

Supapixel Salient Object Detection

by

Jinman Park

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2025

© Jinman Park 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Sven Dickinson
Professor, Department of Computer Science, University of Toronto

Supervisors: Paul Fieguth
Professor, Dept. of Systems Design Engineering, University of Waterloo

David Clausi
Professor, Dept. of Systems Design Engineering, University of Waterloo

Internal Member: Bryan Tripp
Professor, Dept. of Systems Design Engineering, University of Waterloo

Internal-External Member: Yuri Boykov
Professor, Cheriton School of Computer Science, University of Waterloo

Zhou Wang
Professor, Dept. of Electrical and Computer Engineering,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The first paper listed below is used in this thesis, others are not. I co-authored the papers and made significant contributions to the material’s design, development, assessment, and writing.

Park, J., Gharaee, Z., Fieguth, P., and Clausi, D. A. SuperFormer: Superpixel-Based Transformer for Lightweight Salient Object Detection. Pending results for ICCV, 2025. This paper is incorporated in Chapters 2, 3, and 4 of this thesis.

Park, J., Kaai, K., Hossain, S., Sumi, N., Rambhatla, S., and Fieguth, P. 2023. Domain-Guided Spatio-Temporal Self-Attention for Egocentric 3D Pose Estimation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’23). Association for Computing Machinery, New York, NY, USA, 1837–1849. <https://doi.org/10.1145/3580305.3599312>

Chen, X., Patel, M., Pena Cantu, F. J., Park, J., Noa Turnes, J., Xu, L., Scott, K. A., and Clausi, D. A.: MMSeaIce: a collection of techniques for improving sea ice mapping with a multi-task model, The Cryosphere, 18, 1621–1632, <https://doi.org/10.5194/tc-18-1621-2024>, 2024.

Thomson, J. W., Park, J., Magdanz, V., and Clausi, D. A. (2024). Unsupervised Methods for Fly Sperm Motility Analysis. Journal of Computational Vision and Imaging Systems, 9(1), 11–14. <https://doi.org/10.15353/jcvis.v9i1.10003>

Chen, X., Patel, M., Pena Cantu, F. J., Park, J., Turnes, J. N., Jiang, M., Xu, L., Scott, K. A., Clausi, D. A., and Huang, W. The Influence of Input Variable Selection on Deep Learning-Based Sea Ice Parameter Inversion from Multi-Sensor Satellite Data, OCEANS 2023 - MTS/IEEE U.S. Gulf Coast, Biloxi, MS, USA, 2023, pp. 1-4, doi: 10.23919/OCEANS52994.2023.10337212.

Abstract

Salient object detection (SOD) is a core problem in computer vision that involves identifying and segmenting the most visually prominent regions in an image. Its relevance spans a wide range of applications, including image understanding, object recognition, scene parsing, and human-computer interaction, as well as safety-critical domains such as autonomous driving, robotics, and medical imaging. Despite substantial progress, modern SOD models often rely on dense, pixel-level computation that imposes high computational and memory costs, limiting their deployment in resource-constrained environments.

This thesis investigates an alternative paradigm for salient object detection based on superpixel representations—compact, perceptually homogeneous regions that reduce spatial redundancy while preserving boundary structure. Although superpixels offer significant efficiency advantages, their irregular and heterogeneous nature presents unique challenges for integration into modern deep learning frameworks. Furthermore, traditional augmentation strategies and transfer learning pipelines are not readily compatible with superpixel-based models, complicating training and generalization.

To address these challenges, we propose **SuperFormer**, a lightweight vision transformer architecture tailored for superpixel-based saliency detection. Our contributions include: (1) formulating SOD as a superpixel-to-superpixel learning task to reduce computational overhead, (2) introducing a heterogeneity-aware feature representation that fuses color, texture, and shape cues, (3) adapting vision transformer architectures to operate on irregular superpixel inputs via novel positional encodings and mix-attention decoding, (4) designing superpixel-specific augmentation strategies and demonstrating the effectiveness of ImageNet pre-training in this context, and (5) conducting extensive evaluations on seven benchmark datasets, where our models (SF-S, SF-XS, and SF-XXS) achieve state-of-the-art results among lightweight SOD methods in terms of accuracy, FLOPs, and parameter efficiency.

Overall, this thesis demonstrates that with the proper architectural adaptations and training strategies, superpixel representations can enable efficient, interpretable, and high-performing salient object detection, laying the groundwork for broader adoption of structured visual abstractions in deep learning.

Acknowledgements

I would like to sincerely thank my supervisors, Professor Paul Fieguth and Professor David Clausi, for their guidance, trust, and encouragement throughout my doctoral studies. Their mentorship has been instrumental—not only in shaping this thesis, but in helping me grow as an independent scientist and critical thinker. I am especially grateful for the freedom they gave me to explore unconventional ideas, and for always challenging me to push deeper into the “why” behind my work.

I am also deeply appreciative of my committee members—Professor Zhou Wang, Professor Bryan Tripp, Professor Yuri Boykov, and Professor Sven Dickinson—for their time, insight, and thoughtful feedback. Their perspectives helped strengthen both the technical and conceptual foundations of this thesis, and I am thankful for their contributions to this final outcome.

To my colleagues and friends in the Vision and Image Processing Lab: thank you for making this journey intellectually stimulating and personally fulfilling. Whether it was a spontaneous whiteboard session or an offhand conversation that turned into a new idea, I am grateful for the collaborative spirit and genuine curiosity that defined our time together.

Finally, I am also grateful to Microsoft Office Media Group, NSERC Alliance, and the Digital Research Alliance of Canada (alliancecan.ca) for their generous support of this research.

Dedication

To my parents, for their unconditional love, strength, and sacrifices that made everything possible.

To my brother, for always cheering me on without hesitation, and believing in me even when I did not believe in myself.

To Meg, my partner and my best friend—thank you for walking beside me with love, laughter, and unwavering faith. Your presence made the hard days lighter and the good days even brighter.

To Clare, Stefan, and Theo—thank you for the countless conversations, for listening when things were hard, and for helping me take my mind off the stress when I needed it most. Your kindness and presence meant more than you know.

And to my cats Charlie and Nori, whose quiet companionship and playful reminders helped me keep balance in a life that so easily tilted toward work.

Table of Contents

List of Figures	x
List of Tables	xi
List of Abbreviations	xii
1 Introduction	1
1.1 Challenges	3
1.2 Contributions	4
1.3 Thesis Structure	5
2 Background	7
2.1 Superpixels and Segmentations	8
2.1.1 Graph-based Methods	8
2.1.2 Gradient-ascent-based Methods	9
2.1.3 Neural Network Methods	10
2.1.4 Simple Linear Iterative Clustering	11
2.2 Salient Object Detection	14
2.3 Transformers	17
2.3.1 Attention Module	17
2.3.2 Positional Encoding	18
2.3.3 Vision Transformers	20
2.4 Superpixel Attributes in Computer Vision	22

3	Problem Formulation	24
3.1	Supapixel Salient Object Detection	24
3.2	Limitations and Motivations	26
3.3	Datasets and Metrics	29
4	SuperFormer: Supapixel-based Vision Transformer	33
4.1	Addressing Heterogeneity	34
4.1.1	Colour and Texture Representation	34
4.1.2	Shape Representation	34
4.1.3	Dynamic Centroid Positional Embedding	42
4.1.4	Ablation Studies	45
4.2	Model Expansion	46
4.2.1	Encoder	47
4.2.2	Decoder	52
4.2.3	Loss Functions	54
4.2.4	Overall Model Architecture	55
5	Experiments and Results	59
5.1	Feature Augmentation	60
5.1.1	Ablation Studies	69
5.2	ImageNet Pre-training	70
5.3	Overall Results	72
6	Conclusion	79
6.1	Summary of Contributions	80
6.2	Future Work	80
	References	83

List of Figures

1.1	Visualization of Superpixels	2
3.1	Problem Formulation Visualization	25
3.2	Reconstruction Example.	27
3.3	Boxplot of Reconstruction Error.	28
3.4	Upperbound F1-score Accuracy for Superpixel Segments.	29
4.1	SLIC Initialization	35
4.2	Shape Heterogeneity of DUTS [1]	36
4.3	Fourier Coefficients Cumulative Energy	39
4.4	Efficient Shape Representation	40
4.5	Fourier Descriptors	41
4.6	Centroid Variance	43
4.7	Swin Transformer Block	48
4.8	Window Partitioning	49
4.9	Mix-Attention	52
4.10	SuperFormer Overview.	56
5.1	Diagram of Feature Augmentation	61
5.2	Superpixel Colour Augmentations	64
5.3	Horizontal Flip Visualization	67
5.4	Superpixel CutMix	68
5.5	SuperFormer Classification Diagram	72

List of Tables

2.1	Comparison of superpixel segmentation methods.	14
3.1	Summary of datasets	30
4.1	Overview of Heterogeneity Ablations	45
4.2	Pixels versus Superpixels on SuperFormer	57
4.3	Pixels versus Superpixels on CNNs	57
5.1	Overview of Feature Augmentation Ablations	70
5.2	Superpixel Processing Time	70
5.3	Overview of ImageNet Pre-training Ablations	72
5.4	Impact of Encoder Learning Rate	73
5.7	Training Hyperparameters	75

List of Abbreviations

Notation	Convention
a	A scalar
a_i	Scalar entry at index i of vector $\mathbf{a}$
$\mathbf{a}$	A vector
$\mathbf{A}$	A matrix
$\mathbf{A}_i$	The i -th matrix in a set of matrices $\{\mathbf{A}_i\}$
$\mathbf{A}_{ij}$	The j -th matrix from the i -th block or stage $\{\{\mathbf{A}_i\}_j\}$
$\mathbf{A}_{i,j}$	Scalar entry at row i , column j of matrix $\mathbf{A}$
$\mathbf{A}_{:,j}$	A vector from column j of matrix $\mathbf{A}$
$\mathbf{A}_{i,:}$	A vector from row i of matrix $\mathbf{A}$
$\mathbf{A}$	A tensor
$\mathbf{A}_{i,j,k}$	Scalar entry at index (i,j,k) of tensor $\mathbf{A}$, corresponding to the i -th row, j -th column, and k -th channel
$\mathbf{A}_{:,j,k}$	A vector from column j , and channel k of tensor $\mathbf{A}$
$\mathbf{A}_{i,:,k}$	A vector from row i , and channel k of tensor $\mathbf{A}$
$\mathbf{A}_{i,j,:}$	A vector from row i , and column j of tensor $\mathbf{A}$
$\mathcal{A}$	A set
a^{text}	Superscript text indicates a descriptive label (e.g., color space, mode)

Chapter 1

Introduction

Salient object detection (SOD) [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] is a fundamental problem in computer vision that aims to identify and segment the most visually prominent regions in an image. These regions typically correspond to objects that naturally attract human attention due to their distinct appearance, location, or semantic importance. By learning to highlight what matters most in a scene, SOD serves as an important front-end task for a wide range of downstream applications, such as image understanding [14], object recognition [15], visual tracking [16], scene parsing [17], and human-computer interaction [18].

The significance of saliency lies in its alignment with how humans perceive and process visual information. The human visual system excels at filtering out irrelevant background clutter and focusing on semantically meaningful regions almost instantaneously. This ability to rapidly identify salient regions allows us to make sense of complex visual environments with minimal cognitive effort. In the same way, computational models that can mimic this behavior—by isolating foreground objects from background noise—can improve both the efficiency and accuracy of vision systems. SOD, therefore, plays a central role in reducing the dimensionality of input data, allocating computational resources where they matter most, and enhancing interpretability across a variety of vision tasks.

Beyond traditional applications, SOD has growing relevance in modern AI systems that require not just recognition, but attention-aware decision making. For instance, in autonomous driving [19], drones [20], or assistive robotics [21], detecting salient regions helps prioritize scene elements that demand immediate action or further processing. In medical imaging [22], highlighting anomalies such as tumors or lesions can support diagnostic workflows by guiding expert attention to regions of interest. Even in non-critical



Figure 1.1: **Visualization of Superpixels.** An example of superpixel segmentation. (Left) Original input image. (Right) Superpixel representation, where pixels are grouped into compact, perceptually similar regions. This representation reduces spatial complexity while maintaining important object boundary information.

domains like photography [23], SOD supports automatic cropping and aesthetic enhancement by identifying the focal points of an image.

Given its utility, SOD continues to evolve as new challenges arise from diverse data modalities, dynamic environments, and computational constraints. As vision models become increasingly integrated into real-world systems, the need for fast, lightweight, and interpretable saliency models has become more pronounced. This thesis revisits SOD through the lens of spatial abstraction and efficiency, aiming to bridge the gap between high-performance detection and practical deployment.

Inspired by this notion of selective attention and abstraction, this thesis explores an alternative paradigm for SOD—one built on superpixels [24, 25, 26, 27, 28, 29, 30, 31]. Superpixels group perceptually similar pixels into compact, spatially coherent regions, thereby offering a middle ground between raw pixels and high-level objects. They provide a natural mechanism for reducing computational overhead, while retaining boundary-aware structural detail. However, despite their potential, superpixels have been underutilized in deep learning pipelines, primarily due to their irregularity and heterogeneous nature, which complicates their integration into modern architectures such as transformers [32, 33, 34].

The goal of this thesis is to develop a framework that embraces the efficiency and structural richness of superpixels while overcoming the technical barriers that have limited their adoption. Through a series of architectural innovations, novel feature representations, and efficient training strategies, we demonstrate that it is possible to build a lightweight and high-performing salient object detector that operates entirely on superpixel inputs. Our approach not only advances the state of the art in efficient SOD but also paves the way for more structured and resource-aware computer vision models.

1.1 Challenges

Superpixels provide a powerful abstraction for reducing spatial redundancy in visual data. By grouping perceptually similar pixels into compact, semantically meaningful regions, they significantly decrease the input resolution and computational overhead of downstream models. This compression is especially advantageous for SOD, where the goal is to segment foreground objects with high boundary fidelity while maintaining efficiency. Superpixels also offer better alignment with object contours, improving interpretability and reducing over-segmentation artifacts common in dense prediction tasks.

However, while the use of superpixels reduces spatial dimensionality, it introduces a new set of challenges that are fundamentally different from those encountered in conventional pixel-wise methods. Chief among these is the issue of *heterogeneity*: superpixels vary in size, shape, texture, and color, and are not evenly distributed across the image. This diversity in region characteristics poses a significant obstacle to standard neural architectures, particularly convolutional neural networks (CNNs) [35, 36, 37], which rely on translation-equivariant kernels and assume local grid-like regularity. The lack of structural uniformity in superpixel representations disrupts the inductive biases that make CNNs effective, and can lead to unstable learning or degraded performance if not addressed through specialized feature encodings or architectural modifications.

Furthermore, the integration of superpixels into deep learning pipelines complicates the application of standard data augmentation techniques. While pixel-based models can apply augmentations such as cropping, flipping, or color jittering directly to the input tensor, superpixel-based models require recomputing the superpixel segmentation (e.g., via SLIC [31]) after every augmentation. This re-segmentation process is computationally expensive (specifically, applying SLIC on a batch of images over multiple CPU cores), adding substantial overhead (nearly 6 times) during training. As a result, common strategies for improving generalization become far more costly and less scalable in the superpixel setting.

Another critical limitation is the lack of large-scale pre-training tailored to superpixel representations. Pre-training on ImageNet has become a foundational practice for transfer learning in computer vision, enabling models to generalize effectively to downstream tasks. However, all existing ImageNet pre-training [35, 38, 39] pipelines are designed for pixel-based encoders. There is currently no standardized approach or dataset for pre-training networks directly on superpixel inputs, leaving a significant gap in model initialization strategies. This absence makes it more difficult for superpixel-based models to benefit from the generalization advantages that have driven progress in other vision domains.

Together, these challenges underscore the need for new architectures, training strategies, and augmentation pipelines that are specifically designed to handle the irregular, heterogeneous, and augmentation-sensitive nature of superpixels—while still delivering competitive accuracy and efficiency for SOD.

1.2 Contributions

This thesis addresses the aforementioned challenges by rethinking SOD through the lens of superpixel-based representations and proposing a novel architecture—**SuperFormer**—that is explicitly designed to harness the strengths of superpixels while addressing their unique limitations.

The key contributions of this thesis are as follows:

- **Superpixel Salient Object Detection Framework (Section 3.1):** We redefine SOD as a superpixel-to-superpixel learning problem. By leveraging superpixels as the fundamental unit of representation, our approach significantly reduces spatial redundancy and accelerates inference while preserving salient boundaries.
- **Addressing Superpixel Heterogeneity (Section 4.1):** We identify and rigorously analyze the heterogeneity of superpixels in color, shape, and spatial position. To mitigate this issue, we develop a feature representation that fuses RGB statistics, texture features (Local Binary Patterns [40]), and compact shape descriptors (central moments [41, 42, 43] and Fourier descriptors [44, 45, 46]). Additionally, we investigate several positional encoding strategies to adapt transformer models to irregular superpixel layouts.
- **Superpixel-based Vision Transformer (Section 4.2):** We adapt the SWIN Transformer [33] architecture to operate on superpixel representations by modifying

its patch embedding to our superpixel representation and integrating a mix-attention [47] decoder, while preserving the overall model structure.

- **Feature Augmentation and Pre-training Strategies (Section 5.1 & Section 5.2):** We design and evaluate superpixel-specific feature augmentation methods, including Superpixel CutMix, horizontal flip and color perturbations. Furthermore, we demonstrate the benefit of ImageNet pre-training even when using superpixel-based representations, highlighting the importance of transfer learning in this domain.
- **Comprehensive Evaluation (Section 5.3):** We benchmark our method across seven standard SOD datasets spanning both natural scenes and remote sensing imagery. Our proposed models—SF-S, SF-XS, and SF-XXS—achieve state-of-the-art performance among lightweight SOD models, exhibiting favorable trade-offs between accuracy, FLOPs, and model size.

1.3 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2 (Background)** reviews the foundational concepts relevant to this research, including superpixel segmentation methods, SOD techniques, and the application of transformers in computer vision. The chapter also details how superpixel attributes have previously been harnessed for computer vision tasks.
- **Chapter 3 (Problem Formulation)** formally defines the superpixel-based SOD problem, outlines its challenges, and motivates our methodological choices. It also describes the datasets and evaluation metrics used throughout the thesis.
- **Chapter 4 (SuperFormer: Superpixel-based Vision Transformer)** presents our proposed architecture in detail, with a focus on mitigating superpixel heterogeneity and building a scalable encoder-decoder framework. This chapter includes several ablation studies to validate each design choice.
- **Chapter 5 (Experiments and Results)** evaluates SuperFormer across multiple datasets and baselines. We examine the impact of feature augmentation techniques, pre-training strategies, and model scaling, comparing our results with other lightweight state-of-the-art SOD methods.

- **Chapter 6 (Conclusion)** summarizes the main findings and contributions of the thesis and outlines potential directions for future research in superpixel-based learning and efficient saliency detection.

Chapter 2

Background

This chapter provides the foundational concepts and methodologies relevant to the research, focusing on superpixels and segmentation techniques, salient object detection, and the emergence of transformers in machine learning. The following topics form the building blocks for understanding the challenges and advancements addressed in this work.

- [Section 2.1 Superpixels and Segmentations](#): Image segmentation, a fundamental task in computer vision, partitions an image into coherent regions that simplify representation and analysis. Superpixel techniques, which group pixels into perceptually meaningful clusters, have become essential for reducing computational complexity while preserving boundary details. This subsection explores the evolution of superpixel-based methods and their applications in modern segmentation tasks.
- [Section 2.2 Salient Object Detection¹](#): Salient object detection identifies the most visually distinctive regions of an image, playing a pivotal role in various applications, such as image editing, object recognition, and scene understanding. This section reviews the progression of saliency detection methods, emphasizing their reliance on spatial, color, and contextual cues to enhance accuracy.
- [Section 2.3 Transformers](#): Originally developed for natural language processing, transformers have recently revolutionized computer vision by enabling long-range dependency modeling and flexible attention mechanisms. This section introduces the principles of transformer architectures, their adaptation for vision tasks, and the pivotal role of positional encoding for transformers.

¹Section 2.2 and Section 2.4 includes the material from Section 2 of [48].

- [Section 2.4 Superpixel Attributes in Computer Vision.](#): Superpixel attributes, such as color, texture, shape, and spatial relationships, play a pivotal role in enhancing computer vision tasks by providing a compact and meaningful representation of image regions. These attributes are widely utilized for object segmentation, tracking, and scene understanding, enabling algorithms to operate on coherent regions rather than individual pixels. By leveraging superpixel attributes, vision models can preserve critical boundary details, improve computational efficiency, and incorporate regional context for more robust feature extraction. This section delves into how superpixel attributes are harnessed to address diverse challenges in computer vision, highlighting their versatility and impact across various applications.

Together, these elements provide the background context for existing methodologies relevant to superpixel salient object detection and its challenges that will be addressed in this thesis.

2.1 Superpixels and Segmentations

Over the last two decades, many methods [24, 25, 26, 27, 28, 29, 30, 49, 50, 51, 52] have been proposed for superpixel segmentation. In this thesis, simple linear iterative clustering (SLIC) [49] is chosen and fixed as the default method for superpixel segmentation. Although the effect of different superpixel segmentation methods on salient object detection is not covered in the scope of this thesis, a quick overview of other methods and an explanation for why SLIC is chosen as the default method will be covered in this section.

Superpixel segmentation can be largely divided into graph-based, neural network, and gradient-ascent-based methods. In the following three subsections, related works on the respective methods will be presented.

2.1.1 Graph-based Methods

With graph-based algorithms [24, 25, 26, 27], each pixel is a node in a graph, and the similarity between the pixels are proportionally represented as an edge weight between two nodes. The final superpixel segments are generated by minimizing an objective function defined on the graph.

The normalized cuts algorithm by Shi and Malik [24] (NC05) recursively segments a given graph as a function of contour and texture. The objective is to minimizing a cost

function defined on the edges of the segmentation boundaries. Although attempts have been made to make the algorithm more efficient [25], the algorithmic complexity of the NC05 algorithm is still computationally expensive for images with high resolution.

Felzenszwalb and Huttenlocher [26] (GS04) present another graph-based segmentation method to generate superpixels. This method clusters pixels agglomeratively on a graph. As a result, each superpixel is the shortest spanning tree of the inherent pixels. Although GS04 is much faster than NC05, it does not have the ability to explicitly control the number of generated superpixels or their compactness.

A superpixel lattice by Moore et al. [27] (SL08) is generated by calculating optimal vertical or horizontal paths across the image. Specifically, within the vertical or horizontal strips of pixels, it uses graph cuts to find the optimal path. Even though SL08 offers control of size, number, and compactness of the generated superpixels, the quality and inference speed depends heavily on the pre-computed boundary maps.

Normalized cuts algorithm, graph-based segmentation, and superpixel lattice each suffer from inefficiencies that limit their practicality. Normalized cuts, while effective in producing globally optimal segmentations, is computationally expensive, making it unsuitable for real-time applications or high-resolution images. Graph-based segmentation, though more efficient, lacks control over superpixel shape and size, often resulting in irregular regions that fail to consistently align with object boundaries, reducing accuracy in detecting salient objects. Superpixel lattice method, while computationally simpler due to its structured grid-like arrangement, is inflexible and does not adapt to image content, leading to poor boundary adherence and suboptimal saliency representation. Given these limitations, we now explore gradient-ascent-based methods in the next subsection to address some of these challenges.

2.1.2 Gradient-ascent-based Methods

Graph-based and gradient-ascent-based superpixel methods differ in their segmentation approach, computational efficiency, and adaptability. Graph-based methods [24, 25, 26, 27], treat the image as a graph and partition it by optimizing intra-segment similarity and inter-segment contrast. While graph-based methods effectively preserve object boundaries, they can be computationally expensive and produce irregularly shaped superpixels. In contrast, gradient-ascent-based methods [28, 29, 30], iteratively seek high-density regions in feature space, leading to more adaptive, edge-aware segmentation. These methods are generally more efficient but can suffer from over-segmentation if parameters are not well-tuned.

While graph-based superpixels offer strong global segmentation quality, gradient-ascent-based approaches provide better local structure preservation, making them more suitable for applications requiring fine detail extraction, such as salient object detection.

Mean-shift by Comaniciu and Meer [28] (MS02) is a mode-seeking algorithm that generates image segments by merging all data points in the pixel feature space to the kernel smoothed centroid. The number of generated superpixels depends on the input kernel parameters. However, MS02 offers no explicit control over the size, number, and compactness.

Quick-shift by Vedaldi and Soatto [29] (QS08) is similar to MS02 in that it is also a mode-seeking algorithm with no control over the size or number of superpixels, but is faster in practice. The goal of QS08 is to maximize the Parzen density estimate by assigning each data point in the pixel feature space to the nearest neighbour that increases the estimate. Unlike the other methods presented in this subsection, QS08 is non-iterative.

Turbopixels by Levinshtein et al. [30] (TP09) uses computationally efficient level-set based geometric flow to generate superpixels. Specifically, TP09 sequentially dilates a pre-determined number of seeds on the image plane. The geometric flow utilizes local image gradients, and thereby aims to distribute superpixels evenly on image plane.

In summary, the aforementioned methods are less suitable for superpixel SOD for the following reasons. Mean-shift is known for its high computational cost due to iterative density estimation, and its performance is highly sensitive to parameter choices, which can lead to irregular, non-compact regions. Quick-shift, although simpler, requires careful tuning of its kernel parameters and often produces over-segmented results with inconsistent boundary adherence, resulting in superpixels that can be overly fragmented and irregularly shaped. Turbopixels, while leveraging geometric flows for segmentation, can also generate non-uniform superpixels and may struggle in regions with weak edge cues, leading to under-segmentation and less reliable boundary alignment overall. To address these limitations, recent research has therefore explored neural network-based approaches [50, 51, 52] that aim to learn superpixel assignments directly from data. Although the neural network-based methods offer task-specific optimization and improved accuracy, they come at the cost of significantly increased computational complexity, making their suitability for lightweight SOD applications an important point of discussion in the following section.

2.1.3 Neural Network Methods

In recent years, there has been growing interest in integrating superpixel generation into neural network pipelines [50, 51, 52]. Unlike classical heuristic methods [24, 25, 26, 27, 28,

29, 30, 49] that rely on hand-crafted similarity measures, neural network-based approaches learn to generate superpixels directly from data. While these methods demonstrate high performance and adaptability across tasks, they also introduce substantial computational overhead in terms of floating point operations (FLOPs) and model parameters. As such, their applicability to lightweight and real-time SOD systems remains limited.

Superpixel Sampling Networks (SSN) [50] is among the earliest works to propose a differentiable superpixel framework. It introduces a trainable network that learns to generate soft superpixel assignments from pixel features, optimized via reconstruction and compactness losses. The model outputs continuous-valued superpixel masks that can be integrated end-to-end with downstream tasks such as segmentation or depth estimation. Yang et al. [51] proposes a fully convolutional network that predicts superpixel boundaries directly from input images. The method achieves state-of-the-art superpixel segmentation quality by optimizing for boundary adherence and undersegmentation error. An et al. [52] presents a lightweight alternative by focusing on region-level attributes such as color, texture, and spatial proximity. Their Extract and Merge (EAM) framework employs a shallow neural network to extract regional cues, followed by a merge strategy to aggregate pixels into superpixels.

