules, each with a single well-defined responsibility. The following subsections describe the design and key implementation decisions of each module.

Energy Modules (`energy.py`). Gradient energy e_{grad} is computed via Sobel operators and normalized to $[0, 1]$. Local variance e_{var} is derived from the identity

$$\text{Var}(X) = \text{E}\left[X^2\right] - \left(\text{E}[X]\right)^2$$

evaluated with a box filter over a 3×3 neighborhood on the grayscale luminance channel. Forward transition costs C_L, C_U, and C_R are computed through vectorized array slicing on a padded grayscale image. The composite energy $E_{\text{CARVE}} = \alpha \cdot e_{\text{grad}} + \beta \cdot e_{\text{var}} + \gamma \cdot e_{\text{fwd}} + \delta \cdot \mathcal{M}$ is assembled with $e_{\text{fwd}} = \min(C_L, C_U, C_R)$, and $\mathcal{M}$ imposing a penalty of 10^5 on protected pixels. Raw C_L, C_U, C_R values are returned alongside E_{CARVE} for direct use as DP transition costs.

Dynamic Programming Module (`dp.py`). The cumulative cost table $M(i, j)$ is filled row-by-row in bottom-up order. At each row, all three predecessor costs are evaluated simultaneously across all columns using `np.vstack` and `np.argmin`, avoiding per-pixel loops. Boundary conditions are enforced by padding with `np.inf`. Backtracking recovers the optimal seam in $\mathcal{O}(M)$ from the minimum entry in the last row. Horizontal seams are found via the transpose trick, reusing the vertical implementation.

Seam Manipulation (`seam.py`). Removal is performed with boolean indexing in a single vectorized pass, returning an $(M, N-1)$ image while updating the protection mask in the same operation. For expansion, all k target seams are pre-computed from the original image before any insertion; column indices of pending seams are shifted right after each insertion to prevent the parallel-artifact problem [1]. Horizontal operations reuse the vertical functions via transposition.

Main Pipeline (`carve.py`). The `carve_image` function orchestrates all five stages. Vertical seams (width) are processed before horizontal seams (height). For reduction, seams are removed sequentially with energy recomputed each iteration; for expansion, the batch pre-computation strategy is applied.

C. Dataset

Experiments are conducted on a subset of the RetargetMe benchmark [4], a publicly available collection of images commonly used to evaluate content-aware retargeting methods. The full dataset contains over 80 images spanning diverse visual categories including people, animals, architecture, landscape, and indoor scenes.

For this study, 15 images are selected from the dataset, representing a range of visual characteristic, edge-dominant scenes (architecture), texture-rich regions (animal, landscape), and mixed content (people, indoor). Images are loaded in RGB format. Any RGBA input is converted to RGB prior to processing. No further preprocessing such as resizing or normalization is applied, preserving the native resolution of each image.

Each image undergoes width reduction of 25% (i.e., target width = $0.75 \times N$) while height is held constant, consistent with the standard single-axis retargeting evaluation protocol used in prior work.

V. TESTING & RESULTS

A. Evaluation Metrics

Retargeting quality is assessed using two complementary quantitative metrics. **Structural Similarity Index (SSIM).** SSIM [5] measures perceptual similarity between the retargeted output and the original image by jointly evaluating luminance, contrast, and structural information. It is computed on the grayscale representations of both images with `data_range` $= 255$

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma}$$

where l, c, and s denote the luminance, contrast, and structure comparison functions respectively. SSIM ranges from -1 to 1, with higher values indicating greater perceptual fidelity. Values above 0.8 are generally considered excellent, $0.6 - 0.8$ acceptable, and below 0.6 poor.

Energy Preservation Ratio (EPR). EPR quantifies how effectively the retargeting process retains high-energy (visually important) content by comparing the mean gradient energy density of the output against that of the original

$$EPR = \frac{\sum e_{\text{grad}}(I')/(M' \cdot N')}{\sum e_{\text{grad}}(I)/(M \cdot N)}$$

where I and I' denote the original and retargeted images respectively. A value of $EPR > 1$ indicates that the output is more energy-dense per pixel than the original, meaning the removed pixels were predominantly low-energy, the desired behavior for a content-aware method. SSIM captures overall structural fidelity, while EPR directly measures whether the discarded content was visually dispensable.

B. Quantitative Results

Table I reports the mean SSIM and EPR scores across 15 images for all four evaluated methods. Fig. 6 visualizes these results as a grouped bar chart.

TABLE I: Mean SSIM and EPR Across 15 RetargetMe Images

Method	SSIM ↑	EPR ↑
M0 — Naive Crop	0.3087	1.0277
M1 — Backward SC	0.3409	1.1450
M2 — Forward SC	0.3397	1.1449
M3 — CARVE (Ours)	0.3398	1.1449

EPR. All three seam-based methods consistently outperform naive cropping on EPR by a margin of approximately 0.117(1.1449 vs. 1.0277), confirming that seam carving preferentially discards low-energy pixels and preserves visually important content, a property that center cropping fundamentally cannot guarantee. The EPR values of M1, M2, and M3 are nearly identical (difference < 0.001), indicating that the energy component configuration has minimal effect on which pixels are ultimately discarded, as all three methods route seams through similarly low-energy regions.

