

**A Comparative Study of Supervised and Unsupervised Deep Learning Models for
Fabric Defect Localization in Smart Textile Manufacturing**

Submitted By

Md. Mahmudul Hasan Bhuiyan

200012101

Nabila Islam Khan

200012127

Supervised By

Prof. Dr. M. Ahsan Habib

**A Thesis submitted in partial fulfillment of the requirement for the degree of Bachelor
of Science in Industrial and Production Engineering**



Department of Mechanical and Production Engineering (MPE)

Islamic University of Technology (IUT)

October, 2025

Candidate's Declaration

This is to certify that the work presented in this thesis, titled, "A Comparative Study of Supervised and Unsupervised Deep Learning Models for Fabric Defect Localization in Smart Textile Manufacturing", is the outcome of the investigation and research carried out by us under the supervision of Prof. Dr. M. Ahsan Habib, Professor and Head, Department of Mechanical and Production Engineering.

It is also declared that neither this thesis nor any part of it has been submitted elsewhere for the award of any degree or diploma.



Name of the Student: Md. Mahmudul Hasan Bhuiyan
Student No: 200012101



Name of the Student: Nabila Islam Khan
Student No: 200012127

Recommendation of the Thesis Supervisors

The thesis titled, “A Comparative Study of Supervised and Unsupervised Deep Learning Models for Fabric Defect Localization in Smart Textile Manufacturing” submitted by Md. Mahmudul Hasan Bhuiyan, Student No: 200012101, and Nabila Islam Khan, Student No: 200012127 has been accepted as satisfactory in partial fulfillment of the requirements for the degree of B.Sc. in Industrial and Production Engineering on 29 September 2025.

1. _____

Prof. Dr. M. Ahsan Habib
Professor and Head
MPE Dept., IUT, Board Bazar, Gazipur-1704, Bangladesh.

(Supervisor)

CO-PO Mapping of IPE 4800 -Thesis and Project

COs	Course Outcomes (CO) Statement	(PO)	Addressed by	
CO1	Discover and Locate research problems and illustrate them via figures/tables or projections/ideas through field visit and literature review and determine/Setting aim and objectives of the project/work/research in specific, measurable, achievable, realistic and timeframe manner.	PO2 Problem analysis	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO2	Design research solutions of the problems towards achieving the objectives and its application. Design systems, components or processes that meets related needs in the field of mechanical engineering	PO3 Design/development of solutions	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO3	Review, debate, compare and contrast the relevant literature contents. Relevance of this research/study. Methods, tools, and techniques used by past researchers and justification of use of them in this work.	PO4 Investigation	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO4	Analyze data and exhibit results using tables, diagrams, graphs with their interpretation. Investigate the designed solutions to solve the problems through case study/survey study/experimentation/simulation using modern tools and techniques.	PO5 Modern tool usage	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO5	Apply moral values and research/professional ethics throughout the work, and justify genuine referencing on sources, and demonstration of own contribution.	PO8 Ethics	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO6	Perform own self and manage group activities from the beginning to the end of the research/work as a quality work.	PO9 Individual work and teamwork	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO7	Compile and arrange the work outputs, write the report/thesis, a sample journal paper, and present the work to wider audience using modern communication tools and techniques.	PO10 Communication	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√
CO8	Recognize the necessity of life-long learning in career development in dynamic real-world situations from the experience of completing this project.	PO12 Life-long learning	Thesis Book	√
			Performance by research	√
			Presentation and soft skill	√

Student Name /ID:

Signature of the Supervisor:

1. Md. Mahmudul Hasan Bhuiyan
(200012101)

Name of the Supervisor: Prof. Dr. M. Ahsan Habib

2. Nabila Islam Khan
(200012127)

K-P-A Mapping of IPE 4800 -Thesis and Project

COs	POs	Related Ks								Related Ps							Related As				
		K1	K2	K3	K4	K5	K6	K7	K8	P1	P2	P3	P4	P5	P6	P7	A1	A2	A3	A4	A5
CO1	PO2	✓	✓	✓	✓					✓											
CO2	PO3					✓				✓	✓	✓				✓	✓	✓	✓		
CO3	PO4								✓	✓	✓	✓					✓	✓			
CO4	PO5						✓			✓				✓							
CO5	PO8							✓													
CO6	PO9																				
CO7	PO10																				
CO8	PO12																				

Student Name /ID:

1. Md. Mahmudul Hasan Bhuiyan
(200012101)

2. Nabila Islam Khan
(200012127)

Signature of the Supervisor:

Name of the Supervisor: Prof. Dr. M. Ahsan Habib

List of Sustainable Development Goals (SDGs) Addressed in this Project

SDG No.	Goals	Targets	Relevance to the Thesis	Remarks
1	No Poverty	1.1 Eradicate extreme poverty (people living on less than \$1.25/day).		
		1.2 Reduce poverty in all its forms by at least half.		
		1.3 Implement nationally appropriate social protection systems.		
		1.4 Ensure equal rights to economic resources, services, property, inheritance, technology, and financial services.	√	Slight technological relevance via automation
		1.5 Build resilience of the poor and reduce exposure to climate-related and other shocks.		
		1.a Mobilize resources to end poverty.		
		1.b Create pro-poor policy frameworks.		
2	Zero Hunger	2.1 End hunger and ensure access to safe, nutritious food year-round.		
		2.2 End all forms of malnutrition.		
		2.3 Double agricultural productivity and incomes of small-scale producers.	√	Automation in textile could inspire agricultural automation and increase agricultural productivity
		2.4 Ensure sustainable food production systems and resilient agricultural practices.		
		2.5 Maintain genetic diversity of seeds, plants, and animals.		
		2.a Increase investment in rural infrastructure, research, and technology.		
		2.b Correct and prevent trade restrictions/distortions in global food markets.		
3	Good Health and Well-	3.1 Reduce global maternal mortality ratio.		
		3.2 End preventable deaths of newborns and under-5 children.		

		3.3 End epidemics of AIDS, tuberculosis, malaria, and neglected tropical diseases.		
		3.4 Reduce premature mortality from NCDs and promote mental health.	√	Indirect relevance through computer vision's potential in future medical imaging
		3.5 Strengthen prevention and treatment of substance abuse.		
		3.6 Halve global deaths/injuries from road traffic accidents.		
		3.7 Ensure universal access to sexual and reproductive healthcare.		
		3.8 Achieve universal health coverage.		
		3.9 Reduce deaths from hazardous chemicals, pollution, and contamination.		
		3.a Strengthen tobacco control (WHO FCTC).		
		3.b Support R&D of vaccines and medicines.		
		3.c Increase health financing and workforce.		
		3.d Strengthen capacity for early warning and risk management.	√	AI defect detection can be conceptually extended to early warning systems
4	Quality Education	4.1 Ensure all complete free, equitable, quality primary and secondary education.		
		4.2 Ensure access to quality early childhood development and pre-primary education.		
		4.3 Ensure equal access to affordable technical, vocational, and higher education.	√	This thesis contributes to higher education and research training in computer vision
		4.4 Increase skills for employment and entrepreneurship.	√	Develops technical skills in AI and automation that support future employment
		4.5 Eliminate gender disparities in education.		
		4.6 Ensure literacy and numeracy for youth and adults.		

		4.7 Ensure learners acquire knowledge/skills for sustainable development.		
		4.a Build and upgrade education facilities that are inclusive and safe.		
		4.b Expand scholarships for developing countries.		
		4.c Increase supply of qualified teachers.		
5	Gender Equality	5.1 End all forms of discrimination against women and girls.		
		5.2 Eliminate violence against women and girls.		
		5.3 Eliminate harmful practices (child, early, forced marriage, FGM).		
		5.4 Recognize and value unpaid care and domestic work.		
		5.5 Ensure women's participation in leadership and decision-making.	√	Academic research equally accessible regardless of gender
		5.6 Ensure universal access to reproductive health and rights.		
		5.a Undertake reforms to give women equal rights to resources.		
		5.b Enhance use of enabling technology to empower women.	√	AI and computer vision are enabling technologies that can promote inclusion in STEM
		5.c Adopt and strengthen policies and laws for gender equality.		
6	Clean Water and Sanitation	6.1 Achieve universal and equitable access to safe drinking water.		
		6.2 Achieve access to adequate sanitation and hygiene.		
		6.3 Improve water quality by reducing pollution.		
		6.4 Increase water-use efficiency and sustainable withdrawals.		
		6.5 Implement integrated water resources management.		
		6.6 Protect and restore water-related ecosystems.		
		6.a Expand international cooperation in water and sanitation.		
		6.b Support participation of local communities.		

7	Affordable and Clean Energy	7.1 Ensure universal access to affordable, reliable, modern energy services.		
		7.2 Increase substantially the share of renewable energy.		
		7.3 Double global rate of improvement in energy efficiency.	√	This thesis promotes computational efficiency
		7.a Enhance international cooperation on clean energy research/technology.		
		7.b Expand infrastructure and upgrade technology for sustainable energy.		
8	Decent Work and Economic Growth	8.1 Sustain per capita economic growth.	√	Support technical advancement contributing to industrial productivity
		8.2 Achieve higher levels of productivity through diversification, tech, and innovation.	√	This thesis promotes automation and innovation through AI and computer vision for textile inspection
		8.3 Promote policies for decent job creation and entrepreneurship.	√	Automation technologies may create new roles in AI-based industrial systems
		8.4 Improve resource efficiency in production and consumption.	√	The system improves inspection efficiency, reducing material waste and rework in textile production
		8.5 Achieve full and productive employment for all.		
		8.6 Substantially reduce youth not in employment/education/training.		
		8.7 Eradicate forced labour, modern slavery, and child labour.		
		8.8 Protect labour rights and safe working environments.	√	Automation reduces manual defect inspection risks, enhancing worker safety
		8.9 Promote sustainable tourism.		
		8.a Increase aid for trade support.		

		8.b Develop a global youth employment strategy.		
9	Industry, Innovation, and Infrastructure	9.1 Develop quality, reliable, sustainable infrastructure.	√	Contributes to sustainable industrial digitalization via AI inspection
		9.2 Promote inclusive and sustainable industrialization.	√	Encourages automation and modernization in the textile sector
		9.3 Increase access of SMEs to financial services and integration into value chains.	√	The system could support SMEs through improved quality control efficiency
		9.4 Upgrade infrastructure for sustainability and resource efficiency.	√	Directly supports resource-efficient, automated textile inspection reducing manual waste
		9.5 Enhance scientific research and technology development.	√	Advances AI research for industrial defect detection
		9.a Facilitate sustainable infrastructure in developing countries.	√	Provides a low-cost digital approach feasible for developing economies
		9.b Support domestic tech development and value addition.	√	Strengthens domestic research capacity in computer vision and AI for manufacturing
		9.c Increase access to ICT and internet.	√	Related to ICT
10	Reduced Inequalities	10.1 Achieve income growth of bottom 40%.		
		10.2 Empower and promote inclusion regardless of status.		
		10.3 Ensure equal opportunity and reduce inequalities of outcome.		
		10.4 Adopt policies for fiscal, wage, and social protection equality.		
		10.5 Improve regulation of global financial markets.		
		10.6 Ensure enhanced representation in global institutions.		

		10.7 Facilitate safe, regular, and responsible migration.		
		10.a Implement special treatment for developing countries.		
		10.b Encourage development assistance and investment in least developed areas.		
		10.c Reduce remittance costs.		
11	Sustainable Cities and Communities	11.1 Ensure access to adequate, safe, and affordable housing.		
		11.2 Provide sustainable transport systems.		
		11.3 Enhance inclusive urbanization and capacity for planning.		
		11.4 Protect cultural and natural heritage.		
		11.5 Reduce disaster impact and losses.		
		11.6 Reduce environmental impact of cities (air quality, waste).		
		11.7 Provide access to safe, inclusive green/public spaces.		
		11.a Support positive links between urban, peri-urban, rural.		
		11.b Increase disaster risk reduction strategies.		
		11.c Support least developed countries in sustainable building.		
12	Responsible Consumption and Production	12.1 Implement 10-Year Framework on sustainable consumption/production.	√	The thesis promotes sustainable textile manufacturing through AI-based defect reduction
		12.2 Achieve sustainable management and use of resources.	√	Automated defect detection minimizes material waste, ensuring efficient use of fabric
		12.3 Halve per capita global food waste.		
		12.4 Manage chemicals and waste sustainably.		
		12.5 Substantially reduce waste generation.	√	Defect localization reduces production waste and improves process efficiency
		12.6 Encourage companies to adopt sustainable practices.	√	Encourages textile industries to adopt AI-driven

				sustainable inspection
		12.7 Promote sustainable public procurement.		
		12.8 Ensure people have relevant information for sustainable development.		
		12.a Support developing countries' scientific and technological capacity.		
		12.b Develop tools to monitor sustainable tourism impacts.		
		12.c Rationalize inefficient fossil-fuel subsidies.		
13	Climate Action	13.1 Strengthen resilience and adaptive capacity to climate-related hazards.		
		13.2 Integrate climate measures into national policies.		
		13.3 Improve education and awareness on climate change.		
		13.a Implement UNFCCC commitments (mobilize \$100 billion annually).		
		13.b Promote mechanisms for capacity-building in least developed countries.		
14	Life Below Water	14.1 Reduce marine pollution.		
		14.2 Sustainably manage and protect marine ecosystems.		
		14.3 Minimize and address ocean acidification.		
		14.4 Regulate harvesting and end overfishing.		
		14.5 Conserve at least 10% of coastal and marine areas.		
		14.6 Prohibit harmful fisheries subsidies.		
		14.7 Increase economic benefits from sustainable marine resources.		
		14.a Increase scientific knowledge and marine technology transfer.		
		14.b Provide access for small-scale artisanal fishers.		
		14.c Implement international law for oceans.		
15	Life on Land	15.1 Conserve terrestrial and freshwater ecosystems.		
		15.2 Promote sustainable management of forests.		

		15.3 Combat desertification and restore degraded land.		
		15.4 Ensure conservation of mountain ecosystems.		
		15.5 Take urgent action to reduce biodiversity loss.		
		15.6 Promote fair benefit-sharing from genetic resources.		
		15.7 End poaching and trafficking of protected species.		
		15.8 Prevent introduction of invasive alien species.		
		15.9 Integrate ecosystem values into policies/planning.		
		15.a Mobilize resources for biodiversity.		
		15.b Finance sustainable forest management.		
		15.c Support local communities for forest and wildlife.		
16	Peace, Justice and Strong Institutions	16.1 Reduce violence and related death rates.		
		16.2 End abuse, trafficking, and violence against children.		
		16.3 Promote rules of law and equal access to justice.		
		16.4 Reduce illicit financial/arms flows, organized crime.		
		16.5 Reduce corruption and bribery.		
		16.6 Develop effective, accountable institutions.		
		16.7 Ensure inclusive, participatory decision-making.		
		16.8 Broaden participation of developing countries in global governance.		
		16.9 Provide legal identity for all (including birth registration).		
		16.10 Ensure access to information and protect freedoms.		
		16.a Strengthen national institutions for prevention of violence.		
		16.b Promote/enforce non-discriminatory laws and policies.		

17	Partnerships for the Goals	17.1 Strengthen domestic resource mobilization.		
		17.2 Developed countries to implement ODA commitments.		
		17.3 Mobilize additional financial resources.		
		17.4 Assist developing countries with debt sustainability.		
		17.5 Invest in least developed countries.		
		17.6 Enhance access to science, technology, innovation.	√	The thesis is a collaborative academic work contributing to global AI research
		17.7 Promote environmentally sound technologies.	√	The research promotes sustainable and efficient AI technology in manufacturing
		17.8 Fully operationalize technology bank for LDCs.	√	Developed at IUT, advancing research capacity in AI within a developing country
		17.9 Enhance international support for capacity-building.	√	Contributes to skill and research capacity in industrial AI
		17.10 Promote a universal, rules-based trading system (WTO).		
		17.11 Increase exports of developing countries.		
		17.12 Timely implementation of duty-free, quota-free market access.		
		17.13 Enhance global macroeconomic stability.		
		17.14 Enhance policy coherence for sustainable development.		
		17.15 Respect national policy space.		
		17.16 Enhance global partnerships.		
		17.17 Encourage multi-stakeholder partnerships.		
		17.18 Enhance data capacity of developing countries.		
		17.19 Support capacity-building for sustainable development indicators.		

Acknowledgment

First and foremost, all praise and gratitude are due to Almighty Allah, the Most Gracious and the Most Merciful, whose infinite blessings, guidance, and mercy have enabled us to bring this thesis to successful completion. It is only through His divine wisdom and benevolence that we were granted the strength, patience, and perseverance to overcome the challenges faced throughout this academic journey. Every milestone achieved in this work is a manifestation of His boundless grace and kindness.

We would like to express our profound gratitude to Prof. Dr. M. Ahsan Habib, Professor and Head, Department of Mechanical and Production Engineering, Islamic University of Technology (IUT), Bangladesh, for his invaluable supervision, continuous encouragement, and insightful guidance throughout the course of this thesis. His deep expertise, constructive feedback, and constant motivation have been a source of inspiration and have significantly enriched the quality and direction of our work.

We are also deeply grateful to Mr. Abid Hasan of McGill University, Canada for his dedicated mentorship and continuous support throughout the progress of this thesis. His profound expertise in computer vision, insightful guidance, and patient mentorship played a vital role in shaping our understanding and execution of the research. His willingness to share knowledge and provide direction at every critical stage was invaluable to our learning experience and the overall success of this work.

We gratefully acknowledge the Islamic University of Technology (IUT) for providing financial support, which facilitated the procurement of a Google Colab Pro+ subscription valued at 14,480 BDT. This computational resource was essential for conducting the large-scale deep learning experiments required for this thesis.

Finally, we extend our heartfelt gratitude to our families, friends, and well-wishers for their constant encouragement, prayers, and unwavering support. Their patience, understanding, and faith in us have been a continuous source of strength and motivation throughout this endeavor.

Abstract

The increasing demand for automated visual inspection in textile manufacturing has intensified the demand for strong and efficient defect detection systems. Traditional manual inspection methods are often inconsistent and resource-intensive, while the inherent variability of fabric textures creates obstacles for conventional machine vision techniques. To overcome these constraints, this study presents a comparative analysis of supervised semantic segmentation and unsupervised anomaly detection models for fabric defect localization using the ZJU-Leaper benchmark dataset. A total of nine deep learning architectures were evaluated across three methodological categories: (i) CNN-based supervised models (U-Net, PSPNet, FPN, DeepLabV3, U-Net++, and DeepLabV3+), (ii) Transformer-based supervised models (UPerNet and SegFormer), and (iii) an unsupervised anomaly detection model (EfficientAD). All models were trained on standardized preprocessed images and optimized under identical hyperparameter settings to ensure fair benchmarking. The evaluation followed the multi-level protocol defined in the ZJU-Leaper paper, incorporating sample-, pixel-, and region-level metrics, alongside efficiency indicators such as FLOPs, parameter count, and FPS to assess deployment feasibility. The results show distinct trade-offs between accuracy and efficiency across the examined architectures. Transformer-based models demonstrated superior generalization on complex textures, achieving balanced performance between global context capture and computational cost. CNN-based models excelled in fine-grained localization, offering high Dice and IoU scores at moderate inference speeds. Conversely, the unsupervised EfficientAD model achieved competitive detection accuracy without relying on labeled data, underscoring its potential for defect-scarce industrial scenarios. Overall, this research establishes a unified evaluation framework for benchmarking supervised and unsupervised deep learning models in fabric inspection. The results show into the architectural and operational considerations necessary for developing scalable, real-time, and label-efficient quality control systems in smart textile manufacturing environments.

Summary

Ensuring surface quality is a critical requirement in modern manufacturing industries, where even minor defects can lead to product rejection or safety concerns. Conventional visual inspection methods rely heavily on human labor, which is time-consuming, inconsistent, and difficult to scale. Consequently, automated defect detection using artificial intelligence (AI) has gained increasing attention as a reliable and efficient alternative.

This thesis presents a comparative study between supervised learning and unsupervised anomaly detection approaches for surface defect localization. Supervised models, such as Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), require large volumes of annotated defect data, whereas unsupervised methods can identify irregularities by learning from defect-free samples alone. Each approach offers unique advantages and limitations in terms of accuracy, data dependency, and computational cost.

To ensure fair evaluation, all experiments were conducted using the ZJU-Leaper dataset, a large-scale, diverse benchmark specifically designed for industrial defect detection research. The dataset enables consistent comparison across multiple defect types and textures using standardized pixel-, region-, and sample-level metrics.

The study systematically evaluates the performance, efficiency, and practicality of representative supervised and unsupervised models under the same experimental conditions. Findings from this research highlight the trade-offs between accuracy and data requirements, offering actionable insights for industries aiming to implement automated inspection systems. The results serve as a reference framework for selecting suitable AI models and deployment strategies according to specific production environments and resource constraints.

Table of Contents

Acknowledgment	xv
Abstract	xvii
Summary	xviii
Table of Contents	xix
List of Figures	xxii
List of Tables	xxv
Nomenclatures and Symbol	xxvii
Chapter 1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives of the Study	2
1.3 Organization of the Thesis	3
Chapter 2 Literature Review	5
2.1 Rise of Deep Learning in Defect Detection	5
2.1.1 Transition to Transformer-Based Models	6
2.1.2 Unsupervised Learning and Emerging Paradigms	7
2.2 Fundamental Supervised CNN Architectures	8
2.2.1 Early Semantic Segmentation: U-Net	8
2.2.2 Pyramid Scene Parsing Network (PSPNet)	9
2.2.3 Feature Pyramid Network (FPN)	9
2.2.4 U-Net++	10
2.2.5 DeepLabV3	11
2.2.6 DeepLabV3+	11
2.2.7 Comparative Insights	12
2.3 Transformer-Based Semantic Segmentation	12
2.3.1 Evolution from CNNs to Vision Transformers	13
2.3.2 Unified Perceptual Parsing (UPerNet)	13
2.3.3 SegFormer	14

2.3.4 Observations	14
2.4 Data Scarcity and Unsupervised Learning Paradigm.....	15
2.4.1 Challenges in Labeling Industrial Data	15
2.4.2 Unsupervised Anomaly Detection.....	15
2.4.3 Other Notable Methods	16
2.4.4 Application to Fabric Defect Detection.....	16
2.5 Computational Efficiency and Model Complexity	16
2.5.1 Motivation for Efficiency Evaluation.....	16
2.5.2 Implications for Industrial Deployment	17
2.6 Experimental Design Considerations and Architectural Choices	18
2.6.1 Backbones.....	18
2.6.2 Loss Function	20
2.6.3 Optimization Strategies	20
2.7 Summary of Literature and Research Gap	20
Chapter 3 Methodology	23
3.1 Introduction	23
3.2 Data Collection.....	26
3.1.1 Dataset Source	26
3.1.2 Dataset Composition and Characteristics	30
3.3 Dataset Preparation	31
3.2.1 Texture Grouping	31
3.2.2 Dataset Split.....	31
3.2.3 Ethical Considerations	33
3.4 Pre-Processing.....	33
3.2.1 Image Standardization	34
3.2.2 Data Loading and Augmentation.....	35
3.2.3 Unsupervised Model Pre-processing (EfficientAD).....	35
3.5 Model Architectures	36
3.3.1 Supervised CNN-Based Architectures	37
3.3.2 Supervised Transformer-Based Architectures.....	43
3.3.3 Unsupervised Anomaly Detection Architecture.....	45

3.6 Evaluation Metrics	46
3.6.1 Sample-Level Performance Evaluation	47
3.6.2 Pixel-Level Performance Evaluation	48
3.6.3 Region-Level Performance Evaluation	49
3.6.4 Efficiency Metrics	51
3.7 Training Configuration and Hyperparameters	53
Chapter 4 Results and Discussion	55
4.1 Sample Level Performance	55
4.1.1 Group-Wise Performance Analysis	57
4.1.2 Model-Wise Performance Analysis	58
4.2 Pixel-Level Performance	60
4.2.1 PR-Curve Analysis	63
4.2.2 Group-Wise Performance Analysis	69
4.2.3 Model-Wise Performance Analysis	71
4.2.4 Qualitative Pixel-Level Analysis	73
4.3 Region-Level Performance	81
4.3.1 Large Defect Region Performance	84
4.3.2 Medium Region Performance	87
4.3.3 Small Region Performance	90
4.4 Accuracy–Efficiency Trade-off Analysis	93
4.4.1 Model Complexity: Parameters and FLOPs	93
4.4.2 Runtime Performance: Inference Speed and Latency	95
4.4.3 Accuracy vs Efficiency Correlation	97
4.4.5 Discussion	102
Chapter 5 Conclusion and Future Direction	104
5.1 Conclusion	104
5.2 Future Work	105
References	108

List of Figures

Figure 3-1: Overview of the Supervised Fabric Defect Detection Pipeline. The flowchart illustrates the sequential process from data collection and preprocessing to model training, inference, and performance evaluation using the ZJU-Leaper dataset.	24
Figure 3-2: Overview of the Unsupervised Fabric Defect Detection Pipeline. The process, based on the ZJU-Leaper dataset, trains exclusively on normal images to learn texture patterns and detect anomalies during inference.	25
Figure 3-3: Representative Normal Fabric Samples from the Five Texture-Based Groups in the ZJU-Leaper Dataset [4].	27
Figure 3-4: Representative Defective Fabric Samples from the Five Texture-Based Groups in the ZJU-Leaper Dataset.	28
Figure 3-5: Corresponding Ground Truth Pixel-Wise Masks for the Defective Samples Shown in Figure 3-4.	29
Figure 4-1: Comparison of Sample-Level F1 Scores of All Evaluated Models Across Group 1-5.	56
Figure 4-2: Comparison of Pixel-Level F1 Scores of All Evaluated Models Across Group 1-5.	62
Figure 4-3: Precision-Recall Curves of Evaluated Models for Group 1.	65
Figure 4-4: Precision-Recall Curves of Evaluated Models for Group 2.	66
Figure 4-5: Precision-Recall Curves of Evaluated Models for Group 3.	67
Figure 4-6: Precision-Recall Curves of Evaluated Models for Group 4.	68

Figure 4-7: Precision-Recall Curves of Evaluated Models for Group 5.....	69
Figure 4-8: Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 1	74
Figure 4-9: Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 2	75
Figure 4-10: Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 3	76
Figure 4-11: Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 4	77
Figure 4-12: Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 5	78
Figure 4-13: Comparison of Region-Level F1 Scores of All Evaluated Models Across Group 1-5	83
Figure 4-14: Comparison of Region-Level F1 Scores for Large Regions of All Evaluated Models Across Group 1-5.....	86
Figure 4-15: Comparison of Region-Level F1 Scores for Medium Regions of All Evaluated Models Across Group 1-5.....	89
Figure 4-16: Comparison of Region-Level F1 Scores for Small Regions of All Evaluated Models Across Group 1-5.....	92
Figure 4-17: Scatter plots illustrating the trade-off between segmentation accuracy (F1-score) and inference speed (FPS) across Groups 1–5. Each bubble represents a model, with its size proportional to the number of parameters, showing how different architectures balance accuracy and runtime efficiency.	99

Figure 4-18: Scatter plots showing the trade-off between segmentation accuracy (F1-score) and computational complexity (FLOPs) across Groups 1–5. Bubble sizes denote parameter counts, highlighting how model complexity influences segmentation performance. 101