While the above neural network-based superpixel methods offer impressive task-specific optimization and accuracy in terms of reconstruction, their computational demands—measured in FLOPs and parameter count—limit their practicality in scenarios where efficiency is a key requirement. In contrast, SLIC (Simple Linear Iterative Clustering) in Section 2.1.4 provides a computationally efficient alternative, generating compact, adaptive superpixels that balance spatial regularity with strong adherence to object boundaries, making it a superior choice for salient object detection tasks.

Considering the primary limitations of previous graph-based, gradient-ascent-based, and neural network-based approaches for efficient SOD, we now elaborate on the simple linear iterative clustering algorithm in the following subsection.

2.1.4 Simple Linear Iterative Clustering

Simple linear iterative clustering (SLIC) by Achanta et al. [49] generates superpixels by grouping pixels as a function of colour similarity and proximity in the spatial domain. Although SLIC is an adaptation of k-means clustering [53] in the joint colour-spatial space, it is sometimes described as gradient-ascent-like because its iterative updates shift cluster centers toward regions of higher local similarity, analogous to moving toward a local maximum in similarity space. SLIC utilizes five feature attributes where three are the pixel

colour vector in CIELAB [54] colour space $[l, a, b]$ and two are the Euclidean xy-coordinate $[x, y]$ of the pixel position.

Given a desired number of superpixels K , the approximate size of each superpixel is N/K pixels for an image with N pixels. Initially, if the image is divided evenly into K superpixels, then a superpixel centre can be assigned at every grid interval of $S = \sqrt{N/K}$.

K superpixel cluster centres are chosen as $\mathbf{c}_k = [l_k, a_k, b_k, x_k, y_k]$ with $k = \{1, 2, \dots, K\}$ with a regular grid of initial cluster centres spaced in x and y by S . Since the approximate area of each superpixel is S^2 , there is a safe assumption that constituent pixels lie within a $2S \times 2S$ area around the superpixel centre on the xy plane. Hence, pixels within that search area are measured to each cluster centre for assignment. CIELAB colour space is perceptually meaningful when the spatial distance between two pixels is small. However, if the pixel spatial distance starts to exceed the limit of colour distance, the spatial distance starts to outweigh the pixel colour distance. As a result, superpixels are generated without respect to region boundaries. Therefore, instead of utilizing a simple Euclidean norm, a distance measure D (the total distance in lab and normalized xy plane by the grid interval S) between a superpixel centre $\mathbf{c}_k$ and a pixel $\mathbf{p}_i$ with $[l, a, b, x, y]$ feature is defined as follows:

Algorithm 1 Simple Linear Iterative Clustering

- 1: Sample initial cluster centres $\mathbf{c}_k = [l_k, a_k, b_k, x_k, y_k]$ with grid interval S .
 - 2: Reassign cluster centres by searching in an $n \times n$ neighbourhood to the lowest gradient value.
 - 3: $E \leftarrow \infty$
 - 4: **while** $E \geq threshold$ **do**
 - 5: **for** each cluster centre $\mathbf{c}_k$ in $k = \{1, \dots, K\}$ **do**
 - 6: In the search area of $2S \times 2S$, calculate the distance measure equation 2.1 and assign pixels to the superpixel with the lowest distance.
 - 7: **end for**
 - 8: Compute new superpixel centres
 - 9: $E = L1$ distance of previous centres and new centres
 - 10: **end while**
 - 11: Optionally, enforce connectivity
-

$$\begin{aligned}
d^{\text{LAB}}(\mathbf{c}_k, \mathbf{p}_i) &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\
d^{\text{XY}}(\mathbf{c}_k, \mathbf{p}_i) &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\
D(\mathbf{c}_k, \mathbf{p}_i) &= d^{\text{LAB}}(\mathbf{c}_k, \mathbf{p}_i) + \frac{m}{S} d^{\text{XY}}(\mathbf{c}_k, \mathbf{p}_i)
\end{aligned} \tag{2.1}$$

m offers explicit control of the compactness of a superpixel. The greater the value of m , the greater the emphasis on spatial distance resulting in a more compact (square) superpixel.

The SLIC algorithm initializes by sampling K regularly spaced cluster centres. Then, the centres are reassigned to the pixel with the lowest gradient position in a 3×3 neighbourhood. This avoids assigning centres at an edge and reduces the odds of choosing a noisy pixel. Image gradients to reassign initial centres are computed as

$$G(x, y) = \|\mathbf{I}(x + 1, y) - \mathbf{I}(x - 1, y)\|^2 + \|\mathbf{I}(x, y + 1) - \mathbf{I}(x, y - 1)\|^2 \tag{2.2}$$

where $\mathbf{I}(x, y)$ is the CIELAB vector of the pixel at position (x, y) . This equation accounts for both colour and intensity in calculating the gradients.

Each pixel in the image is now assigned to the nearest superpixel centre with the search area overlapping this pixel. Once all pixels are assigned to the nearest superpixel centre based on equation 2.1, new centres are computed by taking the mean of *labxy* vector associated with all pixels assigned to the superpixel. This process is recursively done until convergence. Note that at the end of the iterations, it does not guarantee that all pixels assigned to a superpixel are connected. Although seldom, some cases with discontinuity may arise even with higher focus on spatial distance (high constant m) since connectivity is not enforced during the iterative steps. As such, in the last step of the iteration, connectivity is optionally enforced by reassigning disjoint segments to the largest neighbouring superpixel. Enforcing connectivity is optional, still has $O(N)$ complexity, and is generally comprised of less than 10% of the total time for the SLIC algorithm. Refer to Algorithm 1 for a summary of the algorithm.

The main goal of integrating superpixels with salient object detection is to significantly reduce the degrees of freedom by over-segmenting an image with superpixel segmentation. To be in line with the overall objective of this thesis, SLIC is chosen as the default superpixel segmentation method for three reasons. First, SLIC allows control over the generated number of superpixels which is essential for window attention in SWIN Transformers [33](discussed in Section 2.3.3). Second, the ability to control the compactness is important in reducing the size and shape variations within generated superpixels, thereby

Table 2.1: Comparison of superpixel segmentation methods. N indicates the total number of pixels in the image. d is a small constant used by Vedaldi and Soatto [29]. TP09 also benefits from $O(N)$ complexity but is slower by a factor of 10 compared to SLIC for 481×321 resolution images. SLIC offers the most efficient computational complexity, control over the number of superpixels, and control over compactness.

Properties	Graph-based			Gradient-ascent-based			
	GS04	NC05	SL08	MS02	TP09	QS09	SLIC
Ctrl. of superpixel num.	✗	✓	✓	✗	✓	✗	✓
Ctrl. of compactness	✗	✓	✓	✗	✓	✗	✓
Complexity $O(\cdot)$	$N \log N$	$N^{3/2}$	$N^2 \log N$	N^2	N	dN^2	N
Parameters	2	1	3	3	1	2	1

minimizing the amount of noise in the data. Finally, the computational time and complexity is lowest with SLIC making it the most suitable. Refer to Table 2.1 for a summary of the comparison between the state-of-the-art superpixel segmentation methods.

2.2 Salient Object Detection

Salient Object Detection (SOD)[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] is a fundamental problem in computer vision, focusing on identifying the most visually distinctive objects or regions within an image. These regions often correspond to areas that capture human attention, making SOD a critical precursor for numerous downstream applications such as object recognition, image segmentation, and visual tracking. Among the various modalities explored for SOD, RGB-based methods have garnered significant attention due to the ubiquity of RGB cameras and the rich visual information they provide.

The journey of RGB SOD has transitioned from heuristic-based approaches [2, 3, 4] to deep learning-driven methodologies [5, 6, 7, 8, 9, 10, 11, 12, 13]. Early techniques relied heavily on handcrafted features, including color, texture, and edge information, combined with heuristic rules to approximate saliency maps. While these methods demonstrated some success, they often struggled with complex scenes and variations in lighting and texture.

Handcrafted Features. Early SOD approaches, such as those by Itti et al. [55] and Harel et al. [56], introduced computational models inspired by biological vision systems. These models used low-level features like color, intensity, and orientation to generate saliency

maps. For instance, the Graph-Based Visual Saliency (GBVS) method by Harel et al. extended these ideas by employing graph-based representations to refine saliency estimation. Similarly, Achanta et al. [57] proposed a frequency-tuned approach that combined global contrast with color spatial distribution, improving the detection of salient regions.

Natural Scenes. In recent years, notable progress has been made in SOD for natural scenes [5, 6, 7, 8, 9, 20]. In deep learning frameworks, encoder-decoder architectures are commonly used for saliency analysis, focusing on multi-level feature integration [58, 59]. Pang et al. [60] introduced a multiscale network with interaction and self-interaction modules to blend and merge features. Li et al. [61] proposed a stacked U-shaped network with channel attention and multilevel feedback to highlight salient features. Liu et al. [62] employed a Transformer backbone to improve long-range dependency modeling, outperforming convolution-based methods. Research has also focused on preserving object boundaries [63, 64], using boundary labels in training. For example, Wu et al. [65] developed a network to detect both salient regions and boundaries, while Qin et al. [66] used a hybrid loss combining BCE, IoU, and SSIM.

Remote Sensing. To enhance the application of SOD, researchers have developed specialized models tailored for remote sensing imagery. Zhang et al. [20] introduced a dense attention fluid network with a global context-aware module for long-range semantic capture. Tu et al. [67] proposed a multiscale joint region-boundary model for improved reasoning. Zhou et al. [68] designed an edge-guided recurrent network to preserve boundary details. Zeng et al. [69] created an adaptive edge-aware semantic interaction network for better boundary detection. Additionally, Li et al. [70] developed a multicontent network to integrate local and global clues.

Past research in SOD has predominantly emphasized detection accuracy as the primary goal, driving a focus on optimizing accuracy in model design [5, 6, 9, 71]. To achieve higher accuracy, researchers frequently employ large-scale or complex networks, often leading to adverse impacts on inference efficiency. Typically, as the complexity and size of a model increase, a corresponding decline in inference speed and overall efficiency becomes inevitable. While these models achieve impressive results, their computational complexity often limits their practicality in resource-constrained environments. To address this, lightweight SOD methods [11, 12, 13, 72, 73, 74, 75, 76, 77, 78] have been proposed, aiming to balance efficiency and accuracy by reducing model size and computational cost.

Efficient Salient Object Detection

Natural Scenes. SAMNet [10] employs a Stereoscopically Attentive Multi-scale mechanism that dynamically integrates features across different scales. Similarly, HVPNet [72] introduces Hierarchical Visual Perception, designed to emulate the hierarchical processing of the primate visual cortex for advanced perception learning. CSNet [11] proposes an ultra-lightweight model for SOD with only 100k parameters, achieving competitive performance by leveraging efficient feature aggregation and compression techniques. EDN [12] introduces a network that aggressively downsamples feature maps to reduce computational cost while maintaining detection accuracy through effective upsampling and feature refinement. ADMNet [13] utilizes an attention-guided, multi-scale feature fusion approach to enhance lightweight SOD models, improving both efficiency and detection accuracy. Other works have explored different strategies to improve efficiency in SOD. Gu et al. [79] leverage neural architecture search to optimize efficiency based on device constraints, focusing on how efficiency can be achieved given a model. This contrasts with our approach and other compared methods, which aim to achieve efficiency through handcrafted design rather than automated model search. Similarly, Zhang et al. [80] improve efficiency through knowledge distillation, where a larger teacher model generates pseudo-labels to train a smaller, more efficient student model. However, none of the works we compare against, including ours, rely on pseudo-labels from a larger model, making direct comparison unfair. For these reasons, we do not include INAS and KD-SCFNet in our experimental comparisons, as their efficiency paradigms differ fundamentally from the handcrafted design approaches explored in our work.

Remote Sensing. Li et al. [81] adapted VGG16 [82] to create a compact backbone and introduced CorrNet, which uses a coarse-to-fine approach to refine saliency maps, reducing parameters and improving processing speed. Li et al. [74] developed SeaNet, a lightweight model with semantic matching and edge self-alignment, which further reduces complexity but offers limited accuracy improvement. Liang and Luo [75] introduced MEANet, which uses multiscale edge-embedded attention to enhance saliency representation. Yao and Gao [77] proposed dynamic feature fusion, improving feature learning by adjusting weights based on information quality and reducing noise.

While existing lightweight salient object detection methods in both natural scenes and remote sensing focus on model design optimizations—such as efficient architectures, multi-scale feature fusion, and attention mechanisms—none of them achieve efficiency by reducing the input’s spatial dimension. In contrast, our approach achieves efficiency by reducing input dimensions with superpixel representations, introducing a new paradigm for lightweight SOD.

2.3 Transformers

Transformers have been dominating the field of natural language processing (NLP) [83, 84] since 2017, and now been successful in other domains beyond NLP specifically in computer vision [33, 34, 85]. Much of its success is owed to their attention mechanism. Unlike recurrent neural networks (RNNs) [86, 87, 88], the attention module in Transformers enables the modeling of direct relationships between any pair of tokens—temporal units within a sequence represented as vectors. This capability not only enhances the model’s discriminative power but also significantly speeds up computation through parallelization. However, the parallelized architecture inherently lacks a representation of sequential order, which is essential for many tasks. To address this limitation and provide a sense of order to the features, deterministic positional encoding is introduced [32, 83, 89].

2.3.1 Attention Module

Understanding attention modules is crucial for the later sections of this thesis, particularly in the design of both the encoder (Section 4.2.1) and decoder (Section 4.2.2). In subsequent chapters, we will see how attention mechanisms are integrated into these components to enhance feature extraction and information flow. The encoder leverages attention to capture contextual relationships within superpixel representations, while the decoder utilizes attention-based operations to refine and reconstruct meaningful outputs. By establishing a solid foundation here, readers will be better equipped to follow the discussions on how attention is applied in our proposed approach.

Self-attention maps the relationship between a query and a set of key-value pairs to return an output based on the aggregation weights. Given an input $\mathbf{X} \in \mathbb{R}^{T \times F}$ with number of tokens T and feature depth F , there are three sets of learnable parameters (implemented as a linear layer) $\mathbf{L}^Q \in \mathbb{R}^{F \times D}$, $\mathbf{L}^K \in \mathbb{R}^{F \times D}$, and $\mathbf{L}^V \in \mathbb{R}^{F \times D}$ with some depth D which are used to form query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ for the Transformer module as

$$\mathbf{Q} = \mathbf{X} \cdot \mathbf{L}^Q, \quad \mathbf{K} = \mathbf{X} \cdot \mathbf{L}^K, \quad \mathbf{V} = \mathbf{X} \cdot \mathbf{L}^V. \quad (2.3)$$

Given these matrices, the attention matrix $\mathbf{A} \in \mathbb{R}^{T \times T}$ is computed as

$$\mathbf{A} = \text{Softmax}(\mathbf{Q}\mathbf{K}^\top) \quad (2.4)$$

and the subsequent aggregation $\mathbf{A}^v$ using the value matrix $\mathbf{V}$ as

$$\mathbf{A}^v = \mathbf{A}\mathbf{V}. \quad (2.5)$$

The aggregated features $\mathbf{A}^v$ are then passed through a simple position-wise fully connected feed-forward network with non-linear activations. Given this one attention module, Vaswani et al. [32] propose expanding this module H times to employ multi-headed attention to increase the discriminative power of the network. Finally, skip connection with LayerNorm [90] is utilized in between each sub-layer. Self-attention modules are incorporated in the current proposed model in Section 4.2.

2.3.2 Positional Encoding

Due to the parallelization of feature projections, the model lacks recurrent or convolutional structures to inherently encode order in the input sequence. Consequently, the model requires additional guidance to capture either the absolute or relative positional relationships within the sequence. To address this, *positional encodings* (hand-crafted mathematical formulation) or *positional embeddings* (learned vectorization of position) are incorporated into the input embeddings before passing them to the attention modules. Intuitively, these encodings should exhibit similarity for nearby positions in the sequence and distinctiveness for positions that are far apart. Positional encodings can be implemented either as learned parameters [83] or as fixed representations [91].

The original positional encoding [32] uses sinusoidal functions to inject order into the sequence of word embeddings. Given the i -th position in the sentence, hidden dimension of the model d , and j^{th} depth (of the hidden dimension), the position encoding $\mathbf{P}$ is defined as

$$\mathbf{P}_{i,2j} = \sin(i/10000^{2j/d}) \quad (2.6)$$

$$\mathbf{P}_{i,2j+1} = \cos(i/10000^{2j/d}) \quad (2.7)$$

The even indices are sine functions and the odd indices are cosine functions. The constant 10000 sets the frequency range of the sinusoidal functions. It ensures that (a) each dimension gets a sinusoid of different wavelength, (b) The wavelengths form a geometric progression from 2π to $2\pi \cdot 10000$, and (c) the model uniquely encodes positions up to several thousand tokens and to learn to attend based on relative or absolute positions. Essentially, the choice of 10000 is arbitrary but large enough to cover a wide range of input lengths while keeping the frequency differences meaningful across dimensions. It provided a simple and effective way to inject position information into token embeddings without adding learnable parameters. These encodings use fixed sine and cosine functions, enabling the model to distinguish positions and capture relative positional relationships. This deterministic, parameter-free design allowed for efficient training and generalization

to sequences longer than those seen during training. However, their static nature also introduced limitations. Unlike learned positional embeddings, sinusoidal encodings cannot adapt to task-specific patterns, potentially limiting performance in datasets with complex or structured positional relationships, such as images or graphs. Additionally, while they theoretically support long-range context, their effectiveness can degrade with increasing sequence length. As models evolved to tackle more diverse tasks, alternative approaches like learned embeddings [83], relative positional encodings [89, 92], and rotary positional embeddings [93] emerged to address these limitations.

Learnable positional embeddings first introduced in Devlin et al. [83] improve upon traditional positional encodings by using trainable vectors instead of fixed functions like sine and cosine. Each position in a sequence is assigned a unique, learnable embedding, which is updated during training. This allows the model to adapt to specific tasks and better capture dependencies in long sequences, where fixed positional encodings might fall short. Learnable embeddings are typically combined (element-wise addition) with token embeddings to provide a representation that includes both token content and positional information, which is then used in the model’s subsequent layers.

One of the major drawbacks of both the sinusoidal positional encoding and learnable positional embeddings is that both approaches assume that all positions contribute equally to the model’s objective and are processed uniformly, making them less suitable for tasks with heterogeneous representations. In such cases, the inputs may vary in type, structure, or importance—such as combining text, images, and tabular data—requiring a more flexible positional encoding mechanism to account for the diverse roles and relationships between input elements. Consequently, these standard encoding methods struggle to handle the complexity and interactions inherent in heterogeneous data.

To address the challenge of homogeneity of original positional encoding, some works have proposed using Laplacian positional encodings (LaPE) [94, 95, 96] to better discriminate distance in graph nodes. The eigenvectors U and eigenvalues Λ are computed by factorizing the graph Laplacian matrix with adjacency matrix A , identity matrix I , and degree matrix D :

$$\Delta = I - D^{-1/2} A D^{-1/2} = U^\top \Lambda U \quad (2.8)$$

Since the degree matrix D is a non-negative diagonal matrix, $D^{-1/2}$ is a diagonal matrix whose diagonal entries are the reciprocals of the square roots of the diagonal entries of D . Similarly, $D^{1/2}$ has diagonal entries the square roots of D . The smallest k non-trivial eigenvectors U of a node is used as its positional encoding. LPEs rely on the eigenvectors of the graph Laplacian to embed positional information. However, computing

the eigenvalue decomposition of the Laplacian becomes computationally challenging for large adjacency matrices, as it requires significant memory and processing power due to the eigenvalue decomposition that has a computational complexity of $\mathcal{O}(n^3)$ for a dense $n \times n$ matrix and $\mathcal{O}(kn^2)$ for sparse matrices when computing the top k eigenvalues and eigenvectors [97]. As the size of the graph grows, the cubic or near-quadratic complexity makes LPEs impractical for large-scale graphs or high-dimensional applications. Given the need to efficiently represent the heterogeneous nature of superpixel locations, we address the limitations of LPEs by introducing a novel centroid-based method for representing superpixel locations in Section 4.1.3.

2.3.3 Vision Transformers

Unlike traditional convolutional neural networks (CNNs) [35, 36, 39], Vision Transformers (ViTs) [33, 34, 85, 98, 99, 100] process image data by treating image patches as a sequence of tokens, analogous to words in a sentence. Each image is divided into fixed-size patches, linearly embedded into feature vectors, and combined with positional encodings to preserve spatial information. The self-attention mechanism then enables the model to capture global dependencies between patches, overcoming the locality bias of CNNs.

Dosovitskiy et al. [34] demonstrated the potential of ViTs by achieving state-of-the-art performance on image classification tasks using transformer-based architectures. Their success has spurred a wave of research exploring the use of transformers in various computer vision domains, including object detection, image segmentation, and, more recently, salient object detection (SOD).

Motivated by the success of the original ViT by Dosovitskiy et al. [34], many notable ViTs succeeded its work addressing challenges of ViTs. Yuan et al. [98] improves the tokenization process by modeling local structures in images before applying the transformer encoder. Wu et al. [99] combines the strengths of CNNs and transformers by incorporating convolutions into the tokenization and projection layers. Chen et al. [85] leverages a dual-branch architecture with different patch sizes to capture both fine-grained and coarse-grained features. Xie et al. [100] combined lightweight transformer encoders with a simple decoder for semantic segmentation.

To address the high global attention computational demands of the aforementioned ViTs, Swin Transformers (Shifted Window Transformers) [33] were introduced as a hierarchical and computationally efficient variant of ViTs. Swin Transformers divide an image into non-overlapping patches, like ViTs, but process these patches within local windows

instead of globally across the entire image. This hierarchical structure enables Swin Transformers to efficiently handle high-resolution inputs while maintaining high performance. The key innovation of Swin Transformers lies in their shifted window mechanism, which alternates between fixed local windows and overlapping windows across successive layers. This approach improves information exchange between patches and allows the model to capture both local and global dependencies without the computational cost of global attention. Additionally, Swin Transformers adopt a pyramid structure, similar to CNNs, where the resolution of feature maps is progressively reduced, and the receptive field is expanded at each stage.

In the field of computer vision, decoders [47, 101] play a crucial role in tasks requiring dense predictions, such as semantic segmentation, object detection, and image generation. While ViTs excel in capturing global context and hierarchical features through their self-attention mechanisms, decoders are essential for translating these features into task-specific outputs. Decoders for vision transformers are typically designed to process and refine multi-scale features extracted by transformer backbones, enabling precise predictions at a pixel or instance level.

Decoders for vision transformers have advanced significantly to address dense prediction tasks by integrating multi-scale features, attention mechanisms, and efficient upsampling. Popular architectures like SegFormer [100] provide a lightweight design for real-time segmentation, leveraging a simple yet effective multi-scale feature fusion. UPerNet [101] excels in general-purpose applications, employing pyramid pooling to aggregate hierarchical features. DPT [102] focuses on dense prediction tasks such as depth estimation and segmentation by refining features through multi-resolution fusion. Mask2Former [103] introduces a mask-based query mechanism, improving segmentation across instance, semantic, and panoptic tasks. Among these, the U-MixFormer decoder [47], particularly its U-Mix-Attention mechanism, stands out for its precision and efficiency in segmentation, especially in medical imaging. U-Mix-Attention combines hierarchical feature fusion with attention-based refinement to enhance task-specific features, while its U-shaped architecture ensures effective integration of spatial details and global context. This innovative design excels in capturing fine-grained structures, offering superior performance in high-resolution, dense prediction tasks. In this thesis, we employ Swin Transformers as the backbone and U-Mix-Attention as the decoder. The motivations for choosing Swin Transformers and U-Mix-Attention are further developed in Section 4.2.1 and Section 4.2.2 respectively.

2.4 Superpixel Attributes in Computer Vision

Superpixel segmentation [27, 30, 31] aggregates image pixels based on shared descriptive attributes like colour and spatial similarity. The primary motivation behind this technique is to partition an image into meaningful regions, thereby significantly reducing redundancy in the input space while preserving the natural boundaries between regions.

While not directly used for SOD, superpixels have been employed in parallel with pixels in semantic segmentation—a closely related task—alongside Vision Transformers [104, 105, 106, 107]. Zhu et al. [104] utilize superpixels to reduce spatial redundancy in semantic segmentation but still processes pixel-level features within each superpixel region using a transformer-based model. Mei et al. [105] incorporate superpixel representations into Vision Transformers to improve efficiency, but final computations and feature extractions are still performed at the pixel level. Huang et al. [106] group image patches into “super tokens” to reduce the number of tokens processed by the transformer, but initial feature extraction and learning still occur at the pixel level. Qiu et al. [107] explores contrast relationships using superpixels to enhance feature discrimination, yet maintains pixel-level processing for finer details and object delineation. Despite incorporating superpixels, all these methods ultimately rely on pixel-level processing, and do not address the heterogeneity of superpixels, limiting their efficiency gains. As a result, their models exceed 23M parameters, placing them well outside the category of lightweight SOD. In contrast, our approach fully embraces **superpixel-only** representation to achieve significant reductions in both computational complexity and model size.

Next, we review key works that explore and advance the use of superpixel-only attributes in computer vision, highlighting their contributions and explicitly identifying the limitations of previous methods that our approach aims to address.

He et al. [108] treat the segmented superpixels as a 1D array, and extracts the contextual information from spatial kernels by calculating colour uniqueness. In addition, colour distribution was another key factor that the authors determined to be a crucial factor in detecting saliency. Hence, two input sequences of colour uniqueness and colour distribution are trained with CNNs to detect salient objects.

To overcome the shortcomings of 1D representations, other methods proposed applying CNN on 2D data either before or after superpixel segmentation [109, 110]. Zohourian et al. [109] proposed treating the initial grid generated by the SLIC algorithm as the topological basis in projecting the superpixel lattice. As a result, structure (notion of neighbour) is injected back into the superpixels and the spatial inductive bias is established for CNN operations. In addition, colour attributes, position feature, and Local Binary Pattern

[111] are the features used to describe each superpixel for model fitting. However, the proposed method only works well with low input and output level variance such as road segmentation.

Suzuki et al. [110] then proposed a hybrid model that applies naive CNNs before superpixel segmentation, and generalized convolution on generated superpixels. The generalized convolution aggregates based on the adjacency matrix that indicates which superpixels are neighbours. In addition, dilation is applied to the adjacency matrix to increase the receptive field of the kernels. Although the proposed method shows promising results compared to pixel-wise naive CNN SOD, it still inherits the problem of heavy computation from both the early layers of CNN and generalized CNN.

Monti et al. [112] proposed the MoNET framework for general geometric data. The weighting of neighbourhood aggregation is determined through a learnable scaling factor based on geometric distances. Veličković et al. [113] proposed Graph Attention Networks (GAT) that benefits from self-attention for weighting the neighbourhood aggregation. Avelar et al. [114] then proposed joining superpixels and GAT to classify images. Although their proposed method performed worse than pixel-wise methods, their work opened an interesting branch of research on applying deep learning beyond regularly-gridded images. Dadsetan et al. [115] similarly incorporates GCNs with superpixels to detect nutrient deficiency stress from aerial imagery. Instead of connecting nodes just with its spatial neighbours, the nodes are fully connected so that information is propagated globally. Then, the weighting of the edges is determined by colour similarity. Kosman and Di Castro [116] introduces a video representation method that models temporal and spatial dynamics using a sparse set of nodes. By constructing a graph where nodes correspond to key regions across frames, GraphVid captures long-term dependencies efficiently, reducing computational costs while maintaining strong performance on video understanding tasks.