SSIM. On SSIM, all seam-based methods similarly outperform naive cropping (mean 0.34 vs. 0.31). Among the seam-based methods, however, the differences are marginal: CARVE achieves 0.3398 compared to 0.3409 for Backward SC and 0.3397 for Forward SC, a spread of only 0.0012. At the per-image level, CARVE achieves the highest SSIM on 5 out of 15 images (ArtRoom, DKNYgirl, Deck, JapenessHouse, Lotus), all of which share the characteristic of rich local texture or complex indoor structure where the e_var component contributes additional discriminative power.

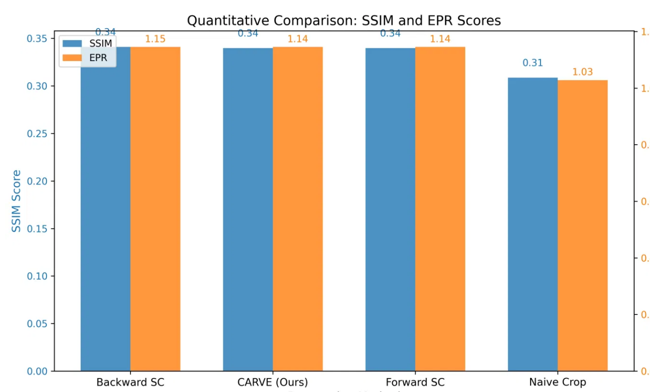

Fig. 6: Bar chart of mean SSIM and EPR across 15 RetargetMe images. M0: Naive Crop, M1: Backward SC, M2: Forward SC, M3: CARVE (ours).

On 5 other images (BedRoom, Fatem, LakeVillage, Marblehead, Sanfrancisco), naive cropping achieves the highest SSIM. These images tend to have compositionally centered subjects or large uniform regions where the center crop already retains the main content; in such cases, SSIM is dominated by global structural similarity rather than local content preservation.

The near-identical EPR and SSIM across M1, M2, and M3 on this dataset reflects the inherent diversity of the RetargetMe benchmark: the dataset spans a wide range of image types, and the benefit of e_var is most pronounced on texture-rich images, which represent only a subset of the 15 images used.

C. Qualitative Analysis

Fig. 7 through Fig. 9 present side-by-side comparisons of the four methods on representative images from the dataset. Each figure shows the original image alongside the outputs of M0 (Naive Crop), M1 (Backward SC), M2 (Forward SC), and M3 (CARVE).

Fig. 7: Fatem. M0: Naive Crop, M1: Backward SC, M2: Forward SC, M3: CARVE (ours).

Fig. 7 Fatem.png (textured background, off-center subject). The original image features a subject positioned slightly right-of-center against a green brick wall with high-contrast graffiti. Naive Crop (SSIM = 0.4738) achieves the highest structural similarity here because the subject's face is naturally close to center, and the hard boundary cut removes only peripheral graffiti without touching the subject. All three seam-based methods, while preserving the face faithfully, introduce mild horizontal compression in the graffiti background as seams must still traverse this high-energy region. CARVE (SSIM = 0.4032) routes seams through the smoother green wall areas guided by e_{var}, but the graffiti is pervasive enough that some compression remains. This image illustrates a known limitation of seam carving: when high-energy content spans the full image width, all content-aware methods yield to the same trade-off.

Fig. 8: Lake Village. M0: Naive Crop, M1: Backward SC, M2: Forward SC, M3: CARVE (ours).

Fig. 8 LakeVillage.png (canal scene, mixed textures). The original contains a waterway with canal houses, dense foliage, and a prominent boat in the foreground with detailed reflections. Naive Crop (SSIM = 0.2580) wins on SSIM here, as the houses and boat are centered and the cropped edges are mostly sky and peripheral water. Among seam-based methods, CARVE achieves SSIM = 0.2248, marginally better than Backward SC (0.2212) and Forward SC (0.2189). The tree foliage and water ripples, both high-variance, low-gradient regions are identified by e_{var} as perceptually important, directing seams toward the calmer sky and uniform wall surfaces. The boat and building contours remain intact across all seam methods, visually indistinguishable in this scene type.

Fig. 9: Marblehead Mass. M0: Naive Crop, M1: Backward SC, M2: Forward SC, M3: CARVE (ours).

Fig. 9 Marblehead.png (row of facades, edge-dominant). This image presents a continuous row of colorful colonial buildings spanning the full image width, creating a particularly challenging case: any retargeting method must either crop an edge building entirely (Naive Crop) or compress the facade widths across the row (seam methods). Naive Crop achieves the highest SSIM (0.1205) by removing the peripheral green building and preserving the center facades intact. All three seam methods produce lower SSIM (0.0926-0.0948) because compressing the building widths even slightly introduces structural deviation across all facades simultaneously. CARVE assigns maximum energy to both building edges (e_{grad}) and textured facades (e_{var}), routing seams through road and sky; the result is visually similar to M1 and M2 but with marginally smaller per-building distortion, not enough to overcome the center-crop advantage in SSIM.