List of Tables

Table 3-1: Distribution of Normal and Defective Images Across 19 Fabric Patterns in the ZJU-Leaper Dataset	30
Table 3-2: Texture Grouping Scheme (based on ZJU-Leaper organization)	31
Table 3-3: Dataset split showing the number of images in the training, validation, and test sets across the five dataset groups for supervised models.	32
Table 3-4: Dataset split showing the number of images in the training, validation, and test sets across the five dataset groups for unsupervised model.....	33
Table 3-5: Summary of CNN-based Architectures.....	43
Table 3-6: Summary of Transformer-Based Architectures	45
Table 3-7: Summary of Unsupervised Architecture	46
Table 3-8: System and Training Configuration Summary.....	54
Table 3-9: Hyperparameter Settings	54
Table 4-1: Sample-Level F1 Scores by Model and Dataset Group	55
Table 4-2: Pixel-Level F1 Scores by Model and Dataset Group	61
Table 4-3: Region-Level F1 Scores by Model and Dataset Group.....	82
Table 4-4: Region-Level F1 Scores for Large Regions by Model and Dataset Group.....	84
Table 4-5: Region-Level F1 Scores for Medium Regions by Model and Dataset Group	87
Table 4-6: Region-Level F1 Scores for Small Regions by Model and Dataset Group.....	90
Table 4-7: Computational Complexity of Segmentation Models in Terms of FLOPs and Parameter Count.....	94

Table 4-8: Inference Speed and Latency of Segmentation Models on Group 1	95
Table 4-9: Inference Speed and Latency of Segmentation Models on Group 2	95
Table 4-10: Inference Speed and Latency of Segmentation Models on Group 3	96
Table 4-11: Inference Speed and Latency of Segmentation Models on Group 4	96
Table 4-12: Inference Speed and Latency of Segmentation Models on Group 5	97

Nomenclatures and Symbol

Abbreviation	Full Form / Description
AD	Anomaly Detection
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
BCE	Binary Cross-Entropy
BCE-Dice	Binary Cross-Entropy with Dice Loss
CNN	Convolutional Neural Network
CV	Computer Vision
DeepLabV3	DeepLab Version 3
DeepLabV3+	DeepLab Version 3 Plus
DL	Deep Learning
EfficientAD	Efficient Anomaly Detection
FLOPs	Floating Point Operations
FPN	Feature Pyramid Network
FPS	Frames Per Second
GPU	Graphics Processing Unit
GT	Ground Truth
IoU	Intersection over Union
LR	Learning Rate
MCDM	Multi-Criteria Decision Making
Param(s)	Parameter(s)
PSPNet	Pyramid Scene Parsing Network
SegFormer	Segmentation Transformer
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
U-Net	U-shaped Convolutional Neural Network
U-Net++	Nested U-Net architecture
UPerNet	Unified Perceptual Parsing Network
ViT	Vision Transformer
ZJU-Leaper	Zhejiang University-Leaper Dataset

Symbol	Definition / Description
TP	True Positives — correctly predicted defective pixels or regions
FP	False Positives — normal pixels or regions predicted as defective
FN	False Negatives — defective pixels or regions missed by the model
TN	True Negatives — correctly predicted normal pixels
P	Predicted binary region or segmentation mask
G	Ground-truth binary region or segmentation mask
IoU	Intersection over Union, a pixel-wise overlap measure between prediction and ground truth
F_1	F1-score or Dice coefficient (harmonic mean of precision and recall)
F_1^{pix}	Pixel-level F1-score
F_1^{sam}	Sample-level F1-score
F_1^{reg}	Region-level F1-score
$F_1^{reg}(s)$	Region-level F1-score for small-sized defects
$F_1^{reg}(m)$	Region-level F1-score for medium-sized defects
$F_1^{reg}(l)$	Region-level F1-score for large-sized defects
TP^{pix}	True positive pixel count
$FP^{\{pix\}}$	False positive pixel count
FN^{pix}	False negative pixel count
TP^{sam}	Matched sample count (true positive samples)
FP^{sam}	Unmatched sample predictions (false positives)
FN^{sam}	Missed defective samples (false negatives)
TP^{reg}	True positive region count
FP^{reg}	False positive region count
FN^{reg}	Missed defect region count

Chapter 1 Introduction

1.1 Background and Motivation

Fabric defect detection plays a fundamental role in quality control within the garment and textile industries, which are vital contributors to a nation's economic growth. Traditionally, defect identification has relied heavily on manual inspection methods. However, manual inspection is inherently limited by factors such as human fatigue, lapses in concentration, and the substantial time required to thoroughly examine fabric surfaces. These limitations negatively impact both the consistency and efficiency of the quality assurance process, possibly resulting in elevated production costs and compromised product quality [1].

Early automated methods based on classical image processing techniques, such as thresholding and morphological filtering, attempted to overcome these challenges. Nevertheless, their performance is often hindered by sensitivity to varying illumination conditions, diverse textile textures, and complex fabric patterns. Recently, deep learning approaches have demonstrated significant advances in defect detection [2]. Supervised semantic segmentation models and unsupervised anomaly detection techniques show promise in accurately localizing defects at the pixel level [3]. In light of these advancements, a thorough comparative analysis of these methods, particularly under uniform evaluation conditions, remains largely unexplored.

Selecting a suitable dataset is crucial for conducting a meaningful comparative study of fabric defect detection methods. The ZJU-Leaper [4] dataset stands out as an ideal choice due to its comprehensive coverage of 19 diverse textile texture classes and multiple defect types, reflecting the complexity found in real industrial settings. Its size, diversity, and annotation

quality collectively provide a robust benchmark that enables fair and extensive evaluation of supervised segmentation and unsupervised anomaly detection methods.

Selecting model efficiency metrics alongside accuracy is also essential for practical deployment. Deep learning advancements have led to larger and more complex models with increased parameters, latency, and resource demands. Thus, evaluating computational footprints such as floating-point operations (FLOPs) and parameter counts provides valuable insight into a model’s feasibility in industrial settings [5]. Recent comprehensive surveys emphasize that addressing efficiency, from modeling techniques through to hardware considerations, is critical not only for improving training and deployment but also for enabling real-world applications in resource-constrained environments. Building on this foundation, this study reviews state-of-the-art supervised convolutional neural networks alongside emerging transformer-based architectures and unsupervised anomaly detection frameworks, with a focus on both accuracy and computational efficiency.

1.2 Objectives of the Study

The primary aim of this study is to conduct a comprehensive comparative analysis of supervised semantic segmentation and unsupervised anomaly detection approaches for fabric defect localization using the ZJU-Leaper [4] benchmark dataset. The research seeks to establish a unified framework that not only assesses detection accuracy but also evaluates scalability and computational efficiency, factors essential for real-time industrial deployment. Through systematic experimentation and standardized evaluation protocols, the study aims to identify how different architectural paradigms perform across varying defect complexities and under resource constraints. By integrating both accuracy-driven and efficiency-driven assessments, this work aspires to provide actionable insights for developing robust, label-efficient, and computationally feasible defect inspection models for the textile industry. This work has the

following objectives:

1. To compare and benchmark supervised semantic segmentation and unsupervised anomaly detection models for fabric defect localization.
2. To evaluate defect localization performance using multi-level and multi-scale metrics, including region-level analysis of small, medium, and large defects.
3. To analyze model efficiency by evaluating computational complexity (FLOPs), parameter count (model size), and practical inference speed, highlighting the trade-offs between accuracy and efficiency.

1.3 Organization of the Thesis

This thesis is structured into five chapters to present the research in a logical and systematic manner. Chapter 1 introduces the overall research background, motivation, and problem statement associated with fabric defect localization. It further outlines the aims and objectives of the study and provides a brief overview of the structure of the thesis. Chapter 2 presents an extensive literature review covering prior research on deep learning-based defect detection. It discusses the evolution from conventional machine vision to modern deep learning techniques, including convolutional neural networks (CNNs), transformer-based models, and unsupervised anomaly detection approaches. The chapter concludes by identifying existing research gaps and establishing the motivation for the present comparative study. Chapter 3 describes the detailed methodology adopted for this research. It includes dataset preparation, preprocessing procedures, model architectures, evaluation protocols, and the experimental setup used for both supervised and unsupervised learning paradigms. Chapter 4 presents and discusses the results obtained from the experiments. It analyzes model performance across sample-, pixel-, and region-level evaluations and investigates the accuracy–efficiency trade-offs among different

architectures. Finally, Chapter 5 concludes the study by summarizing the key findings, highlighting their implications for industrial fabric inspection, and suggesting potential directions for future research.

Chapter 2 Literature Review

2.1 Rise of Deep Learning in Defect Detection

Convolutional Neural Networks (CNN)-based semantic segmentation has emerged as a breakthrough approach in automated visual inspection tasks, including fabric defect detection. Architectures such as U-Net [6], FPN [7], PSPNet [8], and DeepLab [9] have demonstrated strong capabilities in capturing both spatial context and precise localization of defects. U-Net [6], with its encoder–decoder design and skip connections, excels in learning local and global features. This is particularly effective even with limited training samples due to its use of data augmentation strategies. It has been successfully applied beyond biomedical images to industrial defect detection, showing robust segmentation performance.

PSPNet [8] and DeepLab [9] further enhance defect detection performance by integrating multi-scale context through pyramid pooling or atrous spatial pyramid pooling, allowing better handling of varying defect sizes and complex texture backgrounds. These architectures have been validated extensively in related domains such as mineral and medical image segmentation, showcasing high segmentation accuracy and generalization ability.

Despite these advances, supervised segmentation techniques like these require large-scale, pixel-level annotations, which are labor-intensive and costly to obtain in industrial settings. The diversity and complexity of fabric defects require skilled human inspectors for accurate labeling, which poses a major bottleneck for scalable deployment, especially given the challenges of obtaining large-scale, pixel-level annotations. The issue of data imbalance and annotation noise also affects model performance and generalization. Recent work has explored domain adaptation and self-labeling techniques to mitigate labeling costs, but supervised models still hinge on substantial annotated data.

The established effectiveness of U-Net [6], PSPNet [8], FPN [7] and DeepLab[9] in segmentation, combined with the challenges related to annotation cost, provides a strong rationale for focusing on these architectures in comparative studies alongside unsupervised approaches.

2.1.1 Transition to Transformer-Based Models

Vision Transformers (ViTs) [10] mark a significant evolution beyond convolutional neural networks by introducing a global receptive field that enables better contextual modeling in image analysis. Unlike CNNs that rely primarily on local receptive fields, ViTs process images as sequences of fixed-size patches, capturing long-range dependencies through self-attention mechanisms. This attribute enhances the representation of complex textures and spatial relationships, critical for fabric defect detection in industrial settings [10].

Key transformer-based architectures such as UPerNet [11] and SegFormer [12] exemplify this paradigm shift. UPerNet integrates multi-scale feature representations for unified perceptual parsing across varied visual concepts [11], showing strong performance in scene understanding tasks relevant to industrial and texture analysis. SegFormer offers a lightweight yet powerful semantic segmentation framework that combines a hierarchical Transformer encoder with efficient multilayer perceptron decoders [12]. Its design avoids positional encodings, improving robustness to varying input resolutions and delivering superior accuracy and efficiency compared to prior models.

As Transformers continue to gain traction in industrial image analysis, their capacity to model both local and global patterns make them especially suited for fabric inspection tasks where defects often involve subtle, spatially dispersed irregularities. Early applications of ViT in defect detection demonstrate improved accuracy over CNNs by capturing global context while maintaining efficient computation. This emerging evidence motivates the inclusion of

transformer-based models in comparative analyses assessing fabric defect localization methods.

2.1.2 Unsupervised Learning and Emerging Paradigms

Unsupervised anomaly detection has become an important paradigm in fabric defect detection as it circumvents the expensive requirement for pixel-level annotation. These methods train exclusively on defect-free samples to model normality [13] and identify defects as deviations from this learned distribution, thus eliminating the need for labeled anomalous examples.

Deep autoencoders represent one foundational approach, where the model learns to reconstruct normal images, expecting deviations in reconstruction error to signal anomalies [13]. However, vanilla autoencoders sometimes generalize too well, reconstructing anomalies effectively and missing defects. To counter this, memory-augmented autoencoders (MemAE) [14], have been proposed, which incorporate a memory module that retains prototypical normal elements to strengthen reconstruction error on anomalies, thus improving detection robustness across diverse data types.

Patch distribution modeling frameworks such as PaDiM [15] employ pretrained CNN features and model patch-wise embeddings as multivariate Gaussian distributions. By capturing layer-wise semantic correlations, PaDiM [15] achieves fine-grained localization of anomalies and attains state-of-the-art performance on industrial datasets like MVTec AD [16]. PatchCore [17], another contemporary method, optimizes feature extraction and hyperparameters to excel in low-shot anomaly detection scenarios, indicating the potential of adapting full-shot models for practical, data-scarce industrial applications.

Conditional normalizing flow models like CFLOW-AD [18] combine discriminative encoders with multi-scale generative decoders to estimate feature likelihoods, offering real-time and memory-efficient anomaly localization surpassing previous techniques in accuracy and speed.

Building on these advances, EfficientAD [19] introduces a lightweight student-teacher framework that maximizes computational efficiency and maintains high accuracy, detecting both structural and logical anomalies at millisecond latencies, thus making the approach suitable for real-world high-throughput manufacture.

Methods such as PUAD [20] focus on detecting logical anomalies like incorrect object counts by employing out-of-distribution detection in feature space, complementing spatial anomaly localization techniques. Together, these emerging unsupervised paradigms exhibit strong potential for industrial fabric defect detection, enabling scalable and efficient quality control without exhaustive anomaly annotations.

2.2 Fundamental Supervised CNN Architectures

This section traces the evolution of convolutional neural network (CNN)-based semantic segmentation models pivotal for fabric defect localization. The focus is on foundational architectures that have established benchmarks in pixel-level segmentation precision, setting the stage for advanced industrial defect inspection.

2.2.1 Early Semantic Segmentation: U-Net

The U-Net architecture [6] is widely recognized for pioneering an encoder–decoder framework that combines feature extraction with precise localization. Its symmetric U-shaped design includes a contracting path (encoder) that captures context through convolutional and pooling layers while progressively reducing spatial dimensions. A corresponding expansive path (decoder) performs up sampling to restore spatial resolution.

Critical to U-Net’s success are skip connections, which directly transfer feature maps from the encoder to the decoder at matching resolutions. This mechanism mitigates spatial information loss characteristic of deep architectures and enables accurate pixel-wise segmentation, which

is essential for defect localization across diverse fabric textures.

Originally developed for biomedical image segmentation, U-Net has been extensively adopted in industrial contexts due to its efficiency with limited annotated data and its capability to integrate both local texture and global context. It has demonstrated consistent performance in discriminating subtle defects, although its effectiveness can degrade on highly complex textures or low-contrast anomalies.

2.2.2 Pyramid Scene Parsing Network (PSPNet)

Building on this foundation, PSPNet enhances multi-scale contextual aggregation through landmark pyramid pooling modules. By pooling features at varying spatial scales, PSPNet captures diverse contextual information, enabling the network to better differentiate between defect patterns at multiple resolutions and spatial extents [8].

This multi-level feature representation benefits industrial defect detection by improving segmentation consistency across fabrics with varying textures and defect sizes. The pyramid pooling module [21] balances fine-grained detail sensitivity with a broader contextual understanding, mitigating challenges posed by texture heterogeneity.

2.2.3 Feature Pyramid Network (FPN)

Feature Pyramid Networks (FPN), further enrich representation by constructing top-down pathways with lateral connections to fuse high-level semantic features with lower-level spatial features. This hierarchical fusion allows the model to leverage both detailed localization cues and abstract feature maps, a key advantage for fabric defect segmentation where defect scales and appearances vary widely [7].

The top-down architecture efficiently aggregates features across multiple scales without substantial additional computational overhead. FPNs have been shown to improve detection

and segmentation tasks in manufacturing datasets [7], offering a favorable trade-off between accuracy and inference efficiency.

2.2.4 U-Net++

U-Net++ [22] builds upon the foundational U-Net architecture by introducing densely connected, nested skip pathways that progressively refine feature maps and explicitly reduce the semantic gap between encoder and decoder representations. Unlike U-Net’s straightforward skip connections that directly connect high-resolution but semantically dissimilar encoder features to the decoder, U-Net++’s redesigned pathways gradually enrich encoder feature maps before fusion, encouraging greater semantic alignment. This architectural nuance simplifies the optimization process as the model learns to reconcile semantically similar feature maps, ultimately enabling more effective recovery of fine details crucial for accurate segmentation of subtle and small defects.

This deeper supervision mechanism embedded in U-Net++ propagates gradients throughout the network, enhancing feature discrimination at multiple scales and improving segmentation outcomes. In the context of medical imaging ; where precise lesion delineation can critically impact diagnosis, U-Net++ demonstrated significant gains in Intersection over Union (IoU), improving by approximately 3.9 points compared to U-Net and 3.4 points over an expanded wide U-Net variant. These improvements validate its superior capability in capturing fine-grained structures and minimizing segmentation errors that can degrade clinical utility [22]. Translating these advances to industrial fabric defect detection, the dense and nested skip connections offer tangible advantages in handling heterogeneous textures and varying defect scales typical in textiles. By enforcing semantic consistency between encoder and decoder features, U-Net++ [22] enhances boundary delineation, ensuring that complex or low-contrast defects are more reliably localized. Its architecture thus addresses typical weaknesses of baseline U-Net models in textile applications, providing a pathway for improved accuracy and

robustness in fabric defect segmentation tasks.

2.2.5 DeepLabV3

The original DeepLab [9] model introduced by Liang-Chieh Chen et al. innovatively addresses three key challenges in semantic image segmentation: reduced feature resolution, multi-scale object representation, and localization accuracy degradation caused by convolutional neural network (CNN) invariance. DeepLab’s core contribution is atrous convolution (also known as dilated convolution), which explicitly controls the resolution at which feature responses are computed without increasing the number of parameters or computational cost. Atrous convolution enlarges the receptive field, enabling the model to gather larger context while preserving spatial resolution essential for dense prediction tasks [9].

To capture objects at multiple scales, DeepLab proposes Atrous Spatial Pyramid Pooling (ASPP), which applies parallel atrous convolutions with different dilation rates on the feature maps. This multi-scale pooling robustly encodes both local and global context, improving segmentation performance on objects varying in size. DeepLab further enhances object boundary localization by integrating fully connected Conditional Random Fields (CRFs), refining segmentation masks to align better with object edges. This combination established a new state-of-the-art on PASCAL VOC-2012 and other benchmarks, demonstrating substantial practical merit for dense prediction problems involving complex imagery [9].

2.2.6 DeepLabV3+

DeepLabV3+ [23] builds on DeepLab by incorporating a simple yet effective decoder module designed to recover sharp object boundaries, a limitation in its predecessor. The encoder leverages atrous convolution to extract rich multi-scale semantic features efficiently, allowing configurable output feature resolution to balance accuracy and computational budget. The decoder progressively fuses these encoded features with low-level spatial features from earlier

backbone layers to restore spatial details lost during down sampling.

Additionally, DeepLabV3+ employs depth wise separable convolutions, lightweight operations that significantly reduce computation, within both the ASPP and decoder modules. This results in a faster and stronger encoder-decoder framework, achieving superior segmentation accuracy with improved efficiency [23]. This validates DeepLabV3+ [23] as a state-of-the-art architecture well-suited for applications requiring both precise defect localization and computational practicality, such as fabric defect detection in industrial settings.

2.2.7 Comparative Insights

Collectively, these CNN architectures represent a continuum of innovation in feature extraction, multi-scale context integration, and spatial detail preservation. While U-Net [6] offers a robust baseline, enhancements in PSPNet [8], FPN[7], U-Net++[22], and DeepLabV3+ [23] address limitations around scale variance and complex texture modeling.

Trade-offs between segmentation accuracy and computational efficiency are pivotal for fabric defect detection where real-time or resource-constrained deployment is common. Benchmarking studies on manufacturing and texture datasets indicate that no single model universally dominates; instead, model choice depends on application-specific accuracy requirements and processing constraints.

2.3 Transformer-Based Semantic Segmentation

Transformer-Based Semantic Segmentation marks a significant shift from conventional CNN architectures by moving beyond their limited local receptive fields. Using self-attention mechanisms, transformers capture global context and model relationships among all image regions at once [10]. This enables a deeper understanding of complex scenes and improves the detection of fine textures and boundaries. By combining multi-scale features with adaptive

attention, transformer-based models achieve more precise and consistent segmentation results [10]. This section reviews key transformer architectures, their core design ideas, and how they outperform CNN-based methods in accuracy and flexibility.

2.3.1 Evolution from CNNs to Vision Transformers

The advent of Vision Transformers (ViTs) [10] marked a paradigm shift in image recognition, motivated by the limitations of convolutional neural networks (CNNs) in capturing global context due to their localized receptive fields. ViTs, introduced by restructure images as sequences of fixed-size patches and employ self-attention mechanisms to model relationships across the entire image. This attention-centric approach enables ViTs to effectively capture long-range dependencies and complex spatial interactions, resulting in superior performance on multiple vision benchmarks with comparatively fewer training resources than CNNs [10].

2.3.2 Unified Perceptual Parsing (UPerNet)

UPerNet, or Unified Perceptual Parsing Network [11], advances beyond traditional segmentation models like U-Net [6] by addressing the multifaceted nature of visual scene understanding. Inspired by human perception, which simultaneously recognizes scenes, objects, parts, textures, and materials, UPerNet is designed to parse multiple visual concepts concurrently from a single image. Its multi-task architecture leverages heterogeneous image annotations across tasks to learn rich hierarchical representations, enabling effective segmentation of diverse visual elements with compositional relationships. Unlike U-Net, which primarily focuses on pixel-level object segmentation through an encoder-decoder structure and skip connections, UPerNet integrates multi-level features from CNN or Transformer backbones with pyramid pooling to capture global context at multiple scales. This facilitates parsing fine-grained textures and materials along with objects, making it highly suited for complex industrial contexts like fabric defect detection where identifying varied surface patterns and

material properties is essential. By unifying disparate visual parsing tasks into a cohesive framework, UPerNet embodies a more holistic scene understanding approach, positioning it as a powerful tool for fabric defect analysis that demands both detailed segmentation and contextual awareness [11].

2.3.3 SegFormer

SegFormer [12], introduces a simple yet efficient semantic segmentation framework that combines a hierarchically structured Transformer encoder with a lightweight multilayer perceptron decoder. Eschewing explicit positional encoding, SegFormer leverages multiscale features directly from the encoder, and its decoder aggregates information from various layers, effectively blending local and global attention. This design leads to improved segmentation accuracy and model efficiency, surpassing previous state-of-the-art models on challenging datasets. The model's robustness to scale variations and its computational efficiency affirms its suitability for complex tasks including fabric inspection [12].

2.3.4 Observations

Comparative studies consistently demonstrate that transformer-based models outperform CNNs in global context understanding, crucial for segmenting irregular textures and dispersed defects on fabrics. While CNNs excel at local feature extraction, transformers' ability to model global relationships results in higher accuracy and better generalization across diverse fabric types. These attributes underscore the growing adoption of transformer architectures in industrial defect detection applications, marking a significant evolution in semantic segmentation methodologies.

2.4 Data Scarcity and Unsupervised Learning Paradigm

2.4.1 Challenges in Labeling Industrial Data

A fundamental barrier to employing supervised deep learning for fabric defect detection is the difficulty of obtaining comprehensive and accurate annotations. This challenge is an acute, domain-specific manifestation of the general industrial data labeling problem [24]. Industrial defect labeling is labor-intensive and costly due to the need for expert knowledge, as defect types vary widely and often require careful visual inspection to identify subtle anomalies. The inherent subjectivity in defining defect boundaries and severity can lead to inconsistent labeling and reduced model generalization. Moreover, large-scale industrial production environments generate vast amounts of visual data, making manual annotation at scale impractical. This data imbalance problem, where defective samples are rare compared to non-defective ones, further complicates the annotation process and model training. The resulting high annotation costs and inherent data limitations drive the need to investigate alternative approaches, such as unsupervised learning, that minimize dependence on extensive pixel-level labeling [24].

2.4.2 Unsupervised Anomaly Detection

To overcome the limitations of costly annotations, unsupervised anomaly detection methods have garnered significant attention. These approaches model the distribution of normal, defect-free data and identify defects as deviations or anomalies without requiring explicit defect labels. Recent innovations such as Efficient AD [19] leverage lightweight feature extractors coupled with a student-teacher training regime, enabling millisecond-level detection latencies while maintaining low error rates in real-time industrial applications. This method addresses both structural and logical anomalies by coupling local feature prediction failures with global image analysis, demonstrating exceptional efficiency and effectiveness [13].

2.4.3 Other Notable Methods

Additional unsupervised frameworks include reconstruction-based models like memory-augmented autoencoders (MemAE) [14], which improve upon vanilla autoencoders by incorporating memory modules to enhance anomaly reconstruction differences. Embedding-based methods, exemplified by PaDiM [15] and PatchCore [17] use pretrained CNN features and probabilistic modeling or optimized feature extraction for accurate anomaly localization and detection. Teacher-student models, a class to which EfficientAD [19] belongs, employ knowledge distillation concepts to detect discrepancies in feature representation between networks trained on normal data. Conditional normalizing flow models such as CFLOW-AD [18] merge discriminative and generative components to achieve memory-efficient and real-time anomaly uncovering.

2.4.4 Application to Fabric Defect Detection

These unsupervised methods have been adapted successfully to industrial texture datasets like MVTec [16] and ZJU-Leaper [4], achieving high anomaly detection accuracy without reliance on defect annotations. Embedding-based and reconstruction-based unsupervised techniques are particularly adept at localizing subtle defects across varying fabric textures. Studies evaluating these models on fabric inspection benchmarks reveal that while supervised CNNs excel with abundant annotations, unsupervised approaches provide robust alternatives when annotations are sparse or unavailable, making them highly relevant for scalable industrial quality control.

2.5 Computational Efficiency and Model Complexity

2.5.1 Motivation for Efficiency Evaluation

The transition of semantic segmentation models from theoretical research to industrial deployment necessitates a stringent evaluation of their computational footprint, moving beyond mere accuracy maximization. In the domain of real-time industrial inspection, a model's

operational efficiency is paramount, often superseding minor gains in defect detection performance. The three principal metrics governing this efficiency are Floating-Point Operations (FLOPs), the total number of parameters, and inference latency.

FLOPs serve as a hardware-agnostic proxy for the computational complexity inherent in a single forward pass. High FLOP counts translate directly to increased processing time and energy consumption, posing a challenge for the rapid throughput required on an assembly line. Parameters, conversely, quantify the memory footprint of a model. Large parameter counts demand substantial storage capacity and increased memory bandwidth during inference, limiting deployment to specialized, resource-rich devices. Most critically, Latency represents the real-world inference speed, defined as the time elapsed between input acquisition and output generation. This metric is the decisive factor for real-time capability; if a system's latency exceeds the cycle time of the production line, it becomes a severe bottleneck [5]. It has become essential to "pay attention to these footprint metrics of a model as well, not just its quality" due to the significant increase in the resource demands of modern deep learning architectures.

2.5.2 Implications for Industrial Deployment

The constraints of the factory floor environment fundamentally dictate the acceptable bounds for model efficiency. Deployment is typically not in a centralized data center but on edge computing devices proximal to the production line, where resources are inherently limited, making model complexity a critical design consideration.

The requirement for real-time inference speed is non-negotiable in automated visual inspection. Inspection speeds must frequently operate at dozens of frames per second (FPS) to maintain the momentum of the manufacturing process. For example, in a study on PCB inspection, Zhao et al. [2] demonstrated an optimized lightweight model that achieved a viable operational speed

of 65 FPS while utilizing only 0.9 million parameters on the PKU-Market-PCB dataset. This performance profile highlights the essential trade-off: minimizing parameters and optimizing computational flow is the practical path to achieving high-speed, real-time performance in such systems [25].