In summary, despite the benefits of superpixels, there remains a significant gap in research directly utilizing superpixels for SOD. Prior methods, such as Suzuki et al. [117] and Yang et al. [51], leverage convolutional neural networks (CNNs) for feature extraction before applying over-segmentation, which still requires a substantial number of parameters. Other approaches utilize GCNs [113, 114, 115, 118], reducing computational requirements compared to pixel-level processing, but lack reproducibility, standardized datasets, accuracy, and most importantly, methods to represent the heterogeneous nature of superpixels.

Chapter 3

Problem Formulation

Building on the importance of superpixels, the research gap, and the limitations of previous superpixel-based computer vision applications discussed in Chapter 2, this chapter formulates the superpixel-to-superpixel framework (Section 3.1) motivated by pixel-to-pixel salient object detection problem [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. We then outline the key motivations¹ for innovation by addressing the limitations of prior methods in superpixel-based salient object detection (Section 3.2). Finally, in Section 3.3, we present the datasets and metrics used to evaluate the effectiveness and robustness of SuperFormer.

3.1 Superpixel Salient Object Detection

Given an RGB input image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ with height H and width W in pixels, the corresponding pixel-wise label space is defined as $\mathbf{Y} \in \{0, 1\}^{H \times W}$, where each element in $\mathbf{Y}$ represents non-salient (0) or salient (1).

For a desired number K of approximately equal-sized superpixels and compactness parameter m , the SLIC algorithm [31] maps image $\mathbf{X}$ to a superpixel label map $\mathbf{L} \in \mathbb{R}^{H \times W}$, where each pixel $\mathbf{L}_{h,w}$, $1 \leq h \leq H$, $1 \leq w \leq W$ is assigned to one of the superpixel indices $\mathbf{L}_{h,w} \in \{1, 2, \dots, K\}$.

Transformers (Section 2.3) without positional encoding (Section 2.3.2) lack an inherent sense of order or position, which is why they are traditionally paired with positional embeddings. Our observation is that this characteristic of Transformers can be leveraged to

¹Section 3.1 and Section 3.2 includes the material from Section 3.1 of [48].

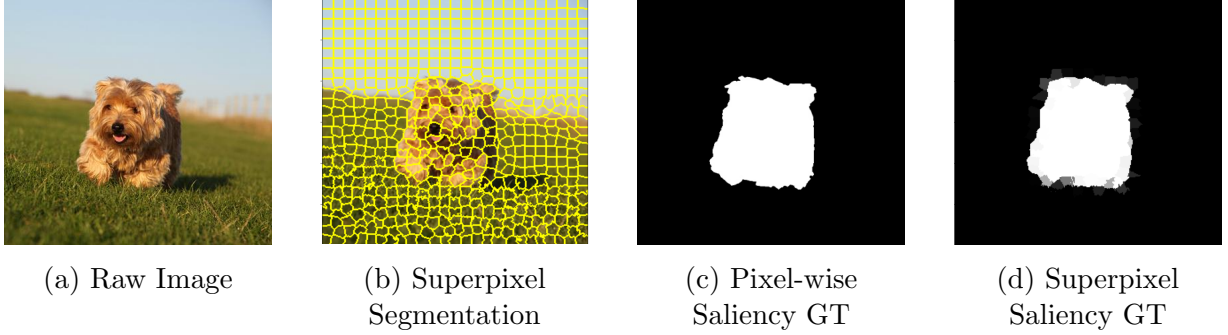


Figure 3.1: **Problem Formulation Visualization.** We oversegment a raw image (a) into superpixels (b), and transform the pixel-wise saliency ground truth (c) into superpixel saliency ground truth (d) guided by the segmentation map from (b). Our goal is to design some representation of (b) that can model the heterogeneity of superpixels, then map the relationship between those representations and the superpixel ground truth (d). Note that while pixel-wise saliency ground truth values are binary (0 or 1), superpixel saliency ground truth values are continuous, ranging between 0 and 1. GT stands for ground truth.

incorporate heterogeneous positional information into the model. Let $\mathbf{C}_k \in \mathbb{Z}^{n_k \times 2}$ represent the Euclidean coordinates of the pixel indices belonging to the k^{th} superpixel, where n_k is the number of pixels in the k^{th} superpixel. We model the positional information with superpixel centroids $\mathbf{P} = [\mathbf{M}(\mathbf{C}_1), \mathbf{M}(\mathbf{C}_2), \dots, \mathbf{M}(\mathbf{C}_K)]$ where the $\mathbf{M}(\cdot)$ operator takes the mean position over the n_k points so that $\mathbf{M}(\mathbf{C}_k) \in \mathbb{Z}^2$.

We define a set of RGB values as a matrix, $\mathbf{X}_k$, for the k^{th} superpixel as

$$\mathbf{X}_k = \mathbf{X}[\mathbf{C}_k] \in \mathbb{R}^{n_k \times 3} \quad (3.1)$$

$$\mathbf{X}[\mathbf{C}_k] = \{\mathbf{x}_{i,j} : (i,j) \in \mathbf{C}_k\} \quad (3.2)$$

for row i and column j . Subsequently, we define a set of all K superpixel RGB values as $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K\}$.

The corresponding superpixel label $\mathbf{y}$ is defined as the average of the pixel-wise saliency binary values from the raw pixel labels associated with $\mathbf{C}_k$, as follows:

$$\mathbf{y} = [\mathbf{M}(\mathbf{Y}[\mathbf{C}_1]), \mathbf{M}(\mathbf{Y}[\mathbf{C}_2]), \dots, \mathbf{M}(\mathbf{Y}[\mathbf{C}_K])] \quad (3.3)$$

where $0 \leq \mathbf{M}(\mathbf{Y}[\mathbf{C}_k]) \leq 1$.

Therefore, given a superpixel segmentation map $\mathbf{L}$, a set of superpixel RGB values $\mathcal{X}$, and superpixel centroids $\mathbf{P}$, our goal is to minimize the difference between label $\mathbf{y}$ and prediction

$\hat{\mathbf{y}}$ by investigating effective strategies for addressing heterogeneity (components g and v in Section 4.1), and by designing a model f capable of handling heterogeneous representations (Section 4.2.4) while ensuring robust training and generalization (Chapter 5):

$$\hat{\mathbf{y}} = f(g(\mathbf{L}, \mathcal{X}), v(\mathbf{P})) \quad (3.4)$$

Superpixel predictions $\hat{\mathbf{y}}$ are mapped back to the regular pixel space $\hat{\mathbf{Y}}$ as follows:

$$\hat{\mathbf{Y}}[C_k] = \hat{\mathbf{y}}. \quad (3.5)$$

Having formulated the problem, we now explore the motivation behind leveraging superpixels for low-loss spatial compression. In doing so, we also articulate the key limitations of prior methods, particularly their inability to effectively handle the irregular and heterogeneous nature of superpixel representations.

3.2 Limitations and Motivations

Our motivation for utilizing superpixel segmentations comes from the fact that computational complexity partly scales to the model input spatial dimension. Superpixel segmentation not only reduces the spatial dimension, but has a superior preservation property that has a much lower reconstruction error compared to a simple down-sampling method (bilinear interpolation). For example, Fig. 3.2 illustrates an example of a comparison between image reconstruction (upsampling) via bilinear interpolation and superpixel segmentation. Bilinear interpolation estimates new pixel values based on a weighted average of the nearest four neighbours (in 2D). It assumes that the underlying function varies smoothly and approximates it using a piecewise bilinear function, which is a first-order approximation. Superpixel segmentation groups pixels into homogeneous regions, assigning all pixels within a segment the same value (e.g., the average colour or intensity). This process effectively approximates the function as a piecewise constant function, which is a zeroth-order approximation. Expanding to the whole DUTS [1] dataset, Fig. 3.3 shows that down-sampling the images from a 300×300 to a 32×32 resolution has a reconstruction mean absolute error (MAE) of 116.0, while superpixel segmentation with a similar compression of 1024 superpixels ($32 \times 32 = 1024$) only has a reconstruction mean absolute error of 6.2. Since superpixel segmentation does not perfectly align with the boundaries of pixel-wise saliency labels, the resulting pixel-wise prediction mapped from superpixel predictions (equation 3.5) may exhibit minor errors—even if the model predicts superpixel saliency with perfect accuracy. As illustrated in Fig. 3.4, our exploratory test reveals that

reducing the spatial resolution from 300×300 pixels to 625 superpixels (25×25) utilizing SLIC, only decreases the maximum achievable F1-score by 0.02, while reducing the spatial dimension by 99.3%. This demonstrates that superpixel representations are a compelling input-level reduction alternative for efficient SOD.

While traditional CNNs leverage pixel-level colour intensities effectively due to their inductive biases [119], colour alone may be insufficient to capture the heterogeneous nature of superpixel segmentations. Previous methods [113, 114, 115, 116, 118, 120] treat a set of superpixels as a homogeneous representation similar to pixels. However, a set of superpixels is heterogeneous in nature (as visually evident from Fig. 3.1 and Fig. 3.2), and its representation needs to reflect the heterogeneity.

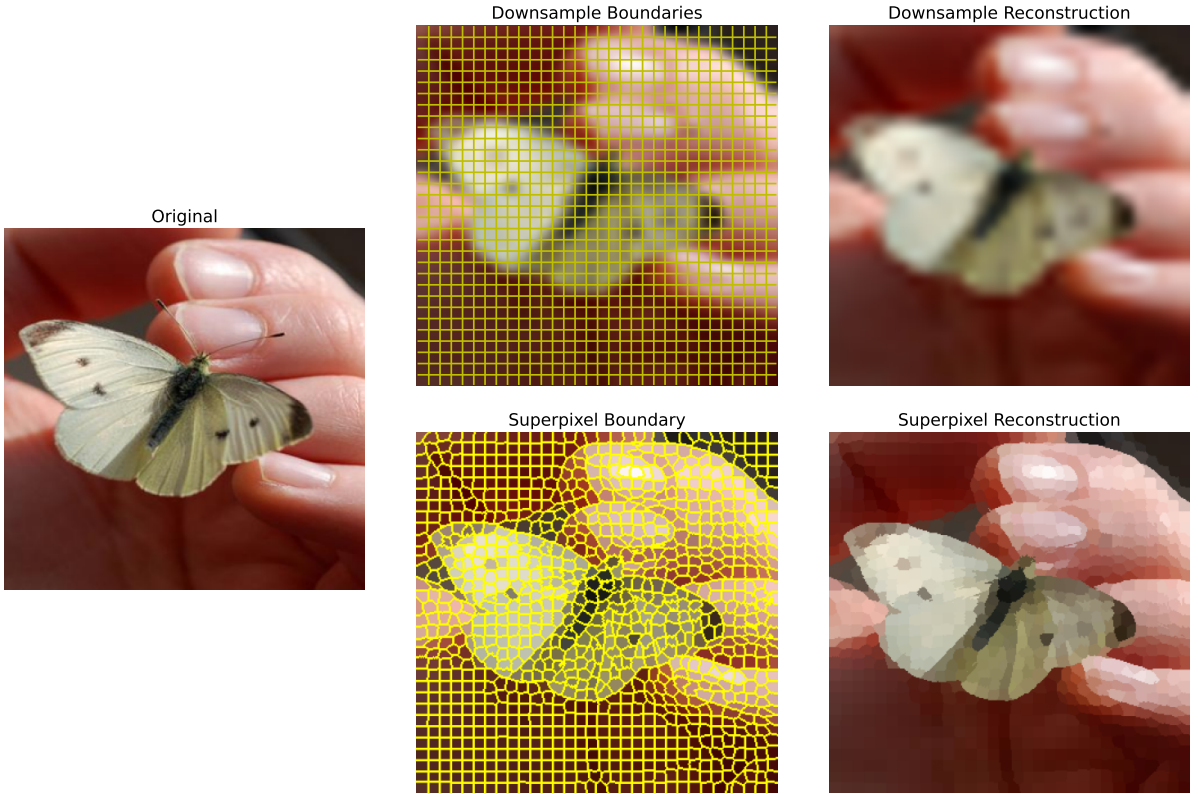


Figure 3.2: **Reconstruction Example.** Visualization of reconstruction via bilinear interpolation from downsample (top middle, top right) and superpixel segmentation (bottom middle, bottom right). Observe that superpixel segmentation captures the natural boundaries of objects superior to downsampling.

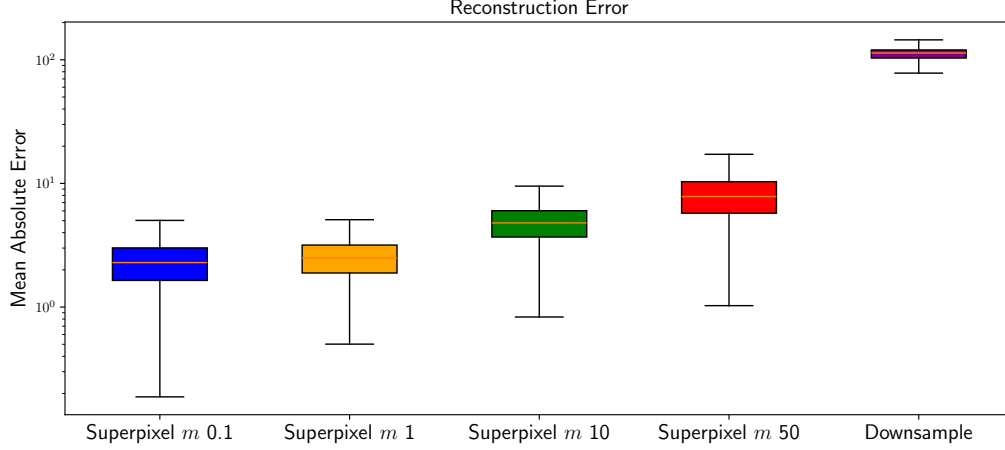


Figure 3.3: **Boxplot of Reconstruction Error.** Observe that superpixels better preserve information, even with zeroth-order reconstruction, compared to bilinear interpolation, which is a second-order approximation. m is the compactness parameter that controls the weight of spatial distance vs colour distance (from equation 2.1).

As mentioned in Section 2.3.2, previous methods such as sinusoidal positional encodings [32] and learnable positional embeddings [83] were designed primarily for homogeneous representations, where input elements share a uniform structure and role, such as words in a sentence or image patches. These methods struggle with heterogeneous representations, where input elements vary in type or importance, because they lack the flexibility to differentiate between diverse roles or account for varying relationships within the input. While Laplacian Positional Embeddings (LPEs) [94, 95, 96] offer a promising alternative for graphs by capturing structural information through eigenvectors of the graph Laplacian, they face significant scalability challenges. The eigenvalue decomposition required to compute LPEs is computationally expensive, with complexity scaling quadratically for graphs with a sufficiently large number of nodes. This makes LPEs impractical for our case where we plan to utilize a minimum of 3000 superpixels.

To address the lack of heterogeneous representation and positional embedding, incorporating feature attributes that account for the heterogeneous nature of superpixels and improving upon existing positional embedding is essential. Specifically, our goal is to address the heterogeneity of superpixels by integrating the size, shape, rotation, and location of each superpixel, thereby creating a more robust representation that enhances model accuracy while taking advantage of the efficiency of superpixel representation.

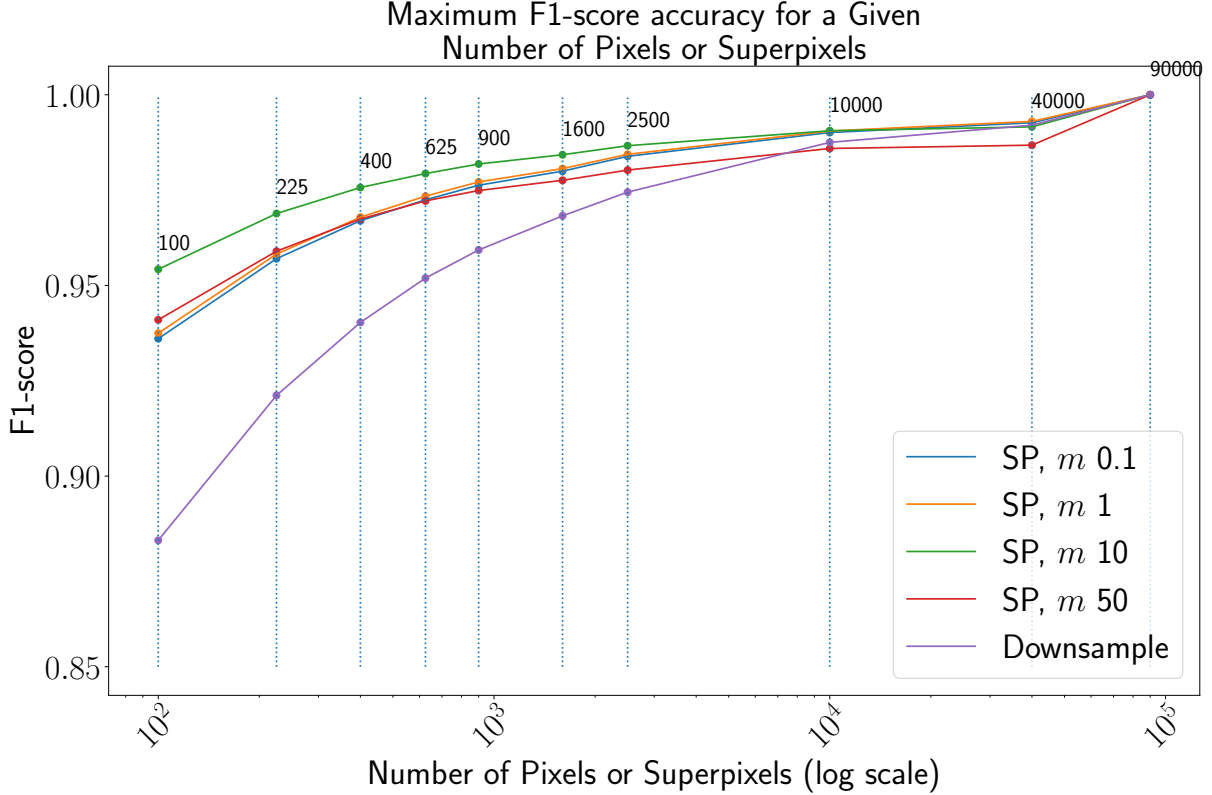


Figure 3.4: **Upperbound F1-score Accuracy for Superpixel Segments.** We empirically examine the relationship between the number of superpixels (SP) or number of pixels and the maximum achievable F1-score. Even with a relatively low number of segments (around 200 to 300), we maintain a high upper-bound F1-score accuracy of approximately 0.96 compared to downsampled images. m is the compactness parameter that controls the weight of spatial distance vs colour distance (from equation 2.1).

3.3 Datasets and Metrics

To evaluate the effectiveness and robustness of our proposed method SuperFormer, we conduct experiments on several widely used SOD datasets, covering both natural scenes and remote sensing images. Below, we provide an overview of each dataset, with a summary in Table 3.1.

Table 3.1: **Summary of datasets.** The datasets consist of natural scenes and remote sensing.

Dataset	N# Images	Split	Source	Additional Information
DUTS-Train [1]	10,553	Train	ImageNet DET	Includes diverse foreground/background variations
DUTS-TE [1]	5,019	Test	ImageNet DET	Same as DUTS-Train with test-specific images
DUTS-O [4]	5168	Test	ImageNet DET	One or more prominent objects set against a relatively complex background.
ECSSD [121]	1,000	Test	Various sources	Scenes that are both structurally intricate and semantically significant
HKU-IS [122]	4,447	Test	Various sources	Two or more objects set against different backgrounds.
PASCAL-S [123]	850	Test	PASCAL VOC	Provides object-centric annotations
ORSSD* [124]	600/200	Train/Test	Various sources	Emphasizes object regions
EORSSD* [20]	1,400/600	Train/Test	Various sources	Extends ORSSD with additional variations

* Remote sensing datasets.

Natural Scene Datasets

These datasets contain diverse images from real-world scenarios, commonly used in benchmarking salient object detection models.

- **DUTS** [1]: The DUTS dataset is the largest SOD dataset, consisting of 10,553 training images (DUTS-TR) and 5,019 test images (DUTS-TE). The dataset features complex backgrounds and varying object sizes, making it a challenging benchmark.
- **DUT-OMRON** [4]: This dataset contains 5,168 images with objects of diverse structures and cluttered backgrounds. It is particularly useful for evaluating the generalization of SOD models in challenging scenarios.
- **HKU-IS** [122]: The HKU-IS dataset comprises 4,447 images with multiple salient objects per image and minimal background noise, making it useful for assessing multi-object saliency detection.
- **ECSSD** [121]: The Extended Complex Scene Saliency Dataset (ECSSD) includes 1,000 images with visually salient and semantically meaningful objects in complex backgrounds, making it a challenging dataset for model evaluation.
- **PASCAL-S** [123]: This dataset consists of 850 images derived from the PASCAL VOC dataset. It includes human-eye fixation maps for saliency evaluation and is used for benchmarking models under human-like attention settings.

Remote Sensing Datasets

To extend our evaluation beyond natural scenes, we also assess performance on salient object detection in aerial imagery, where saliency characteristics differ significantly due to scale, resolution, and background complexity.

- **ORSSD** [124]: The Optical Remote Sensing for Salient Object Detection (ORSSD) dataset contains 800 images, featuring high-resolution aerial scenes with objects such as ships, airplanes, and buildings. The dataset presents a unique challenge due to small object sizes and high background clutter.
- **EORSSD** [20]: An extension of ORSSD, the Extended Optical Remote Sensing for Salient Object Detection (EORSSD) dataset includes 2,000 images with improved annotation quality, providing a more diverse and challenging benchmark for remote sensing SOD.

Metrics

To ensure a comprehensive evaluation, we focus on two primary metrics following the works of [10, 11, 12, 13, 77]. We report the maximum F1-score (F_β) which balances precision and recall with a 0.3 weight (β) on precision. Given predictions $\hat{\mathbf{Y}}$ with threshold τ , ground truth $\mathbf{Y}$, and indicator function $\mathbb{1}(\cdot)$ (which returns 1 if the condition inside is true, otherwise 0), we get true positive (TP), false positive (FP), and false negative (FN) as follows:

$$\hat{\mathbf{Y}}_{i,j}^\tau = \mathbb{1}(\hat{\mathbf{Y}}_{i,j} \geq \tau) \quad (3.6)$$

$$\text{TP} = \sum_i \sum_j \mathbb{1}(\hat{\mathbf{Y}}_{i,j}^\tau = 1 \wedge \mathbf{Y}_{i,j} = 1) \quad (3.7)$$

$$\text{FP} = \sum_i \sum_j \mathbb{1}(\hat{\mathbf{Y}}_{i,j}^\tau = 1 \wedge \mathbf{Y}_{i,j} = 0) \quad (3.8)$$

$$\text{FN} = \sum_i \sum_j \mathbb{1}(\hat{\mathbf{Y}}_{i,j}^\tau = 0 \wedge \mathbf{Y}_{i,j} = 1) \quad (3.9)$$

Then, we obtain Precision, Recall, and finally F1-score F_β as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.10)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.11)$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad (3.12)$$

τ is evenly spaced between 0 and 1 with 256 bins, so the maximum F1-score is the maximum of the 256 F_β values that are averaged across a dataset. We also report on mean absolute error (MAE)

$$\text{MAE} = \frac{1}{HW} \sum_i \sum_j |\mathbf{Y}_{i,j} - \hat{\mathbf{Y}}_{i,j}| \quad (3.13)$$

to measure the overall error.

To assess the model’s complexity and resource usage, we report metrics such as floating-point operations (FLOPs) and the number of model parameters. These metrics help in understanding the trade-offs between model accuracy and computational efficiency (memory and inference speed). Inference speeds (ms) are tested on single samples on an NVIDIA RTX 3090 GPU with an Intel i7 12700KF CPU.

In this chapter, we redefined pixel-to-pixel salient object detection as a superpixel-to-superpixel framework. We highlighted the advantages of superpixel representations, particularly their efficiency and low reconstruction error, while addressing the limitations of previous superpixel-based SOD methods. Additionally, we discussed diverse datasets and main metrics used to evaluate the effectiveness and robustness of SuperFormer. In the next chapter, we explore our approach to overcoming these limitations and examine the impact of our contributions.

Chapter 4

SuperFormer: Superpixel-based Vision Transformer

This chapter outlines the methodological framework developed to investigate **SuperFormer**, a superpixel-based vision Transformer. The approach highlights the efficiency of superpixels, while addressing fundamental questions about the heterogeneity of superpixels to ensure a comprehensive understanding of the problem domain.

This chapter is structured to achieve the following objectives:

- To design representation methods that improve salient object detection accuracy by specifically addressing the heterogeneity of superpixels.
- To transition an existing encoder-decoder framework with non-overlapping patches of pixels to superpixels.

To achieve these objectives, this chapter is organized into the following sections:

- [Section 4.1 Addressing Heterogeneity](#)¹: This section introduces methods to address the heterogeneity of superpixels, and validate its effectiveness through experiments.
- [Section 4.2 Model Expansion](#): This section elaborates on the model architecture and its motivation behind the design.

¹Section 4.1 and Section 4.2 includes the material from Section 3.2 and Section 3.3 of [48].

The rationale behind the selection of methods and tools is also presented, emphasizing the trade-offs made to balance theoretical rigor, computational efficiency, and practical applicability. This chapter aims to provide a clear and reproducible account of the research design, ensuring transparency and alignment with the broader research goals.

4.1 Addressing Heterogeneity

Heterogeneity in visual data poses a significant challenge for accurate and robust image analysis. Unlike pixels which are represented in an even grid-like structure, superpixels have variations in texture, shape, and positional attributes. This problem necessitate methods that can effectively capture and integrate such diverse features. This section discusses the strategies employed to address these variations, ensuring that the proposed approach can generalize across complex and diverse images. Specifically, we address how colour and texture are incorporated in Subsection 4.1.1, superpixel shapes in Subsection 4.1.2, and superpixel locations in Subsection 4.1.3.

4.1.1 Colour and Texture Representation

Recall from Section 3.2 that using the mean RGB values of superpixels retains significantly more information than traditional image down-sampling. Building on this finding, we use the mean and standard deviation of the intensity of each channel for each superpixel segment to represent both the overall RGB values and their variance.

One major drawback of averaging RGB values is the loss of texture information. To mitigate this, we follow the approach of [120] and integrate Local Binary Patterns (LBP) [40]. Specifically, we construct a 16-dimensional feature vector $g^c(\mathbf{X}_k) \in \mathbb{R}^{16}$, comprising the mean (3D), standard deviation (3D), and LBP features (10D), ensuring a richer representation of each superpixel. $g^c(\mathbf{X}_k)$ will be the baseline model as we cumulatively add features that address the heterogeneity of superpixels in Subsection 4.1.4.