D. Discussion

When CARVE performs best. CARVE demonstrates the most consistent advantage over baseline methods on images containing texture-rich regions with moderate gradient magnitude *such as* fabric patterns, foliage, and indoor surfaces *where* e_{var} provides discriminative power that gradient-only methods lack. On the 15-image dataset, CARVE achieves the highest SSIM on images with these characteristics (ArtRoom, DKNYgirl, Deck, JapenessHouse, Lotus). On images where the primary subject is naturally centered and background is uniform, naive cropping remains competitive, as observed in BedRoom, Fatem, and Sanfrancisco.

Limitations. Three limitations are worth noting. First, CARVE does not incorporate a saliency detector. Objects that lack both strong edges and rich texture such as a smooth human face against a similarly smooth background may not receive sufficient energy to be protected from seam traversal. Second, the energy weights α, β, γ are treated as global constants across the image, whereas the optimal configuration likely varies by scene type. Third, when high-energy content spans the full image width (e.g., Marblehead), all seam-based methods including CARVE are constrained to traverse content regions, and no energy formulation can avoid this trade-off.

VI. CONCLUSIONS

Image retargeting across heterogeneous display devices remains a practical challenge. Uniform scaling distorts object proportions, while naive cropping discards content without regard for its visual importance. Seam carving [1] addressed this through dynamic-programming-based removal of minimum-energy pixel paths, but its gradient-based energy function fails to protect texture-rich regions that carry perceptual importance without producing strong directional gradients. The forward energy refinement of Rubinstein et al. [2] reduced visual artifacts but retained the same limitation.

This paper proposed CARVE, a content-aware retargeting framework that augments the existing energy formulation with a local variance component e_{var} and integrates gradient, variance, and forward energy into a single unified DP recurrence. Experiments on 15 images from the RetargetMe benchmark showed that all seam-based methods, including CARVE, consistently outperform naive cropping on EPR by a margin of 0.117, confirming that seam-based approaches preferentially discard low-energy content. Among seam methods, CARVE achieved the highest SSIM on 5 out of 15 images, specifically those containing rich local texture, validating the discriminative contribution of e_{var} in these cases.

The primary strength of CARVE lies in its ability to detect and preserve texture-rich content without requiring an external saliency detector, at no additional asymptotic cost. Its main limitations are the global nature of the energy weights α, β, γ, which may require per-image tuning, and the inherent trade-off shared by all seam methods when high-energy content spans the full image width.

Future work may address these limitations through automatic weight adaptation based on scene classification, integration of a lightweight saliency map for protecting smooth but semantically important regions, and extension of the framework to temporal retargeting for video sequences.

APPENDIX

The full implementation of this program can be seen in the following Kaggle repository Kaggle Link. Explanation video regarding this paper can be seen in the following YouTube link YouTube Link.

ACKNOWLEDGMENT

All praise and gratitude are due to Allah Subhanahu wa Ta'ala, whose blessings and mercy have enabled the author to complete this paper. The author extends sincere thanks to Prof. Dr. Ir. Rinaldi Munir, M.T, the lecturer of IF2211 Strategi Algoritma, for his guidance and support throughout the writing process. The author is also grateful to family and friends for their continuous encouragement, and especially to Athilla Zaidan Zidna Fann, Bryan Pratama Putra Hendra, Philipp Hamara, Kurt Mikhael Purba, and other "AYCE" group member who provided valuable assistance and support during the completion of this work. The author would also like to thank Naufal Agam Ardiansyah, Muhammad Athala Nandana Zulmi, Nehan Radiftio, and other "Info Staycation" group member who although separated by distance, still provides emotional support to the author.

REFERENCES

[1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, Art. no. 10, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276390

[2] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," *ACM Trans. Graph.*, vol. 27, no. 3, Art. no. 16, Aug. 2008. [Online]. Available: https://doi.org/10.1145/1360612.1360615

[3] M. Rubinstein, A. Shamir, and S. Avidan, "Multi-operator media retargeting," *ACM Trans. Graph.*, vol. 28, no. 3, Art. no. 23, Jul. 2009. [Online]. Available: https://doi.org/10.1145/1531326.1531329

[4] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, "A comparative study of image retargeting," *ACM Trans. Graph.*, vol. 29, no. 6, Art. no. 160, Dec. 2010. [Online]. Available: https://doi.org/10.1145/1882261.1866186

[5] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004. [Online]. Available: https://doi.org/10.1109/TIP.2003.819861

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. New York, NY, USA: Pearson, 2018.

[7] K. Nahrstedt, "Digital image representation," lecture notes for CS 414: Multimedia Systems Design, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, Spring 2009.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.

[9] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[10] R. Munir, "Program dinamis (dynamic programming) bagian 1," bahan kuliah IF2211 Strategi Algoritma, Program Studi Teknik Informatika, STEI, Institut Teknologi Bandung, Bandung, Indonesia.

STATEMENT

I hereby declare that this paper is an original work, written entirely on my own, and does not involve adaptation, translation, or plagiarism of any other individual's work.

Bandung, 19 Juni 2026

Muhammad Aufar Rizqi Kusuma, 13524061