Furthermore, constraints related to GPU availability and power consumption restrict the use of high-end accelerators. Industrial settings often mandate the use of lower-power, cost-effective devices. Models must therefore be structured to utilize memory efficiently and avoid computationally dense operations that necessitate expensive hardware [26]. The drive toward lower FLOPs and parameter counts is not merely an academic exercise; it is a necessity for reducing the Total Cost of Ownership (TCO) and ensuring the system's longevity, minimal heat generation, and scalability across numerous inspection stations. Consequently, comprehensive reporting of computational metrics alongside detection performance allows industrial end-users to select a model architecture that optimally balances accuracy, cost, and throughput given their specific factory floor constraints.

2.6 Experimental Design Considerations and Architectural Choices

Designing a reliable defect detection system requires not only selecting an appropriate network architecture but also making informed choices about backbone networks, loss functions, optimization strategies, and training configurations. These design elements jointly determine model generalization, convergence and computational feasibility during deployment.

2.6.1 Backbones

Backbone selection plays a critical role in determining a model's feature extraction capability, generalization power, and computational cost.

Convolutional Neural Networks (CNNs) remain the foundation of most supervised

segmentation models due to their efficiency in hierarchical spatial feature learning. Earlier architectures such as VGG networks [27], characterized by deep stacks of small 3×3 convolutional filters, demonstrating that increased depth substantially improves representational power without dramatic parameter inflation [27]. However suffered from vanishing gradients [28] and redundant parameters, making them computationally expensive for high-resolution defect imagery.

Subsequent models like ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-110 [29] addressed these issues through residual learning, allowing much deeper networks without degradation. Among these, ResNet-50 has emerged as an optimal trade-off between performance and efficiency, offering rich multi-scale features with manageable parameter size and training time.

Empirical comparisons show that while ResNet-101 and ResNet-110 [30] achieve slightly higher accuracy on large-scale datasets, their higher computational demands and GPU memory usage make them less suitable for real-time or resource-constrained industrial settings. Conversely, ResNet-18/34 are faster but yield weaker feature representations for complex textures and subtle defects, common in woven and knitted fabric surfaces.

Recently, Transformer-based architectures like the Swin Transformer [31] and Mix Transformer (MiT) [32] have emerged as powerful alternatives. These models employ hierarchical, multi-scale attention mechanisms such as shifted window attention and multi-resolution patch embedding. This design not only captures long-range dependency patterns vital for recognizing fine-grained defects but also maintains computational efficiency compatible with GPU memory constraints. Benchmark results, such as from the ZJU-Leaper [4] dataset, attest to the competitiveness of hybrid backbones that integrate convolutional and attention mechanisms.

2.6.2 Loss Function

Training for semantic segmentation often tackles class imbalance through composite loss functions. A widely adopted choice is the combination of Binary Cross-Entropy (BCE) [33] with Dice Loss, which together enforce pixel-level classification accuracy and enhance region-level overlap precision. BCE penalizes misclassified pixels, while Dice Loss [34] emphasizes boundary precision and overlap, resulting in more robust segmentation particularly when defect pixels constitute a minor class fraction. This composite loss [34] has demonstrated efficacy and interpretability in industrial defect detection and medical imaging context.

2.6.3 Optimization Strategies

The Adam optimizer [35] remains the default for efficiently training segmentation models. An initial learning rate of 1×10^{-4} has shown consistent effectiveness across diverse datasets, supported by adaptive learning rate schedules such as ReduceLROnPlateau that lower learning rates upon validation stagnation to prevent overfitting [35]. Early stopping further safeguards against unnecessary training, yielding better generalization and optimized hardware utilization in line with best practices.

2.7 Summary of Literature and Research Gap

Recent advances in fabric defect detection have been driven primarily by deep learning-based methods, particularly supervised semantic segmentation networks such as CNNs and Transformer architectures. These models have demonstrated impressive accuracy in localizing fabric anomalies under controlled conditions and benchmark datasets. In parallel, unsupervised anomaly detection techniques have emerged as a promising alternative that eliminates the need for defect-labeled data, instead learning normal patterns from clean samples. Together, these approaches mark substantial progress toward automated textile inspection.

However, a critical limitation in the existing literature lies in the fragmented evaluation of these two paradigms. Most prior works have examined either supervised or unsupervised models independently, often using different datasets, evaluation metrics, and experimental conditions. This lack of methodological uniformity prevents direct performance comparison and obscures the trade-offs between annotation requirements, model interpretability, and deployment feasibility. Furthermore, while several studies have introduced novel architectures or loss functions, comparative investigations across architectures under identical industrial settings remain scarce.

The current body of research also provides limited insight into the relationship between model accuracy and computational efficiency—an essential consideration for real-time inspection environments. Model scalability, inference latency, and computational load (in terms of FLOPs and parameters) are often reported inconsistently or overlooked entirely. As a result, there remains an incomplete understanding of which model families best balance precision and speed for practical manufacturing integration.

This gap is significant because fabric inspection systems must not only be accurate but also efficient, interpretable, and deployable on resource-constrained factory hardware. Without standardized comparative studies, it is difficult for researchers and industry practitioners to determine which methods are most suitable for specific operational requirements.

To address these shortcomings, the present study conducts a systematic comparative analysis of supervised and unsupervised fabric defect detection models on a unified benchmark—the ZJU-Leaper dataset. By employing consistent evaluation metrics, standardized training protocols, and efficiency assessments (including FLOPs, inference speed, and latency), this thesis bridges the divide between academic development and industrial application. The outcomes are intended to provide actionable insights for selecting models that optimize both

performance and computational feasibility, thereby contributing to scalable, sustainable, and intelligent fabric inspection in modern textile manufacturing.

Chapter 3 Methodology

3.1 Introduction

This chapter presents the methodological workflow followed for fabric defect localization using the ZJU-Leaper dataset [4]. The process began with dataset collection and study to understand its structure, resolution, and annotation format, followed by dataset preparation according to the official benchmark protocol, where all 19 fabric textures were grouped into five representative texture categories. Nine models were employed for experimentation—PSPNet, FPN, U-Net, U-Net++, DeepLabV3, DeepLabV3+, UPerNet, SegFormer, and EfficientAD; among which EfficientAD represents the unsupervised approach, while the remaining eight are supervised semantic segmentation models. Accordingly, two distinct workflows were developed: one for supervised semantic segmentation and another for unsupervised anomaly detection. Prior to training, all images underwent preprocessing such as resizing, normalization, and format standardization to ensure uniformity across architectures. Each model was then trained using the prepared dataset, and the resulting trained models were subsequently used for inference on the test images to generate predictions. The predicted outputs were compared with the ground truth masks to evaluate performance using quantitative and qualitative metrics. The complete methodological structure, including supervised and unsupervised experimental pipelines, is illustrated in Figure 3-1 and Figure 3-2, providing a concise overview of the workflow adopted in this study.

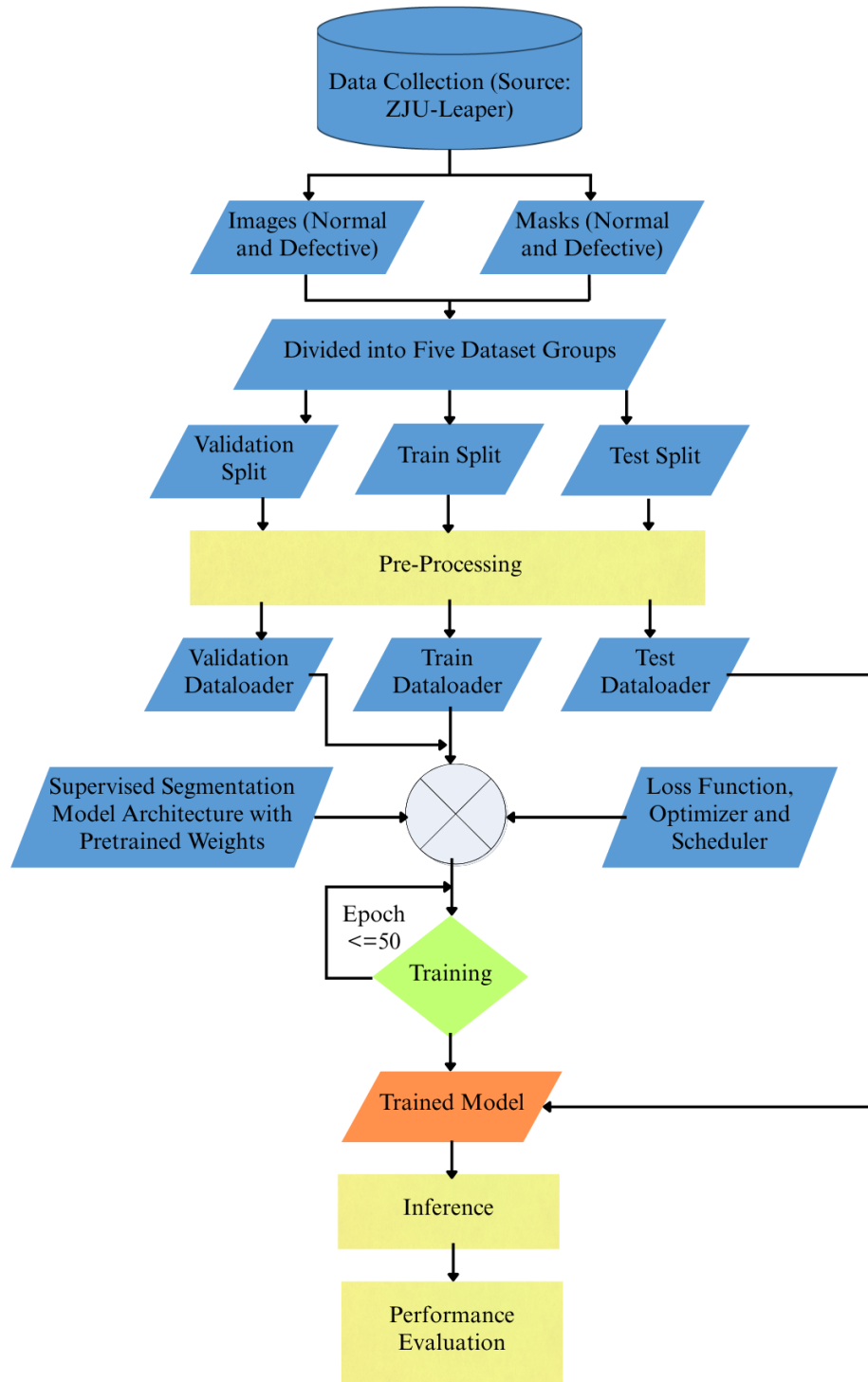


Figure 3-1: Overview of the Supervised Fabric Defect Detection Pipeline. The flowchart illustrates the sequential process from data collection and preprocessing to model training, inference, and performance evaluation using the ZJU-Leaper dataset.

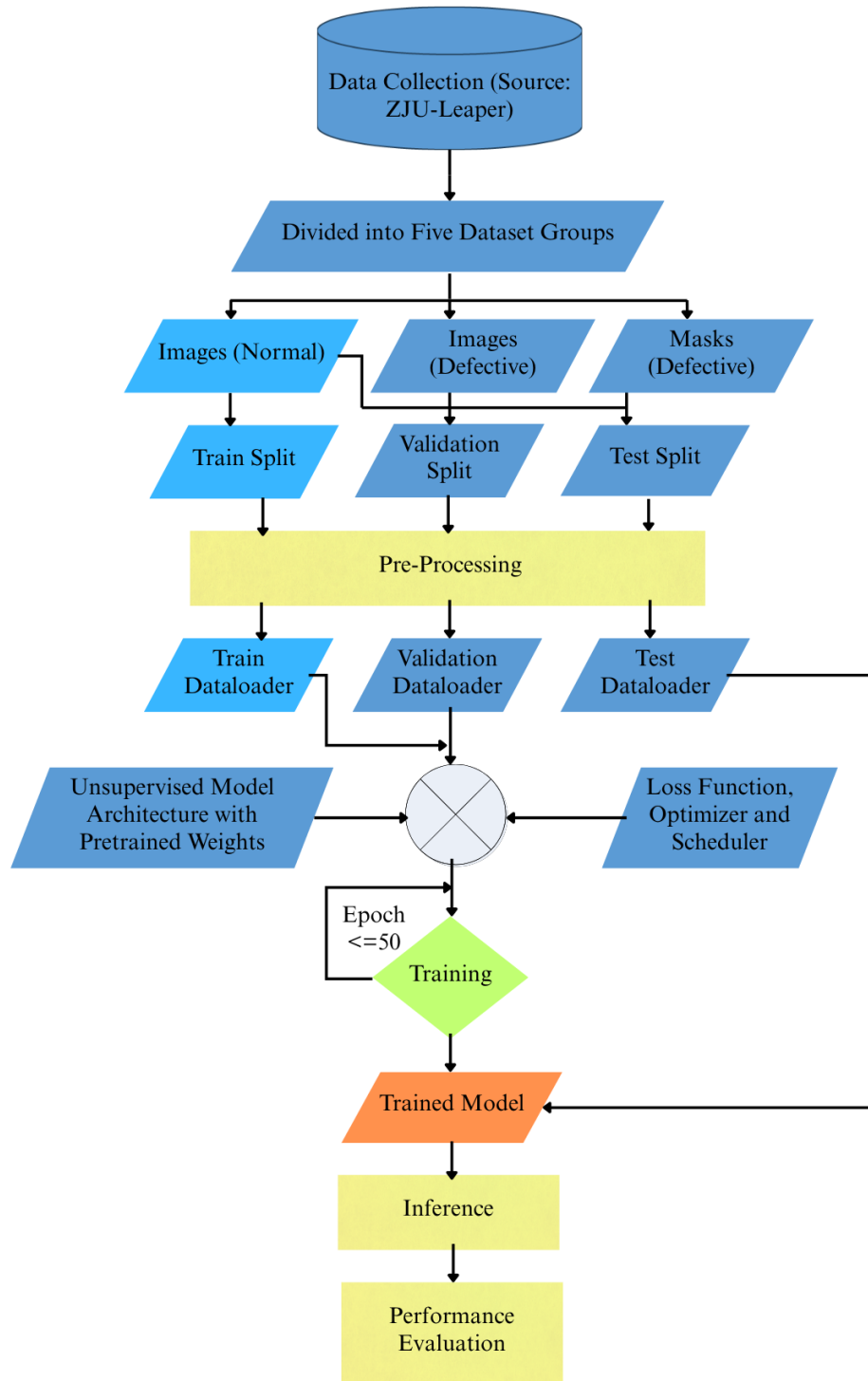


Figure 3-2: Overview of the Unsupervised Fabric Defect Detection Pipeline. The process, based on the ZJU-Leaper dataset, trains exclusively on normal images to learn texture patterns and detect anomalies during inference.

3.2 Data Collection

3.1.1 Dataset Source

The experiments in this study utilize the ZJU-Leaper Fabric Defect Dataset [4], developed by Zhejiang University. This large-scale benchmark contains approximately 94,833 high-resolution color images covering 19 distinct fabric patterns. Each pattern consists of around 5,000 samples on average, encompassing both defective and non-defective fabric surfaces.

Images were captured in a controlled laboratory environment with identical camera configurations and consistent illumination to ensure uniform image quality. The dataset is characterized by its detailed annotations and rich texture diversity, including multiple defect types such as broken ends, oil stains, mispicks, and holes.

To minimize data redundancy, normal (non-defective) samples were randomly selected in a 3:1 ratio relative to the number of defect images for each pattern. Annotation was performed manually by six trained individuals with normal vision. Each annotator independently identified the presence of defects and delineated their boundaries. The final segmentation masks were generated by averaging the outputs of three annotators to improve labeling consistency and reduce subjective variation.

The dataset supports multiple inspection tasks by providing two levels of annotation granularity:

1. **Image-level labels** – binary classification (defective or non-defective)
2. **Pixel-wise masks** – for semantic segmentation and precise defect boundary delineation.

Each image is further associated with metadata specifying its pattern name, pattern number, group name, and group number, allowing hierarchical analysis across different fabric categories and texture complexities.

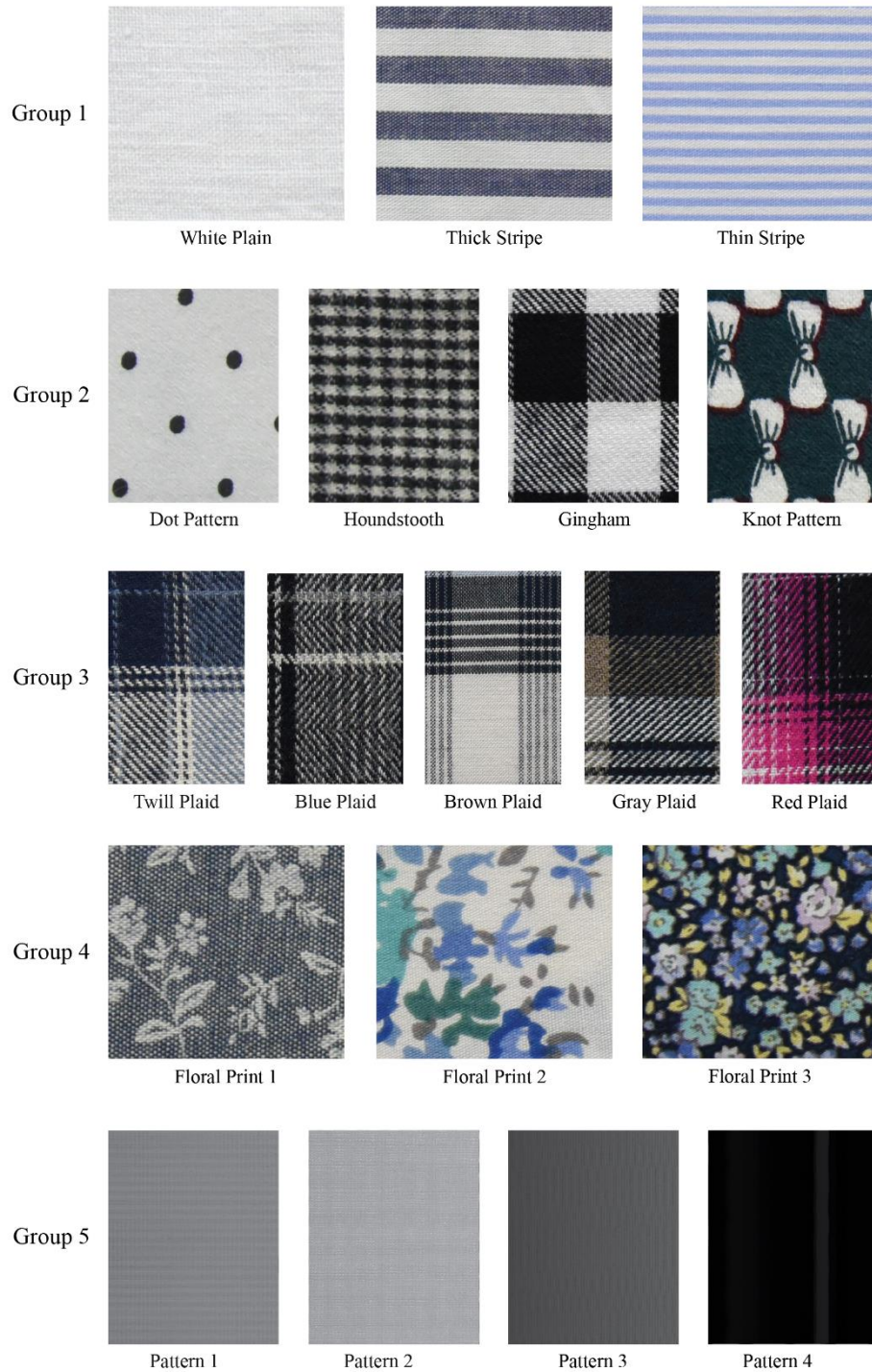


Figure 3-3: *Representative Normal Fabric Samples from the Five Texture-Based Groups in the ZJU-Leaper Dataset [4].*

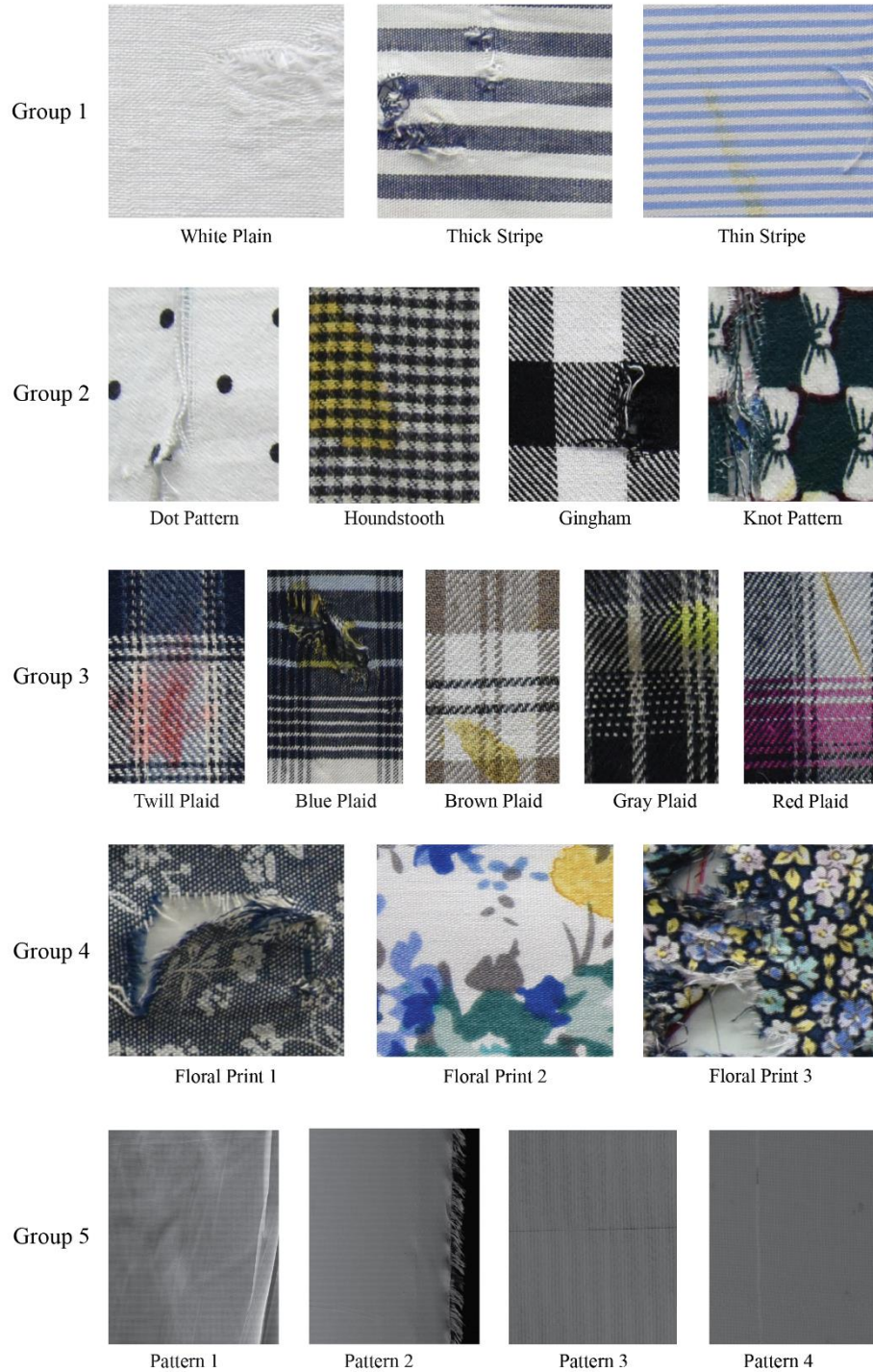


Figure 3-4: *Representative Defective Fabric Samples from the Five Texture-Based Groups in the ZJU-Leaper Dataset.*

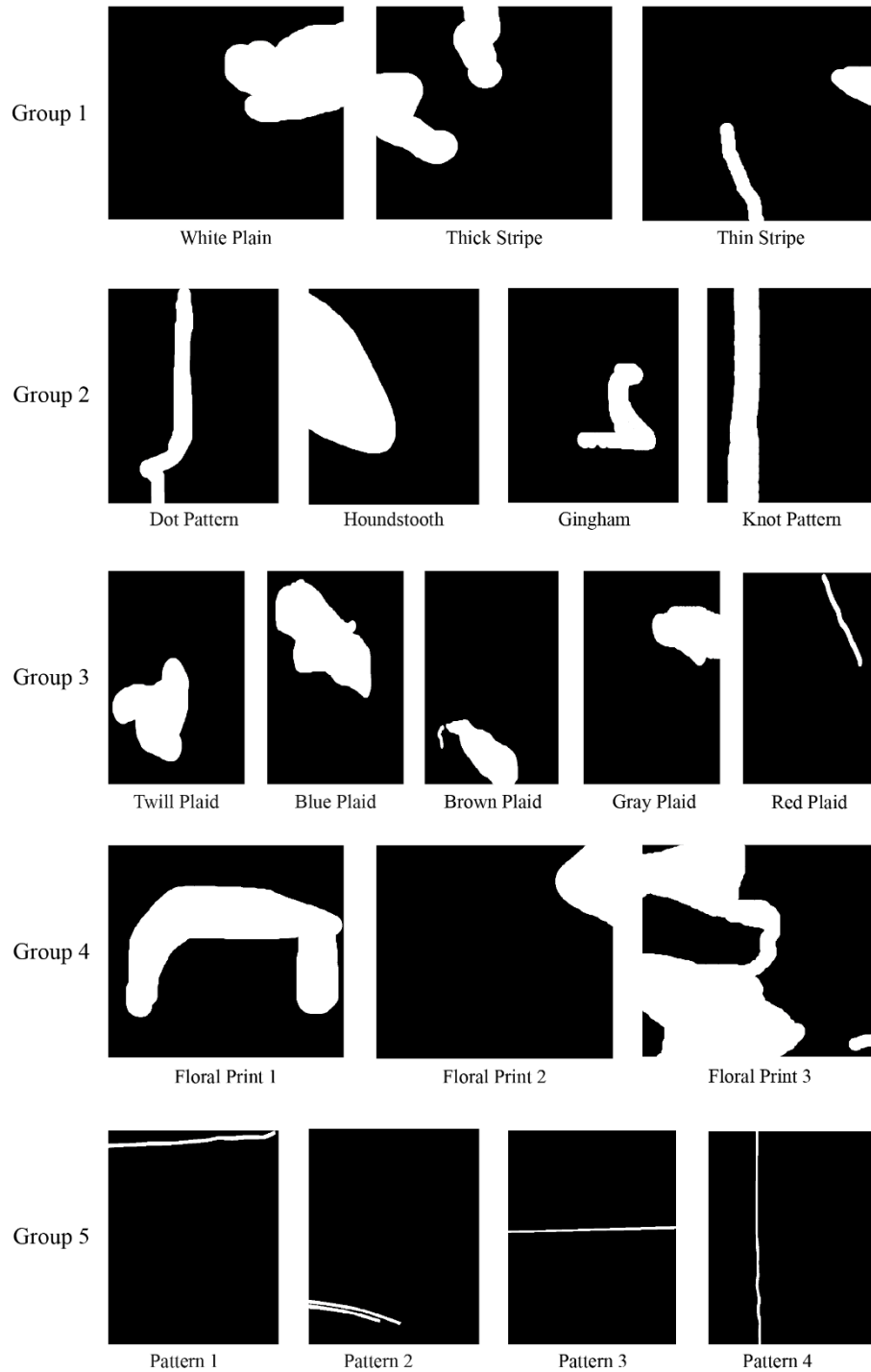


Figure 3-5: *Corresponding Ground Truth Pixel-Wise Masks for the Defective Samples Shown in Figure 3-4.*

3.1.2 Dataset Composition and Characteristics

The ZJU-Leaper dataset [4] comprises a total of 23,706 defective and 71,127 normal fabric images. Compared to other public textile defect datasets, ZJU-Leaper demonstrates superior scale, pattern diversity, and annotation richness. Moreover, it exhibits minimal center bias, a phenomenon where defects tend to cluster near the image center, thereby providing a more uniform spatial defect distribution. This balanced spatial distribution improves the robustness and generalization potential of defect detection models trained on this dataset [4].