4.1.2 Shape Representation

Superpixels, by design, aim to provide a compact and perceptually meaningful representation of an image by grouping pixels into coherent regions. However, the shapes of these superpixels are inherently heterogeneous because they represent small, irregularly segmented

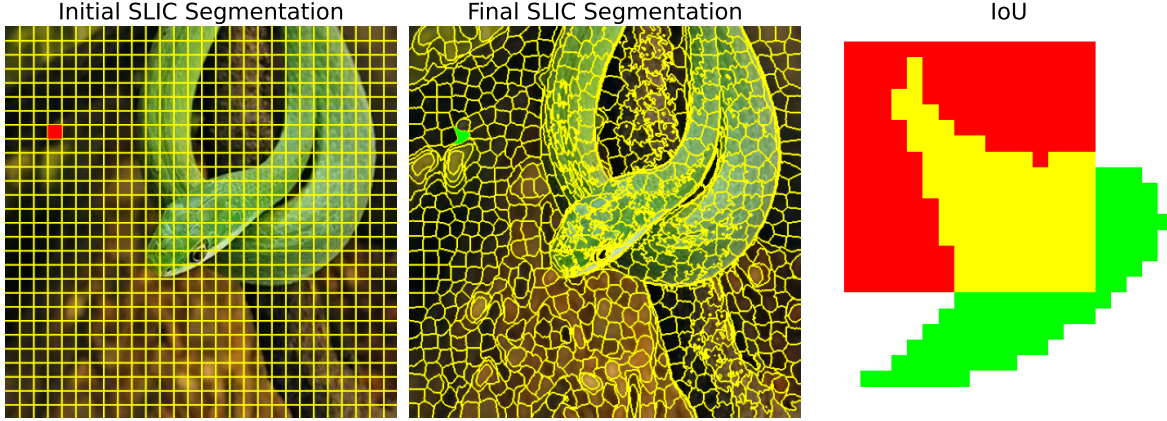


Figure 4.1: **SLIC Initialization.** This figure demonstrates how we define heterogeneity for a given superpixel. SLIC initializes the segmentation (left) and fits to the contour of the image (middle). We utilize the initial SLIC segmentation to quantify the heterogeneity (right) by computing the complement of intersection-over-union between the initialized segmentation and the final segmentation. Red represents an arbitrary superpixel from the initial SLIC segmentation, green represents the same superpixel from the final SLIC segmentation, and yellow represent the intersection between the superpixels from the initial and final SLIC segmentation. Zooming in is recommended to identify the red superpixel (left) and green superpixel (middle).

regions of the underlying image. Unlike uniform grid-based representations, superpixels have spatial irregularity as they adapt to local image structures.

We hypothesize that the heterogeneity in superpixel shapes is strongly correlated with the underlying saliency of the image. To quantify this correlation, we define shape heterogeneity as the complement of the intersection-over-union (IoU) between a superpixel in the initial SLIC segmentation and its corresponding region in the final SLIC segmentation. Given a final SLIC segmentation map $\mathbf{L}$ from Section 3.1, an initial SLIC segmentation map $\mathbf{L}^g$, a set of pixels belonging to the k^{th} final SLIC segmentation map $\mathbf{L}_k$, a set of pixels belonging to the k^{th} initial SLIC segmentation map $\mathbf{L}_k^g$, cardinality (number of elements) $|\cdot|$, IoU complement is defined as

$$\text{IoU}_k^c = 1 - \frac{|\mathbf{L}_k \cap \mathbf{L}_k^g|}{|\mathbf{L}_k \cup \mathbf{L}_k^g|}. \quad (4.1)$$

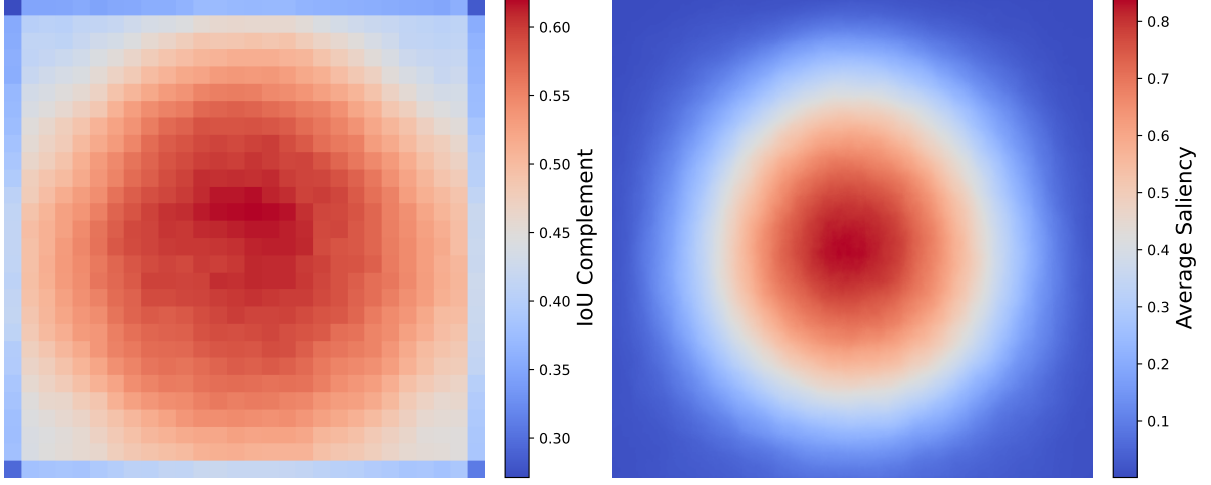


Figure 4.2: **Shape Heterogeneity of DUTS** [1]. Observe that the shape heterogeneity of superpixels (left) are correlated to the saliency (right). Specifically, salient objects are dense around the center of the image, and the heterogeneity is similarly dense around the center of the image. This underlying correlation indicates that variations in superpixel boundaries may carry meaningful information about object saliency.

where a value closer to 1 indicates a superpixel that undergoes significant deformation relative to its initial square-like SLIC initialization, while a value closer to 0 indicates a superpixel that remains largely unchanged in shape. To aid with the visualization of shape heterogeneity, we turn to Fig. 4.1 for an example. The IoU complement is applied for each superpixel label k , where $k \in \{1, 2, \dots, K\}$, and over all of K we have

$$\text{IoU}^c = [\text{IoU}_1^c, \text{IoU}_2^c, \dots, \text{IoU}_K^c]. \quad (4.2)$$

To understand how superpixel heterogeneity varies with respect to their initialization location within the image, the $\overline{\text{IoU}^c}$ vector is averaged over the whole dataset (with N images) as a final step:

$$\overline{\text{IoU}^c} = \frac{\sum_i^N \text{IoU}_i^c}{N} \quad (4.3)$$

Similarly, to analyze the spatial distribution of salient objects within an image, we compute the average saliency map $\bar{\mathbf{Y}}$ across the dataset by taking the mean of the ground truth saliency:

$$\bar{\mathbf{Y}} = \frac{\sum_i^N \mathbf{Y}_i}{N} \quad (4.4)$$

In Fig. 4.2, we can visually observe that superpixels in high-saliency regions (i.e., those containing objects of interest) often exhibit more irregular and complex boundaries, as segmentation methods attempt to capture fine details and object edges. In contrast, background regions tend to have larger, more uniform superpixels, reflecting smoother, less informative areas. This implicit correlation suggests that superpixel shape boundary variation encodes valuable cues about object saliency.

However, without explicitly addressing this variability, a model may struggle to leverage shape information effectively, leading to inconsistencies in saliency prediction. Therefore, a robust shape representation is crucial—not only to preserve the discriminative nature of superpixel geometries but also to ensure that saliency cues embedded in shape heterogeneity are utilized in an effective and computationally efficient manner.

Fourier Descriptors

Fourier descriptors [44, 45, 46, 125, 126] are an effective tool for capturing the intrinsic geometric properties of shapes, such as size, shape, and rotation. By transforming a shape into a frequency domain representation, Fourier descriptors provide a compact yet robust way to describe complex contours. This makes them particularly well-suited for characterizing the variability of superpixels, which often exhibit irregular shapes and varying orientations. Low-frequency coefficients captures the overall shape, whereas high-frequency coefficients captures finer details, texture, and noise. Keeping only the low frequency components smooths out irregularities, minimizing the impact of noise and variations in sampling density, thus providing a more stable representation compared to raw coordinates. The amplitude of Fourier Transform captures the **size** and **shape** of the superpixel.

To represent size and shape, we first sample R points from each superpixel contour as a set of Euclidean coordinates and project each coordinate onto the complex plane:

$$\begin{aligned} \mathbf{U} &= \text{ComplexContour}(\mathbf{L}) \\ &= [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K], \\ \mathbf{u}_k &= [\mathbf{u}_{k,1}^x + i\mathbf{u}_{k,1}^y, \mathbf{u}_{k,2}^x + i\mathbf{u}_{k,2}^y, \dots, \mathbf{u}_{k,R}^x + i\mathbf{u}_{k,R}^y] \in \mathbb{C}^R \end{aligned} \tag{4.5}$$

Then, we apply the Fourier Transform to extract the amplitude, but remove the DC (zero-frequency) coefficient. The coefficients that remain, after removing DC coefficient, will be translation invariant (location will be addressed in Section 4.1.3). Applying the Fourier Transform and then extracting the amplitude ($|\cdot|$) effectively captures the size, and shape of the superpixels. Fig. 4.5 aids visualization by illustrating the behavior of amplitude under various transformations. Column 2 highlights the translation invariance of amplitude

when the DC coefficient is removed, a feature accounted for in Section 4.1.3. Column 3 demonstrates how the amplitude reflects the size of the superpixel, while Column 4 shows the minimal amplitude changes (due to discretizations) associated with superpixel rotation.

The Fourier Transform decomposes a signal (or shape) into a set of frequency components, and the energy of these components indicates how much information each coefficient contributes. The cumulative energy can be measured to determine how many coefficients are sufficient to capture most of the signal’s content. To analyze the energy distribution of Fourier Transform coefficients, we computed the Fourier Transform for approximately 660,000 superpixels extracted from 1,000 DUTS-TE [1] images. For each superpixel, the Fourier coefficients were calculated and used to evaluate the cumulative energy preservation, allowing us to determine the proportion of signal energy captured by the top r coefficients.

The cumulative energy preserved by the first r coefficients (omitting k from the notation for simplicity) is defined as

$$E_r = \sum_{i=0}^{r-1} \text{amp}_i. \quad (4.6)$$

To measure the proportion of energy preserved, the normalized cumulative energy is computed as

$$\text{Cumulative Energy Ratio} = \frac{E_r}{E_R}. \quad (4.7)$$

Fig. 4.3 illustrates the effect of selecting the first r Fourier coefficients on the representation of superpixel contours using Fourier Descriptors. Fourier coefficients are ordered by frequency, with low-frequency coefficients encoding global structural information and high-frequency coefficients capturing finer details of the contour. We choose r to be 10, which preserves most of the global structure while removing the fine details of the superpixel shapes.

In Fig. 4.4, we observe that retaining only a small number of coefficients (e.g., the first 2) captures the general shape of the contour, preserving its global features. However, as r increases, more high-frequency coefficients are included, leading to a more detailed and accurate reconstruction of the contour. Although using more Fourier coefficients enhances the detail captured in the representation, it also risks overfitting by capturing noise. The diminishing returns in representation improvement beyond a certain number of coefficients are evident from Fig. 4.3. The majority of the contour’s energy and structure is concentrated in the low-frequency coefficients, with higher-order coefficients contributing

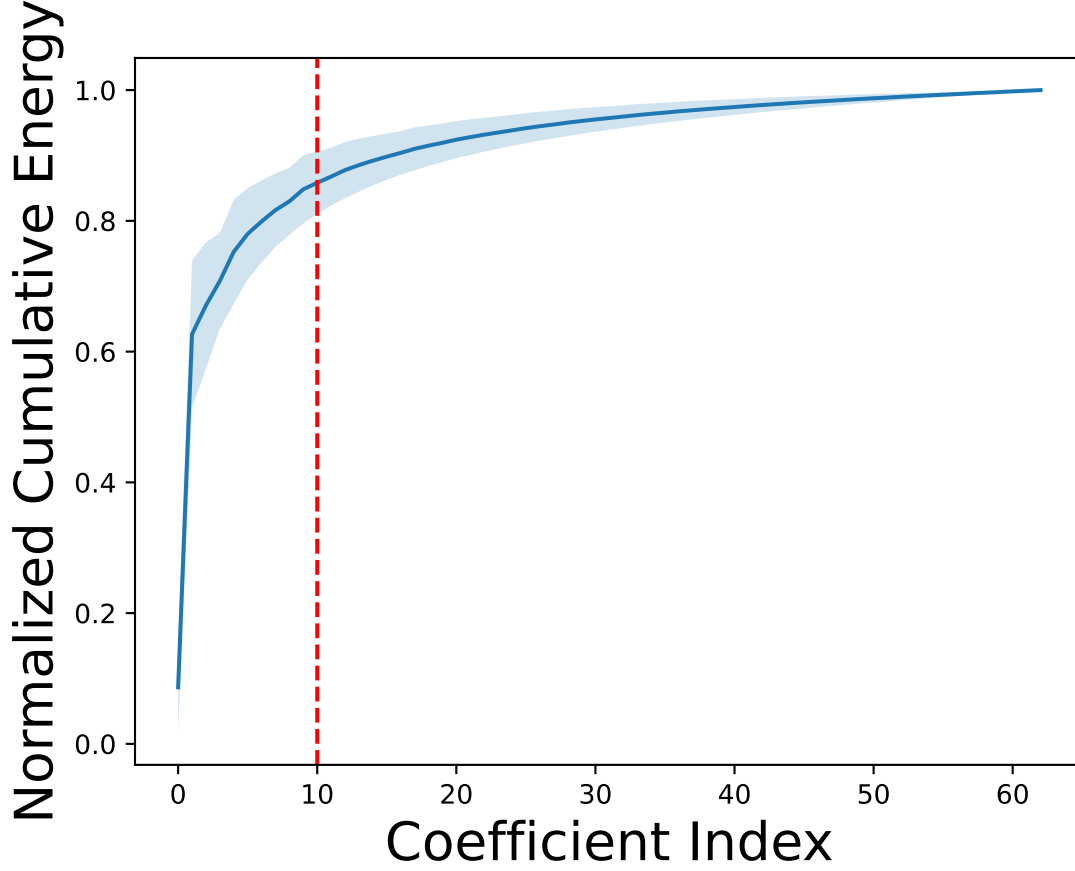


Figure 4.3: **Fourier Coefficients Cumulative Energy**. Normalized cumulative energy distribution of Fourier coefficients across approximately 660,000 superpixels (1000 images) from DUTS-TE [1]. The plot highlights that the majority of the energy is concentrated in the low-frequency coefficients (i.e. 10 coefficients noted by the dotted red line), indicating that these coefficients capture the most significant information about the superpixel contours. This observation supports the use of a reduced number of coefficients for efficient representation without significant loss of detail. The shaded region above and below each data point represents one standard deviation from the 660,000 superpixels.

marginally to finer details. This reinforces the importance of using methods like cumulative energy analysis to determine an optimal r that captures a desired proportion of the contour’s information while minimizing noise.

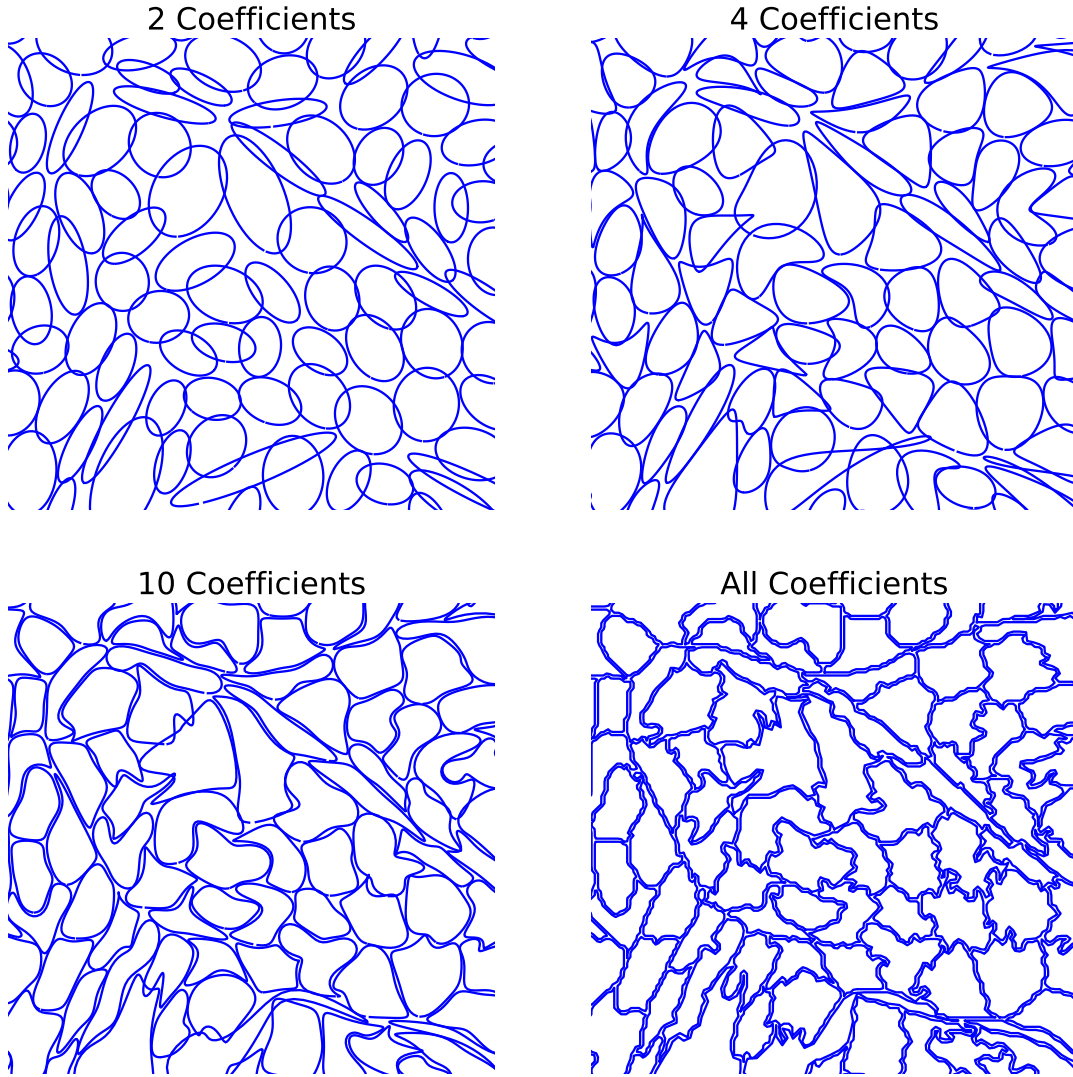


Figure 4.4: **Efficient Shape Representation.** The impact of selecting the first r Fourier coefficients on the representation of superpixel contours using Fourier Descriptors. The plot demonstrates how increasing r progressively captures more details of the contour, with the low-frequency coefficients preserving the global structure and the high-frequency coefficients refining finer details. Based on the findings from Fig. 4.3, we use 10 coefficients in this thesis to preserve most of the structure without overfitting to the fine details of the superpixel shapes.

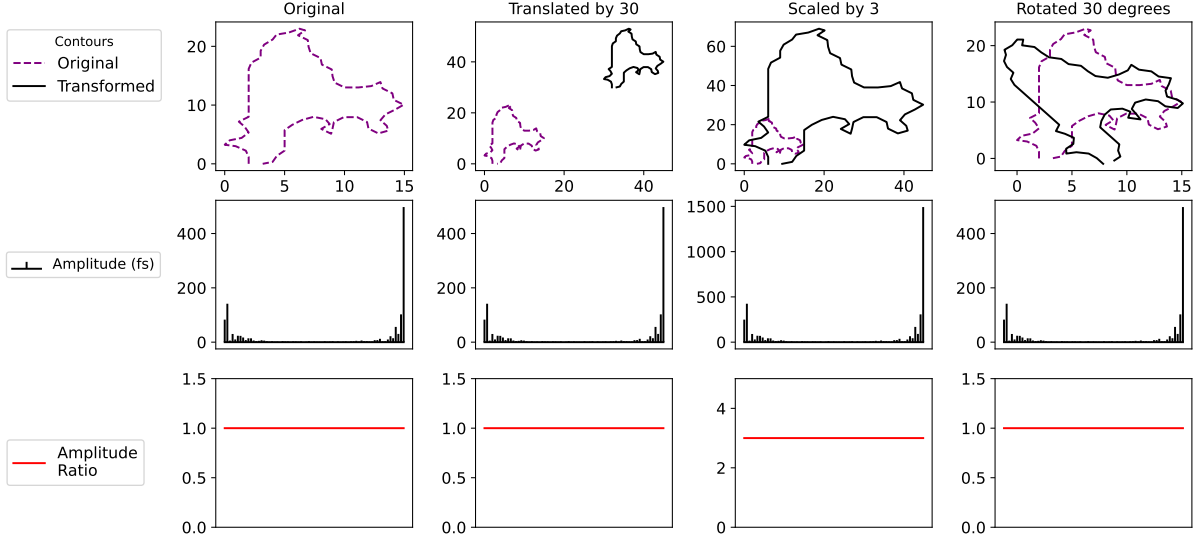


Figure 4.5: **Fourier Descriptors**. Given a sample superpixel contour (1st column), the amplitudes of the Fourier Transform are shown for translation (2nd column), scale (3rd column), and rotation (4th column) in the second row. The amplitude ratio between the original and transformed contours (third row) reveal that translation causes no changes, scaling affects the amplitude, and rotation causes no changes.

The phase component of Fourier descriptors is highly sensitive to the starting point of the contour sequence, meaning that any change in the starting position results in a phase shift. This lack of invariance can introduce inconsistencies when applying feature augmentations (Section 5.1) such as rotation, flipping, or shifting, as the representation may vary significantly based on the chosen starting point. Therefore, to leverage the size and shape information captured by Fourier descriptors while maintaining invariance to the contour’s starting point, we exclude the phase component and instead incorporate central moments [41, 42, 43].

Central Moments

Central moments provide a more stable alternative for capturing shape characteristics, as they remain invariant to translation and changes in sequence ordering. This makes them particularly well-suited for representing attributes such as mirror symmetry and rotation, ensuring that the extracted features remain consistent regardless of the contour’s starting point.

To account for the rotation and mirror symmetry of the superpixels, we employ central moments. Moments are widely used in shape analysis as they provide a compact and invariant representation of an object’s geometry. Central moments serve as essential tools for describing shape properties while ensuring invariance to translation (which will be addressed in Section 4.1.3). Central moments μ_{pq} with order of x -deviation p and y -deviation q are computed relative to the centroid of the shape $(\bar{x}, \bar{y})$ to achieve translation invariance. Given a set of pixel coordinates in superpixel k , $\mathbf{C}_k$, a central moment is defined as

$$\mu_{pq,k} = \sum_{(x,y) \in \mathbf{C}_k} (x - \bar{x})^p (y - \bar{y})^q, \quad (4.8)$$

We use up to order three, where first order is removed since it always returns zero. The zeroth order (μ_{00}) measures the area of the superpixel. The second order (variance, $\mu_{11}, \mu_{20}, \mu_{02}$) quantifies the spread or dispersion of data points around the mean. A higher variance indicate that the shape is more spread out in that respective direction. The third order (skewness, $\mu_{03}, \mu_{30}, \mu_{12}, \mu_{21}$) measures the asymmetry of a distribution. The values of these orders provide information about the orientation of the shapes and offer advantageous properties, enabling direct transformations of the moments without requiring reapplication of superpixel segmentation after augmenting the images (Section 5.1). Combining the amplitude of Fourier descriptors and central moments, we have

$$g^s(\mathbf{L}) = [\text{amp}_{k,1}, \text{amp}_{k,2}, \dots, \text{amp}_{k,R-1}, \mu_{00,k}, \mu_{11,k}, \dots, \mu_{21,k}]. \quad (4.9)$$

The shape representation g^s and colour representation g^c are concatenated to form the input embedding $\mathbf{X}^e$. Then the input embeddings $\mathbf{X}^e$ are linearly projected with $\mathbf{W}^a$, to form the input for the Transformer-based model in Section 4.2.1:

$$\mathbf{X}^e = \text{Concat}(g^s(\mathbf{L}), g^c(\mathcal{X})) \quad (4.10)$$

$$g(\mathbf{L}, \mathcal{X}) = \mathbf{X}^e \mathbf{W}^a \quad (4.11)$$

In summary, Fourier descriptors and central moments provide a compact yet expressive encoding of each superpixel’s shape, capturing both global contours and spatial distribution. While these shape descriptors offer critical geometric cues, effective modeling of superpixel relationships also requires a precise understanding of their spatial arrangement—motivating our next component: positional embeddings based on superpixel centroids.

4.1.3 Dynamic Centroid Positional Embedding

In addition to the heterogeneity of superpixel shapes, we hypothesize that spatial location—specifically, where each superpixel is positioned within the image—is another im-

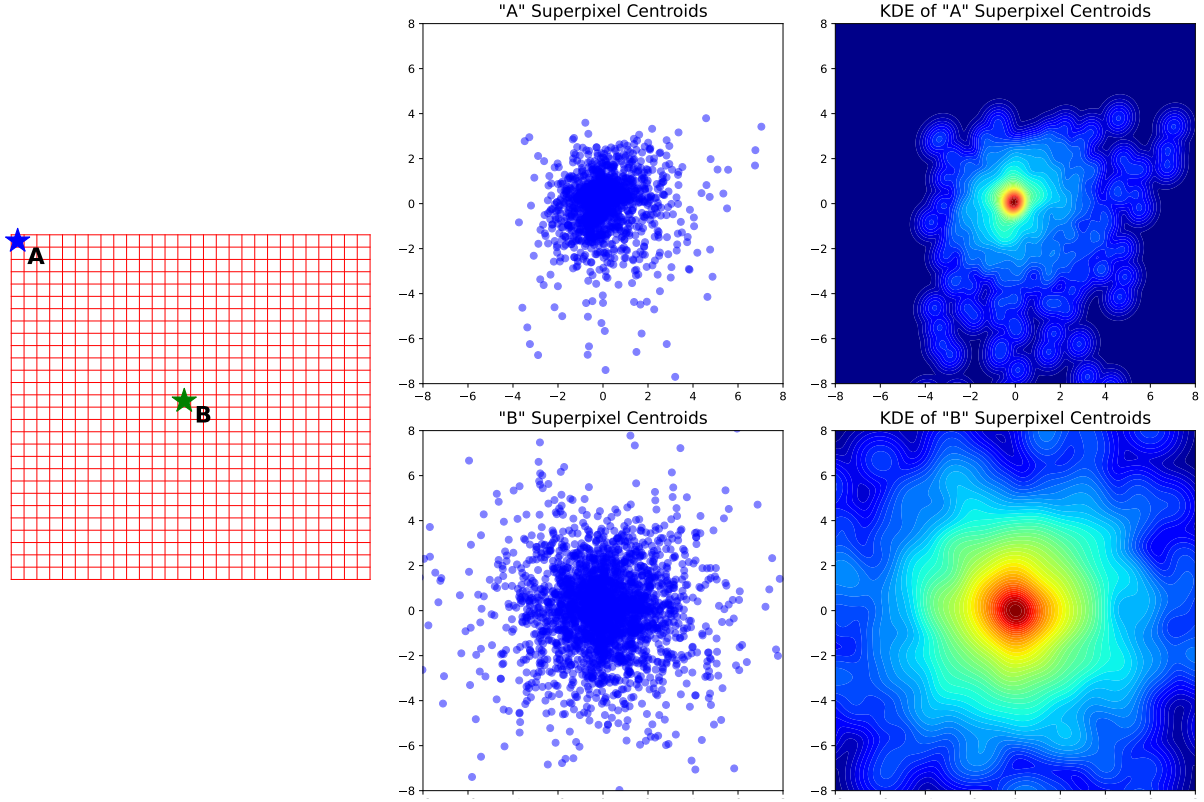


Figure 4.6: **Centroid Variance.** On the left, we show the locations of superpixel indices for “A” and “B.” “A” corresponds to the first superpixel index, initialized in the top-left corner of the image, while “B” represents a middle superpixel index, initialized near the center of the image. In the middle, scatterplots display the mean-shifted centroids for “A” and “B” across the DUTS dataset [1]. On the right, we present the continuous kernel density estimation (KDE) contour, with quadratic root scaling applied to enhance distribution distinctions. The variation in centroid distribution reveals how the centroids of a given superpixel index shift based on image location. Notably, these variations align with the shape heterogeneity observed in Fig. 4.2.

portant source of heterogeneity that must be addressed to improve performance on salient object detection (SOD). This is because the superpixel corresponding to a given index can vary in location across different images. Specifically, with homogeneous positional encodings, the k^{th} superpixel will always be perturbed by the same positional vector, but the position (centroids) of the k^{th} superpixel varies from image to image. In Fig. 4.6, we

illustrate the variance for two different centroid locations consistent with the findings from Fig. 4.2 where the superpixels segments on the corner of the image (noted “A”) have a much lower variance than superpixel segments in the middle of the image (noted “B”). Superpixel heterogeneity varies across the image, with greater diversity in the center due to the prevalence of salient objects. This variation presents a challenge for homogeneous positional embeddings, which typically apply the same positional perturbation to identical tokens. To address this, the model should incorporate spatially adaptive mechanisms that account for the differing levels of heterogeneity, ensuring more effective feature representation and improved performance.