Table 3-1: *Distribution of Normal and Defective Images Across 19 Fabric Patterns in the ZJU-Leaper Dataset*

Pattern ID	Pattern Names	Normal	Defective	Total
T1	White Plain	2673	891	3564
T2	Thick Stripe	3080	1026	4106
T3	Thin Stripe	3080	1026	4106
T4	Dot Pattern	3080	1026	4106
T5	Houndstooth	4086	1362	5448
T6	Gingham	4167	1389	5556
T7	Knot Pattern	4575	1525	6100
T8	Twill Plaid	4269	1423	5692
T9	Blue Plaid	4479	1493	5972
T10	Brown Plaid	4401	1467	5868
T11	Gray Plaid	3507	1169	4676
T12	Red Plaid	3978	1326	5304
T13	Floral Print1	3936	1312	5248
T14	Floral Print2	3696	1232	4928
T15	Floral Print3	3531	1177	4708
T16	Pattern1	3190	1063	4253
T17	Pattern2	5433	1811	7244
T18	Pattern3	1769	589	2358
T19	Pattern4	4197	1399	5596
Total		71127	23706	94833

3.3 Dataset Preparation

3.2.1 Texture Grouping

To facilitate texture-sensitive analysis and fair comparison across fabric types, the 19 original fabric patterns were organized into five major texture groups based on visual similarity and weave characteristics, following the grouping strategy described by the ZJU-Leaper authors [4]. Each group contains fabrics with related structural or visual attributes to support consistent benchmarking.

Table 3-2: *Texture Grouping Scheme (based on ZJU-Leaper organization)*

Groups	Texture Description	Included Patterns (Pattern ID)
Group 1	Simple repetitive patterns (e.g., plain, striped)	T1–T3
Group 2	Small repeated motifs (e.g., checks, dots)	T4–T7
Group 3	Complex patterns such as plaid fabrics	T8–T12
Group 4	Floral and intricate textured designs	T13–T15
Group 5	Greige (unbleached, undyed) factory fabrics	T16–T19

Group 5 samples generally possess lower image resolution and annotation quality compared to laboratory samples but are retained for comparative evaluation. Grouping textures in this way ensures balanced representation across fabrics of varying smoothness, pattern density, and visual complexity, aligning with best practices in industrial visual inspection research.

3.2.2 Dataset Split

A well-defined dataset splitting strategy is essential to ensure the reliability, reproducibility, and fairness of model evaluation. In this study, the dataset partitioning procedure was designed to remain consistent with the official ZJU-Leaper [4] benchmark configuration while supporting both supervised and unsupervised learning paradigms. The benchmark dataset

provides detailed metadata in XML format, which specifies the indices and identifiers of images originally used for training and testing. To maintain experimental consistency and enable a direct comparative analysis with previously reported benchmark results, the same image indices designated for testing in the benchmark metadata were adopted as the test set in this research.

For the supervised learning experiments, the dataset was divided into three distinct subsets: training, validation, and testing. Approximately 33% of the total images were reserved for testing, following the benchmark’s official split ratio. From the remaining images, 15% were further separated from the training set to form the validation set, which was used to monitor the model’s performance and prevent overfitting during training [4]. This hierarchical division ensured that the testing data remained completely unseen throughout the training and validation phases, preserving the objectivity of the evaluation process. The resulting distribution of samples across the five texture-based dataset groups is presented in Table 3-3.

Table 3-3: *Dataset split showing the number of images in the training, validation, and test sets across the five dataset groups for supervised models.*

Split	Group 1	Group 2	Group 3	Group 4	Group 5
Train	6654	12076	15558	8421	10999
Validation	1174	2131	2745	1485	1941
Test	3948	7003	9209	4978	6511

In contrast, the unsupervised learning configuration, applied exclusively to the EfficientAD model, followed a fundamentally different data utilization approach. Since unsupervised anomaly detection relies on learning the representation of normal textures and identifying deviations as defects, only normal samples were included in the training set. This ensures that the model captures the intrinsic appearance characteristics of defect-free fabrics. Defective images were entirely excluded from training and used solely for testing and validation, allowing

the model’s ability to identify unseen anomalies to be quantitatively and qualitatively assessed.

The sample distribution for this configuration is detailed in Table 3-4.

Table 3-4: *Dataset split showing the number of images in the training, validation, and test sets across the five dataset groups for unsupervised model.*

Split	Group 1	Group 2	Group 3	Group 4	Group 5
Train (Normal Only)	5890	10606	13756	7442	9727
Validation	1176	2120	2750	1488	1944
Test	4710	8484	11006	5954	7780

This structured splitting methodology not only adheres to the official ZJU-Leaper[4] benchmark but also provides a standardized framework for evaluating both supervised and unsupervised approaches under comparable conditions, ensuring fairness, reproducibility, and interpretability of results across different model architectures.

3.2.3 Ethical Considerations

The ZJU-Leaper [4] dataset used in this study is publicly available and does not contain any personal, identifiable, or proprietary information. All images represent fabric textures collected under controlled laboratory conditions for research purposes. All experimental procedures conducted in this work fully comply with institutional research ethics and data usage policies. No human participants, animal subjects, or sensitive materials were involved at any stage of this research. The dataset was used solely for academic and non-commercial purposes, following the licensing and citation requirements specified by the original authors.

3.4 Pre-Processing

To ensure comparability across all nine models, a unified preprocessing pipeline was employed. The dataset, consisting of 19 surface pattern types, was divided into five groups based on pattern complexity. Each group was treated as an independent dataset, but identical

preprocessing steps were applied across all experiments to eliminate input-induced biases and maintain consistency in learning settings.

3.2.1 Image Standardization

Standardization refers to converting all input images into a uniform numerical, spatial, and color format before model training. This process ensures that each image follows the same structural and statistical properties, which helps stabilize training and improves generalization.

In practice, standardization involves three main aspects:

1. ***Spatial Standardization:*** Every image and its corresponding mask were resized to a fixed spatial dimension of 256×256 pixels. This ensures consistent input size across architectures, from convolutional neural network-based architectures (e.g., U-Net, U-Net++, DeepLabV3, DeepLabV3+) to transformer-based models (e.g., SegFormer, UPerNet,). A consistent input size simplifies model design and guarantees that feature maps remain comparable in scale across experiments.
2. ***Color and Channel Standardization:*** Using the Pillow (PIL) library, each image was opened and explicitly converted to RGB format, producing three standardized color channels (red, green, blue). Corresponding defect masks were converted to grayscale (L mode), as they represent binary segmentation targets. PIL's internal color conversion ensures accurate mapping of pixel intensities and consistent handling of color depth across all samples, preventing discrepancies due to varying image encodings in the raw dataset.
3. ***Intensity Standardization:*** Following color conversion, images were transformed into PyTorch tensors using `torchvision.transforms.ToTensor()`. This operation automatically rescales pixel values from the original range $[0, 255]$ to $[0, 1]$ floating-point numbers. This rescaling acts as a normalization step, ensuring that all pixel intensities are expressed within a bounded numerical domain. It prevents gradient

explosion during training and aligns with the expected input scale of most pretrained architectures. While ImageNet-style channel normalization (subtracting mean and dividing by standard deviation) was not explicitly applied, the tensor conversion inherently provides a standardized value distribution sufficient for stable training across all supervised models. This process collectively guarantees that variations in image brightness, contrast, or acquisition conditions do not affect the model’s ability to detect or segment surface defects.

4. **Mask Binarization:** Each mask tensor was thresholded at 0.5, converting all values above the threshold to 1 (defective) and below to 0 (non-defective). This binarization ensures clean, discrete boundaries for segmentation tasks and prevents ambiguity in model predictions.

3.2.2 Data Loading and Augmentation

The standardized tensors were organized into PyTorch Dataset objects and loaded through Data Loader instances. Each loader enabled data shuffling for training to promote stochasticity. Although augmentation was kept minimal to preserve the natural structure of surface patterns, the consistent preprocessing ensured robust and unbiased learning across all models.

3.2.3 Unsupervised Model Pre-processing (EfficientAD)

For the unsupervised experiment, the EfficientAD architecture (implemented using the *Anomalib* framework) was trained on the same dataset groups. Anomalib provides an integrated preprocessing module (Preprocessor), which performs equivalent operations internally:

- Images are read using OpenCV/PIL via `read_image()`.
- They are resized to 256×256 pixels through a `torchvision.transforms.v2.Compose` pipeline.

- Normalization and tensor conversion occur automatically within the framework, standardizing image intensity distributions to $[0, 1]$.
- Data splits (train, validation, test) are handled through the Folder data module, ensuring consistent transformation across all phases.

Although implemented differently, the EfficientAD [19] preprocessing pipeline is conceptually identical to that of the supervised models, ensuring comparable input distributions across both paradigms.

Across both supervised and unsupervised experiments, all input images were standardized in size, scale, format, and color space. This consistent preprocessing ensured that performance differences among models were driven by architectural and learning variations rather than discrepancies in data preparation.

3.5 Model Architectures

The selection of model architectures plays a key role in balancing accuracy, generalization, and computational efficiency in fabric defect localization. Convolutional neural networks (CNNs) have been widely adopted for pixel-wise segmentation due to their strong spatial feature extraction and hierarchical representation capabilities. However, their local receptive field and dependence on large annotated datasets often limit performance on complex or fine-grained defect textures.

To overcome these challenges, transformer-based architectures have been introduced, leveraging self-attention to capture long-range dependencies and global context for improved defect boundary delineation. In parallel, unsupervised anomaly detection models have gained traction for industrial inspection tasks where labeled data are scarce. These models learn normal texture distributions and identify deviations as potential defects without pixel-level supervision.

This study evaluates representative models from these three paradigms to benchmark supervised and unsupervised approaches on the ZJU-Leaper dataset.

3.3.1 Supervised CNN-Based Architectures

Convolutional Neural Networks (CNNs) remain the backbone of industrial surface inspection due to their efficiency and strong feature localization ability. In this study, six representative CNN-based segmentation architectures were trained and compared. U-Net, PSPNet, FPN, DeepLabV3, U-Net++, and DeepLabV3+, all utilizing ResNet-50 encoders pretrained on ImageNet. The goal was to benchmark hierarchical and multi-scale convolutional designs for their ability to detect and localize textile surface defects across varying texture complexities.

U-Net

The U-Net [6] architecture is a fully convolutional neural network designed for pixel-level image segmentation, especially effective for biomedical images. It consists of two symmetrical paths: a contracting (encoder) path and an expansive (decoder) path. The contracting path captures context by applying repeated blocks of two unpadded 3×3 convolutions, each followed by a rectified linear unit (ReLU) activation, and a 2×2 max pooling operation with stride 2 for down sampling. At every down sampling step, the number of feature channels doubles to capture increasingly complex features with reduced spatial resolution.

The expansive path performs up sampling of feature maps through 2×2 transposed convolutions that halve the number of feature channels. These up-sample features are concatenated with correspondingly cropped high-resolution feature maps from the contracting path via skip connections, enabling the network to combine localization information with contextual understanding. Each concatenated feature map undergoes two successive 3×3 convolutions followed by ReLU activations. The final layer is a 1×1 convolution mapping the 64-channel feature vector at each pixel to the desired number of output classes. The network overall comprises 23 convolutional layers.

The architecture allows seamless segmentation of images via an overlap-tile strategy, handling border effects through mirroring to extrapolate missing input context. Training employs a pixel-wise softmax activation combined with cross-entropy loss weighted to emphasize difficult or small borders, such as those separating touching cells. The loss at pixel $[x \in \Omega \subset \mathbb{Z}^2]$ is:

Pixel-wise Loss Function:

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)) \quad (3.1)$$

Softmax Probability:

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{j=1}^K \exp(a_j(x))} \quad (3.2)$$

Weight Function:

$$w(x) = w_c(x) + w_0 \cdot \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \quad (3.3)$$

Weight Initialization:

$$\sigma = \sqrt{\frac{2}{N}} \quad (3.4)$$

PSPNet

The Pyramid Scene Parsing Network (PSPNet) [8] architecture addresses semantic segmentation challenges by integrating global contextual information at multiple scales. The model utilizes a dilated ResNet as a backbone for feature extraction, producing feature maps at 1/8th of the input resolutions with an enlarged receptive field due to dilated convolutions.

Central to PSPNet is the Pyramid Pooling Module (PPM), which partitions the feature map into regions at four scales: 1×1 , 2×2 , 3×3 , and 6×6 . Each region undergoes adaptive average pooling and is then processed by a 1×1 convolution to reduce channel dimensions. The pooled features are up sampled to the original feature map size and concatenated:

$$f_i = \text{Upsample} \left(\text{Conv}_{1 \times 1} (\text{Pool}_i (\text{Feature Map})) \right) \quad (3.5)$$

Where F is the input feature map and Pool_i denotes pooling at scale i [8].

The concatenated features provide a rich, multi-scale representation, enhancing pixel-wise classification by leveraging both local detail and broad context. These are further refined through convolutional layers to generate the final segmentation output. This hierarchical approach is particularly effective for complex scenarios like fabric defect detection, where texture and spatial context variations are significant.

FPN

Feature Pyramid Network (FPN) [7] architecture builds upon a convolutional backbone (e.g., ResNet) to extract hierarchical feature maps at different spatial resolutions denoted as $\{C_2, C_3, C_4, C_5\}$, corresponding to output of successive stages with strides $\{4, 8, 16, 32\}$ pixels relative to the input image. The bottom-up pathway computes these increasingly semantic but spatially coarser maps.

The top-down pathway reconstructs higher resolution feature maps by progressive nearest-neighbor upsampling. These upsampled maps are merged with lateral connections that

perform 1×1 convolutions on the corresponding bottom-up feature map to align the channel dimensions. The merging operation is element-wise addition:

$$P_i = \text{Upsample}(P_{i+1}) + \text{Conv}_{1 \times 1}(C_i) \quad (3.6)$$

for $i=4,3,2$, with $[P_5 = \text{Conv}_{1 \times 1}(C_5)]$ as the initial top-down feature map. Each merged feature map P_i is further refined by a 3×3 convolution to reduce aliasing effects.

This design results in a feature pyramid $\{P_2, P_3, P_4, P_5\}$ containing semantically strong features at multiple spatial scales, enhancing object detection and segmentation across diverse object sizes. The channel dimensionality in each P_i is fixed (commonly 256) for efficiency [7].

DeeplabV3

DeepLab [9] introduces atrous (dilated) convolution, which controls the resolution of feature responses without increasing computational cost.

$$y[i] = \sum_{k=1}^K x[i + r \cdot k] w[k] \quad (3.7)$$

Here, r determines the dilation rate (stride with which the input is sampled). Standard convolution is a special case with $r = 1$.

Atrous convolution enlarges the kernel's field of view without increasing parameters, calculated as:

$$k_e = k + (k - 1)(r - 1) \quad (3.8)$$

For a one-dimensional input signal $x[i]$ and filter $w[k]$ of length K , the output of atrous convolution $y[i]$ with rate r is defined as:

To capture multi-scale information, DeepLab uses Atrous Spatial Pyramid Pooling (ASPP).

ASPP applies multiple parallel atrous convolutions with different dilation rates to the feature map, then fuses the results. This allows the network to segment objects of varying sizes effectively.

To improve localization accuracy, DeepLab integrates fully connected CRFs.

The energy function over label assignment x is:

$$E(x) = \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i, x_j) \quad (3.9)$$

The unary potentials are:

$$\theta_i(x_i) = -\log P(x_i) \quad (3.10)$$

This integration allows precise boundary recovery beyond what DCNN outputs alone can achieve [9].

UNet++

UNet++ [22] starts with an encoder sub-network followed by a decoder sub-network. What distinguishes UNet++ from U-Net are the re-designed skip pathways that connect the two sub-networks and the use of deep supervision.

The main idea behind UNet++ is to bridge the semantic gap between the feature maps of the encoder and decoder prior to fusion. For example, the semantic gap between $X_{\{0,0\}}$ and $X_{\{1,3\}}$ is bridged using a dense convolution block with three convolution layers.

The operation of a dense convolution block along the skip pathway can be expressed as:

$$X_{i,j} = H([X_{i,0}, X_{i,1}, \dots, X_{i,j-1}]) \quad (3.11)$$

Here, H represents a composite function of convolution, and denotes concatenation of feature maps [22].

DeepLabV3+

DeepLabv3+ [23] architecture builds upon the original DeepLabv3 by integrating an encoder-decoder structure and employing atrous separable convolution for efficient and high-performance semantic segmentation. The encoder utilizes atrous convolution with rate r , which expands the receptive field without increasing the number of parameters, formulated as:

$$y[i] = \sum_{k=1}^K x[i + r \cdot k] w[k] \quad (3.12)$$

This allows the model to capture long-range contextual information. Depthwise separable convolution further decomposes a standard convolution into a depthwise convolution followed by a pointwise 1×1 convolution:

Depthwise Convolution:

$$y_d[i] = \sum_{k=1}^K x_d[i + r \cdot k] w_d[k] \quad (3.13)$$

Pointwise Convolution:

$$y[i] = \sum_{d=1}^D y_d[i] \cdot w_p[d] \quad (3.14)$$

This approach significantly reduces computational complexity and the number of parameters, leading to faster inference. The decoder module up-samples the encoder output and fuses it with low-level features, refining spatial details [23]. By combining atrous convolution with the depth wise separable structure, DeepLabV3+ [23] achieves accuracy with optimized efficiency, making it suitable for real-time applications such as autonomous driving and high-resolution medical imaging classification tasks.

Table 3-5: Summary of CNN-based Architectures

Model	Encoder Type	Core Mechanism	Key Strength
U-Net	ResNet-50	Symmetric encoder–decoder with skip connections	Fine-grained localization
PSPNet	ResNet-50	Pyramid pooling for multi-scale context	Robust contextual feature capture
FPN	ResNet-50	Top-down feature pyramid with lateral fusion	Multi-scale consistency
DeepLabV3	ResNet-50	Atrous Spatial Pyramid Pooling (ASPP)	Strong scale adaptability
U-Net++	ResNet-50	Nested dense skip pathways	Improved boundary refinement
DeepLabV3+	ResNet-50	ASPP + decoder refinement	Accurate edge delineation

3.3.2 Supervised Transformer-Based Architectures

With the advent of Vision Transformers (ViTs) [10], attention-based models have demonstrated superior capability in capturing long-range dependencies and global context ; aspects crucial for identifying subtle or spatially sparse fabric defects.

Two transformer-based segmentation models were implemented: UPerNet [11] and SegFormer [12].

UPerNet

The Unified Perceptual Parsing Network (UPerNet) [11] as a unified framework capable of parsing multiple visual attributes—scene, object, part, material, and texture using a single end-to-end trainable network. As illustrated in Figure X, UPerNet builds upon the Feature Pyramid Network (FPN) [7] and integrates a Pyramid Pooling Module (PPM) [21] to capture both global contextual information and multi-scale feature representations.

The FPN [7] component acts as a generic feature extractor that leverages a top-down architecture with lateral connections, enabling the fusion of high-level semantic features into lower-level spatially detailed representations. This design allows UPerNet to efficiently exploit hierarchical information from multiple semantic levels with minimal computational overhead.

To overcome the limitation of the empirical receptive field in deep convolutional neural networks (CNNs), which tends to be smaller than the theoretical receptive field, the Pyramid Pooling Module (PPM) from PSPNet [16] is applied to the last feature map of the backbone network before entering the top-down branch of the FPN. This enhances the receptive field and introduces effective global priors, thereby improving contextual awareness [11].

SegFormer

The encoder in SegFormer [12] employs Mix Transformer (MiT) [32] blocks to generate hierarchical, CNN-like multi-scale features represented as progressively lower-resolution feature maps. To preserve local continuity and spatial coherence, overlapping patch merging is applied during tokenization using large kernels and strides in the early layers and smaller ones in deeper layers.

A key feature of the encoder is its efficient self-attention mechanism, which reduces the computational cost of the standard attention operation. By introducing a sequence reduction ratio R , SegFormer lowers the complexity from quadratic to nearly linear without compromising global context modeling. This allows the encoder to effectively capture long-range dependencies while maintaining efficiency on high-resolution inputs.

SegFormer also replaces conventional positional embeddings with a Mix Feed-Forward Network (Mix-FFN). This component integrates a 3×3 convolution layer within the MLP to inject spatial awareness directly into feature representations, enhancing the encoder's ability to localize fine details and structural boundaries.

The decoder complements the encoder with an efficient, convolution-free design that fuses multi-scale features using only linear operations. Each encoder output is projected to a uniform embedding dimension and upsampled to a common spatial size before being concatenated. These combined features are then linearly transformed to produce the final segmentation mask.

This streamlined decoder design eliminates the need for complex convolutional or attention modules, significantly reducing computational overhead. Despite its simplicity, the All-MLP decoder achieves high segmentation accuracy by leveraging the rich hierarchical features produced by the encoder [12].

Table 3-6: Summary of Transformer-Based Architectures

Model	Encoder Type	Core Mechanism	Key Strength
UPerNet	Swin Transformer	Unified PPM + FPN feature aggregation	Robust fusion of global and local context
SegFormer	MiT-B1	Hierarchical transformer encoder + MLP decoder	Efficient, resolution-adaptive representation

These transformer-based [10] models were evaluated to determine whether attention-driven architectures can outperform CNNs in capturing complex periodic and non-periodic fabric structures, especially in cases where global context [10] plays a critical role.

3.3.3 Unsupervised Anomaly Detection Architecture

While supervised models require annotated masks for each defect instance, real-world industrial environments often lack large-scale labeled data. To address this limitation, EfficientAD [19], an unsupervised anomaly detection framework was integrated into this comparative study.

EfficientAD

EfficientAD [19] is a lightweight yet high-performing anomaly detection framework designed for industrial visual inspection. It integrates three main components: a Patch Description Network (PDN), a student–teacher (S–T) model, and an autoencoder-based logical anomaly detector.

The PDN serves as an efficient feature extractor, composed of four convolutional layers with intermediate pooling to reduce feature map dimensions. Each output neuron corresponds to a

33×33 patch of the input image, generating compact yet expressive descriptors in a single forward pass. To retain discriminative power while minimizing depth, the PDN is trained by distilling features from a pretrained WideResNet-101 model on ImageNet through a mean squared feature alignment loss.

For anomaly detection, EfficientAD employs a lightweight student–teacher model, where both networks share the PDN architecture. The student is trained to replicate the teacher’s features on normal data using a *hard feature loss*, which focuses only on the most mismatched feature regions to enhance sensitivity to defects. The anomaly score map at inference is computed as the channel-wise mean squared difference between the teacher and student feature maps:

$$M_{w,h} = \frac{1}{C} \sum_{c=1}^C (T(I)_{c,w,h} - S(I)_{c,w,h})^2 \quad (3.15)$$

An auxiliary **autoencoder** captures global structural dependencies to detect logical anomalies (e.g., missing or misplaced components). Its reconstruction discrepancies with the student output form a *global anomaly map*, which is averaged with the *local map* from the S–T model to yield the final anomaly score. This hybrid design balances accuracy and efficiency, achieving sub-millisecond inference per image while maintaining robust localization performance [19].

Table 3-7: Summary of Unsupervised Architecture

Model	Learning Paradigm	Core Mechanism	Key Strength
EfficientAD	Student–Teacher Self-Supervised	Feature discrepancy + reconstruction scoring	Label-free, fast, efficient for defect-scarce data

3.6 Evaluation Metrics

To ensure a comprehensive and fair evaluation of model performance, results were analyzed at three hierarchical levels sample, pixel, and region following the evaluation protocol introduced

in the ZJU-Leaper benchmark paper [4].

Additionally, efficiency metrics were included to capture computational trade-offs between model families.

This multi-level evaluation framework provides a balanced view of both detection quality and operational feasibility in fabric defect localization.

3.6.1 Sample-Level Performance Evaluation

The sample-level [4] evaluation provides an overall measure of a model's ability to correctly identify whether a fabric image is defective or normal. Unlike metrics that operate at the pixel or region scale, this level treats each image as a single decision unit, making it particularly effective for assessing a model's image-wise classification capability in practical inspection scenarios. In this evaluation, if any defect is correctly detected within an image, the model's prediction for that image is considered positive; conversely, if no defect is detected, the prediction is regarded as negative.

To quantify sample-level performance, the sample-level F1-score (F_1^{sam}) is employed. This metric captures the harmonic mean between precision and recall, thereby providing a balanced representation of both defect detection sensitivity and false alarm control across all samples. It is defined as follows:

$$F_1^{sam} = \frac{TP^{sam}}{TP^{sam} + \frac{1}{2}(FP^{sam} + FN^{sam})} \quad (3.16)$$

where TP^{sam} represents the number of correctly predicted defective samples that match the ground truth, FP^{sam} denotes the number of normal samples incorrectly classified as defective, and FN^{sam} corresponds to defective samples that are missed by the model.

This metric reflects the overall reliability of a model in determining the presence or absence of fabric defects at the image level [4]. By establishing a high-level accuracy indicator, it serves as the foundation for deeper analysis, which is further extended through pixel-level and region-level evaluations to explore localization precision and scale-dependent behavior.

3.6.2 Pixel-Level Performance Evaluation

Pixel-level performance [4] evaluation assesses segmentation accuracy at the most granular scale by comparing each pixel in the predicted mask with its corresponding ground truth label. In this evaluation, pixels belonging to defective regions are treated as *positive samples*, while non-defective pixels are considered *negative samples*. Let TP , FP , TN , and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively. Specifically, TP^{pix} and TN^{pix} represent the number of pixels correctly classified as defective or non-defective, whereas FP^{pix} and FN^{pix} indicate the number of misclassified pixels.

Among the several pixel-wise evaluation metrics used in segmentation tasks, Intersection over Union (IoU) and the Dice coefficient (also known as the pixel-level F1-score) are the most common. IoU measures the overlap between the predicted and ground truth defective regions, while the Dice coefficient evaluates the balance between precision and recall, providing a stable measure even under class imbalance conditions. Their mathematical formulations are as follows:

$$IoU = \frac{TP^{pix}}{TP^{pix} + FP^{pix} + FN^{pix}} \quad (3.17)$$

$$Dice = F_1^{pix} = \frac{TP^{pix}}{TP^{pix} + \frac{1}{2}(FP^{pix} + FN^{pix})} = \frac{2 \cdot IoU}{IoU + 1} \quad (3.18)$$

From Equation (3.18), it can be observed that IoU can be derived from the Dice coefficient, implying that both metrics are mathematically correlated. Therefore, using either one is sufficient for pixel-level evaluation. In this study, the Dice coefficient is preferred as the primary metric since it captures pixel-level overlap more sensitively and aligns better with perceptual segmentation quality, especially in cases of class imbalance where defect regions are small compared to the background [4].

3.6.3 Region-Level Performance Evaluation

Region-level [4] performance evaluation measures how effectively a model identifies and localizes defective regions, rather than evaluating at the pixel level. While pixel-level metrics assess the overlap of individual pixels, region-level metrics consider each *defect instance* (a connected region of defective pixels) as a unit of evaluation. This approach ensures that both the completeness of detected regions and the avoidance of false positives are reflected in the evaluation, aligning more closely with human visual inspection of defect localization.