As illustrated in Section 3.2 and Section 2.3.2, current SOTA learnable positional embeddings [83, 92] or sinusoidal positional encodings [32] typically rely on two approaches: they either assume a uniform positional relationship, where token indices are consistent across spatial or sequential domains, or they use a dictionary of learnable parameters to cover all possible absolute and relative positions. However, both sinusoidal positional encodings and learnable positional embeddings were designed for homogeneous representations where an arbitrary index implied that the position of an index remained the same for each input sequence. To address the heterogeneity of graphs, Laplacian positional embeddings [94, 95, 96] have been introduced, but the eigenvalue decomposition in its process does not scale well (cubic or quadratic to the number of nodes), making them impractical for lightweight deployment with large number of nodes. Thus, a method that effectively encodes positional information given the heterogeneous arrangement of superpixels that can scale well to the number of nodes (i.e. superpixels) is needed. To address these challenges, we propose Dynamic Centroid Positional Embedding (DCPE) that offers a solution by projecting positional embeddings based on the centroids of each superpixel, adapting to their irregular spatial placement.

Given the Euclidean coordinates of all superpixel centroids $\mathbf{P} \in \mathbb{R}^{K \times 2}$, our goal is to further address the heterogeneity by infusing positional information to each token as a function of the centroids rather than homogeneous index. To specifically address the locational heterogeneity of superpixels, we employ learnable parameters $\mathbf{W}^c \in \mathbb{R}^{2 \times d_1}$. These parameters allow for a linear projection of the centroids into a d_1 -dimensional vector space:

$$v(\mathbf{P}) = \mathbf{P} \cdot \mathbf{W}^c \quad (4.12)$$

$v(\mathbf{P})$ can now perturb each token of (4.11) to its corresponding centroid (heterogeneous), and not to a deterministic index in a sequence (homogeneous).

In summary, this section provided an in-depth discussion on addressing the heterogeneity of superpixels using Fourier descriptors, central moments, and dynamic centroid

Table 4.1: **Overview of Heterogeneity Ablations.** This table illustrates the impact of addressing the heterogeneity of superpixels. Each model variant (SF-XS discussed in Section 5.3) is reported with sample standard deviation from five different seeds on the DUTS-TE [1] dataset. Our contributions are highlighted in **Bold**.

Model Variants	MAE ↓	F1-score ↑
Base	$0.099 \pm 1\text{e-}3$	$0.742 \pm 3\text{e-}3$
Base + Sinusoidal [32]	$0.095 \pm 2\text{e-}3$	$0.742 \pm 2\text{e-}3$
Base + LPE [83]	$0.094 \pm 9\text{e-}4$	$0.743 \pm 4\text{e-}3$
Base + LaPE [94]	N/A	N/A
Base + DCPE	$0.088 \pm 4\text{e-}4$	$0.747 \pm 2\text{e-}3$
Base + DCPE + FD	$0.081 \pm 6\text{e-}4$	$0.762 \pm 5\text{e-}4$

positional embeddings, highlighting their potential to capture diverse spatial and structural variations effectively. In the next section, we will present ablation results to evaluate the contributions of these methods, demonstrating their individual and combined impact on improving performance across SOD.

4.1.4 Ablation Studies

We conduct a series of ablation studies on a version of SuperFormer (SuperFormer-XS, details in Section 5.3) to assess the incremental impact of each contribution from this chapter. We start by providing brief descriptions of these components.

- **Base.** Base model. Mean, standard deviation of RGB values and texture (LBP) from Section 4.1.1 and rotary positional embeddings from Section 4.2.1.
- **Base + Sinusoidal.** Base model with sinusoidal positional encoding from Vaswani et al. [32].
- **Base + LPE.** Base model with learnable positional embeddings from Devlin et al. [83].
- **Base + LaPE.** Base model with laplacian positional embeddings from Dwivedi and Bresson [94].
- **Base + DCPE.** Base model with DCPE from Section 4.1.3.

- **Base + DCPE + FD.** Base model with DCPE, Fourier descriptors, and central moments from Section 4.1.2.

Table 4.1 presents the performance of each incremental model, highlighting their impact on the performance of SuperFormer, averaged over five random seeds. The following observations can be made.

Importance of DCPE. To understand the impact of DCPE (Section 4.1.3), we compare DCPE with two mainstream homogeneous positional embeddings and a well-known positional embedding used in graph transformers. Note that sinusoidal PE and LPE still improves the performance by a small margin since superpixels inherently have a small degree of structure. However, when we replace these homogeneous positional embeddings with DCPE, we observe a much larger F1-score and MAE improvement, because we model the positional embedding as a function of its centroids as opposed to its index. Note that results for Laplacian embeddings are omitted, as the eigendecomposition of an adjacency matrix with over 3,000 nodes exceeds 2 seconds during inference. This is several orders of magnitude slower than the typical inference time of lightweight models which generally operate in the range of tens of milliseconds. As such, Laplacian positional embeddings are impractical within this performance regime.

Importance of Shape Descriptors. To validate the importance of superpixel shape attributes, we incorporate Fourier descriptors and central moments from Section 4.1.2 into the model variant with DCPE. The results show a significant improvement upon adding shape descriptors, with performance gains comparable to those from DCPE.

Overall, our results address the fundamental questions of superpixel heterogeneity while achieving high efficiency and high performance through the use of superpixels. This underscores the crucial role of both DCPE and shape descriptors in the effectiveness of SuperFormer. Having explored the importance of addressing heterogeneity in this section, we now turn to the design of the model.

4.2 Model Expansion

In this section, we introduce the model framework, which builds on foundational principles to enhance representational capacity, and computational efficiency. The model is structured into distinct yet interdependent components that work in tandem to achieve these goals. The overview of the subsections is as follows:

- [Section 4.3.1 Encoder](#): The encoder is responsible for extracting meaningful features from raw input data, transforming it into a compact and informative representation. This subsection details the encoder’s structure, including its layers, mechanisms for capturing local and global patterns, and strategies to handle variability in input data.
- [Section 4.3.2 Decoder](#): The decoder maps the encoded representation back into the target output domain, ensuring accurate prediction. This subsection explores the design of the decoder, focusing on its capacity for contextual reasoning, precision in output generation, and its integration with the encoder.
- [Section 4.3.1 Overall Model Architecture](#): The integration of the encoder and decoder within the broader model architecture is crucial for cohesive operation. This subsection provides a comprehensive overview of the architecture, and discussing the flow of information. We also demonstrate two key findings: first, the performance gain does not stem solely from the transformer architecture, but from the synergy between it and our superpixel-based representation, particularly the design of the superpixel extraction module. Second, the transformer-based network is essential, as CNNs do not perform well with superpixel representations.

Through this structure, the model expansion enhances the ability to maintain high efficiency and accuracy.

4.2.1 Encoder

The decision to employ Swin Transformers [33] is rooted in efficiency. With its linearly scaling computational complexity relative to spatial dimensions, the Swin Transformer plays a key role in the overall efficiency of SuperFormer. Unlike standard Swin Transformers, we do not split the input tensor into non-overlapping patches, as the spatial dimensions have already been significantly reduced. Instead, each token represents the descriptive attributes of a superpixel.

Our key contribution to the model is replacing non-overlapping pixel patches with superpixels. As observed in Section 3.2, this approach reduces token complexity by grouping similar-coloured pixels while incorporating additional information to address the heterogeneity introduced by transforming pixel patches into superpixels. This ensures that each token effectively represents the descriptive attributes of a superpixel. Notably, the core structure of the SWIN Transformer and Mix-Attention remains unchanged from the original design.

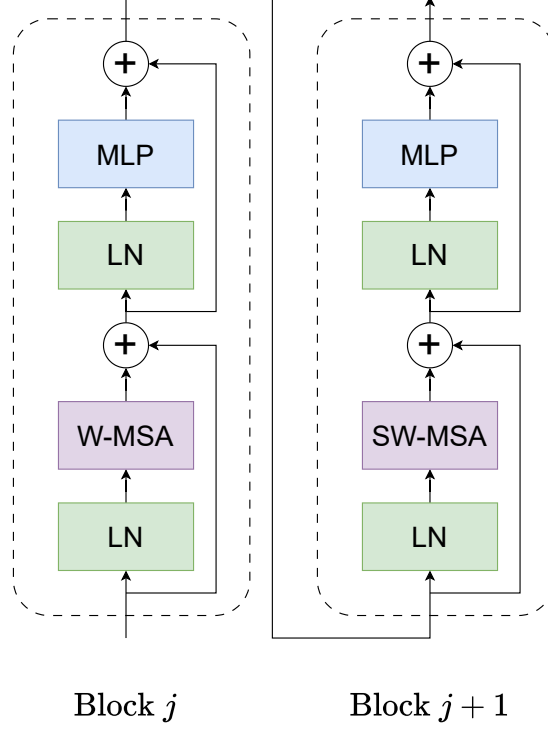


Figure 4.7: **Swin Transformer Block.** Two consecutive Swin Transformer blocks ($j, j + 1$) are used, where W-MSA and SW-MSA represent multi-head self-attention modules utilizing regular and shifted windowing configurations, respectively. This diagram is adopted from [33] (Fig.3b, page 4)

Given K superpixels, we assume H^s, W^s to be the height and width of the superpixel segmentation based on the SLIC initialization where $H^s \times W^s = K$. Multiple Transformer blocks utilizing a modified self-attention mechanism, known as Swin Transformer blocks, are applied to these superpixel tokens. These blocks preserve the token count and, along with the linear embedding, collectively form what is termed “Stage i ”.

To construct a hierarchical representation, the number of tokens is progressively reduced by patch merging layers as the network deepens. The first patch merging layer combines the features of each (2×2) group of neighboring patches and applies a linear layer to the resulting $4d_i$ -dimensional concatenated features. This process decreases the token

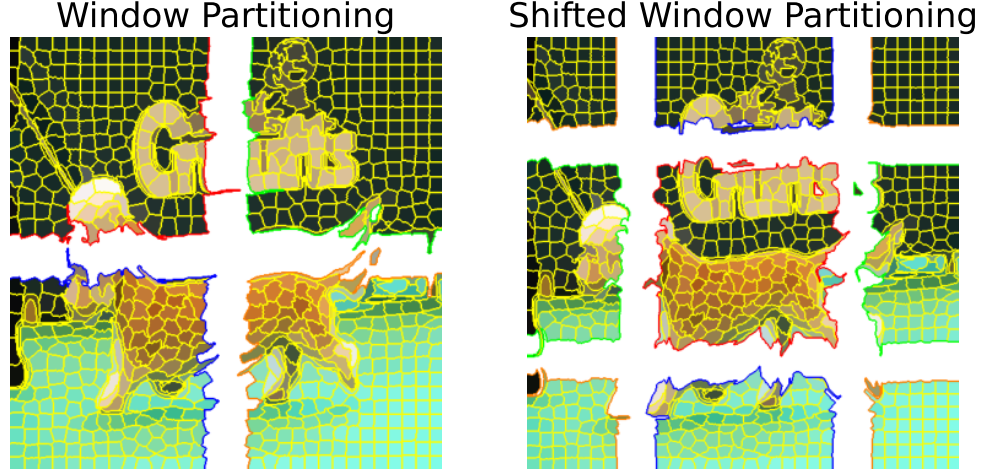


Figure 4.8: **Window Partitioning.** The shifted window approach for self-attention computation in the Swin Transformer architecture is illustrated here. In layer j (left), a standard window partitioning method is used, where self-attention is applied within each window. In the subsequent layer $j+1$ (right), the partitioning shifts, forming new windows. This shift allows self-attention in the new windows to span across the boundaries of the previous windows from layer j , enabling inter-window connections.

count by a factor of 4 (equivalent to $2\times$ downsampling of the resolution), producing an output dimension of d_{i+1} (except for the last stage, which does not apply downsampling). These stages collectively generate a hierarchical representation, maintaining feature map resolutions similar to conventional convolutional networks such as VGG [82] and ResNet [36].

The Swin Transformer block is designed by substituting the standard multi-head self-attention (MSA) module in a traditional Transformer block with a shifted window-based attention mechanism, while keeping the other layers unchanged. As shown in Fig. 4.7, a Swin Transformer block comprises a shifted window MSA module, followed by a two-layer multilayer perceptron (MLP) with a GELU activation [127] function in between. Additionally, LayerNorm (LN) [90] is applied before both the MSA module and the MLP, with residual connections incorporated after each module.

The standard Transformer architecture [32] and its adaptation for image classification

[34] employ global self-attention, where each token interacts with all other tokens. This global computation results in quadratic complexity concerning the number of tokens, making it impractical for many vision tasks that require handling a large number of tokens for dense predictions or high-resolution images.

As shown in Fig. 4.8, the initial module employs a standard window partitioning method, beginning from the top-left pixel. Given a window size of M , the 28×28 feature map is uniformly divided into 2×2 windows, each measuring 14×14 ($M = 14$). In the subsequent module, the window configuration is shifted relative to the previous layer by offsetting the windows by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels from the original partitioning.

With W-MSA, SW-MSA, $\hat{\mathbf{E}}_j$, and $\mathbf{E}_j$ referring to window-based multi-head self-attention, shifted window-based multi-head self-attention, output features of the (S)W-MSA module, and output features of the MLP module, respectively, consecutive Swin Transformer blocks with the window and shifted window partitioning are computed as described in [33]

$$\begin{aligned}\hat{\mathbf{E}}_j &= \text{W-MSA}(\text{LN}(\mathbf{E}_{j-1})) + \mathbf{E}_{j-1} \\ \mathbf{E}_j &= \text{MLP}(\text{LN}(\hat{\mathbf{E}}_j)) + \hat{\mathbf{E}}_j \\ \hat{\mathbf{E}}_{j+1} &= \text{SW-MSA}(\text{LN}(\mathbf{E}_j)) + \mathbf{E}_j \\ \mathbf{E}_{j+1} &= \text{MLP}(\text{LN}(\hat{\mathbf{E}}_{j+1})) + \hat{\mathbf{E}}_{j+1}\end{aligned}\tag{4.13}$$

In the next section, the hierarchical intermediate feature maps $\mathbf{E}_j$ will be utilized in the decoder stage to upsample the feature maps and predict the final saliency.

Using both absolute and relative positional encodings can provide complementary benefits. Absolute positional embeddings (APE) explicitly encode position indices, ensuring that each token has a unique positional representation. This helps when absolute positions carry semantic meaning, such as in structured data or specific sequence tasks. Relative positional embeddings (RPE), like RoPE, encode relationships between tokens rather than fixed positions. This is useful for capturing translational invariance, allowing models to generalize better to sequences of different lengths and enabling better contextual understanding. Together, APE provides a global position reference, while RPE ensures local positional relationships are captured within self-attention.

Recall from Section 4.1.3 that DCPE is an **absolute** positional embedding based on the centroids of the superpixels. To further refine positional representations, we explore Rotary Positional Embeddings (RoPE) [93] (specifically 2D RoPE [128]), which provide a **relative** approach to encode positional relationships within self-attention mechanisms.

Rotary Positional Embeddings

Traditional APEs [32, 83], explicitly encode position indices but do not naturally model relative dependencies. To address this, RoPE introduce a rotation-based mechanism that encodes relative positional information directly within the dot-product attention mechanism.

In standard transformers, APEs are additively combined with input tokens, which can limit the model’s ability to generalize to unseen sequence lengths and struggle with extrapolation beyond trained positions. Instead, RoPE incorporates positional information multiplicatively, allowing self-attention to naturally capture relative positional dependencies. This property aligns well with our proposed DCPE, which aims to dynamically adjust position representations based on structural relationships in the data.

Given a d_i -dimensional input query and key representations from an arbitrary aggregation window $\mathbf{E}^q, \mathbf{E}^k \in \mathbb{R}^{M^2 \times d_i}$ (omitting the layer index notation j for clarity), and a rotation matrix $\mathbf{R} \in \mathbb{C}^{M^2 \times \frac{d_i}{2}}$, the embedding transformation is then applied following [128] as

$$\text{Complex}(\mathbf{x}) = [x_1 + ix_2, \dots, x_{d_i-1} + ix_{d_i}] \text{ for some real vector } \mathbf{x} \quad (4.14)$$

$$\text{Real}(\mathbf{x}) = [x_1, x_2, \dots, x_{d_i-1}, x_{d_i}] \text{ for some complex vector } \mathbf{x} \quad (4.15)$$

$$\mathbf{E}^{q'} = [\text{Complex}(\mathbf{E}_1^q), \dots, \text{Complex}(\mathbf{E}_{M^2}^q)] \quad (4.16)$$

$$\mathbf{E}^{k'} = [\text{Complex}(\mathbf{E}_1^k), \dots, \text{Complex}(\mathbf{E}_{M^2}^k)] \quad (4.17)$$

$$\mathbf{E}^{q''} = [\text{Real}(\mathbf{E}_1^{q'} \cdot \mathbf{R}), \dots, \text{Real}(\mathbf{E}_{M^2}^{q'} \cdot \mathbf{R})] \quad (4.18)$$

$$\mathbf{E}^{k''} = [\text{Real}(\mathbf{E}_1^{k'} \cdot \mathbf{R}), \dots, \text{Real}(\mathbf{E}_{M^2}^{k'} \cdot \mathbf{R})] \quad (4.19)$$

where we transform the query and key representations into a complex matrix, then transform it back to a real-valued matrix after taking the dot product with the rotation matrix. This formulation enables the self-attention mechanism to naturally incorporate relative positional information without explicitly computing positional differences, thereby enhancing local positional differentiation.

Our proposed dynamic centroid positional embedding from Section 4.1.3 further refines positional encoding by dynamically adjusting embeddings based on structural centroids in the data. By integrating RoPE, we ensure that relative positional information is preserved, while DCPE introduces a more flexible mechanism that adapts to underlying spatial distributions. This hybrid approach enhances positional awareness in transformer models for tasks requiring adaptive spatial relationships, such as superpixel-based representations.

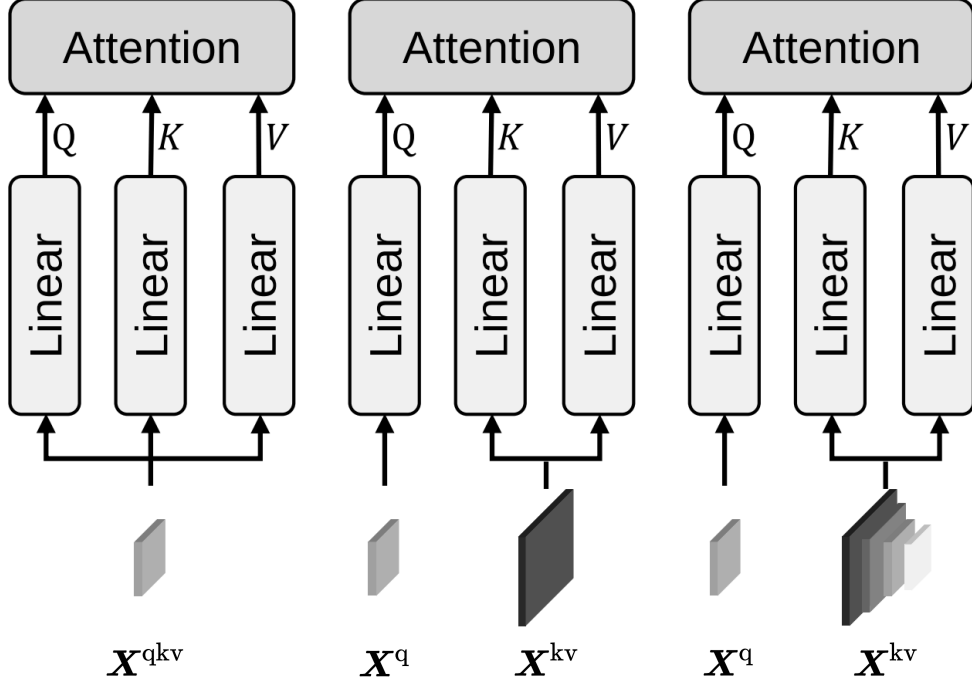


Figure 4.9: **Mix-Attention**. Comparison of self-attention (left), cross-attention (middle), and mix-attention (right). Diagram directly taken from Yeom and von Klitzing [47] (Figure 3, page 4).

Building on the hierarchical features maps developed from the encoder, we upsample the semantically rich feature maps in the next section.

4.2.2 Decoder

The primary advantage of Mix-attention [47] lies in its ability to leverage hierarchical representations. Unlike standard UNets [64, 129, 130], which discard feature maps after upsampling during decoding, Mix-attention preserves all hierarchical information throughout the decoding process and incorporates it into the final saliency prediction. In self-attention, the same features ($\mathbf{X}^{\text{qkv}}$) are used to generate queries, keys, and values, all originating from a single source, such as the same encoder or decoder stage. Cross-attention, on the other hand, utilizes two distinct feature sets: $\mathbf{X}^{\text{q}}$ and $\mathbf{X}^{\text{kv}}$, each coming from a separate source. In contrast, our mix-attention mechanism constructs $\mathbf{X}^{\text{kv}}$ from a combination of

multiple multi-scale stages. This approach enables the query to identify matches across various stages, capturing different levels of contextual granularity and thereby enhancing feature refinement. A comparison between traditional self-attention, cross-attention, and the mix-attention proposed in Yeom and von Klitzing [47] is displayed in Fig. 4.9.

A feature set $\mathcal{F}_i$ for a decoder stage i with N encoder stages is defined as follows:

$$\mathcal{F}_i = \begin{cases} \{\mathbf{E}_j\}_{j=1}^N & \text{if } i = 1 \\ \{\mathbf{E}_j\}_{j=1}^{N-i+1} \cup \{\mathbf{D}_j\}_{j=N-i+2}^N & \text{otherwise} \end{cases} \quad (4.20)$$

where all the encoder features ($\mathbf{E}_j$) are selected for the first decoder stage ($i = 1$). In the following stages, the outputs from the previous decoder stages ($\mathbf{D}_j$) are carried forward by substituting their lateral encoder counterparts in $\mathcal{F}_i$.

To ensure alignment of the spatial dimensions within $\mathcal{F}_i$, we adopt the same spatial reduction method introduced by Wang et al. [131]:

$$\hat{\mathbf{F}}_{ij} = \text{AvgPool}(\text{pr}_j, \text{pr}_j)(\mathcal{F}_{ij}), \forall j \in \{1, 2, \dots, N-1\} \quad (4.21)$$

$$\mathbf{F}_{ij} = \text{MLP}(d_j, d_j)(\hat{\mathbf{F}}_{ij}), \forall j \in \{1, 2, \dots, N-1\} \quad (4.22)$$

where $\mathcal{F}_{ij}$ is the j -th feature map of the feature set $\mathcal{F}_i$, and pr_j denote the pooling ratio that aligns the spatial dimension to the smallest resolution feature map $\mathcal{F}_{iN}$. The average pool function (AvgPool) and the linear mapping function (MLP) are configured as $\text{AvgPool}(\text{kernel size}, \text{stride})(\cdot)$ and $\text{MLP}(\text{input channels}, \text{output channels})(\cdot)$, respectively. The spatially aligned features are concatenated along the channel dimension to generate a mixed feature $\mathbf{X}_i^{\text{kv}}$ for keys and values:

$$\mathbf{X}_i^{\text{kv}} = \text{Concat}(\{\mathbf{F}_{ij}\}_{j=1}^{N-1} \cup \mathcal{F}_{iN}) \quad (4.23)$$

With layer normalization (LN) and a feedforward network (FFN), the output of decoder stage j ($\mathbf{D}_j$) is computed as follows:

$$\mathbf{A}_j = \text{LN}(\text{MixAttn}(\text{LN}(\mathbf{X}_j^{\text{q}}), \text{LN}(\mathbf{X}_j^{\text{kv}}))) + \text{LN}(\mathbf{X}_j^{\text{q}}) \quad (4.24)$$

$$\mathbf{D}_j = \text{FFN}(\mathbf{A}_j) + \mathbf{A}_j \quad (4.25)$$

However, our motivation for adopting Mix-attention extends beyond its effective use of hierarchical representation. We also observed that the computational complexity of its attention mechanism is

$$\mathcal{O}(d^{\text{q}} \cdot N^{\text{q}} \cdot N^{\text{kv}}), \quad (4.26)$$

where d^q is the hidden dimension of the query, N^q is the number of tokens of the query, and N^{kv} is the number of tokens of keys/values. Notably, N^{kv} reduces to

$$\frac{H^s \cdot W^s}{16}, \quad (4.27)$$

ensuring that the computational complexity remains low across all stages of the decoder.

Once we have the updated hierarchical feature maps, we employ the saliency head $\mathbf{W}^h$ to predict the superpixels' saliency map.

4.2.3 Loss Functions

In this thesis, we use Dice loss [132] and Focal loss [133], following the works of [134, 135, 136, 137], who have demonstrated their effectiveness in SOD. Dice loss directly optimizes the overlap between the predicted and ground truth masks, making it well-suited for addressing foreground-background imbalance. Focal loss, on the other hand, down-weights easy negatives and focuses learning on hard examples, improving convergence and precision in challenging regions. These loss functions have been widely adopted in the SOD literature for their ability to improve both training stability and segmentation accuracy.

Dice Loss

Dice loss [132] is derived from the Dice coefficient [138], a statistical metric used to quantify the similarity—or overlap—between two sets. Originally introduced to measure agreement between samples, it has since become widely used in segmentation tasks to evaluate the overlap between predicted and ground truth masks. For binary segmentation, the Dice coefficient between the predicted superpixel saliency map $\hat{\mathbf{y}} \in [0, 1]^K$ and the ground truth superpixel mask $\mathbf{y} \in [0, 1]^K$ is defined as

$$\text{Dice}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{2 \sum_{i=1}^K y_i \hat{y}_i}{\sum_{i=1}^K y_i + \sum_{i=1}^K \hat{y}_i + \epsilon}, \quad (4.28)$$

where $\hat{y}_i$ and y_i are the predicted and ground truth values at superpixel i , respectively, K is the total number of superpixels, and ϵ is a small constant to avoid division by zero.

The Dice loss is then formulated as

$$\mathcal{L}^D = 1 - \text{Dice}(\hat{\mathbf{y}}, \mathbf{y}). \quad (4.29)$$

Unlike pixel-wise losses such as binary cross-entropy (BCE), Dice loss considers the global overlap between the predicted saliency mask and the ground truth saliency mask, making it particularly effective for segmenting small or sparse salient regions. It has been successfully used in many SOD frameworks, including F³Net [136] and PoolNet [139].

Focal Loss

Focal loss [133], originally introduced for dense object detection, dynamically down-weights well-classified examples and focuses training on hard negatives. The standard binary cross-entropy loss is modified with a modulating factor:

$$\mathcal{L}^F = -\alpha(1 - \hat{y}_i)^\gamma y_i \log(\hat{y}_i) - (1 - \alpha)\hat{y}_i^\gamma (1 - y_i) \log(1 - \hat{y}_i), \quad (4.30)$$

where $\alpha \in [0, 1]$ is a balancing factor, and $\gamma \geq 0$ is the focusing parameter that adjusts the rate at which easy examples are down-weighted.

In SOD, focal loss helps mitigate the dominance of background pixels during training, thus enhancing the learning signal from hard-to-classify salient regions. Prior works such as CPD [137] and EGNet [135] have shown that incorporating focal loss leads to improved boundary precision and recall in complex scenes.

Combined Objective

In our framework, we combine Dice loss and Focal loss to jointly leverage their strengths:

$$\mathcal{L} = \lambda_1 \mathcal{L}^D + \lambda_2 \mathcal{L}^F, \quad (4.31)$$

where λ_1 and λ_2 are scalar weights controlling the contribution of each term. This combination encourages both accurate foreground localization and robust learning under class imbalance, which are crucial for high-quality saliency maps. In this thesis, we choose λ_1 and λ_2 to be both 1 following previous works [134, 135, 136, 137].

4.2.4 Overall Model Architecture

Finally, we present the complete SuperFormer architecture in Fig. 4.10, incorporating all components discussed in the previous sections. The linearly projected descriptive attributes (which accounts for the colour, texture, area, rotation, and shape) from (4.11) are added element-wise with the positional embedding (which accounts for the location) from (4.12)

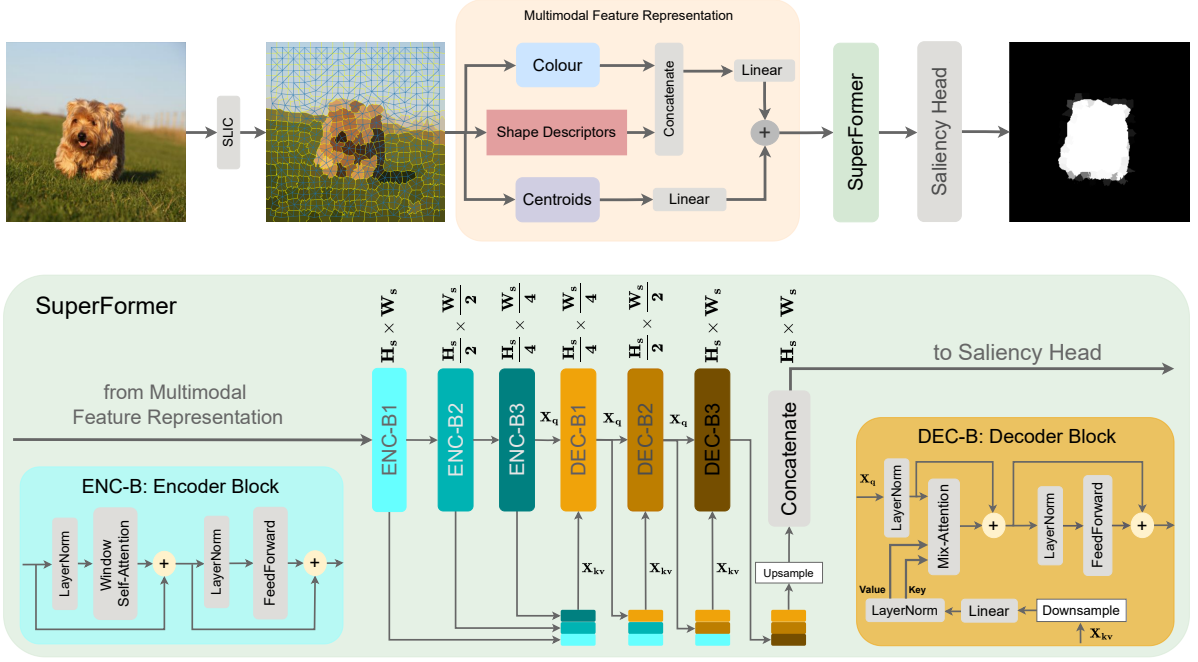


Figure 4.10: **SuperFormer Overview.** Using SLIC, we segment a raw image into superpixels, each capturing low-dimensional features. The colour attributes of each superpixel are combined with its Fourier-transformed shape, and then projected and element-wise added to the centroids’ linear projection (DCPE). Observe that both Fourier Descriptors and DCPE incorporate heterogeneous spatial information. The resulting features are processed through SWIN Transformer encoders with Mix-attention decoders and passed to a saliency head for SOD.

and then processed through SuperFormer (SF), a UNet-like [129] architecture with SWIN Transformer [33] encoder and Mix-attention [47] decoder.

It is crucial to distinguish whether our performance improvements arise primarily from our superpixel-based design, rather than the use of the vision transformer architecture. Since most existing methods rely on CNN-based architectures, we must determine whether our gains stem from the superpixel representation itself, rather than the inherent advantages of vision transformers. To address this, we conduct an ablation study by applying our transformer-based models (SF-XS and SF-S) to regular pixel-based inputs and take the average across the seven datasets from Section 3.3. This comparison allows us to isolate the contributions of our superpixel pipeline and demonstrate that the observed improvements

Metric	Pixel-Based		Superspixel-Based	
	SF-XS	SF-S	SF-XS	SF-S
MAE ↓	0.061	0.056	0.046	0.036
maxF ↑	0.829	0.837	0.868	0.892

Table 4.2: **Pixels versus Superspixels on SuperFormer.** This table demonstrates that the Vision transformer architecture is not the source of performance gains. Observe that SuperFormer performs worse on pixel-based inputs. However, when integrated with our superspixel extraction design and DCPE, it achieves significantly higher accuracy, outperforming previous methods and setting new state-of-the-art (SOTA) results.

Metric	Pixel-Based		Superspixel-Based	
	SAMNet [10]	HVPNet [72]	SAMNet [10]	HVPNet [72]
MAE ↓	0.049	0.049	0.062	0.058
maxF ↑	0.843	0.844	0.818	0.825

Table 4.3: **Pixels versus Superspixels on CNNs.** This table shows that CNNs are less effective when applied to superspixel representations, further underscoring the effectiveness of SuperFormer in leveraging these representations.

are not merely a result of using the vision transformer architecture.

As shown in Table 4.2, the combination of Swin Transformer and Mix-Attention decoder with pixel-based representations does not yield competitive performance. This confirms that the **performance gains stem from our superspixel extraction design rather than the use of the vision transformer architecture itself**. The substantial improvements highlight the effectiveness of integrating these modules with our superspixel representation, emphasizing the novelty of our framework.

In parallel, to highlight the synergy between the transformer-based architecture and superspixel representations, we compare two CNN-based models [10, 72] trained on superspixel inputs against the same architectures trained on regular pixel inputs. As shown in Table 4.3, the CNN models perform worse when using superspixel representations, underscoring the transformer’s superior ability to model such representations. This further validates our choice of using a transformer-based architecture to fully leverage the benefits of superspixels.

In this chapter, we introduced key contributions aimed at addressing the challenges posed by the heterogeneity of superspixels. To capture structural variations, we leveraged Fourier descriptors and central moments, ensuring robust shape representations. Additionally, we proposed dynamic centroid positional embedding to effectively encode positional

information despite the irregular placement of superpixels. Beyond these enhancements, we expanded our model design by replacing conventional non-overlapping pixel patches with a superpixel-based representation, enabling a more efficient encoding of spatial and colour information. Furthermore, we incorporated SWIN Transformers as the encoder backbone and Mix-Attention in the decoder to enhance feature aggregation. Together, these advancements establish a framework that balances efficiency and accuracy, paving the way for more structured and interpretable superpixel-based modeling. The next chapter focuses on additional methods for generalization, balancing computational complexity with accuracy, and comparison with SOTA lightweight SOD methods.

Chapter 5

Experiments and Results

In the preceding chapter, we established the methodological foundation for the development of SuperFormer, a superpixel-based vision Transformer. The investigation centered on leveraging the computational advantages of superpixels while rigorously addressing the challenges posed by their inherent heterogeneity. Two primary objectives guided the framework: first, to design representation methods that enhance salient object detection performance by explicitly accounting for superpixel heterogeneity; and second, to extend an existing encoder-decoder framework from fixed, non-overlapping pixel patches to a superpixel-based representation. The chapter was structured around two major components: a systematic exploration of methods to address heterogeneity, supported by empirical validation, and a detailed exposition of the model architecture and its design rationale. Throughout, methodological choices were critically motivated to balance theoretical soundness, and computational efficiency. This foundation provides the necessary context for the work presented in this chapter, where we focus on further improving the efficiency, generalizability, and practical applicability of SuperFormer.

The goals for this chapter are as follows:

- To address the slow training time of image augmentation by employing feature augmentation and improving its generalizability.
- To investigate the importance of ImageNet pre-training with superpixel representations.
- To validate the high efficiency and accuracy of SuperFormer by comparing it to SOTA lightweight SOD models using real-world datasets.

This chapter is organized as follows to achieve these goals:

- **Section 5.1 Feature Augmentation:** This section discusses the importance of feature augmentation on training time, and our proposed method to improve the generalizability of feature augmentation.
- **Section 5.2 ImageNet Pre-training:** This section outlines the importance of ImageNet pre-training, demonstrating how leveraging pre-trained weights improves model performance and convergence.
- **Section 5.3 Overall Results¹:** This section brings all the components in the previous sections together to validate the high efficiency and accuracy of SuperFormer compared to state-of-the-art efficient SOD methods.

These experiments collectively provide a thorough evaluation of our framework, validating its effectiveness and identifying key factors that influence its performance.

5.1 Feature Augmentation

Data augmentation [38, 140, 141, 142] is a widely used technique in machine learning to improve model generalization and robustness by introducing variations in the training data. In computer vision, image augmentation methods such as horizontal flips, rotations, colour augmentations, and CutMix [141] are commonly applied to raw input images. While effective, combining image augmentation with superpixel segmentation is computationally impractical during training. As depicted in Fig. 5.1, extracting superpixel attributes require re-computation of the superpixels segmentations (and the corresponding superpixel attributes) for each augmented image, significantly increasing processing time and slowing down the training pipeline.

To address this inefficiency, feature augmentation provides a more practical alternative. Instead of augmenting the raw images $\mathbf{X}$, the augmentations are applied directly to the features $\mathbf{X}^e$ extracted from the superpixels (superpixel attributes need to be extracted just once), drastically reducing computational overhead. In this section, we augment the superpixel-based feature representations by applying transformations such as colour space adjustments, horizontal flips, and CutMix directly to the nodes representing the superpixels.

¹Section 5.3 includes the material from Section 4.3 of [48].

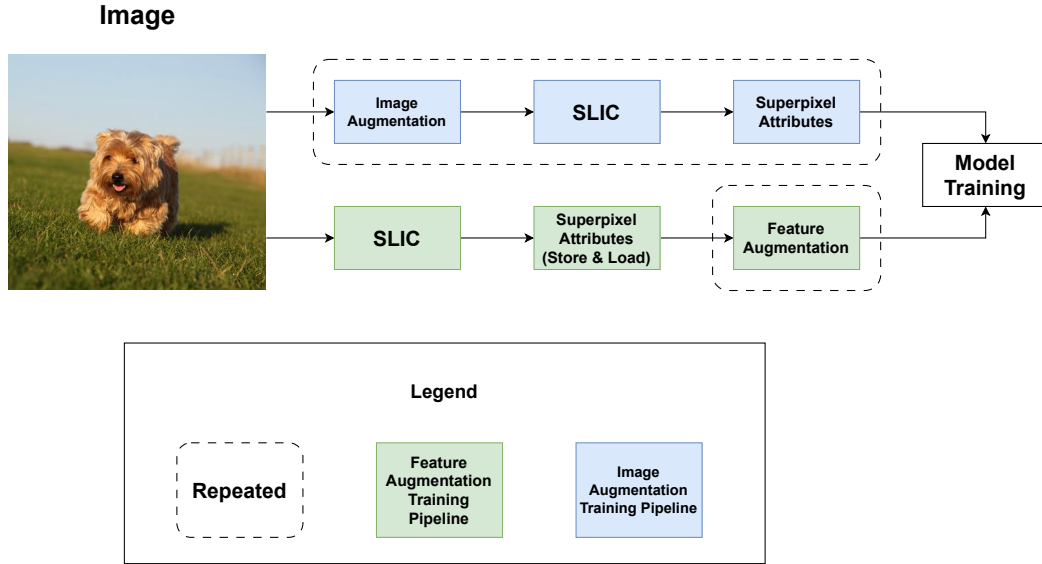


Figure 5.1: **Diagram of Feature Augmentation.** Observe that with image augmentation, SLIC and superpixel feature extraction needs to be applied on every augmented image, thereby requiring to repeat the SLIC segmentation and superpixel attribute extraction on each iteration. Meanwhile, the feature augmentation pipeline loads the stored superpixel features each iteration and applies the feature augmentation.

Feature augmentation not only maintains the benefits of augmentation but also leverages the compactness of the superpixel representation to perform the transformations more efficiently. Feature augmentation allows the model to explore diverse variations in the data without the need for repeated segmentation computations, ensuring faster and more practical training while preserving the integrity of the superpixel-based representation.

Colour Augmentation

RandAugment [143] is a data augmentation technique introduced to enhance the generalization of machine learning models by applying a random subset of augmentation operations to input data. Unlike manual data augmentation strategies [38, 144, 145], RandAugment simplifies the search for effective augmentation policies by relying on two key hyperparameters: the number of transformations (N) and the magnitude of the transformations (m). This eliminates the need for a search over augmentation policies, making it

computationally efficient and easy to integrate into training pipelines.

Let the mean colour features $g^c(\mathbf{X}_k)$ from Section 4.1.1 (without the standard deviation and texture) be noted as $\mathbf{I}$ for simplicity ($g^c(\mathbf{X}_k) = \mathbf{I}$), the augmented colour features from a single operation as $\mathbf{I}'$, and the augmented colour features from multiple operations as $\mathbf{I}''$. Given an input superpixel colour features $\mathbf{I}$, RandAugment applies a sequence of N operations $\{a_1, a_2, \dots, a_N\} \subset \mathcal{A}$, where $a_j(\mathbf{I}, \rho)$ denotes the application of the j -th operation with magnitude ρ on input $\mathbf{I}$. The colour augmented superpixel feature representation $\mathbf{I}''$ can be expressed as

$$\mathbf{I}'' = a_N(a_{N-1}(\dots a_1(\mathbf{I}, \rho) \dots, \rho), \rho),$$

The colour augmentation a_j are randomly sampled from the set of transformations introduced in RandAugment.

1. **Brightness augmentation** modifies the intensity of an image by scaling its RGB values:

$$\mathbf{I}' = (1 + \beta) \cdot \mathbf{I},$$

$\beta \in [-1, 1]$ controls the brightness adjustment ($\beta > 0$ for brighter, $\beta < 0$ for darker).

2. **Colour augmentation** adjusts the colour saturation by interpolating between the grayscale version of the colour features and the original:

$$\mathbf{I}' = (1 - \gamma) \cdot \mathbf{I}^g + \gamma \cdot \mathbf{I},$$

where $\mathbf{I}^g$ is the grayscale version of $\mathbf{I}$, and $\gamma \in [0, 2]$ controls the colour intensity ($\gamma > 1$ increases saturation, $\gamma < 1$ decreases it).

3. **Contrast augmentation** scales the difference between the colour features and the mean of the colour features:

$$\mathbf{I}' = (1 + \alpha) \cdot (\mathbf{I} - \mu) + \mu,$$

where μ is the mean of the colour features, and $\alpha \in [-1, 1]$ controls the contrast adjustment ($\alpha > 0$ for higher contrast, $\alpha < 0$ for lower contrast).

4. **Sharpness augmentation** enhances or reduces the edges in the colour features by blending the original colour features with an edge-enhanced version:

$$\mathbf{I}' = (1 - \lambda) \cdot \mathbf{I} + \lambda \cdot \mathbf{I}^s,$$

where $\mathbf{I}^s$ is an edge-enhanced version of $\mathbf{I}$, and $\lambda \in [0, 2]$ controls the sharpness ($\lambda > 1$ increases sharpness, $\lambda < 1$ decreases it).

5. **Posterize** reduces the number of bits used to represent each colour channel:

$$\mathbf{I}' = \lfloor \mathbf{I}/2^b \rfloor \cdot 2^b,$$

where b is the number of bits to keep ($b \leq 8$), and $\lfloor \cdot \rfloor$ denotes the floor operation.

6. **Solarize** inverts the colour feature values above a threshold:

$$\mathbf{I}'(k) = \begin{cases} \mathbf{I}(k) & \text{if } \mathbf{I}(k) < T, \\ 255 - \mathbf{I}(k) & \text{if } \mathbf{I}(k) \geq T, \end{cases}$$

where T is the threshold, and the colour feature value for the k^{th} superpixel as $\mathbf{I}(k)$.

7. **Autocontrast** stretches the intensity range of the colour features to cover the full 0–255 range:

$$\mathbf{I}'(k) = \frac{\mathbf{I}(k) - \min(\mathbf{I})}{\max(\mathbf{I}) - \min(\mathbf{I})} \cdot 255,$$

where $\min(\mathbf{I})$ and $\max(\mathbf{I})$ are the minimum and maximum colour feature values in $\mathbf{I}$.

8. **Equalize** adjusts the colour feature values based on the cumulative distribution function (CDF) of their histogram:

$$\mathbf{I}'(k) = \text{CDF}(\mathbf{I}(k)) \cdot 255,$$

where $\text{CDF}(v)$ is the normalized cumulative histogram value for colour feature value v .

Fig. 5.2 illustrates examples of color augmentation applied to superpixels, showing how these transformations maintain the structural coherence of the superpixels while introducing significant variations in their color properties.

Spatial Augmentation

Applying spatial augmentations [141, 144, 145] to superpixels poses unique challenges compared to pixel-based methods. Unlike traditional images, superpixels represent compact, perceptually meaningful regions with irregular shapes and boundaries. Mainly, the relationship between spatial augmentations applied to the image space and their corresponding effects on the superpixel contours is not always straightforward. Transformations such as rotation or scaling can lead to unpredictable shifts in contour positions or changes in



Figure 5.2: **Superpixel Colour Augmentations.** The visualizations demonstrate the eight transformations from RandAugment applied at the superpixel level. These augmentations introduce diverse variations while preserving the structural integrity of the superpixels, enabling robust training for models in tasks involving superpixel-based representations.

boundary alignments, making it difficult to ensure that augmented contours accurately reflect the underlying transformations. This lack of clarity complicates the design of aug-

mentation pipelines for superpixel-based representations, as any misalignment between the image space and the superpixel space can introduce noise or inconsistencies in the data. These challenges highlight the need for carefully designed augmentation methods that respect the unique properties of superpixels, preserve their structural integrity, and maintain consistency with the augmented image space.

Horizontal Flip

Horizontal flipping is a crucial augmentation technique in computer vision, widely used to enhance model generalization by introducing left-right invariance [35, 36, 146]. This transformation is particularly beneficial for tasks like object detection and segmentation, where objects may appear in different orientations. While horizontal flipping is commonly applied at the pixel level, its integration into superpixel-based representations remains underexplored. To bridge this gap, we propose extending horizontal flipping to the superpixel domain by flipping both the superpixel centroids and their corresponding central moments. Note that the amplitude of the Fourier descriptors are unaffected by horizontal flips, so it does not need to be transformed. By mirroring the centroids across the vertical axis (center of the image) and adjusting the central moments accordingly, we ensure that the geometric and moment representations remain consistent with the transformed image. This augmentation strategy aims to improve robustness and performance in salient object detection by leveraging the benefits of symmetry-aware learning in the superpixel space.

When a binary geometric shape is flipped horizontally, its coordinates transform as

$$(x, y) \rightarrow (-x, y). \quad (5.1)$$

The central moments are defined as

$$\mu_{pq} = \sum (x - \bar{x})^p (y - \bar{y})^q \quad (5.2)$$

where $\bar{x}$ and $\bar{y}$ are the centroid coordinates. After a horizontal flip, the new centroid becomes $(\bar{x}', \bar{y}') = (-\bar{x}, \bar{y})$, and substituting $x' = -x$ gives

$$\mu'_{pq} = \sum (-x - (-\bar{x}))^p (y - \bar{y})^q. \quad (5.3)$$

So we have

$$\mu'_{pq} = \sum (-1)^p (x - \bar{x})^p (y - \bar{y})^q = (-1)^p \mu_{pq} \quad (5.4)$$

Thus, the central moments transform as

$$\mu'_{pq} = (-1)^p \mu_{pq} \quad (5.5)$$

indicating that moments with an odd p change sign, while those with even p remain unchanged.

In summary, equation 5.5 demonstrates that applying this transformation to the central moments is equivalent to horizontally flipping an image. Next, we explore the necessary transformation to horizontally flip the centroids.

Given an image of width W , the centroid of each superpixel needs to be reflected across the vertical axis that runs through the center of the image.

Following the notation from Section 4.1.3, let the centroid of the k -th superpixel be denoted as

$$\mathbf{P}_{k,:} = (x_k, y_k) \quad (5.6)$$

where x_k and y_k are the horizontal and vertical coordinates, respectively.

To perform a horizontal flip, the new centroid position $\mathbf{P}'_{k,:} = (x'_k, y'_k)$ is computed as:

$$x'_k = W - x_k \quad (5.7)$$

$$y'_k = y_k \quad (5.8)$$

This transformation ensures that each centroid is symmetrically mapped to its corresponding position on the opposite side of the image, while preserving its vertical coordinate. The set of flipped centroids, denoted as p'_k , can then be used in subsequent processing steps to maintain geometric consistency with the flipped image.

Applying this transformation allows the superpixel-based model to benefit from horizontal flipping as a data augmentation technique, improving robustness and generalization. In Fig. 5.3, we demonstrate that the transformed central moments using equation 5.5 is equal to the central moments of a horizontally flipped superpixel.

Superpixel CutMix

CutMix [141], a widely adopted augmentation technique in deep learning, enhances model generalization by replacing regions of an image with patches from another image, while adjusting the corresponding labels proportionally. This strategy encourages the model to focus on discriminative features across mixed regions, leading to improved robustness. While CutMix has been extensively explored in pixel-based tasks [147, 148, 149], its direct application to superpixel-based SOD remains underexplored.

In this section, we propose a novel adaptation of CutMix for superpixel-based SOD, wherein augmentation is performed at the superpixel level rather than the pixel level.

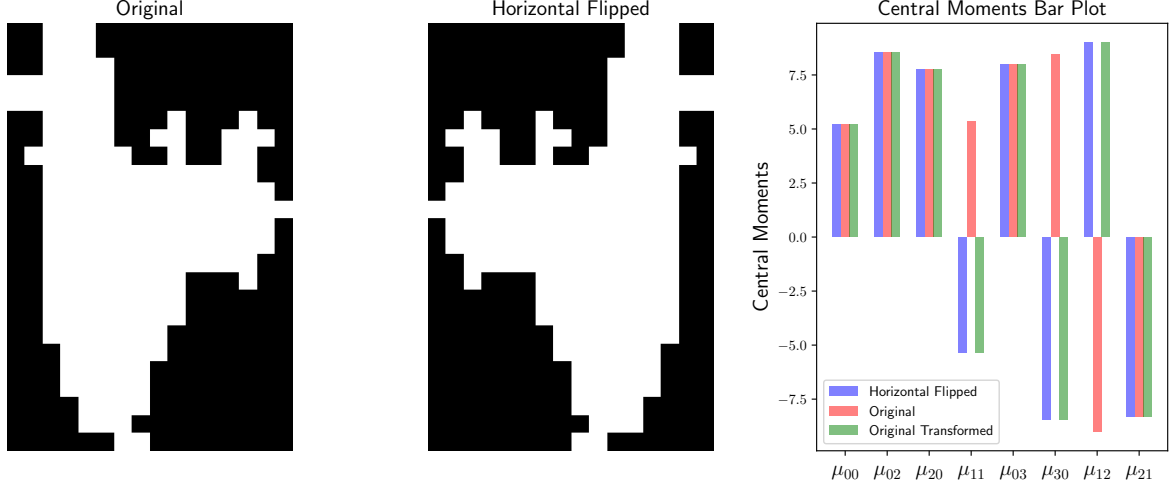


Figure 5.3: **Horizontal Flip Visualization.** On the left, we have a sample binary geometric shape (an arbitrary superpixel). In the middle, a horizontal flipped version of the original shape. On the right, we visualize the central moments for the original superpixel image, central moments for the horizontal flipped superpixel image, and the transformed central moments from the original superpixel image using equation 5.5. Observe that the horizontal flipped central moments are identical to the transformed original central moments.

By strategically mixing superpixels between different samples, we encourage the model to learn more transferable feature representations while preserving local structure. Our approach ensures that the generated samples remain semantically meaningful, promoting spatial coherence in SOD predictions.