Let TP^{reg} , FP^{reg} , and FN^{reg} denote the numbers of *true positive*, *false positive*, and *false negative* regions, respectively. A predicted region is considered a *true positive* if it matches a ground-truth defective region based on an overlap criterion such as Intersection over Union (IoU) or Area Ratio (ARR). Regions failing to meet this condition are counted as false positives or false negatives. To determine whether a predicted region matches a ground-truth region, the matching criteria follow the benchmark definition from the ZJU-Leaper paper:

$$IoU > 0.3 \text{ or } ARR > 0.51$$

The region-level precision, recall, and F1-score are formulated as follows:

$$Precision^{reg} = \frac{TP^{reg}}{TP^{reg} + FP^{reg}} \quad (3.19)$$

$$Recall^{reg} = \frac{TP^{reg}}{TP^{reg} + FN^{reg}} \quad (3.20)$$

$$F_1^{reg} = \frac{2 \times Precision^{reg} \times Recall^{reg}}{Precision^{reg} + Recall^{reg}} \quad (3.21)$$

Furthermore, region-level performance is disaggregated by defect size categories, *small* (*s*), *medium* (*m*), and *large* (*l*), to evaluate the sensitivity of each model across defect scales:

$$F_1^{reg\{s,m,l\}} = \frac{2 \times Precision^{reg\{s,m,l\}} \times Recall^{reg\{s,m,l\}}}{Precision^{reg\{s,m,l\}} + Recall^{reg\{s,m,l\}}} \quad (3.22)$$

Defect detection and localization performance can vary significantly with the physical size of the defective regions. Small defects are often subtle, low-contrast, and irregular in shape, whereas larger defects are more distinct and easier to identify. To obtain a comprehensive understanding of model performance across different defect scales, the region-level evaluation is further divided into small, medium, and large categories based on the pixel area of each annotated defect region [4].

Following the definition provided in the ZJU-Leaper [4] benchmark, the classification of defect sizes is based on the total number of defective pixels $|G|$ in each ground-truth region. Two thresholds, 50^2 and 100^2 pixels, are used to segment defects into three distinct categories:

Small: $|G| < 50^2$

Medium: $50^2 \leq |G| < 100^2$

Large: $|G| \geq 100^2$

Similarly, the predicted regions $|P|$ obtained from the model outputs are categorized using the same thresholds to ensure consistent evaluation. Thus, the number of proposed and ground-truth regions for each size group can be formally expressed as:

- Small proposed regions: $\{ P \mid |P| < 50^2 \}$
- Small ground-truth regions: $\{ G \mid |G| < 50^2 \}$
- Medium proposed regions: $\{ P \mid 50^2 \leq |P| < 100^2 \}$
- Medium ground-truth regions: $\{ G \mid 50^2 \leq |G| < 100^2 \}$
- Large proposed regions: $\{ P \mid |P| \geq 100^2 \}$
- Medium ground-truth regions: $\{ G \mid |G| \geq 100^2 \}$

Here, P and G represent the binary masks of the predicted and ground-truth regions, respectively, and $|*|$ indicates the cardinality, i.e., the number of defective pixels within a region.

Once the regions are grouped by size, the region-level F1-score (as defined in Equation 3.22) are calculated separately for each defect size group, resulting in F_1^{small} , F_1^{medium} , and F_1^{large} . This stratified evaluation provides a deeper insight into how well each model generalizes across varying spatial scales of defects. Typically, small regions are found to be the most challenging due to their limited spatial cues and lower signal-to-noise ratio, whereas medium and large regions yield higher and more stable detection accuracy [4].

3.6.4 Efficiency Metrics

While segmentation accuracy provides insight into a model's defect localization capability, computational efficiency determines its applicability in real-world manufacturing environments, where inspection systems must operate under strict latency and resource constraints. Therefore, in addition to accuracy-based evaluations, a set of efficiency metrics was employed to assess the computational cost [5], memory footprint, and inference

performance of each model.

Floating Point Operations (FLOPs)

FLOPs [5] denote the total number of floating-point computations required for a single forward pass of the network. This metric provides an architecture-independent measure of computational complexity, allowing for fair comparison across models implemented on different hardware or frameworks. Lower FLOPs indicate a more lightweight model capable of faster inference and reduced energy consumption, attributes critical for real-time fabric inspection systems that process hundreds of images per second on production lines. Since FLOPs correlate strongly with computational latency [5], they are widely adopted as a proxy for algorithmic efficiency in both academic and industrial deep learning benchmarks.

Model Parameters (M)

The total number of trainable parameters [36], expressed in millions (M), reflects a model's capacity and memory requirement. A high parameter count typically enables greater feature abstraction and accuracy but comes at the cost of increased memory usage, longer training times, and higher deployment overhead [37]. In industrial inspection contexts, where embedded GPUs or edge accelerators are often used, models with fewer parameters offer practical advantages such as lower inference latency and energy consumption [38]. Therefore, the parameter count was considered a critical indicator of deployment feasibility and scalability.

Inference Speed - Frames Per Second (FPS)

Frames Per Second (FPS) quantifies the number of images a model can process per second during inference, providing a direct measure of real-time performance. It integrates both theoretical efficiency (as reflected by FLOPs and parameters) and hardware-specific throughput, capturing the effect of memory bandwidth, parallelization, and GPU utilization [39]. Higher FPS values correspond to shorter per-image inference times, which are

essential for online defect inspection systems deployed in textile manufacturing environments, where conveyor-based imaging requires decisions within milliseconds.

To ensure fairness, FPS was measured on identical hardware (NVIDIA L4 GPU) using consistent batch sizes and runtime conditions across all models.

3.7 Training Configuration and Hyperparameters

All experiments were conducted using a consistent and reproducible computational setup in the Google Colab Pro+ environment. The system was configured with Python 3.12.11 running on Linux kernel 6.1.123+, utilizing PyTorch 2.8.0 with CUDA 12.6 support. The hardware environment comprised an NVIDIA L4 GPU (22.16 GB VRAM, CUDA capability 8.9) and an Intel Xeon CPU @ 2.20 GHz with 6 physical cores and 12 logical threads, supported by 52.96 GB of system RAM. This configuration ensured sufficient computational throughput and memory bandwidth for large-scale semantic segmentation training.

Each model was initialized with ImageNet-pretrained [40] encoder weights, and all input images and masks were uniformly resized to 256×256 pixels. The optimization objective combined Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss [32]) and Dice Loss [34], which jointly balance pixel-level accuracy and region-level segmentation precision. The Adam optimizer [35] was employed with an initial learning rate of 1×10^{-4} , while a ReduceLROnPlateau scheduler was used to adaptively lower the learning rate when validation performance stagnated [35]. The maximum number of epochs was set to 50, with scheduler patience of 5 epochs and early stopping patience of 10 epochs to mitigate overfitting and ensure convergence stability.

Batch size was varied across models according to their memory requirements and computational complexity, with each configuration optimized for the available GPU memory.

Checkpoints were saved based on the best validation pixel-level F1-score [4], and final evaluations were performed on the test split defined in the benchmark metadata.

Table 3-8: System and Training Configuration Summary

Parameter	Description	Configuration
Environment	Platform	Google Colab Pro+
OS	Operating system	Linux 6.6.97+
Python Version	Interpreter	3.12.11
Framework	Deep learning library	PyTorch 2.8.0
CUDA Version	GPU compute backend	12.6
GPU	Graphics Processing Unit	NVIDIA L4
GPU VRAM	GPU memory capacity	22.16 GB
CUDA Capability	GPU architecture	8.9
CPU	Processor	Intel Xeon @ 2.20 GHz (6C/12T)
System RAM	Host memory	52.96 GB
Input Dimension	Image and mask resolution	256 × 256

Table 3-9: Hyperparameter Settings

Hyperparameter	Description	Value
Pretrained Weights	Encoder initialization	ImageNet
Loss Function	Optimization objective	BCEWithLogitsLoss + Dice Loss
Optimizer	Optimization algorithm	Adam
Learning Rate	Initial learning rate	1×10^{-4}
Scheduler	Learning rate policy	ReduceLROnPlateau
Scheduler Patience	Epochs before LR decay	5
Maximum Epochs	Training iterations	50
Early Stopping Patience	Epochs before stopping	10
Batch Size	Images per iteration	Variable (model-dependent)

Note: For the EfficientAD unsupervised model, only normal (defect-free) images were used for training. Some hyperparameters were defined as built-in components within the Anomalib library and were therefore retained as originally configured.

Chapter 4 Results and Discussion

4.1 Sample Level Performance

The sample-level performance analysis evaluates each model’s capability to determine whether a fabric image contains any visible defect, irrespective of its location, size, or type. As discussed in the Methodology chapter, this metric represents the highest level of abstraction in the defect detection hierarchy, focusing purely on binary classification accuracy—defective versus non-defective samples. This stage is particularly important for industrial inspection systems, where rapid screening and rejection of faulty materials must occur before more detailed localization processes.

The quantitative results of this evaluation are summarized in Table 4-1, which presents the F1-scores achieved by each model across five dataset groups. These five groups were formed by clustering the nineteen fabric textures from the ZJU-Leaper dataset into categories of similar visual and structural characteristics, thereby ensuring a balanced assessment of the models under increasing levels of textural complexity. The visual comparison of the same data is depicted in Figure 4-1, where a grouped histogram illustrates the variation in performance among models and across texture groups.

Table 4-1: *Sample-Level F1 Scores by Model and Dataset Group*

Models	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.9797	0.969	0.9555	0.9582	0.7869
FPN	0.9718	0.9643	0.9580	0.9648	0.8449
U-Net	0.9790	0.9674	0.9588	0.9658	0.8465
DeepLabV3	0.9780	0.9649	0.9598	0.9659	0.846
U-Net++	0.9796	0.9704	0.9645	0.9684	0.8403
DeepLabV3+	0.9742	0.9681	0.9536	0.9618	0.8472
UPerNet	0.9790	0.9601	0.9704	0.9658	0.7334
SegFormer	0.9790	0.9602	0.9385	0.9669	0.7073
EfficientAD	0.9027	0.9309	0.8424	0.8670	0.7037

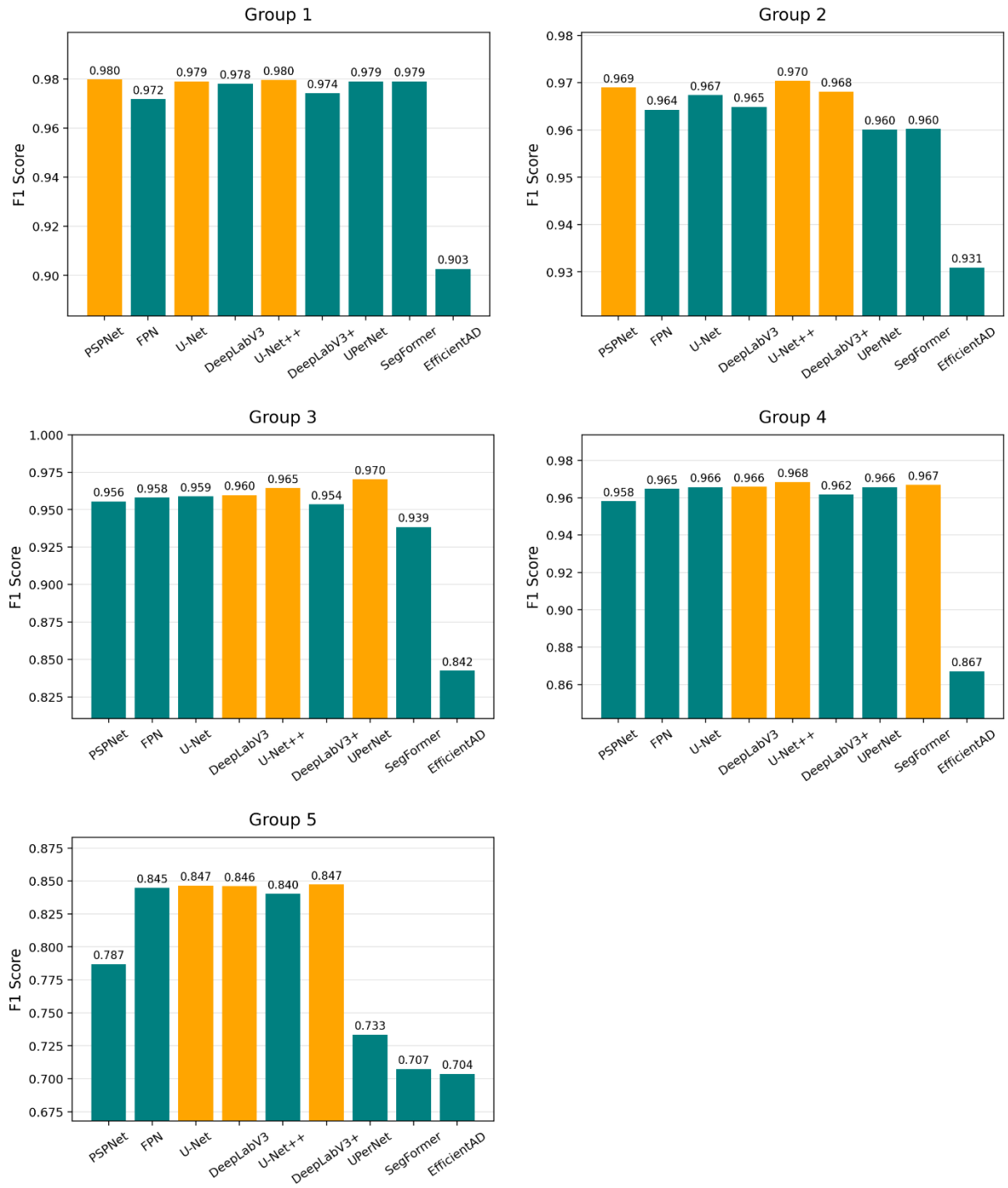


Figure 4-1: Comparison of Sample-Level F1 Scores of All Evaluated Models Across Group 1-5

4.1.1 Group-Wise Performance Analysis

A group-wise evaluation provides deeper insight into how each model responds to variations in fabric texture complexity. As the ZJU-Leaper dataset is divided into five groups of increasing visual irregularity, this analysis helps reveal the robustness and adaptability of the models to structural diversity and noise within textile surfaces.

Group 1 and Group 2 represent fabrics with simple or moderately repetitive structures, and all supervised models perform exceptionally well here, achieving F1-scores above 0.96. This suggests that convolutional architectures are highly effective at capturing low-level spatial features such as weave consistency and directional line patterns. The unsupervised EfficientAD also records relatively strong results in these groups (0.90–0.93), implying that its reconstruction-based learning is reliable when the normal pattern distribution is highly stable and defects are visually distinct.

Group 3 maintains a similar trend, though a slight reduction in performance (0.95–0.96) is observed as pattern variability increases. The group’s plaid and twill textures introduce multiscale periodicity and color variance, which can slightly reduce sensitivity to subtle defect cues. Nonetheless, all supervised models continue to demonstrate robust generalization, especially U-Net++, DeepLabV3+, and UPerNet, which retain high scores above 0.96.

In Group 4, which contains printed floral fabrics with irregular contours and tonal gradients, performance remains comparable to Group 3 but shows slightly higher variance between models. This indicates that the less repetitive and more stochastic pattern distribution challenges feature extraction layers, particularly in transformer-based architectures. However, the CNN-based methods still achieve stable results, with DeepLabV3+ and U-Net++ maintaining F1-scores around 0.97.

Group 5 displays the most significant decline in all models’ performance, with F1-scores dropping to approximately 0.79–0.85 for CNN-based models and as low as 0.70 for

transformer-based and unsupervised approaches. These fabrics, being unpatterned greige materials, exhibit low contrast and fine, uniform textures that provide minimal cues for defect localization. The pronounced drop across all methods highlights that global and local feature extractors alike struggle when both defect and background share similar intensity and frequency characteristics.

From a comparative standpoint, this group-wise analysis demonstrates that texture entropy—the degree of variation and complexity in visual structure—has a direct influence on sample-level classification accuracy. Models perform best on low-to-moderate entropy textures, while highly uniform or highly stochastic fabrics still present significant challenges. The consistent patterns observed here also validate the hierarchical evaluation approach, motivating a closer examination of fine-grained performance at the pixel and region levels in subsequent sections.

4.1.2 Model-Wise Performance Analysis

A model-wise comparison provides a comprehensive understanding of how architectural design, feature extraction strategy, and learning paradigm influence sample-level defect detection accuracy. The nine models considered in this study represent three categories of deep learning frameworks—convolutional encoder-decoders, pyramid-based feature aggregation networks, and transformer-based vision models—along with an unsupervised anomaly detection approach.

Among the supervised architectures, U-Net++ and DeepLabV3+ consistently achieve the highest F1-scores across almost all dataset groups. This superior performance can be attributed to their enhanced capability for multi-scale feature fusion. The dense skip connections in U-Net++ effectively bridge the semantic gap between encoder and decoder layers, ensuring that both fine texture details and contextual features are preserved. Similarly, DeepLabV3+ benefits from atrous spatial pyramid pooling (ASPP), which captures contextual information at multiple receptive fields, enabling the model to detect subtle texture deviations that signify fabric

defects. These two architectures demonstrate a strong balance between precision and recall, particularly in Groups 1–4, where patterns are moderately complex.

Models such as U-Net, DeepLabV3, PSPNet, and FPN also exhibit strong and stable performance. Their results indicate that encoder-decoder-based CNNs are inherently well-suited for binary classification tasks involving visual uniformity. U-Net’s symmetrical structure facilitates effective pixel-to-image-level reasoning, while PSPNet’s pyramid pooling module enriches the global context representation. FPN’s top-down feature fusion also contributes to consistency, particularly in moderately variable textures. Together, these networks maintain minimal performance variation between texture groups, highlighting their reliability in industrial defect screening.

In contrast, transformer-based architectures—UPerNet and SegFormer—achieve competitive accuracy on simpler groups but experience notable performance degradation in Groups 4 and 5. This decline can be explained by their heavy reliance on global attention mechanisms, which can over-smooth localized information when defects are small or low in contrast. Although transformers provide superior contextual understanding, they may inadvertently weaken sensitivity to subtle local anomalies in homogeneous regions. These observations reinforce that pure attention-based systems require texture-aware calibration or hybrid convolutional integration to reach CNN-level consistency in fine-grained defect identification.

Finally, the unsupervised EfficientAD model, despite its comparatively lower overall accuracy, demonstrates meaningful competence in Groups 1 and 2, where the normal texture distribution is well-defined. Its reconstruction-based approach enables effective identification of out-of-distribution patterns. However, in more visually complex groups, the model’s reliance on reconstruction residuals becomes less discriminative, leading to misclassification of defect-free textures as anomalies. This limitation underscores the current gap between unsupervised and supervised paradigms in industrial defect detection tasks.

In summary, the model-wise analysis reveals that architecture plays a pivotal role in determining classification robustness. CNN-based architectures remain dominant at the sample level due to their ability to capture both local and global texture features. Transformers, while promising, require structural refinement for texture-sensitive domains, and unsupervised methods—though efficient—still lag behind in environments with high intra-class variability. These comparative insights establish a foundation for examining spatially detailed behaviors in the upcoming pixel-level and region-level analyses.

4.2 Pixel-Level Performance

While the sample-level analysis provided a high-level understanding of each model’s capacity to classify fabrics as defective or non-defective, pixel-level evaluation delves deeper into spatial accuracy and the models’ ability to delineate defect boundaries with precision. This level of analysis is crucial for industrial inspection systems, where not only identifying the presence of defects but also accurately localizing them determines the effectiveness of automated quality control.

As detailed in the methodology chapter, the pixel-level F1-score (interpreted here as the Dice coefficient) quantifies the overlap between predicted defect regions and the ground-truth masks. A higher F1-score indicates both high precision in capturing defect boundaries and high recall in covering the entire defective area. Unlike sample-level classification, pixel-level segmentation directly evaluates how well the models learn spatial context, boundary sharpness, and region consistency.

The quantitative pixel-level results are presented in Table 4-2, which reports F1-scores for all models across the five texture groups. These findings are visually summarized in Figure 4-2, where grouped histograms illustrate how model performance varies with increasing texture irregularity. Additionally, Figure 4-3 to Figure 4-7 show the corresponding Precision–Recall

(PR) curves for each group, providing a more comprehensive view of the balance between defect detection precision and completeness.

In summary, the pixel-level results highlight the strengths and weaknesses of each architectural family under spatially detailed evaluation. The following subsections analyze these findings in greater depth through Precision–Recall curve interpretation, group-wise performance trends, and model-specific architectural insights, followed by qualitative visualization of segmentation outputs.

Table 4-2: *Pixel-Level F1 Scores by Model and Dataset Group*

Model	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.8488	0.8271	0.8303	0.8467	0.603
FPN	0.8497	0.8356	0.8431	0.8417	0.6652
U-Net	0.8751	0.8578	0.8622	0.8682	0.6708
DeepLabV3	0.8572	0.8472	0.8454	0.8534	0.6543
U-Net++	0.8749	0.8664	0.8702	0.8783	0.6743
DeepLabV3+	0.858	0.8474	0.8447	0.8525	0.6642
UPerNet	0.8734	0.8486	0.8545	0.8707	0.6915
SegFormer	0.8688	0.8334	0.8484	0.8542	0.6804
EfficientAD	0.5707	0.5502	0.4598	0.5437	0.0983

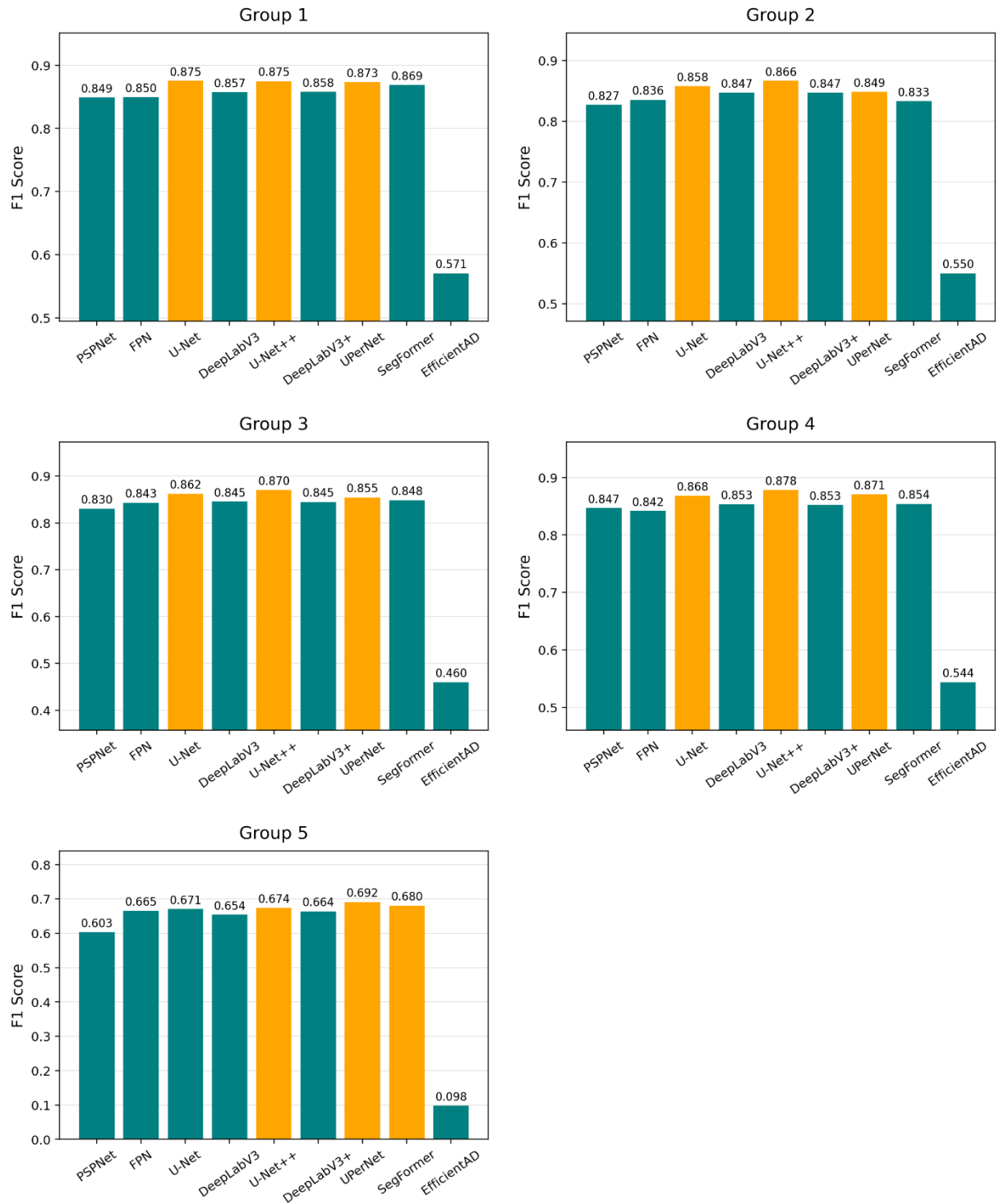


Figure 4-2: Comparison of Pixel-Level F1 Scores of All Evaluated Models Across Group 1-5

4.2.1 PR-Curve Analysis

Precision–Recall (PR) curves provide an in-depth perspective on each model’s ability to maintain a balance between defect detection accuracy and completeness across varying decision thresholds. Unlike the F1-score, which represents a single-point summary, PR curves reveal how precision and recall interact over the full prediction spectrum, making them essential for assessing segmentation reliability in defect localization tasks.

Across the dataset, the supervised models demonstrate highly stable and well-formed PR curves, with AUC (Area Under the Curve) values typically ranging between 0.87 and 0.92 in structured and moderately complex textures. These nearly rectangular curves indicate that models such as PSPNet, FPN, U-Net, U-Net++, DeepLabV3, and DeepLabV3+ maintain strong defect discrimination even under changing thresholds, consistently preserving precision at high recall levels. Among them, SegFormer and UPerNet achieve some of the highest AUCs, reaching up to 0.92, reflecting the advantage of transformer-based attention in maintaining global spatial coherence during segmentation. However, their slight precision decline at full recall suggests that global context modeling can occasionally blur small, localized defects when the background structure is highly repetitive.

In contrast, the EfficientAD model exhibits a distinctly different behavior, characterized by rapidly declining precision as recall increases and overall AUC values falling well below 0.6 across all textures. This pattern underscores the inherent limitation of unsupervised anomaly detection approaches that rely on reconstruction error rather than pixel-level classification. Without direct supervision, EfficientAD struggles to distinguish subtle structural differences, often misclassifying normal texture variations as defects or missing low-contrast anomalies entirely.

As fabric complexity increases—from regular striped and checkered patterns to irregular

printed and near-uniform greige textures—the overall AUC values gradually decline for all models. In highly structured textures, the CNN-based architectures perform exceptionally well, leveraging spatial hierarchies to retain boundary precision. For semi-random or printed fabrics, transformer-based models exhibit a modest edge due to their global dependency modeling, which helps contextualize irregular defect shapes. However, when texture contrast becomes minimal, as in the most uniform fabrics, both architectural paradigms experience a substantial drop in PR performance, with AUC values falling below 0.70, and the gap between defective and non-defective confidence scores narrowing sharply.

Overall, the PR-curve analysis reveals a clear performance hierarchy: supervised convolutional and hybrid transformer models sustain a strong balance between precision and recall under most fabric conditions, while unsupervised reconstruction-based methods fail to generalize when texture entropy and contrast are low. These observations affirm that threshold calibration and context-aware feature extraction are vital for reliable segmentation in industrial defect inspection, particularly when applied to diverse textile surfaces.