Following the notation from equation 4.10 and equation 3.3, let $\mathbf{X}^e$ and $\mathbf{y}$ be a superpixel representation of an image and the superpixel saliency ground truth respectively. Similar to the original CutMix, superpixel CutMix aims to generate a new training sample $(\tilde{\mathbf{X}}^e, \mathbf{y})$ by mixing two training samples $(\mathbf{X}_A^e, \mathbf{y}_A)$ and $(\mathbf{X}_B^e, \mathbf{y}_B)$. The mixed training sample $(\tilde{\mathbf{X}}^e, \mathbf{y})$ is used to train the model with identical objective loss function. The mixing process is defined as

$$\tilde{\mathbf{X}}^e = \mathbf{M} \odot \mathbf{X}_A^e + (\mathbf{1} - \mathbf{M}) \odot \mathbf{X}_B^e \quad (5.9)$$

$$\tilde{\mathbf{y}} = \lambda \mathbf{y}_A + (\mathbf{1} - \lambda) \mathbf{y}_B \quad (5.10)$$

where $\mathbf{M} \in \{0, 1\}^K$ denotes a binary mask indicating which of the superpixel representation is filled, $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is an element-wise multiplication

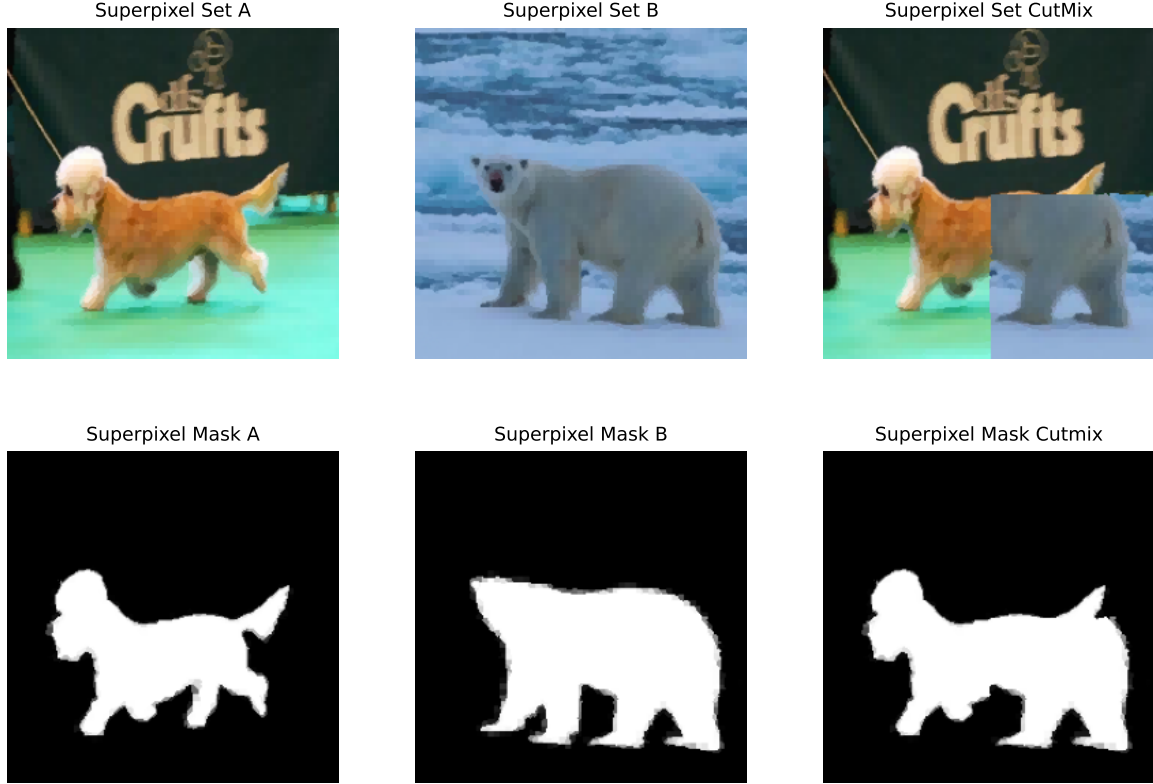


Figure 5.4: **Superpixel CutMix**. This figure demonstrates the generalization of CutMix [141] into the superpixel domain. In particular, the cutmix superpixels are generated in the superpixel domain and do not require regenerating SLIC segmentations. The total number of superpixels is large (3136), therefore zooming in is recommended as relatively fine scrutiny is required. Note that we cannot directly visualize the associated image after performing superpixel CutMix, since superpixels from different images are likely to overlap. However, in the feature domain this is not an issue, as the operation simply swaps the feature indices according to the superpixel index.

operator. Sampling of the binary mask $\mathbf{M}$, and the aspect ratio of the crop box are adapted from [141].

In Fig. 5.4, we present a visualization of superpixel CutMix. Note that in addition to colour in the visualization, additional features such as centroids, shape descriptors, and texture (LBP) are also incorporated in the superpixel CutMix process. Since the superpixel segmentations generated by SLIC exhibit a certain degree of spatial homogeneity, we

leverage this property to select bounding boxes for cropping based on its SLIC initialization (as demonstrated in Fig. 4.1).

5.1.1 Ablation Studies

Continuing on from Section 4.1.4, We conduct a series of ablation studies to assess the incremental impact of each contribution from feature augmentation. We start by providing brief descriptions of these components.

- **Base + DCPE + FD + HF.** Our contributions from Section 4.1 with horizontal flip on the central moments and centroids.
- **Base + DCPE + FD + HF + SPCutmix.** Cumulatively adding superpixel cutmix.
- **Base + DCPE + FD + HF + SPCutmix + RandAug.** Cumulatively adding RandAug operated on mean superpixel RGB values.
- **Base + DCPE + FD + ImageAug.** Our contributions from Section 4.1 with image augmentations (horizontal flip, CutMix, and RandAug).

Table 5.1 presents the performance of each incrementally augmented model, highlighting the impact of individual and combined feature augmentations on SuperFormer, averaged over five random seeds. The cumulative effect of horizontal flip, Superpixel CutMix, and RandAug consistently improves both F1-score and MAE. Notably, in our experiments, we found that the combination of these augmentations achieves performance comparable to traditional image-level augmentation methods, while providing a $6\times$ speedup in training—effectively addressing the computational bottleneck of applying image augmentations in conjunction with SLIC. For reference, training on the DUTS dataset ($\sim 10k$ images) for 400 epochs takes approximately 4 hours with feature augmentation, but nearly 24 hours with image augmentation on a single NVIDIA A100 GPU for accelerated training with AMD Milan 7413 CPU available via the shared high performance computing infrastructure. As shown in Table 5.2, superpixel processing (SLIC segmentation and attribute extraction) incurs negligible overhead during inference on single-image streams, but becomes substantially slower when processing large batches in parallel. Feature augmentation mitigates this slowdown, yielding much faster processing. The discrepancy is due to the much larger number of images processed in parallel during training: inference processes at most 20 images concurrently (one per CPU core), while training processes batches of 32 images

Table 5.1: **Overview of Feature Augmentation Ablations.** This table is a cumulative continuation from Table. 4.1, highlighting the impact of feature augmentation on top of addressing heterogeneity from Section 4.1. Our contributions are highlighted in **Bold**.

Model Variants	MAE ↓	F1-score ↑
Base + DCPE + FD	$0.081 \pm 6\text{e-}4$	$0.762 \pm 5\text{e-}4$
Base + DCPE + FD + HF	$0.074 \pm 1\text{e-}3$	$0.784 \pm 3\text{e-}3$
Base + DCPE + FD + HF + SPCutMix	$0.074 \pm 5\text{e-}4$	$0.805 \pm 3\text{e-}3$
Base + DCPE + FD + HF + SPCutMix + RandAug	$0.073 \pm 1\text{e-}3$	$0.807 \pm 2\text{e-}3$
Base + DCPE + FD + ImageAug	$0.073 \pm 3\text{e-}3$	$0.806 \pm 5\text{e-}3$

Table 5.2: **Superpixel Processing Time.** Processing time for SLIC segmentation and superpixel attribute extraction (Section 4.1) incurs minimal overhead during inference on single-image streams. However, it becomes a bottleneck when applying image-based augmentation. In contrast, feature augmentation significantly reduces processing time. Reported values indicate the average processing time per batch of images (DUTS dataset [1]), measured with 20 CPU cores. The number of images per batch is specified in the table.

Method	Processing Time (ms) ↓	Batch size
Inferencing	2	1
Image Augmentation	785	32
Feature Augmentation	50	32

on each of the 20 cores, yielding up to $32 \times 20 = 640$ images simultaneously. Having established the importance of feature-level augmentations, we now turn to the final component for enhancing generalization: ImageNet pre-training.

5.2 ImageNet Pre-training

ImageNet [150] pre-training has been instrumental in advancing computer vision, providing models with robust feature representations that enhance performance across various tasks. This approach involves training models on the extensive ImageNet dataset, enabling them to capture diverse visual patterns applicable to numerous applications.

Studies have demonstrated the efficacy of ImageNet pre-training. Huh et al. [151] found that pre-trained models not only accelerate convergence during training but also achieve superior accuracy compared to training from scratch. Similarly, Studer et al. [152] observed

that in historical document analysis, models utilizing ImageNet pre-training outperformed those without, particularly in classification and retrieval tasks. To support these findings, state-of-the-art (SOTA) methods in SOD consistently employ feature extractors pre-trained on ImageNet [10, 72, 74, 75, 77, 81, 153]. This practice highlights the widespread adoption of ImageNet pre-training in cutting-edge research, and underscore the value of leveraging pre-trained weights to enhance model performance.

Despite its prevalence, there is a notable absence of pre-trained networks specifically tailored for superpixel representations. The lack of dedicated pre-trained models for superpixels presents a potential gap in reaching competitive results to SOTA SOD methods. To address this, we pre-train on superpixel representations on ImageNet to assess its impact on model accuracy. By developing a pre-training framework tailored to superpixels, we aim to provide models with a nuanced understanding of these representations. Diagram of the ImageNet classification task can be found in Fig. 5.5.

We report two key findings. First, we highlight the importance of utilizing ImageNet pre-trained weights for the SOD task. As shown in Table 5.3, models incorporating superpixel pre-training consistently outperform those without it, demonstrating that this step significantly boosts final accuracy. This not only addresses a gap in existing research but also underscores the value of task-specific pre-training for models operating on superpixel representations.

Second, given that our architecture follows an encoder-decoder structure—with only the encoder initialized from pre-trained weights—using a single learning rate across the entire model is suboptimal. Our grid search, summarized in Table 5.4, explores separate learning rates for the encoder and decoder, revealing that applying a scaled learning rate to the encoder leads to further performance gains.

In summary, these findings underscore the critical role of ImageNet pre-training in enhancing the performance of superpixel-based models for SOD. By bridging the gap in representation-specific pre-training and optimizing learning dynamics through tailored learning rates, our approach lays the groundwork for more effective and efficient training pipelines. These insights not only align with established practices in the broader vision community but also extend them to the relatively underexplored domain of superpixel representations, paving the way for future advancements.

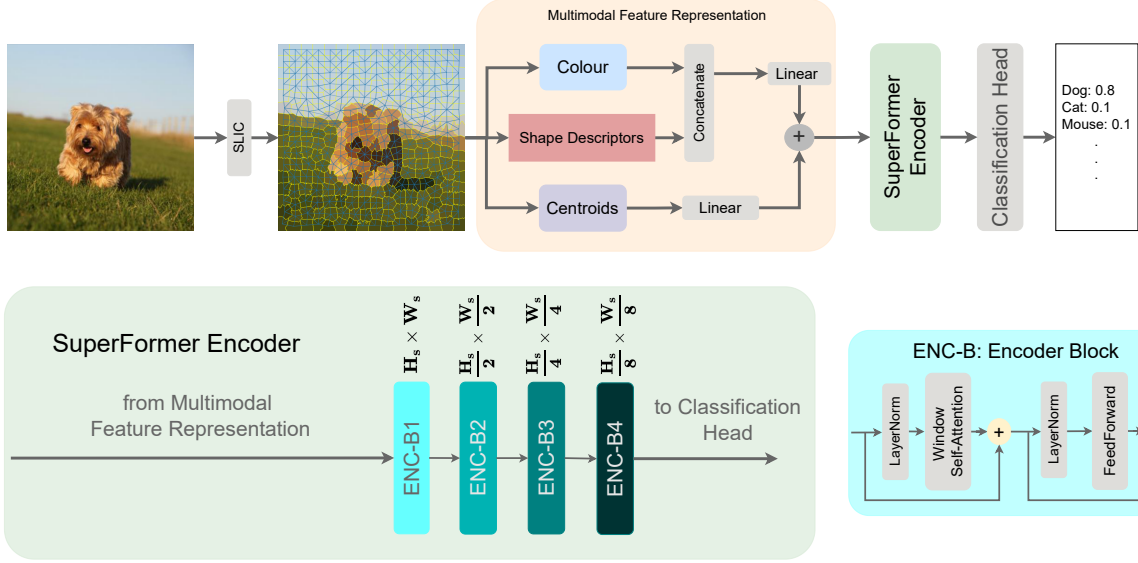


Figure 5.5: **SuperFormer Classification Diagram.** Visualization of the SuperFormer model for classification task. The main difference to Fig. 4.10 for salient object detection is that only the SuperFormer encoder blocks are utilized, with a classification head that replaces the saliency head.

Table 5.3: **Overview of ImageNet Pre-training Ablations.** This table is a cumulative continuation from Table. 5.1, highlighting the impact of ImageNet pre-training on top of feature augmentation from Section 5.1. Our contributions are highlighted in **Bold**.

Model Variants	MAE ↓	F1-score ↑
Base + DCPE + FD + HF + SPCutMix + RandAug	$0.073 \pm 1e-3$	$0.807 \pm 2e-3$
Base + DCPE + FD + HF + SPCutMix + RandAug + ImageNet Pre-training	$0.055 \pm 2e-3$	$0.847 \pm 1e-3$

5.3 Overall Results

Given all the contributions from the preceding chapters, we now evaluate the performance of SuperFormer and demonstrate its superiority in both accuracy and efficiency compared to existing methods. In lightweight SOD, efficiency is as crucial as accuracy—particularly for deployment in real-world, resource-constrained environments. To this end, we consider not only traditional accuracy metrics such as F1-score and MAE, but also report model

Table 5.4: **Impact of Encoder Learning Rate.** This table presents the results of a grid search over decoder and encoder learning rates, highlighting the importance of tuning these hyperparameters to achieve optimal performance. The top table reports F1-scores, while the bottom table presents MAE results. ‘Enc LR’ and ‘Dec LR’ refer to the encoder and decoder learning rates, respectively.

Dec LR \ Enc LR	Dec LR	1e-3	1e-4
	Enc LR		
1e-4		$0.837 \pm 4e-3$	$0.829 \pm 5e-3$
1e-5		$0.847 \pm 2e-3$	$0.843 \pm 3e-3$
1e-6		$0.841 \pm 1e-3$	$0.827 \pm 3e-3$

Dec LR \ Enc LR	Dec LR	1e-3	1e-4
	Enc LR		
1e-4		$0.059 \pm 2e-3$	$0.062 \pm 8e-4$
1e-5		$0.055 \pm 1e-3$	$0.058 \pm 1e-3$
1e-6		$0.057 \pm 2e-3$	$0.063 \pm 9e-4$

Table 5.5: **Natural Scene Results.** A comprehensive evaluation of our proposed SuperFormer (SF) against *ten* state-of-the-art models across *five* diverse natural scene datasets. We compare performance in terms of model parameters (millions), GFLOPs (billions), inference time (milliseconds), and accuracy (F1-score and MAE). Notably, our proposed SF-XXS, SF-XS, and SF-S achieve SOTA results with fewer parameters and FLOPs in their respective model classes. **Best** and **second best** models’ performance and efficiency are shown in the table.

Dataset	Metric	XX-Small Models			X-Small Models				Small Models					
		CSNet [11]	ADMNet [13]	SF-XXS (Ours)	SAMNet [10]	HVPNet [72]	EDN [12]	SF-XS (Ours)	CorrNet [81]	SeaNet [74]	MEANet [75]	MSHNet [153]	ISAANet [77]	SF-S (Ours)
	Params. (M) ↓	0.14	0.84	0.14	1.33	1.23	1.80	1.12	4.09	2.76	3.27	4.07	3.59	2.18
	GFLOPs (B) ↓	0.72	0.80	0.40	0.54	1.13	1.02	0.89	21.09	1.83	11.29	6.11	4.34	1.63
	Infer. (ms) ↓	8.9	9.8	2.7	9.3	6.2	7.1	2.9	14.5	3.4	8.5	13.3	22.1	3.0
DUTS-TE [1]	MAE ↓	0.074	0.057	0.052	0.058	0.058	0.045	0.042	0.047	0.045	0.045	0.125	0.043	0.036
	maxF ↑	0.819	0.795	0.840	0.835	0.839	0.856	0.864	0.847	0.854	0.863	0.705	0.875	0.886
DUTS-O [4]	MAE ↓	0.080	0.063	0.061	0.065	0.064	0.058	0.055	0.059	0.058	0.057	0.144	0.058	0.052
	maxF ↑	0.792	0.744	0.804	0.797	0.799	0.783	0.815	0.773	0.794	0.806	0.674	0.816	0.827
ECSSD [121]	MAE ↓	0.066	0.051	0.049	0.053	0.055	0.043	0.042	0.043	0.042	0.043	0.093	0.035	0.038
	maxF ↑	0.916	0.897	0.927	0.925	0.925	0.934	0.936	0.932	0.931	0.937	0.830	0.945	0.940
HKU-IS [122]	MAE ↓	0.059	0.040	0.041	0.045	0.045	0.034	0.032	0.033	0.035	0.035	0.082	0.030	0.030
	maxF ↑	0.899	0.892	0.912	0.915	0.915	0.924	0.926	0.921	0.917	0.924	0.830	0.934	0.926
PASCAL-S [123]	MAE ↓	0.102	0.094	0.091	0.093	0.090	0.073	0.071	0.081	0.076	0.080	0.146	0.076	0.069
	maxF ↑	0.835	0.774	0.834	0.832	0.832	0.852	0.858	0.845	0.852	0.848	0.732	0.852	0.862
	Avg. MAE ↓	0.076	0.061	0.058	0.063	0.062	0.051	0.048	0.052	0.051	0.052	0.118	0.048	0.045
	Avg. maxF ↑	0.852	0.820	0.863	0.861	0.862	0.870	0.880	0.864	0.870	0.876	0.754	0.884	0.888

Table 5.6: **Remote Sensing Results.** A comprehensive evaluation of our proposed SuperFormer (SF) against *ten* state-of-the-art models across *two* diverse remote sensing datasets. The table shows **Best** and **second best** models performance.

Dataset	Metric	XX-Small Models			X-Small Models				Small Models					
		CSNet [11]	ADMNet [13]	SF-XXS (Ours)	SAMNet [10]	HVPNet [72]	EDN [12]	SF-XS (Ours)	CorrNet [81]	SeaNet [74]	MEANet [75]	MSHNet [153]	ISAANet [77]	SF-S (Ours)
ORSSD [124]	MAE ↓	0.045	0.027	0.021	0.022	0.023	0.026	0.015	0.010	0.011	0.010	0.052	0.008	0.009
	maxF ↑	0.771	0.807	0.865	0.814	0.794	0.825	0.904	0.913	0.894	0.903	0.714	0.917	0.918
EORSSD [20] [4]	MAE ↓	0.024	0.015	0.012	0.013	0.011	0.012	0.009	0.008	0.007	0.007	0.025	0.006	0.007
	maxF ↑	0.809	0.819	0.860	0.781	0.803	0.804	0.891	0.878	0.865	0.877	0.753	0.887	0.892
Avg. MAE ↓		0.034	0.021	0.017	0.017	0.017	0.019	0.012	0.009	0.009	0.009	0.039	0.007	0.008
Avg. maxF ↑		.790	0.813	0.862	0.797	0.799	0.815	0.898	0.895	0.880	0.890	0.733	0.901	0.905

parameters, FLOPs (floating-point operations), and inference speed to provide a holistic assessment.

Model parameters reflect the size and memory footprint of a model, with smaller models being more suitable for edge deployment. FLOPs serve as a proxy for computational complexity, where lower values indicate faster processing and reduced power consumption. Inference speed, measured in milliseconds per image, is critical for real-time applications such as video analysis and interactive systems.

We classify models with fewer than 1 M parameters as extra-extra-small, between 1 M and 2 M as extra-small, and between 2 M and 5 M parameters as small. Accordingly, we present three variants of SuperFormer—SF-S (Small), SF-XS (Extra-Small), and SF-XXS (Extra-Extra-Small)—and compare them against ten state-of-the-art (SOTA) lightweight SOD models [10, 11, 12, 13, 72, 74, 75, 77, 81, 153]. Evaluations are conducted across five natural scene datasets (Table 5.5) and two remote sensing datasets (Table 5.6), demonstrating SuperFormer’s ability to deliver high accuracy with a significantly reduced computational footprint.

On average, across all datasets, SF-XXS, SF-XS and SF-S extend the Pareto front of SOTA results in both accuracy and efficiency. This demonstrates that a combination of addressing the superpixel heterogeneity, feature augmentation, and ImageNet pre-training for superpixel representation allows us to improve the accuracy while taking advantage of the inherent efficiency of superpixel representations, thereby reaching SOTA lightweight performance. For a qualitative comparison, we show the estimation results on a variety of images in Fig. 5.6, further demonstrating the superior performance of SuperFormer over the SOTA SOD methods. We also demonstrate some failure modes of SuperFormer in Fig. 5.7 which point to potential directions for future improvement. Full details of the training hyperparameters can be found in Table 5.7.

In this chapter, we presented a comprehensive strategy to enhance superpixel-based

SOD through feature augmentation and robust pre-training. By introducing data augmentation techniques tailored to the superpixel representation—such as horizontal flipping, RandAugment, and Superpixel CutMix—we improved the model’s generalization. Furthermore, leveraging ImageNet pre-training allowed us to inject rich visual priors into the encoder, significantly boosting performance, especially in lower-data regimes. Extensive experiments demonstrated the superiority of our proposed SuperFormer architecture across seven benchmark datasets, consistently outperforming state-of-the-art lightweight SOD methods in both accuracy and efficiency. SuperFormer achieves higher F1-scores and lower MAE while maintaining a smaller model footprint and faster inference times, making

Table 5.7: **Training Hyperparameters.** Detailed hyperparameter settings of SuperFormer. Values that are separated by “/” imply {Natural Scenes}/{Remote Sensing} otherwise the values are assumed to be shared for other settings.

Parameters	SF-XXS	SF-XS	SF-S
P	10	10	10
K	5184	5184	5184
# Heads	1,2,3	2,4,8	3,6,11
# Hidden dim	16,32,48	32,64,128	48,96,176
# layers	2, 2, 6	2, 2, 6	2, 2, 6
Epochs	400	400	400
# Contour Points	16	16	16
# Fourier Coefficients	10	10	10
Optimizer	AdamW	AdamW	AdamW
LR	$1 \times 10^{-3} / 5 \times 10^{-4}$	$1 \times 10^{-3} / 5 \times 10^{-4}$	$1 \times 10^{-3} / 5 \times 10^{-4}$
Cosine decay	5×10^{-8}	5×10^{-8}	5×10^{-8}
Window size	18	9	9
Drop path	0.1	0.1	0.1
Image Resolution	288	288	288
MLP ratio	1	2	2
Attention drop rate	0	0	0
Dropout	0	0	0
Cutmix	Yes / No	Yes / No	Yes / No
Feature HF	Yes	Yes	Yes
Feature RandAug	Yes	Yes	Yes

it well-suited for deployment on edge devices. These results underscore the effectiveness of combining carefully designed augmentations with a strong pre-training backbone to unlock the full potential of superpixel-based representations.

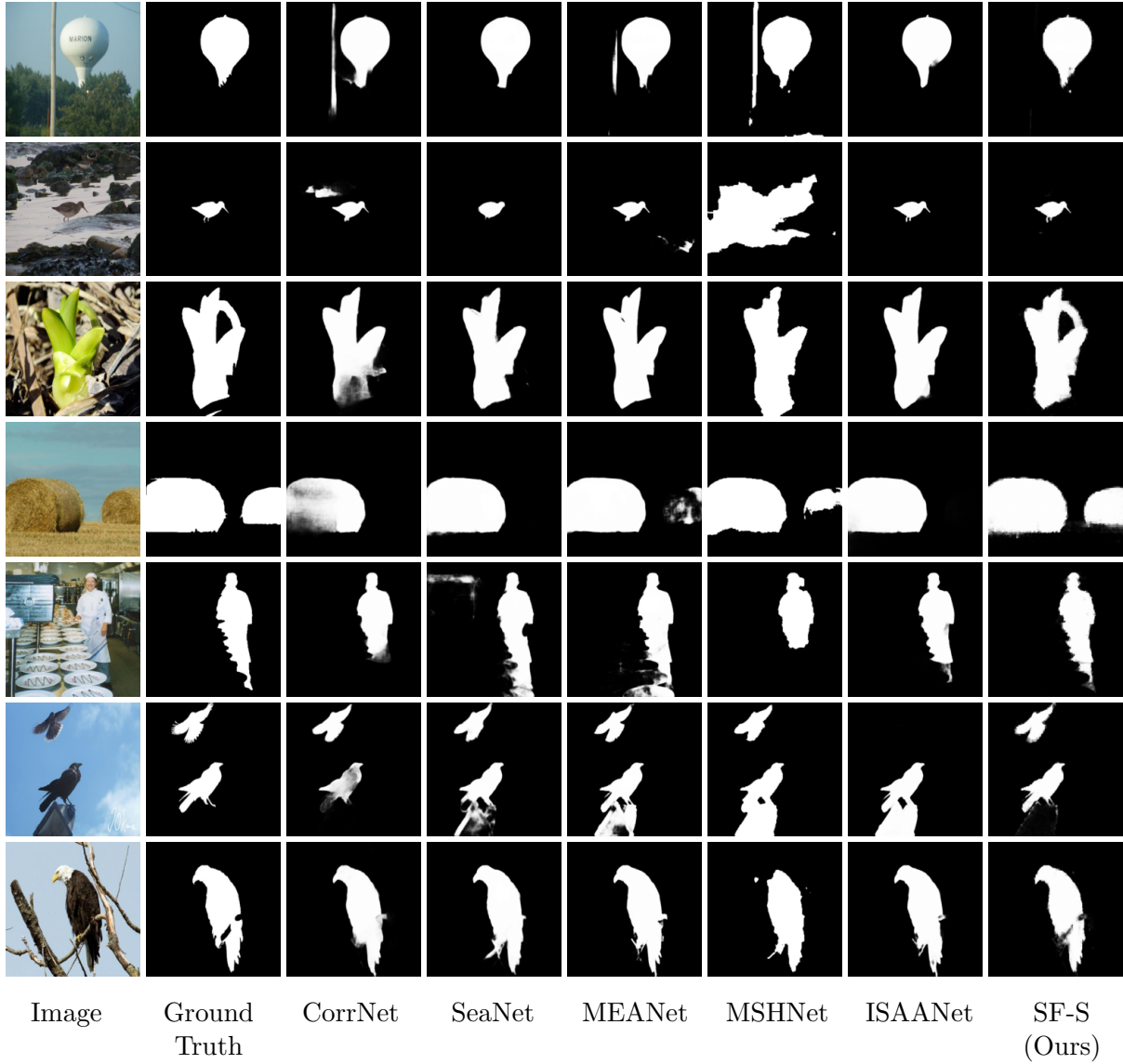


Figure 5.6: **Qualitative saliency map comparison of the small models.** Observe that the boundary of our predicted saliency map seamlessly aligns with the ground truth boundary, even with lower model complexity compared to other pixel-wise methods.