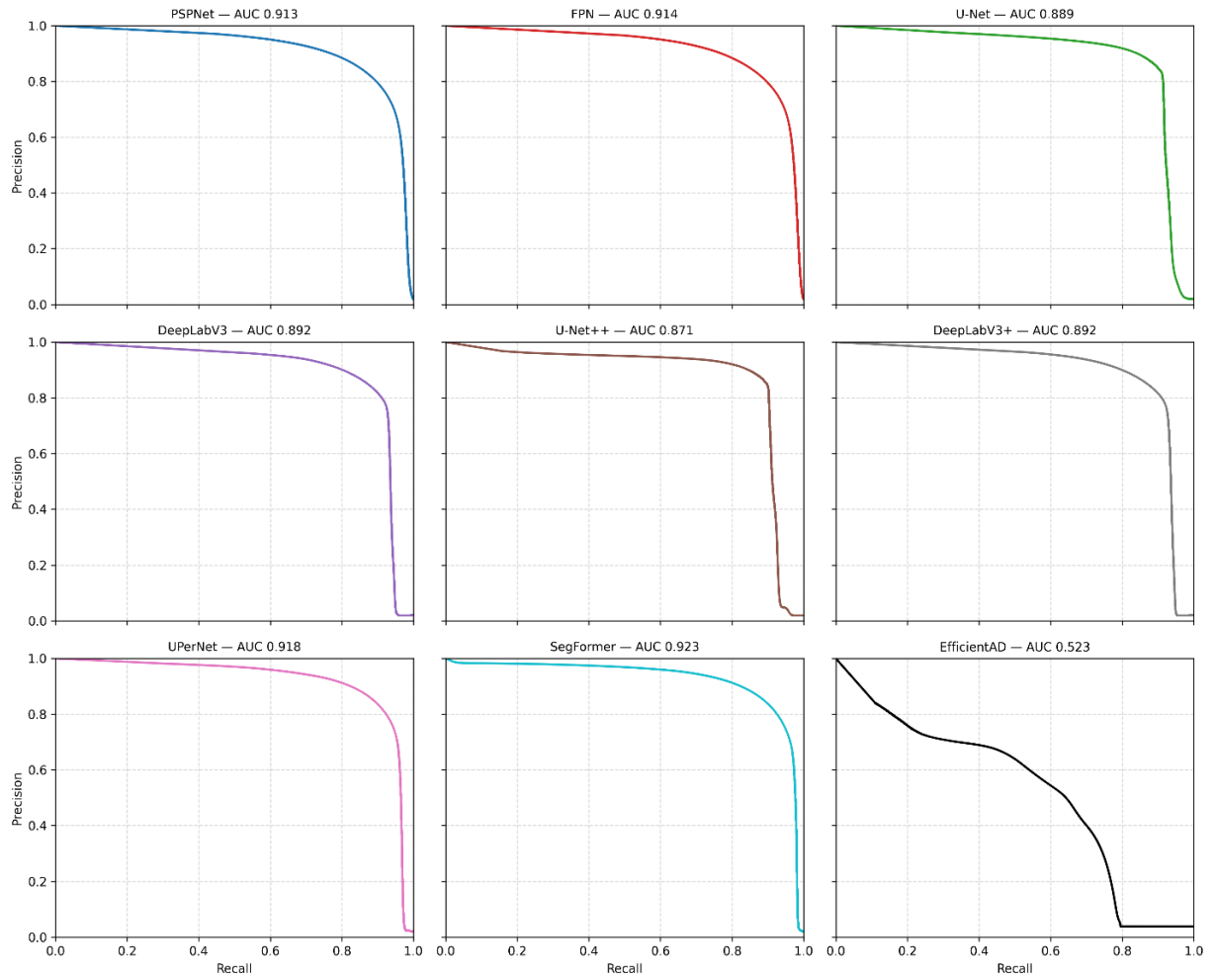


Figure 4-3: *Precision-Recall Curves of Evaluated Models for Group 1*

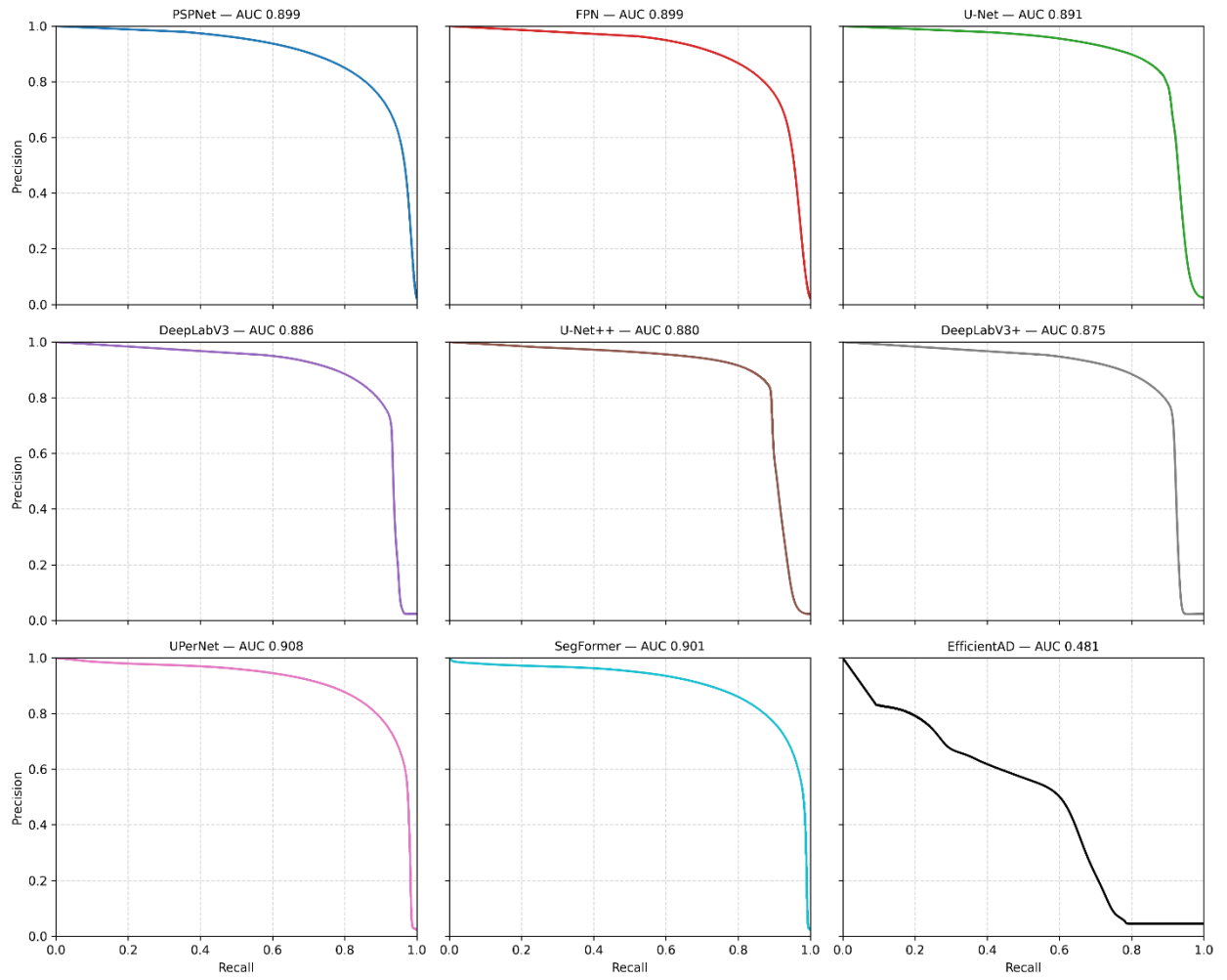


Figure 4-4: *Precision-Recall Curves of Evaluated Models for Group 2*

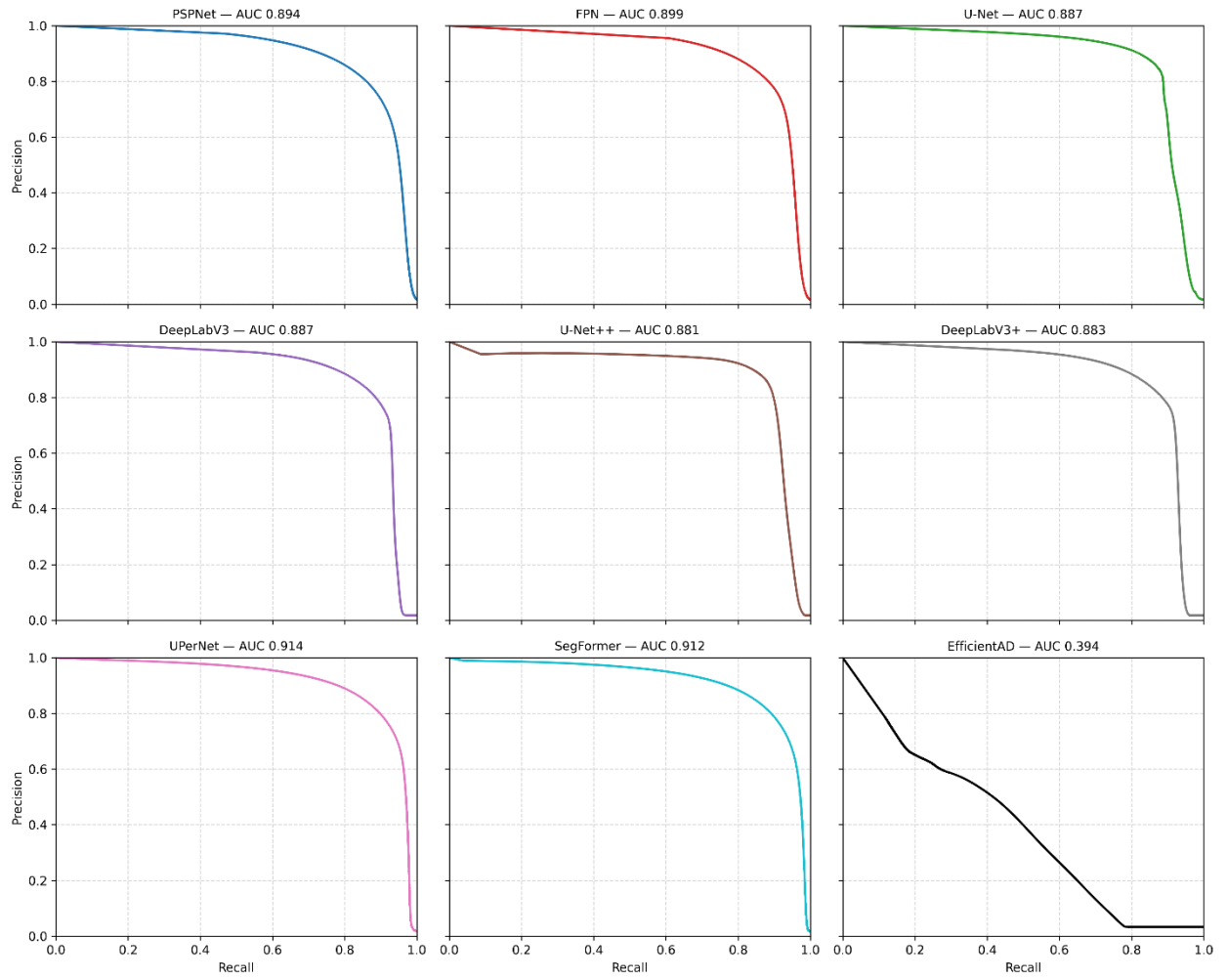


Figure 4-5: Precision-Recall Curves of Evaluated Models for Group 3

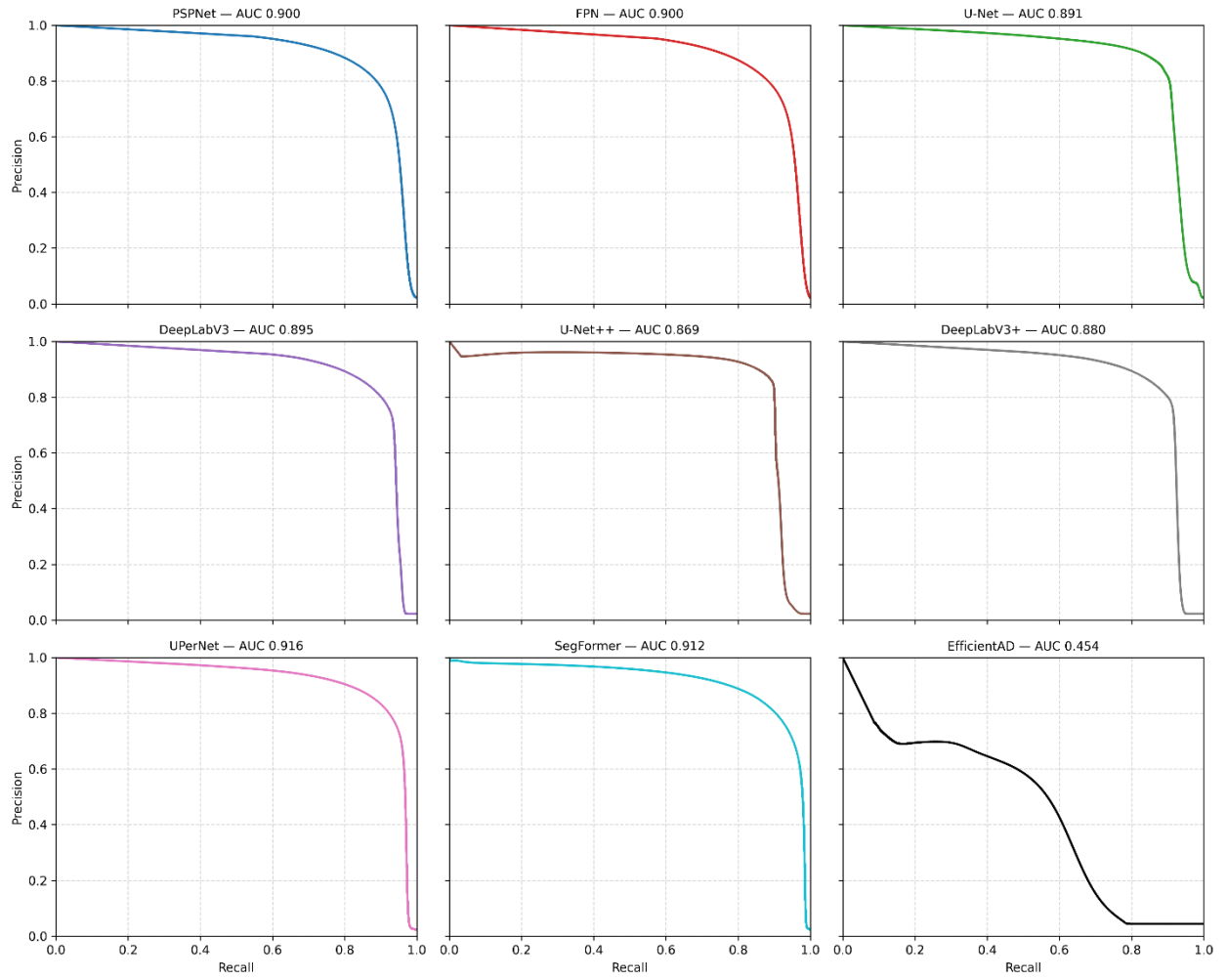


Figure 4-6: *Precision-Recall Curves of Evaluated Models for Group 4*

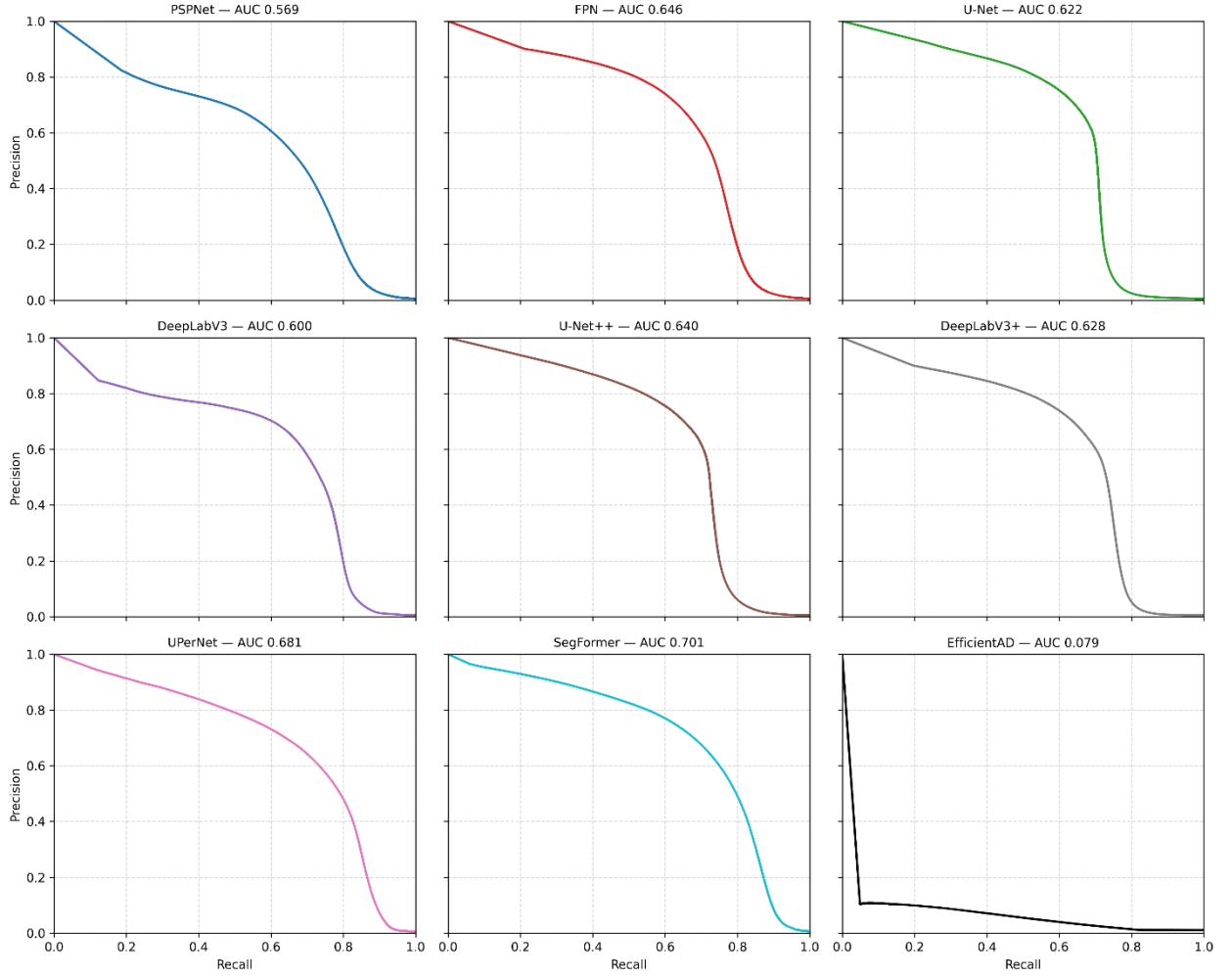


Figure 4-7: Precision-Recall Curves of Evaluated Models for Group 5

4.2.2 Group-Wise Performance Analysis

The group-wise analysis of pixel-level F1-scores provides a detailed understanding of how segmentation accuracy varies across fabric textures of different structural complexities. Unlike the sample-level results, which measured overall defect detectability, this evaluation focuses on how accurately each model delineates the actual defect regions at the pixel level. The table and histogram clearly show that performance trends are closely tied to texture regularity and surface uniformity.

In the first two texture groups, where the fabric patterns are well-structured and display strong spatial consistency, all supervised models achieve high pixel-level F1-scores, generally between 0.84 and 0.88. This reflects stable segmentation behavior in scenarios where the

background exhibits periodic repetition and defects disrupt clearly defined texture boundaries. The presence of strong edges and uniform spatial cues allows both convolutional and transformer-based architectures to accurately localize defect regions with minimal false positives. Even the unsupervised EfficientAD achieves moderate performance in these groups (around 0.55–0.57), benefiting from the predictable reconstruction of regular textures.

In the mid-range texture groups, segmentation performance remains strong but exhibits slight variability across models. The F1-scores for most supervised approaches stay within 0.84–0.87, with UPerNet showing the most consistent behavior across these groups. The stability of performance in these moderately irregular patterns suggests that the models effectively generalize to more diverse visual conditions, leveraging multi-scale context to maintain defect boundary precision. However, small fluctuations in accuracy highlight that color and illumination variations begin to challenge the discriminative power of the learned representations, especially for defects with blurred or diffuse edges.

A more noticeable decline in performance appears in the final texture group, characterized by low-contrast greige fabrics with weak structural cues. Here, pixel-level F1-scores drop significantly across all supervised models, falling to the 0.60–0.69 range, while EfficientAD experiences a sharp degradation to 0.09. This pattern underscores the difficulty of segmenting defects in homogeneous materials where defect-to-background contrast is minimal and texture continuity provides limited reference for spatial differentiation. The results indicate that both supervised and unsupervised methods struggle to capture defect boundaries when local gradients and intensity variations are nearly absent.

Overall, the group-wise pixel-level analysis reveals a clear correlation between texture complexity and segmentation reliability. As texture regularity decreases and visual entropy increases, the precision of defect localization declines steadily. Despite this, models such as

UPerNet and U-Net++ demonstrate strong adaptability across groups, maintaining balanced accuracy even in challenging texture conditions. These results highlight that the inherent visual properties of fabrics—particularly structural uniformity and contrast—play a dominant role in determining segmentation success, independent of architectural design.

4.2.3 Model-Wise Performance Analysis

At the model level, distinct trends emerge in how different architectural families respond to the challenges of defect segmentation on the ZJU-Leaper dataset. The observed variations in pixel-level F1-scores and precision–recall (PR) curve behaviors are closely tied to each model’s design philosophy, including feature hierarchy depth, receptive field expansion, and the treatment of fine-grained texture information.

Among the supervised approaches, U-Net and U-Net++ maintain high segmentation accuracy across all texture groups, reflecting the advantage of their dense skip connections and encoder–decoder symmetry. These connections facilitate precise recovery of spatial details lost during down sampling, allowing the models to delineate defect boundaries sharply even in complex weave structures. The incremental refinement in U-Net++, achieved through nested convolutional blocks, enhances feature reuse and stabilizes learning across scales—this contributes to its strong performance consistency ($F1 \approx 0.87$) across regular and moderately irregular textures.

DeepLabV3 and DeepLabV3+, with their atrous spatial pyramid pooling (ASPP) modules, effectively capture multi-scale contextual information, leading to reliable performance in textures with varying defect sizes. Their slightly lower F1-scores compared to U-Net++ can be attributed to coarse boundary predictions, a common limitation of dilated convolutions when fine defect details are subtle. PSPNet and FPN also deliver robust results, leveraging pyramid-based feature aggregation; however, their global pooling and top-down fusion strategies can sometimes over-smooth defect edges, slightly reducing pixel-wise precision in low-contrast

fabrics.

Transformer-based architectures, particularly UPerNet, exhibit outstanding overall generalization and stability. With F1-scores frequently exceeding 0.87 in regular textures and maintaining strong performance (~ 0.85) in more irregular groups, UPerNet demonstrates its ability to integrate hierarchical CNN features with global self-attention. This combination enables effective modeling of long-range spatial dependencies—allowing the network to perceive texture continuity and contextual contrast beyond the local receptive field. SegFormer, though conceptually similar, produces marginally lower scores, likely due to its lighter-weight decoder and reliance on pure attention mechanisms without sufficient convolutional refinement. The PR-curves for these models show steep, high-precision regions that sustain recall until near the saturation point, confirming their stable defect localization behavior.

In contrast, the EfficientAD model, representing the unsupervised class, exhibits a substantial gap in segmentation performance. Its F1-scores remain below 0.60 across most texture groups and collapse to near 0.10 on the most homogeneous fabrics. This degradation stems from its reliance on reconstruction-based anomaly detection: when background textures are repetitive, reconstruction errors become minimal even for defective regions, making anomaly separation unreliable. The corresponding PR-curves further confirm this limitation, showing weak precision–recall trade-offs and near-linear patterns indicative of indistinguishable normal and defective pixel distributions.

Overall, this comparative analysis emphasizes that supervised CNN–Transformer hybrids remain the most effective for precise pixel-level defect localization. Architectures that blend multi-scale convolutional features with attention-driven global reasoning, such as UPerNet and U-Net++, offer the optimal balance between spatial fidelity and contextual awareness. Unsupervised models like EfficientAD, while efficient and label-independent, still lack the

representational power required for accurate segmentation under diverse and visually subtle fabric conditions.

4.2.4 Qualitative Pixel-Level Analysis

The qualitative evaluation complements the quantitative findings by visually comparing predicted segmentation masks from all nine models against the corresponding ground-truth defect annotations. One representative sample from each dataset group was selected to illustrate how model performance varies across texture regularity, color complexity, and contrast conditions. Each visualization presents the defective image, its ground-truth mask, and the predicted masks from PSPNet, FPN, U-Net, DeepLabV3, U-Net++, DeepLabV3+, UPerNet, SegFormer, and EfficientAD.

At a general level, the qualitative results strongly align with the quantitative F1-scores and PR-curve behaviors. Supervised CNN–Transformer hybrids, particularly U-Net++ and UPerNet, consistently produce segmentation masks that are visually precise and closely follow the true defect contours. These models preserve spatial fidelity and maintain consistent mask thickness even in patterned or colored textures, confirming their ability to learn robust multi-scale features. DeepLabV3+ also performs well, although its predictions tend to slightly over-smooth defect edges due to its atrous spatial pooling mechanism.

Conversely, PSPNet and FPN occasionally generate thicker boundary regions because their pyramid pooling layers integrate high-level contextual cues at the expense of localized precision. SegFormer captures the overall defect structure but can under-segment finer details, while EfficientAD shows substantial over-segmentation or missed detection, reflecting its unsupervised nature and reliance on reconstruction errors that fail under subtle contrast variations.

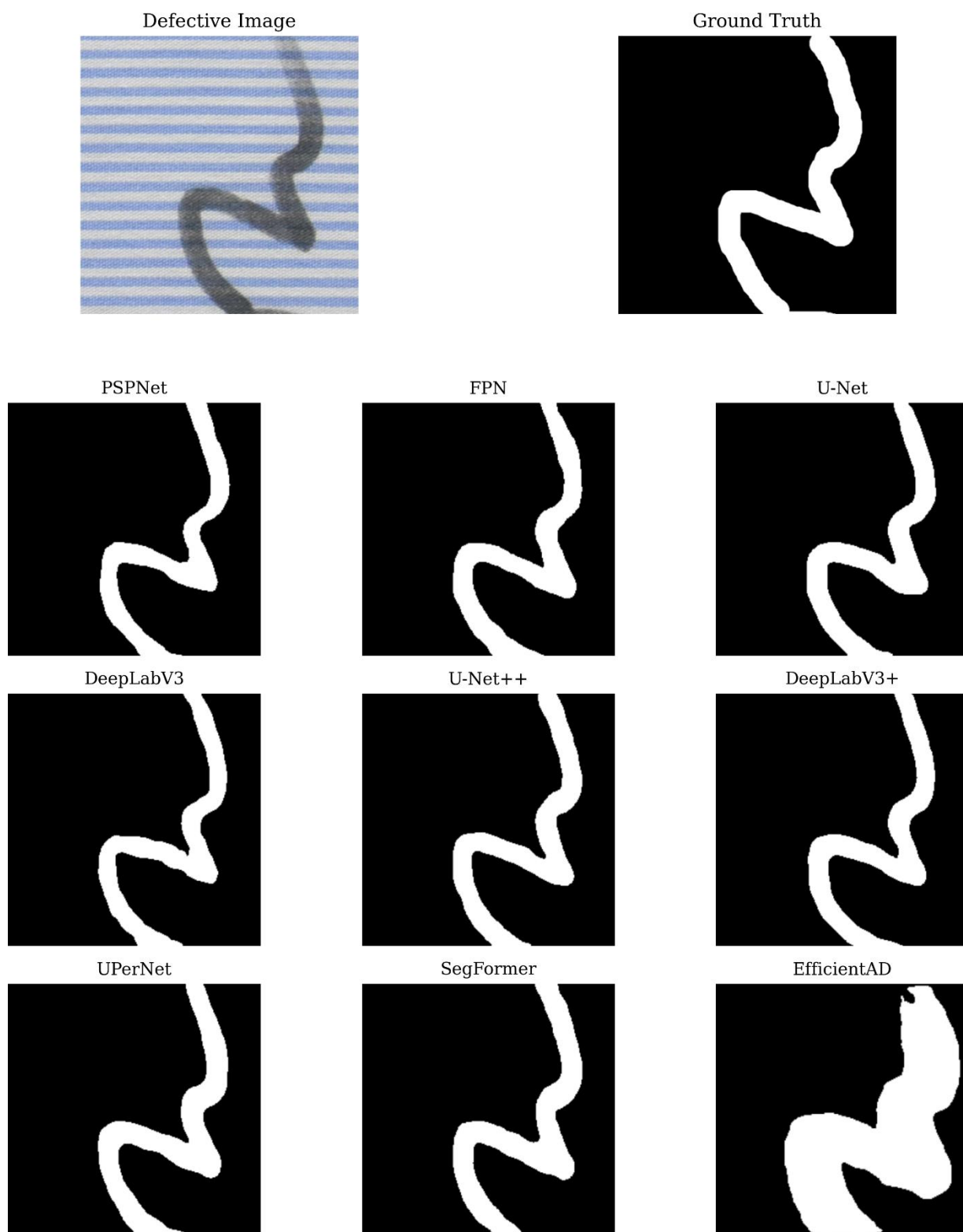


Figure 4-8: *Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 1*

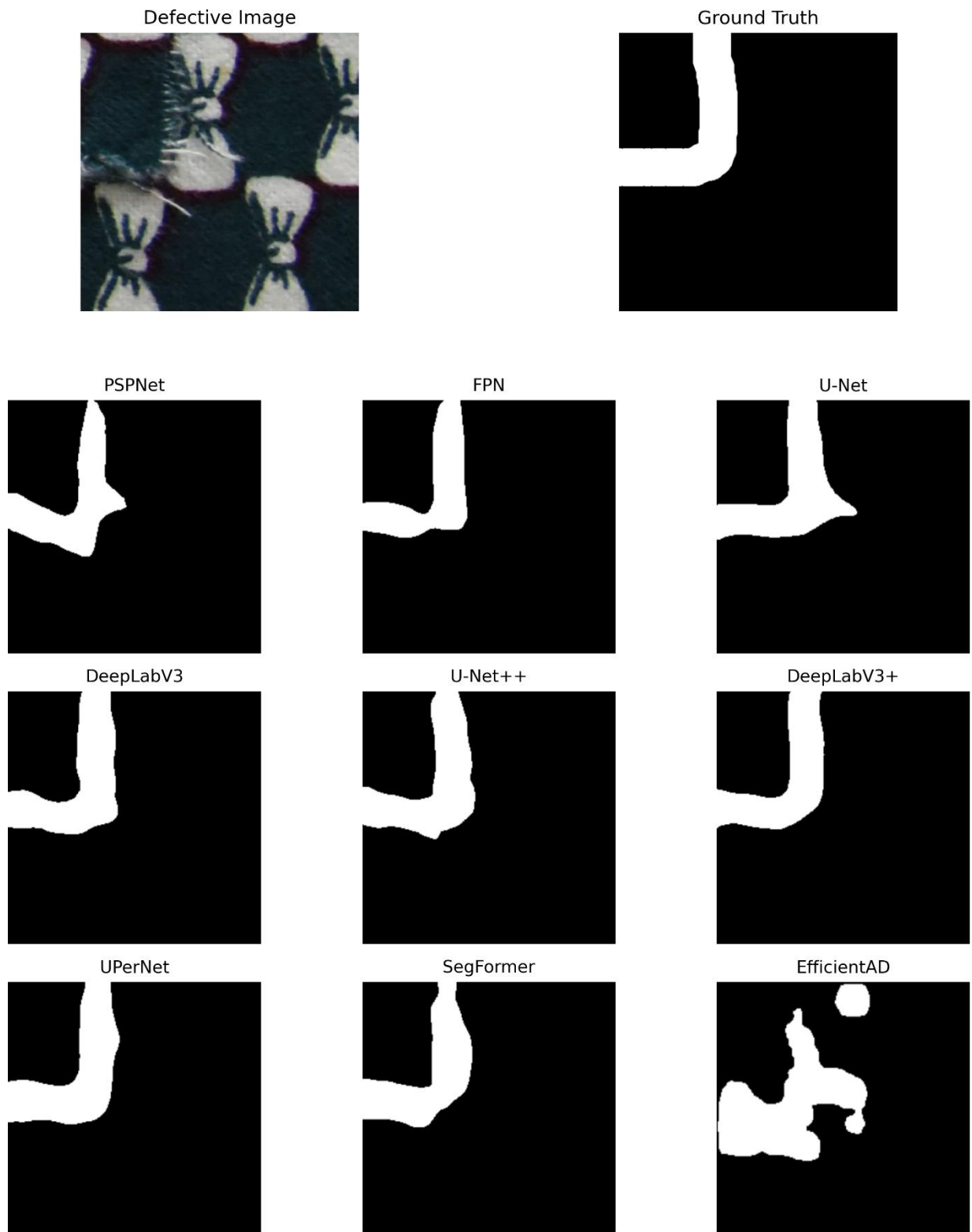


Figure 4-9: *Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 2*

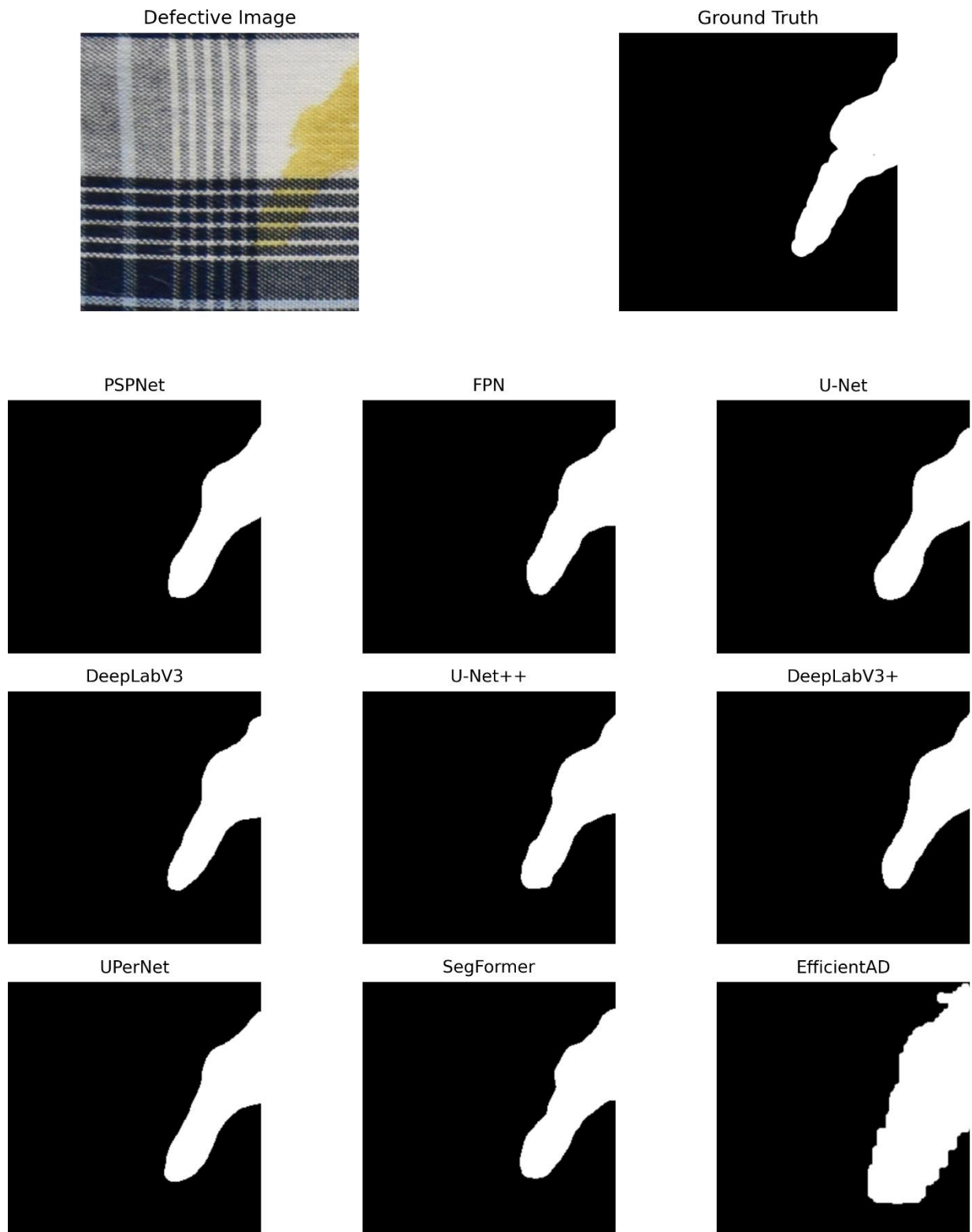


Figure 4-10: *Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 3*

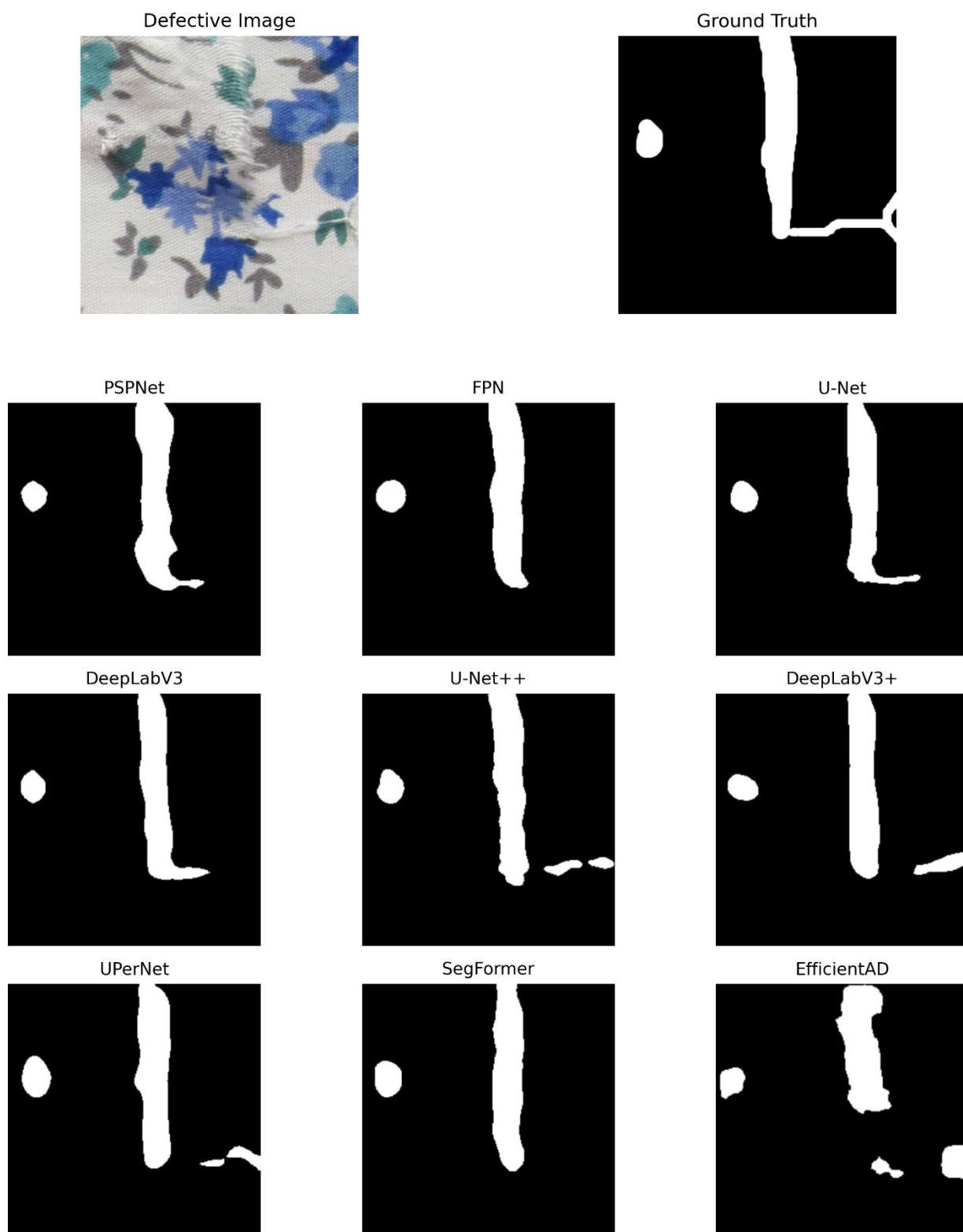


Figure 4-11: *Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 4*

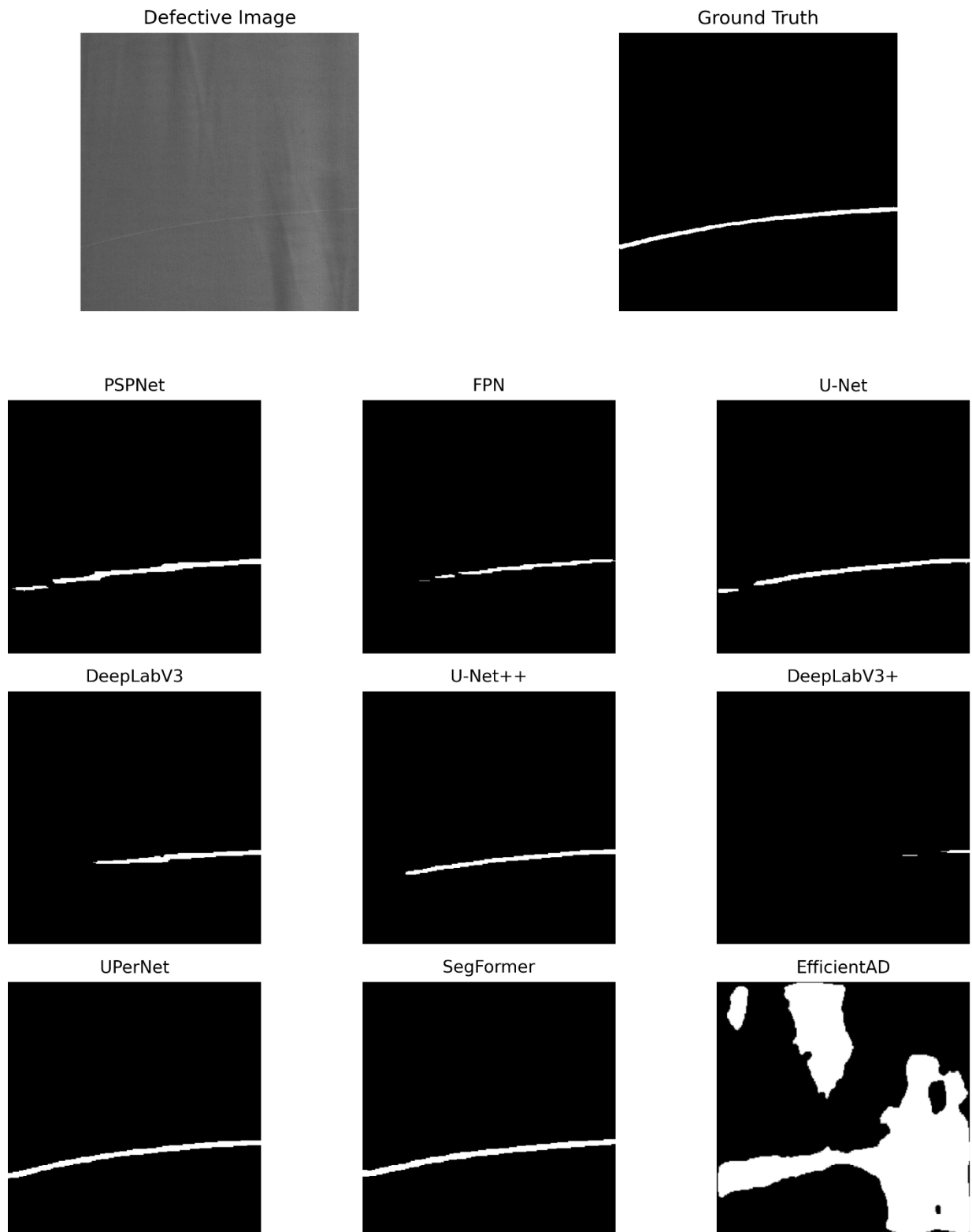


Figure 4-12: *Visual Comparison of Ground Truth and Predicted Masks from All Models on a Representative Defective Sample from Group 5*

In Figure 4-8, the sample represents a highly regular fabric with repetitive horizontal textures and strong local contrast around the defect. Almost all supervised models—including U-Net, U-Net++, DeepLabV3+, and UPerNet—accurately trace the defect contour, producing masks nearly identical to the ground truth. Boundary precision is particularly high, showing that these models effectively leverage consistent structural cues. PSPNet and FPN slightly widen the defect boundaries due to spatial pooling’s global averaging. EfficientAD, however, over-segments by extending predictions outside the true region, illustrating its instability in structured patterns. This observation aligns with its lower F1-score (≈ 0.57).

In Figure 4-9, depicting a moderately complex patterned fabric, CNN-based architectures retain strong localization accuracy: U-Net++ and DeepLabV3+ produce clean and connected defect masks that align closely with the ground truth. UPerNet maintains similar accuracy, confirming its capability to preserve context across patterned textures. SegFormer captures the overall shape but with slightly thicker boundaries. PSPNet and FPN under-detect thinner defect areas, while EfficientAD produces diffuse and inaccurate masks. The visual differences across models reflect the numerical gap between the supervised models ($F1 \approx 0.84\text{--}0.87$) and EfficientAD ($F1 \approx 0.55$).

In Figure 4-10, which illustrates a geometric-weave fabric with moderate irregularity, the segmentation challenge lies in distinguishing true defects from complex woven patterns. U-Net++ and UPerNet continue to outperform others, producing nearly perfect structural continuity and boundary alignment. DeepLabV3 and DeepLabV3+ maintain adequate recall but slightly thicker contours. PSPNet, FPN, and SegFormer blur defect boundaries at points where the background pattern overlaps. EfficientAD shows a large overestimation of the defect region, demonstrating its sensitivity to texture variance. This qualitative behavior reinforces the quantitative finding that while most supervised models sustain $F1 \approx 0.85$, unsupervised

detection drops below 0.46.

In Figure 4-11, showing a printed fabric with strong color variations, chromatic noise and irregular patterns create greater complexity for segmentation. Despite these challenges, UPerNet and U-Net++ produce highly accurate and visually clean segmentation masks, closely following the ground truth. DeepLabV3+ also achieves stable performance, though its boundaries show mild thickening. PSPNet and FPN create slightly wider segmentation regions due to their strong reliance on global pooling. SegFormer achieves reasonable coverage but omits small localized defects. EfficientAD, in contrast, produces disconnected and noisy masks, misclassifying multiple non-defective regions. These results correspond to the F1-scores where UPerNet leads with ≈ 0.87 , while EfficientAD remains below 0.55.

In Figure 4-12, the most challenging example, a low-contrast greige fabric, all models experience notable performance degradation. Here, all models face significant degradation in localization precision. U-Net, U-Net++, and UPerNet retain partial detection accuracy by capturing the general defect trajectory, but smaller or lighter sections remain missed. SegFormer and DeepLabV3+ produce thinner, fragmented traces, while PSPNet and FPN underperform due to pooling-induced smoothing. EfficientAD misclassifies the background almost entirely, producing broad irregular masks. This figure demonstrates how homogeneous textures weaken spatial cues, leading to the lowest group-wise F1 scores (≈ 0.60 – 0.68 for supervised, ≈ 0.10 for unsupervised).

Across all figures, the qualitative outcomes strongly corroborate the quantitative and PR-curve results. The supervised models, particularly U-Net++ and UPerNet, deliver superior boundary fidelity and structural completeness, confirming their strong capacity for multi-scale feature extraction and global contextual reasoning. DeepLabV3+ remains effective but occasionally sacrifices edge sharpness for contextual integration, while SegFormer offers solid

generalization with minor boundary blurring. PSPNet and FPN retain high precision in simple textures but are less reliable in irregular or low-contrast fabrics.

In contrast, EfficientAD fails to maintain consistent defect localization across texture conditions, reinforcing the weakness of reconstruction-based unsupervised learning when discriminative supervision is absent. Overall, the qualitative findings affirm that high F1-scores translate directly into perceptual accuracy—models with hierarchical convolutional refinement and attention-based context modeling produce predictions that are not only numerically superior but also visually aligned with human-perceived defect boundaries.

4.3 Region-Level Performance

While sample-level and pixel-level analyses provide insight into overall classification reliability and boundary precision, region-level evaluation offers a more detailed understanding of how segmentation models respond to defects of different spatial scales. This level of analysis measures performance across small, medium, and large defect regions, enabling assessment of each model’s ability to localize and segment defects with varying sizes, shapes, and boundary complexities.

Region-level metrics are particularly important in industrial inspection since the visual and structural characteristics of fabric defects often vary dramatically across scales. Small defects, such as tiny holes or micro-contaminations, challenge models’ sensitivity, whereas large-area defects demand accurate boundary delineation and robust contextual modeling. The F1-scores reported in this section are computed from the test set by categorizing defect masks according to their pixel area, following the standard criteria defined in the ZJU-Leaper benchmark.

The results are summarized in Table 4-3, which reports the region-level F1-scores of all nine models across the three defect-size categories. The trends are further visualized in Figure 4-13,

showing the comparative performance distribution. These results allow a multi-scale interpretation of model behavior, bridging quantitative accuracy with practical inspection robustness.

Table 4-3: *Region-Level F1 Scores by Model and Dataset Group*

Model	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.798	0.7761	0.7588	0.7624	0.4012
FPN	0.8204	0.8135	0.8023	0.7884	0.491
U-Net	0.8464	0.8436	0.8024	0.8185	0.5327
DeepLabV3	0.7887	0.813	0.7801	0.7821	0.4978
U-Net++	0.813	0.8071	0.795	0.8118	0.5098
DeepLabV3+	0.8447	0.8514	0.8329	0.821	0.4886
UPerNet	0.8691	0.8444	0.8462	0.8295	0.4844
SegFormer	0.8605	0.8365	0.8075	0.8105	0.4441
EfficientAD	0.4622	0.4214	0.3391	0.3263	0.2254

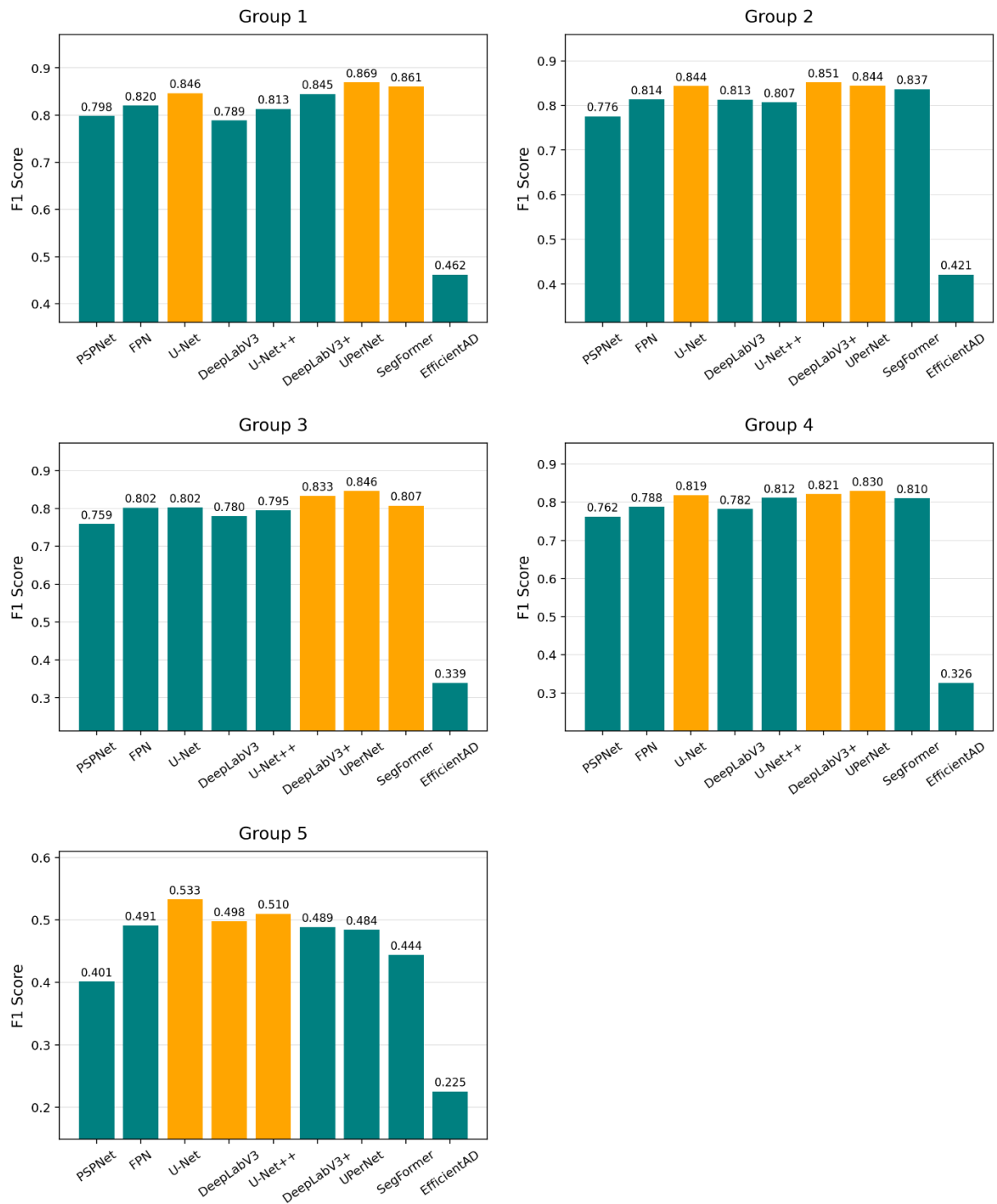


Figure 4-13: Comparison of Region-Level F1 Scores of All Evaluated Models Across Group 1-5

4.3.1 Large Defect Region Performance

Large defects are defined as regions exceeding 10,000 pixels in area, representing extensive surface irregularities, torn yarns, or large contamination marks. These defects typically provide strong visual contrast and structural cues, allowing models to achieve high segmentation consistency across most texture groups.