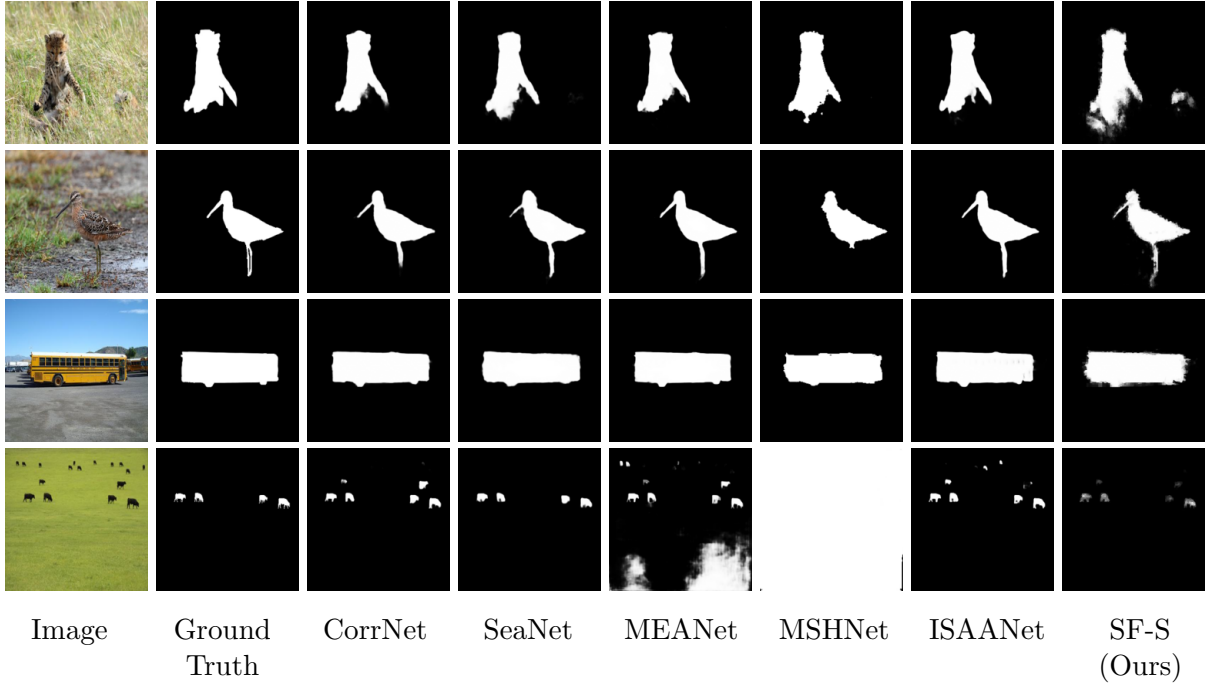


Figure 5.7: **Qualitative saliency map comparison for small models (failure modes)**. SuperFormer shows boundary bleeding in cases where SLIC superpixels misalign with true object edges: (row 1) texture-rich backgrounds (grass), (rows 2–3) low contrast/colour similarity and background clutter (bird against similar background; buses behind the salient bus), and (row 4) small objects. In these scenes, state-of-the-art pixel-based SOD baselines delineate object boundaries more cleanly.

Chapter 6

Conclusion

In this thesis, we revisited the problem of salient object detection (SOD) through the lens of spatial efficiency and structural representation. While conventional methods traditionally rely on pixel-wise processing, we proposed an alternative approach grounded in superpixel representations—regions that group perceptually similar pixels and better align with the human visual system’s abstraction of salient regions.

We introduced **SuperFormer**, a novel Vision Transformer architecture designed to operate directly on superpixel inputs. SuperFormer is built around the premise that saliency can be effectively predicted by learning over a compact set of heterogeneous superpixel tokens, as opposed to dense pixel grids. To this end, we developed a series of innovations including a heterogeneity-aware superpixel representation, compact shape and texture encodings, dynamic centroid positional embeddings, and a mix-attention decoder architecture. Collectively, these components enable SuperFormer to produce high-quality saliency maps with significantly fewer parameters and computational operations.

The experimental results demonstrate that SuperFormer not only achieves state-of-the-art accuracy among lightweight SOD models but also scales effectively across a wide range of input complexities. Our design choices—particularly the inclusion of feature augmentations such as Superpixel CutMix, ImageNet pre-training, and horizontal flip—prove essential in bridging the performance gap between compact models and their high-capacity counterparts. SuperFormer models (SF-S, SF-XS, SF-XXS) consistently outperform competitive lightweight baselines across seven benchmark datasets, both in terms of F1-score and MAE, while maintaining faster inference and reduced model footprint.

6.1 Summary of Contributions

The key contributions of this thesis are as follows:

- We redefined salient object detection from a pixel-based segmentation task to a superpixel-to-superpixel classification problem, offering improved spatial efficiency and boundary preservation (Section 3.1).
- We proposed a heterogeneity-aware representation of superpixels that encodes color statistics, texture (LBP), and shape descriptors (Fourier coefficients and central moments) enabling robust modeling despite irregularity (Subsection 4.1.1 and Subsection 4.1.2).
- We introduced compact and efficient positional encoding methods using centroid-based coordinates, offering a scalable alternative to Laplacian embeddings that are computationally intensive on large superpixel graphs, and homogeneous embeddings that are ill suited for superpixels (Subsection 4.1.3).
- We developed a fully superpixel-based Vision Transformer architecture (SuperFormer) with a U-Mix-Attention decoder, designed to process heterogeneous superpixel tokens while maintaining a small computational footprint (Section 4.2).
- We proposed superpixel-compatible augmentations including Superpixel CutMix and feature-level horizontal flipping, enabling effective generalization without requiring re-segmentation during each augmentation (Section 5.1).
- We demonstrated, for the first time, the applicability and impact of ImageNet pre-training in the superpixel setting (Section 5.2).
- We conducted extensive experiments on natural scene and remote sensing datasets, showing that our approach outperforms state-of-the-art lightweight SOD models in both accuracy and efficiency (Section 5.3).

6.2 Future Work

While this thesis addresses many of the limitations associated with superpixel-based SOD, several avenues for future exploration remain:

- **Supapixel Pre-training Paradigms:** Current pre-training strategies such as ImageNet are not inherently tailored for non-pixel inputs. Developing large-scale pre-training pipelines directly on superpixel graphs or augmenting existing datasets with superpixel-compatible representations could further improve transferability and downstream performance. For example, there are self-supervised pre-training methods such as MAE [154] and SimCLR [146] that could be utilized to further improve the pre-training impact on downstream tasks.
- **Dynamic Supapixel Generation:** Our approach relies on fixed SLIC-based superpixel segmentation. Integrating a learnable or task-adaptive superpixel generation mechanism that minimizes the boundary blurring within the network could enable end-to-end optimization and better handle intra-object variability.
- **Relative Centroid Positional Encoding:** An interesting direction for future work is the design of a relative positional encoding scheme based on superpixel centroids, which could more explicitly model spatial relationships between regions while maintaining translation invariance.
- **Generalization to Other Structured Inputs:** The principles developed here for heterogeneity aware representation, centroid encoding, and transformer adaptation could be extended to other structured input domains, such as medical image regions, video segments, or spatio-temporal representations where computational efficiency is even more critical.
- **Real-Time Applications and Hardware Deployment:** Given the efficiency of SuperFormer, future work could focus on deployment in real-time systems, including UAV imaging, robotic vision, and mobile applications. Optimizing for specific hardware targets such as field-programmable gate arrays or tensor processing units would be a natural extension.
- **Improved Boundary Refinement:** While superpixels generally improve boundary adherence over coarse CNN predictions, the failure modes illustrated in Fig. 5.7, particularly boundary bleeding when SLIC superpixels misalign with true object edges, suggest room for further refinement. Incorporating lightweight post-processing modules such as CRFs, contour decoders, or attention-guided refinement could help sharpen boundaries and mitigate these artifacts.

The methods and insights developed in this thesis reaffirm the potential of superpixels as a fundamental representation in computer vision. By embracing structured abstraction

and compact computation, we hope this work contributes to the development of faster, adaptability to heterogeneity, and scalable models for visual understanding.

References

- [1] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3796–3805, 2017. doi: 10.1109/CVPR.2017.404.
- [2] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.
- [3] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090, 2013.
- [4] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2013. doi: 10.1109/CVPR.2013.407.
- [5] Min Seok Lee, WooSeok Shin, and Sung Won Han. Tracer: Extreme attention guided salient object tracing network (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12993–12994, 2022.
- [6] Yao Yuan, Pan Gao, and XiaoYang Tan. M³ Net: Multilevel, Mixed and Multistage Attention Network for Salient Object Detection. *arXiv preprint arXiv:2309.08365*, 2023.
- [7] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Caver: Cross-modal view-mixed transformer for bi-modal salient object detection. *IEEE Transactions on Image Processing*, 32:892–904, 2023.

- [8] Wei Gao, Songlin Fan, Ge Li, and Weisi Lin. A thorough benchmark and a new model for light field saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8003–8019, 2023.
- [9] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *arXiv preprint arXiv:2401.03407*, 2024.
- [10] Yun Liu, Xin-Yu Zhang, Jia-Wang Bian, Le Zhang, and Ming-Ming Cheng. Samnet: Stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Transactions on Image Processing*, 30:3804–3814, 2021.
- [11] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *European conference on computer vision*, pages 702–721. Springer, 2020.
- [12] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing*, 31:3125–3136, 2022.
- [13] Xiaofei Zhou, Kunye Shen, and Zhi Liu. Admnet: Attention-guided densely multi-scale network for lightweight salient object detection. *IEEE Transactions on Multimedia*, 2024.
- [14] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2376–2383, 2010.
- [15] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 35, pages 185–207, 2012.
- [16] Xi Li, Jun Zhu, Weiming He, Qiang Zhang, Jin Tang, and Liang Lin. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.
- [17] Hengshuang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Anurag Tyagi, and Amit Agrawal. Exfuse: Enhancing feature fusion for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 269–284, 2018.

- [18] Matei Mancas, Johan Wagemans, and Bernard Gosselin. Visual saliency in human-computer interaction: Applications and challenges. In *ACM International Conference on Multimodal Interaction (ICMI)*, pages 436–443, 2016.
- [19] Nawon Kim, Young-Woo Cho, Hyun Soo Kim, and Sung-Eui Yoon. Multi-task learning using saliency maps for autonomous driving. *IEEE Robotics and Automation Letters*, 4(3):3149–3156, 2019.
- [20] Qijian Zhang, Runmin Cong, Chongyi Li, Ming-Ming Cheng, Yuming Fang, Xiaochun Cao, Yao Zhao, and Sam Kwong. Dense attention fluid network for salient object detection in optical remote sensing images. *IEEE Transactions on Image Processing*, 30:1305–1317, 2020.
- [21] Sourav Bak, Young Min Baek, and H Jin Kim. Visual saliency in robotics: A review and a look ahead. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 270–275. IEEE, 2018.
- [22] Zhijie Wang, Deng-Ping Fan, Ming-Ming Cheng, Ling Zhang, and Ling Shao. Hierarchical saliency-aware network for lesion detection in diabetic retinopathy screening. *Medical Image Analysis*, 67:101832, 2021.
- [23] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Learning saliency maps for image and video annotation. *IEEE Transactions on Cybernetics*, 43(6):1867–1879, 2013.
- [24] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [25] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multi-scale graph decomposition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 1124–1131. IEEE, 2005.
- [26] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.
- [27] Alastair P Moore, Simon JD Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. In *2008 IEEE conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [28] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

- [29] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision*, pages 705–718. Springer, 2008.
- [30] Alex Levinshtein, Adrian Stere, Kiriakos N Kutulakos, David J Fleet, Sven J Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–2297, 2009.
- [31] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical Report 149300, EPFL Technical Report, 2010.
- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.
- [36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

- [39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [40] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
- [41] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.
- [42] Jan Flusser, Barbara Zitova, and Tomas Suk. *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009.
- [43] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.
- [44] Charles T Zahn and Ralph Z Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281, 1972.
- [45] Pete E Lestrel. *Fourier descriptors and their applications in biology*. Cambridge University Press, 1997.
- [46] Eric Persoon and King-Sun Fu. Shape discrimination using fourier descriptors. *IEEE Transactions on systems, man, and cybernetics*, 7(3):170–179, 1977.
- [47] Seul-Ki Yeom and Julian von Klitzing. U-mixer: Unet-like transformer with mix-attention for efficient semantic segmentation. *arXiv preprint arXiv:2312.06272*, 2023.
- [48] Jinman Park, Zahra Gharaee, Paul Fieguth, and David A. Clausi. Superformer: Superpixel-based transformer for lightweight salient object detection, 2025. Under review at the IEEE/CVF International Conference on Computer Vision (ICCV), 2025.
- [49] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

- [50] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.
- [51] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 13964–13973, 2020.
- [52] Jianqiao An, Yucheng Shi, Yahong Han, Meijun Sun, and Qi Tian. Extract and merge: Superpixel segmentation with regional attributes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 155–170. Springer, 2020.
- [53] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [54] CIE Colorimetry. Report no: Cie pub no 15. *Vienna: CIE Central Bureau*, 2004.
- [55] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.
- [56] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19, 2006.
- [57] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1597–1604. IEEE, 2009.
- [58] Zhaojian Yao and Luping Wang. Multi-pathway feature integration network for salient object detection. *Neurocomputing*, 461:462–478, 2021.
- [59] Jun Wei, Shuhui Wang, and Qingming Huang. F³net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12321–12328, 2020.
- [60] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9413–9422, 2020.

- [61] Junxia Li, Zefeng Pan, Qingshan Liu, and Ziyang Wang. Stacked u-shape network with channel-wise attention for salient object detection. *IEEE Transactions on Multimedia*, 23:1397–1409, 2020.
- [62] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4722–4732, 2021.
- [63] Zhaojian Yao and Luping Wang. Boundary information progressive guidance network for salient object detection. *IEEE Transactions on Multimedia*, 24:4236–4249, 2021.
- [64] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8779–8788, 2019.
- [65] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7264–7273, 2019.
- [66] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 7479–7489, 2019.
- [67] Zhengzheng Tu, Chao Wang, Chenglong Li, Minghao Fan, Haifeng Zhao, and Bin Luo. Orsi salient object detection via multiscale joint region and boundary model. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.
- [68] Xiaofei Zhou, Kunye Shen, Li Weng, Runmin Cong, Bolun Zheng, Jiyong Zhang, and Chenggang Yan. Edge-guided recurrent positioning network for salient object detection in optical remote sensing images. *IEEE Transactions on Cybernetics*, 53(1):539–552, 2022.
- [69] Xiangyu Zeng, Mingzhu Xu, Yijun Hu, Haoyu Tang, Yupeng Hu, and Liqiang Nie. Adaptive edge-aware semantic interaction network for salient object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

- [70] Gongyang Li, Zhi Liu, Weisi Lin, and Haibin Ling. Multi-content complementation network for salient object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.
- [71] Taehun Kim, Kunhee Kim, Joonyeong Lee, Dongmin Cha, Jiho Lee, and Daijin Kim. Revisiting image pyramid structure for high resolution salient object detection. In *Proceedings of the Asian Conference on Computer Vision*, pages 108–124, 2022.
- [72] Yun Liu, Yu-Chao Gu, Xin-Yu Zhang, Weiwei Wang, and Ming-Ming Cheng. Lightweight salient object detection via hierarchical visual perception learning. *IEEE transactions on cybernetics*, 51(9):4439–4449, 2020.
- [73] Jia Li, Shengye Qiao, Zhirui Zhao, Chenxi Xie, Xiaowu Chen, and Changqun Xia. Rethinking lightweight salient object detection via network depth-width tradeoff. *IEEE Transactions on Image Processing*, 2023.
- [74] Gongyang Li, Zhi Liu, Xinpeng Zhang, and Weisi Lin. Lightweight salient object detection in optical remote-sensing images via semantic matching and edge alignment. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023.
- [75] Bocheng Liang and Huilan Luo. Meanet: An effective and lightweight solution for salient object detection in optical remote sensing images. *Expert Systems with Applications*, page 121778, 2023.
- [76] Qiankun Liu, Rui Liu, Bolun Zheng, Hongkui Wang, and Ying Fu. Infrared small target detection with scale and location sensitivity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17490–17499, 2024.
- [77] Zhaojian Yao and Wei Gao. Iterative saliency aggregation and assignment network for efficient salient object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.
- [78] HuiLan Luo, JianQin Wang, and BoCheng Liang. Spatial attention feedback iteration for lightweight salient object detection in optical remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.
- [79] Yu-Chao Gu, Shang-Hua Gao, Xu-Sheng Cao, Peng Du, Shao-Ping Lu, and Ming-Ming Cheng. Inas: integral nas for device-aware salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4934–4944, 2021.

- [80] Jin Zhang, Qiuwei Liang, and Yanjiao Shi. Kd-scfnet: Towards more accurate and efficient salient object detection via knowledge distillation. *arXiv preprint arXiv:2208.02178*, 2022.
- [81] Gongyang Li, Zhi Liu, Zhen Bai, Weisi Lin, and Haibin Ling. Lightweight salient object detection in optical remote sensing images via feature correlation. *IEEE Trans. Geosci. Remote Sens.*, 60, 2022.
- [82] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [83] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [84] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
- [85] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 357–366, 2021.
- [86] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [87] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [88] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.
- [89] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.
- [90] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

- [91] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.
- [92] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.
- [93] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [94] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.
- [95] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.
- [96] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.
- [97] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- [98] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.
- [99] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.
- [100] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.
- [101] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

- [102] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.
- [103] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [104] Alex Zihao Zhu, Jieru Mei, Siyuan Qiao, Hang Yan, Yukun Zhu, Liang-Chieh Chen, and Henrik Kretzschmar. Superpixel transformers for efficient semantic segmentation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7651–7658. IEEE, 2023.
- [105] Jieru Mei, Liang-Chieh Chen, Alan Yuille, and Cihang Xie. Spformer: Enhancing vision transformer with superpixel representation. *arXiv preprint arXiv:2401.02931*, 2024.
- [106] Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. Vision transformer with super token sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22690–22699, 2023.
- [107] Yu Qiu, Jie Mei, and Jing Xu. Superpixel-wise contrast exploration for salient object detection. *Knowledge-Based Systems*, 292:111617, 2024.
- [108] Shengfeng He, Rynson WH Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International journal of computer vision*, 115(3):330–344, 2015.
- [109] Farnoush Zohourian, Borislav Antic, Jan Siegemund, Mirko Meuter, and Josef Pauli. Superpixel-based road segmentation for real-time systems using cnn. In *VISIGRAPP (5: VISAPP)*, pages 257–265, 2018.
- [110] Teppei Suzuki, Shuichi Akizuki, Naoki Kato, and Yoshimitsu Aoki. Superpixel convolution for segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3249–3253. IEEE, 2018.
- [111] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585. IEEE, 1994.

- [112] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.
- [113] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [114] Pedro HC Avelar, Anderson R Tavares, Thiago LT da Silveira, Cláudio R Jung, and Luís C Lamb. Superpixel image classification with graph attention networks. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 203–209. IEEE, 2020.
- [115] Saba Dadsetan, David Pichler, David Wilson, Naira Hovakimyan, and Jennifer Hobbs. Superpixels and graph convolutional neural networks for efficient detection of nutrient deficiency stress from aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2950–2959, 2021.
- [116] Eitan Kosman and Dotan Di Castro. Graphvid: It only takes a few nodes to understand a video. In *European Conference on Computer Vision*, pages 195–212. Springer, 2022.
- [117] Teppei Suzuki, Shuichi Akizuki, Naoki Kato, and Yoshimitsu Aoki. Superpixel convolution for segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3249–3253, 2018. doi: 10.1109/ICIP.2018.8451721.
- [118] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [119] Benjamin R Mitchell et al. *The spatial inductive bias of deep learning*. PhD thesis, Johns Hopkins University, 2017.
- [120] Farnoush Zohourian, Borislav Antic, Jan Siegemund, Mirko Meuter, and Josef Pauli. Superpixel-based road segmentation for real-time systems using cnn. In *VISIGRAPP*, 2018.
- [121] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013. doi: 10.1109/CVPR.2013.153.

- [122] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.
- [123] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.
- [124] Chongyi Li, Runmin Cong, Junhui Hou, Sanyi Zhang, Yue Qian, and Sam Kwong. Nested network with two-stream pyramid for salient object detection in optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):9156–9166, 2019.
- [125] Mihir Durve, Jean-Michel Tucny, Deepesh Bhamre, Adriano Tiribocchi, Marco Lauricella, Andrea Montessori, and Sauro Succi. Droplet shape representation using fourier series and autoencoders. *AIAA Journal*, 63(2):806–810, 2025.
- [126] Vigneswararajah Mithulavan, Tharaka Samarasinghe, Ragupathyraj Valluvan, Arumairasa Karnan, Navaratnarajah Sathiparan, and Daniel Niruban Subramaniam. Investigation on the effectiveness of fourier shape analysis in classifying milled aggregates. *Construction and Building Materials*, 458:139504, 2025.
- [127] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [128] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.
- [129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [130] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.
- [131] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

- [132] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 565–571, 2016.
- [133] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [134] Xiaoqi Wang, Cheng Liu, Qibin Hou, Ming-Ming Cheng, and Huchuan Lu. Salient object detection with pyramid attention and salient edges. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [135] Shuhan Zhao, Xi Liu, Deng-Ping Fan, Ming-Ming Yang, Jin Tang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [136] Wei Liu, Siyuan Qiao, and Alan Yuille. F3net: Fusion, feedback and focus for salient object detection. In *AAAI Conference on Artificial Intelligence*, 2021.
- [137] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [138] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [139] Jing Liu, Qibin Hou, Ming-Ming Cheng, Jianming Feng, and Jiang Jiang. A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [140] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- [141] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.
- [142] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

- [143] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [144] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. Edinburgh, 2003.
- [145] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
- [146] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.
- [147] Mengzhao Chen, Mingbao Lin, Zhihang Lin, Yuxin Zhang, Fei Chao, and Rongrong Ji. Smmix: Self-motivated image mixing for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17260–17270, 2023.
- [148] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *European conference on computer vision*, pages 455–471. Springer, 2022.
- [149] Chang Liu, Fuxin Fan, Annette Schwarz, and Andreas Maier. Cut to the mix: Simple data augmentation outperforms elaborate ones in limited organ segmentation datasets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 145–154. Springer, 2024.
- [150] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.
- [151] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.
- [152] Linda Studer, Michele Alberti, Vinaychandran Pondenkandath, Pinar Goktepe, Thomas Kolonko, Andreas Fischer, Marcus Liwicki, and Rolf Ingold. A comprehensive study of imagenet pre-training for historical document image analysis. In

- 2019 *International Conference on Document Analysis and Recognition (ICDAR)*, pages 720–725. IEEE, 2019.
- [153] Gongyang Li, Zhen Bai, Zhi Liu, Xinpeng Zhang, and Haibin Ling. Salient object detection in optical remote sensing images driven by transformer. *IEEE Transactions on Image Processing*, 2023.
 - [154] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
 - [155] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
 - [156] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. Mobile-dets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3825–3834, 2021.
 - [157] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
 - [158] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.
 - [159] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2003.07.008>. URL <https://www.sciencedirect.com/science/article/pii/S0031320303002759>.
 - [160] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
 - [161] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

- [162] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *European Conference on Computer Vision*, pages 275–292. Springer, 2020.
- [163] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4019–4028, 2017.
- [164] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *European Conference on Computer Vision*, pages 35–51. Springer, 2020.
- [165] Zuyao Chen, Runmin Cong, Qianqian Xu, and Qingming Huang. Dpanet: Depth potentiality-aware gated attention network for rgb-d salient object detection. *IEEE Transactions on Image Processing*, 30:7012–7024, 2020.
- [166] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7254–7263, 2019.
- [167] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 714–722, 2018.
- [168] Shuhan Chen and Yun Fu. Progressively guided alternate refinement network for rgb-d salient object detection. In *European Conference on Computer Vision*, pages 520–538. Springer, 2020.
- [169] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R3net: Recurrent residual refinement network for saliency detection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 684–690. AAAI Press Menlo Park, CA, USA, 2018.
- [170] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.
- [171] Zhengyi Liu, Song Shi, Quntao Duan, Wei Zhang, and Peng Zhao. Salient object detection for rgb-d image by single stream recurrent convolution neural network. *Neurocomputing*, 363:46–57, 2019.

- [172] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Salient object detection with recurrent fully convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1734–1746, 2018.
- [173] Wenguan Wang, Jianbing Shen, Xingping Dong, and Ali Borji. Salient object detection driven by fixation prediction. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1711–1720, 2018.
- [174] Lu Zhang, Jianming Zhang, Zhe Lin, Huchuan Lu, and You He. Capsal: Leveraging captioning to boost semantics for salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6024–6033, 2019.
- [175] Wei Ji, Jingjing Li, Miao Zhang, Yongri Piao, and Huchuan Lu. Accurate rgb-d salient object detection via collaborative learning. In *European Conference on Computer Vision*, pages 52–69. Springer, 2020.
- [176] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 13025–13034, 2020.
- [177] Miao Zhang, Weisong Ren, Yongri Piao, Zhengkun Rong, and Huchuan Lu. Select, supplement and focus for rgb-d saliency detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 3472–3481, 2020.
- [178] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for rgb-d salient object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3051–3060, 2018.
- [179] Hao Chen and Youfu Li. Three-stream attention-aware network for rgb-d salient object detection. *IEEE Transactions on Image Processing*, 28(6):2825–2835, 2019.
- [180] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 3052–3062, 2020.
- [181] Gongyang Li, Zhi Liu, and Haibin Ling. Icnnet: Information conversion network for rgb-d based salient object detection. *IEEE Transactions on Image Processing*, 29: 4873–4884, 2020.

- [182] Gongyang Li, Zhi Liu, Linwei Ye, Yang Wang, and Haibin Ling. Cross-modal weighting network for rgb-d salient object detection. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020.
- [183] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for rgb-d salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3927–3936, 2019.
- [184] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.
- [185] James Atwood and Don Towsley. Diffusion-convolutional neural networks. *Advances in Neural Information Processing Systems*, 29, 2016.
- [186] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, 28, 2015.
- [187] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.
- [188] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 29, 2016.
- [189] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.
- [190] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [191] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.
- [192] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

- [193] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9141–9150, 2020.
- [194] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.
- [195] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pages 2712–2721. PMLR, 2019.
- [196] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.
- [197] Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11717–11726, 2022.
- [198] Zhenyu Wu, Shuai Li, Chenglizhao Chen, Hong Qin, and Aimin Hao. Salient object detection via dynamic scale routing. *IEEE Transactions on Image Processing*, 31: 6649–6663, 2022.
- [199] Ruochen Fan, Qibin Hou, Ming-Ming Cheng, Gang Yu, Ralph R Martin, and Shi-Min Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 367–383, 2018.
- [200] Vijay Mahadevan and Nuno Vasconcelos. Saliency-based discriminant tracking. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1007–1013. IEEE, 2009.
- [201] Yuming Fang, Zhenzhong Chen, Weisi Lin, and Chia-Wen Lin. Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Transactions on Image Processing*, 21(9):3888–3901, 2012.

- [202] Luca Marchesotti, Claudio Cifarelli, and Gabriela Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2232–2239. IEEE, 2009.
- [203] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
- [204] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [205] Nicholas J Butko, Lingyun Zhang, Garrison W Cottrell, and Javier R Movellan. Visual saliency model for robot cameras. In *2008 IEEE International Conference on Robotics and Automation*, pages 2398–2403. IEEE, 2008.
- [206] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Xiang Ruan. Self-supervised pretraining for rgb-d salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 3463–3471, 2022.
- [207] Xinhao Deng, Pingping Zhang, Wei Liu, and Huchuan Lu. Recurrent multi-scale transformer for high-resolution salient object detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7413–7423, 2023.
- [208] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.
- [209] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.
- [210] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [211] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

- [212] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Kevin Whitehead, Spencer Clegg, Josh Burton, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.