Table 4-4: *Region-Level F1 Scores for Large Regions by Model and Dataset Group*

Model	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.986	0.99	0.99	0.981	0.714
FPN	0.995	0.992	0.99	0.984	0.8
U-Net	0.99	0.994	0.99	0.986	0.714
DeepLabV3	0.99	0.994	0.992	0.989	0.8
U-Net++	0.99	0.99	0.997	0.992	0.714
DeepLabV3+	0.995	0.992	0.992	0.992	0.8
UPerNet	1	0.987	0.992	0.986	0.8
SegFormer	0.996	0.98	0.995	0.99	0.75
EfficientAD	0.86	0.859	0.677	0.869	0.056

The region-level results for large defects, summarized in Table 4-4 and visualized in Figure 4-14 show uniformly high F1-scores across nearly all supervised models, generally above 0.98 in Groups 1–4. This consistency indicates that when defects occupy substantial areas, models can exploit richer contextual and spatial information to capture the boundaries effectively. The transformer-based UPerNet achieves the highest or near-highest scores in multiple groups, reaching perfect segmentation ($F1 = 1.00$) in some textures. This performance reflects its strong global attention mechanism and superior capacity to model wide contextual dependencies—an advantage particularly relevant for large-scale irregularities. Similarly, CNN-based architectures such as DeepLabV3+, U-Net++, and PSPNet maintain strong and nearly indistinguishable accuracy, with only marginal variations in the third decimal place.

Across texture groups, the results follow a clear trend. Groups 1–3, characterized by structured and repetitive patterns, exhibit the highest segmentation accuracy across all models ($F1 > 0.99$).

Group 4 shows a minor reduction (average F1 $\approx$ 0.985–0.99), suggesting that moderate color or weave irregularities introduce slight boundary noise but do not significantly affect detection. A more noticeable decline occurs in Group 5, where performance drops sharply (F1 $\approx$ 0.71–0.80 for most models). This group’s homogeneous and low-contrast texture presents fewer visual gradients for feature differentiation, causing both CNN and transformer models to underperform. The unsupervised EfficientAD, which depends on reconstruction-based anomaly detection, struggles severely here, with its F1-scores plummeting below 0.10 in Group 5—evidence of its limited capacity to represent broad, non-local anomalies.

Overall, the large defect analysis indicates that all supervised models exhibit saturation-level accuracy for prominent, well-structured anomalies. Their minimal variance across architectures suggests that large-scale defect localization is largely a solved problem within this dataset. However, the significant drop in low-contrast regions implies that even for large-area defects, models still rely heavily on local texture variation rather than absolute geometric or contextual cues—a limitation that becomes more pronounced as defect size decreases, as explored in the next section.

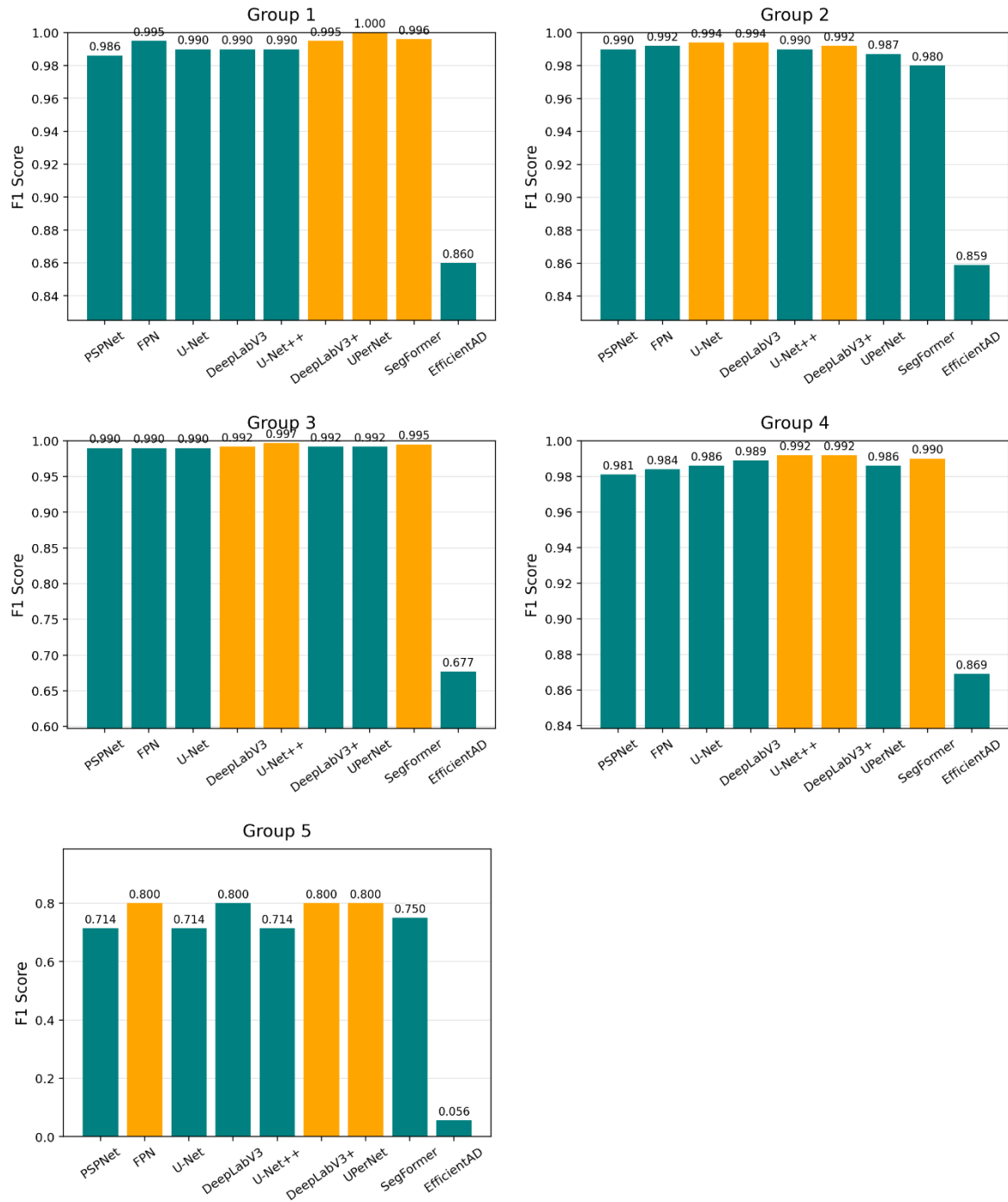


Figure 4-14: Comparison of Region-Level F1 Scores for Large Regions of All Evaluated Models Across Group 1-5

4.3.2 Medium Region Performance

Medium defects are defined as regions covering between 2,500 and 10,000 pixels, typically representing localized distortions such as weave irregularities, moderate stains, or partial thread misalignments. These defects are visually prominent but occupy smaller portions of the fabric than large anomalies, making segmentation performance sensitive to both contextual reasoning and edge precision.

Table 4-5: *Region-Level F1 Scores for Medium Regions by Model and Dataset Group*

Model	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.974	0.956	0.969	0.952	0.606
FPN	0.981	0.967	0.971	0.962	0.653
U-Net	0.978	0.972	0.975	0.969	0.699
DeepLabV3	0.97	0.97	0.982	0.965	0.673
U-Net++	0.977	0.967	0.98	0.972	0.68
DeepLabV3+	0.978	0.967	0.98	0.967	0.673
UPerNet	0.98	0.971	0.984	0.968	0.719
SegFormer	0.979	0.963	0.974	0.97	0.748
EfficientAD	0.774	0.702	0.632	0.727	0.08

As summarized in Table 4-5 and illustrated in Figure 4-15, the overall F1-scores for medium defects remain consistently high across most supervised models, ranging between 0.95 and 0.98 in the first four texture groups. This reflects that all architectures—particularly UPerNet, DeepLabV3+, and U-Net++—can effectively balance contextual awareness with detailed boundary reconstruction when the defect occupies a moderate area. Across the dataset, the top-performing models maintain stable segmentation accuracy, with UPerNet frequently yielding the highest F1 values (up to 0.984), followed closely by DeepLabV3+ and FPN. These results demonstrate that both transformer-based and CNN-based architectures are capable of extracting sufficient contextual cues while retaining precise region boundaries in this defect size range.

Examining the texture-specific behavior reveals distinct group-level variations. Groups 1–3,

consisting of regular or semi-regular textures, exhibit minimal deviation between models, with nearly saturated accuracy (average $F1 \approx 0.97\text{--}0.98$). These results indicate that moderate-scale defects in structured environments are effectively captured, as sufficient texture contrast aids spatial differentiation. In Group 4, a slight decline is observed (average $F1 \approx 0.96\text{--}0.97$), likely caused by complex weave orientations or subtle color transitions that increase boundary ambiguity. The lowest performance again occurs in Group 5, where $F1$ -scores drop to approximately $0.60\text{--}0.75$ for most models. The near-homogeneous appearance and minimal gradient variation of this group hinder segmentation precision, reflecting the inherent difficulty of isolating medium defects in visually uniform textures.

The EfficientAD model continues to perform significantly below the supervised models, maintaining moderate correlation in highly structured groups ($F1 \approx 0.63\text{--}0.77$) but collapsing to extremely low accuracy ($F1 = 0.08$) in unstructured textures. This behavior further validates its reliance on global reconstruction differences, which fail to capture localized mid-scale irregularities.

In summary, medium-sized defect segmentation reveals a transition zone of model sensitivity—large enough to be spatially contextualized yet small enough to challenge precise boundary detection. The strong and consistent performance of both UPerNet and DeepLabV3+ underscores their balanced design, combining multi-scale feature aggregation and contextual encoding.

However, the steady degradation from Groups 1–4 to Group 5 demonstrates that texture uniformity remains a primary limiting factor, affecting all models regardless of architecture. These observations provide a bridge toward the next section, where the challenge intensifies as the analysis shifts to small defects, which demand even finer spatial sensitivity and higher detection granularity.

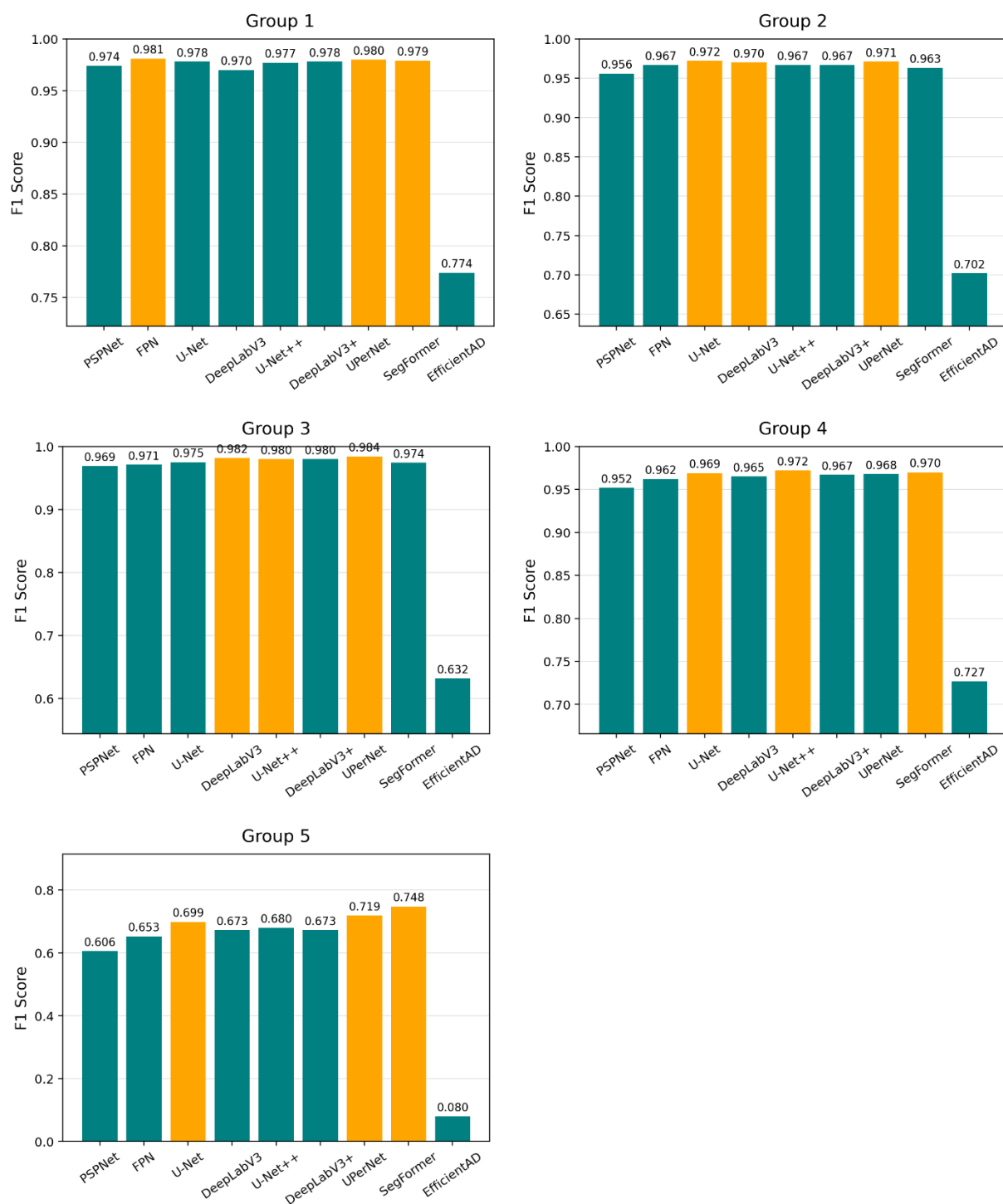


Figure 4-15: Comparison of Region-Level F1 Scores for Medium Regions of All Evaluated Models Across Group 1-5

4.3.3 Small Region Performance

Small defects are defined as regions containing fewer than 2,500 pixels, corresponding to subtle imperfections such as micro-holes, minor thread distortions, lint, or fine surface contaminants. These small-scale anomalies pose the greatest challenge for automated inspection, as they often blend with fabric texture noise and provide limited pixel-level contrast.

Table 4-6: *Region-Level F1 Scores for Small Regions by Model and Dataset Group*

Model	Group 1	Group 2	Group 3	Group 4	Group 5
PSPNet	0.688	0.604	0.603	0.595	0.397
FPN	0.712	0.664	0.666	0.622	0.488
U-Net	0.758	0.711	0.669	0.666	0.529
DeepLabV3	0.674	0.659	0.626	0.611	0.493
U-Net++	0.71	0.659	0.661	0.659	0.506
DeepLabV3+	0.75	0.728	0.708	0.665	0.485
UPerNet	0.791	0.726	0.74	0.702	0.485
SegFormer	0.777	0.707	0.68	0.66	0.438
EfficientAD	0.254	0.183	0.164	0.109	0.245

The summarized results in Table 4-6 and the comparative visualization in Figure 4-16 reveal a consistent performance drop across all models when addressing small defects. The overall F1-scores for supervised methods range between 0.60 and 0.75, representing a substantial decrease compared to the medium and large defect cases. This trend underscores the intrinsic difficulty of detecting fine-grained anomalies with limited contextual evidence.

A clear pattern emerges across the dataset: all models perform relatively better in textures with higher structural regularity, while performance declines as visual complexity increases. For instance, in the most structured groups, segmentation accuracy remains moderately stable (F1 $\approx$ 0.70–0.79), led by UPerNet, SegFormer, and DeepLabV3+. The transformer-based UPerNet consistently achieves the best or near-best performance across most texture types (up to F1 = 0.79), likely due to its global attention mechanism, which enhances contextual inference even when defects occupy sparse or disconnected pixels. However, as texture irregularity and low

contrast dominate (as seen in the less structured groups), all models experience substantial degradation, with F1-scores commonly dropping below 0.65. Here, models struggle to differentiate subtle local intensity variations from normal weave noise, causing misclassification and incomplete boundary recovery.

From an architectural perspective, CNN-based segmentation networks such as U-Net, DeepLabV3, and U-Net++ show relatively consistent but slightly lower F1 values (≈ 0.65 – 0.72), reflecting their reliance on localized feature hierarchies that can miss sub-pixel details in smooth textures. In contrast, transformer-based models, particularly UPerNet and SegFormer, display slightly higher resilience, retaining more spatial context for small, dispersed anomalies. Nevertheless, even these architectures plateau below ideal performance, indicating that global attention alone cannot compensate for insufficient pixel-level contrast or annotation-level granularity.

The EfficientAD model demonstrates the weakest results, with F1-scores dropping to as low as 0.10–0.25 in multiple groups. Its generative reconstruction-based approach struggles to represent fine-scale deviations, often misidentifying small defects as noise. This reinforces the limitations of unsupervised anomaly detection in fine defect segmentation tasks, particularly where contextual cues are subtle and heterogeneous.

In summary, the analysis of small defect regions highlights the most critical limitation of current segmentation and detection frameworks: insufficient sensitivity to small, low-contrast irregularities. Despite high overall capability at larger scales, model accuracy declines sharply as spatial granularity decreases. The consistent underperformance across all architectures emphasizes the need for enhanced fine-detail feature extraction, such as multi-resolution fusion, attention refinement, or hybrid CNN-transformer pipelines, which could improve robustness in identifying fine-scale anomalies within industrial fabric inspection systems.

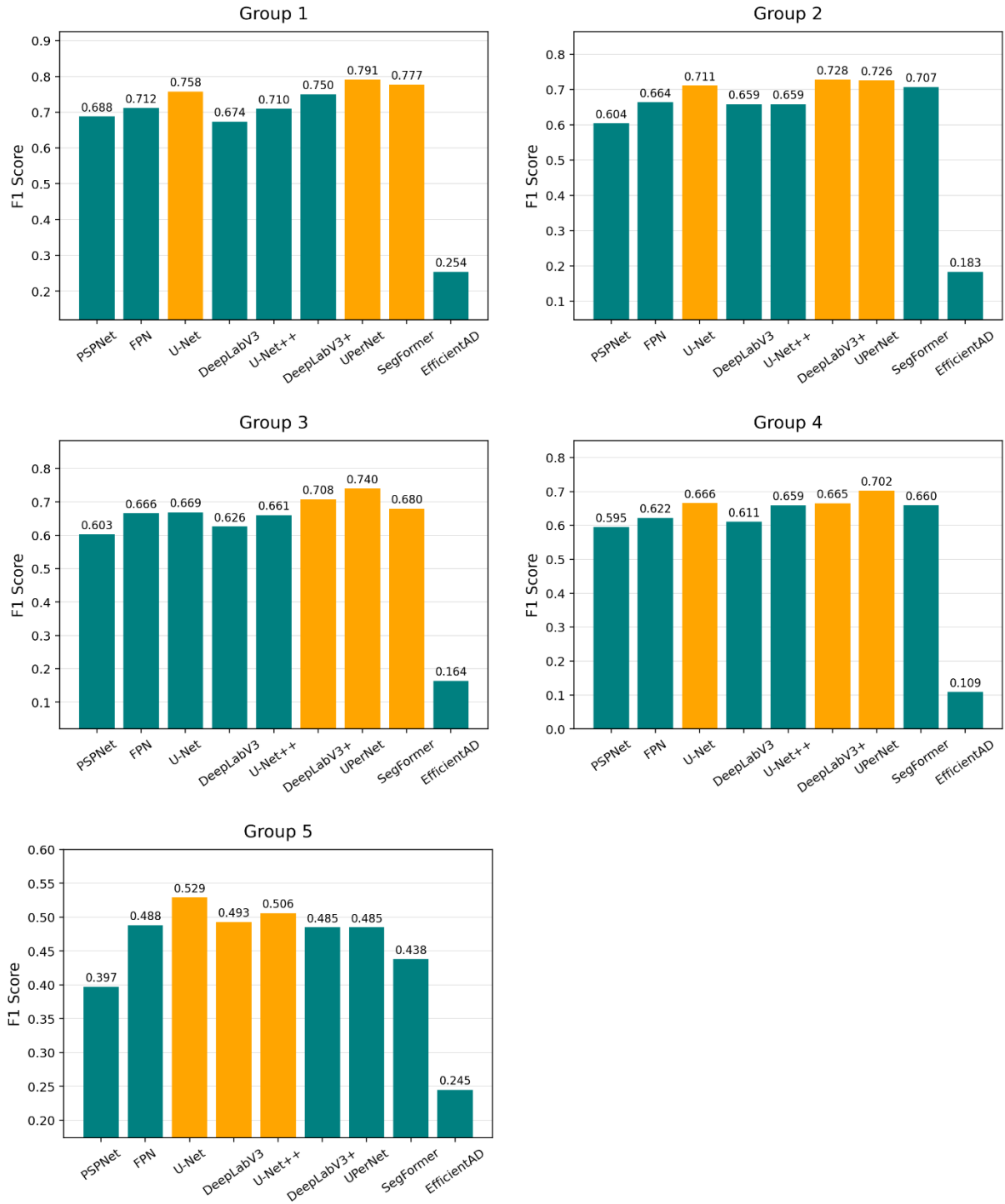


Figure 4-16: Comparison of Region-Level F1 Scores for Small Regions of All Evaluated Models Across Group 1-5

4.4 Accuracy–Efficiency Trade-off Analysis

Up to this point, the analyses have focused primarily on the accuracy of segmentation models—evaluated through sample-, pixel-, and region-level F1-scores—to determine how effectively each architecture localizes fabric defects.

In real-world industrial inspection, however, computational efficiency is equally critical. Models must not only be accurate but also lightweight, fast, and deployable within the processing constraints of automated production lines.

This section therefore presents an Accuracy–Efficiency Trade-off Analysis to investigate how each model’s computational complexity and runtime behavior influence its practical usability.

Four efficiency indicators are considered:

- 1) Parameter count (M),
- 2) Floating-point operations (GFLOPs),
- 3) Inference speed (frames per second, FPS), and
- 4) Latency (ms per image).

The EfficientAD model is excluded from this analysis. Although it serves as an unsupervised baseline, its fundamentally different reconstruction-based inference mechanism and markedly lower segmentation accuracy make direct efficiency comparison irrelevant. The discussion below therefore focuses on the supervised segmentation architectures only.

4.4.1 Model Complexity: Parameters and FLOPs

Table 4-7 summarizes the computational complexity of all evaluated models in terms of floating-point operations (GFLOPs) and parameter count (millions). These two metrics collectively describe the resource requirements of each architecture where FLOPs represent the total arithmetic operations per forward pass, and parameters reflect memory footprint and model size.

The results reveal substantial variability in complexity across architectural families. PSPNet

stands out as the most lightweight model, requiring only 2.96 GFLOPs and 2.24 million parameters, making it highly suitable for low-resource deployments. FPN and DeepLabV3+ also maintain moderate complexity (7.8–9.2 GFLOPs) despite having larger parameter counts, due to efficient feature pyramid and depth-wise separable convolution designs. In contrast, U-Net and U-Net++ exhibit a significant increase in computational load (10.7 and 57.6 GFLOPs, respectively), largely attributable to their dense skip connections and multi-scale feature aggregation. DeepLabV3, though employing a similar encoder–decoder structure, incurs a high 41 GFLOPs due to its atrous spatial pyramid pooling (ASPP) modules.

Transformer-based architectures demonstrate the most diverse computational profiles. SegFormer, despite being transformer-based, achieves a remarkably efficient structure with only 3.31 GFLOPs and 13.68 million parameters—thanks to its lightweight Mix Transformer backbone and hierarchical token reduction. In contrast, UPerNet demands 79.87 GFLOPs and over 121 million parameters, positioning it as the heaviest model in the supervised category, driven by its multi-stage attention fusion and deep decoder.

Overall, this analysis establishes a clear computational hierarchy—from lightweight convolutional designs suitable for embedded inspection systems to transformer-based architectures optimized for high-fidelity defect segmentation at greater computational expense.

Table 4-7: *Computational Complexity of Segmentation Models in Terms of FLOPs and Parameter Count*

Model	FLOPs (G)	Params (M)
PSPNet	2.962	2.238
FPN	7.845	26.114
U-Net	10.707	32.521
DeepLabV3	41.031	39.634
U-Net++	57.561	48.986
DeepLabV3+	9.226	26.678
UPerNet	79.869	121.101
SegFormer	3.31	13.678

4.4.2 Runtime Performance: Inference Speed and Latency

To complement the model complexity evaluation, this section presents the runtime performance of the supervised segmentation models in terms of inference speed (frames per second, FPS) and latency (milliseconds per image). These metrics directly reflect how efficiently each model executes during deployment, especially under the high-throughput demands of automated fabric inspection systems.

Table 4-8 to Table 4-12 summarize the inference speed and latency measured across the five dataset groups. Each model was evaluated under identical hardware conditions using the same input resolution and batch size. As the results show, performance trends remain consistent across groups, with only marginal fluctuations (<5%), confirming that fabric texture variation does not significantly affect inference efficiency.

Table 4-8: *Inference Speed and Latency of Segmentation Models on Group 1*

Model	Inference Speed (FPS)	Latency (ms)
PSPNet	617.05	1.62
FPN	386.06	2.59
U-Net	328.62	3.04
DeepLabV3+	372.91	2.68
SegFormer	393.71	2.54
DeepLabV3	154.15	6.49
U-Net++	98.48	10.15
UPerNet	46.67	21.43

Table 4-9: *Inference Speed and Latency of Segmentation Models on Group 2*

Model	Inference Speed (FPS)	Latency (ms)
PSPNet	570.19	1.75
FPN	380.86	2.63
U-Net	326.01	3.07
DeepLabV3+	385.69	2.59
SegFormer	390.25	2.56
DeepLabV3	152.5	6.56
U-Net++	99.13	10.09
UPerNet	45.75	21.86

Table 4-10: Inference Speed and Latency of Segmentation Models on Group 3

Model	Inference Speed (FPS)	Latency (ms)
PSPNet	597.59	1.67
FPN	402.92	2.48
U-Net	329.61	3.03
DeepLabV3+	393.51	2.54
SegFormer	378.75	2.64
DeepLabV3	154.33	6.48
U-Net++	99.38	10.06
UPerNet	46.52	21.49

Across all groups, PSPNet consistently delivers the fastest inference, exceeding 600 FPS with a latency around 1.6 ms per frame. This makes it the most efficient model in real-time performance. FPN, SegFormer, and DeepLabV3+ follow closely, achieving speeds between 380–410 FPS and maintaining latency below 3 ms, striking a good balance between accuracy and speed. U-Net exhibits moderate performance (~325 FPS, 3 ms latency), while its deeper variant U-Net++ incurs significantly higher latency (~10 ms) due to the dense skip connections and expanded decoder depth. DeepLabV3 demonstrates relatively slower processing (~155 FPS, 6.5 ms latency), as its ASPP module increases the number of convolutional operations. Finally, UPerNet, the heaviest model, records the lowest speed (~46 FPS) and highest latency (~21 ms), highlighting the computational overhead introduced by transformer-based multi-scale attention.

Table 4-11: Inference Speed and Latency of Segmentation Models on Group 4

Model	Inference Speed (FPS)	Latency (ms)
PSPNet	535.74	1.87
FPN	385.21	2.6
U-Net	322.4	3.1
DeepLabV3+	379.43	2.64
SegFormer	402.6	2.48
DeepLabV3	154.75	6.46
U-Net++	98.88	10.11
UPerNet	46.14	21.68

Table 4-12: Inference Speed and Latency of Segmentation Models on Group 5

Model	Inference Speed (FPS)	Latency (ms)
PSPNet	602.16	1.66
FPN	408.07	2.45
U-Net	327.77	3.05
DeepLabV3+	384.05	2.6
SegFormer	409.21	2.44
DeepLabV3	154.28	6.48
U-Net++	98.82	10.12
UPerNet	46.26	21.62

Overall, these results reveal that PSPNet, FPN, and SegFormer offer the best runtime performance, making them attractive for time-sensitive inspection pipelines.

U-Net++, DeepLabV3, and UPerNet trade off speed for improved feature representation and segmentation precision—suitable for offline or batch defect analysis rather than live production monitoring.

4.4.3 Accuracy vs Efficiency Correlation

While the previous analyses evaluated model accuracy and efficiency independently, practical deployment requires a holistic understanding of how segmentation accuracy correlates with computational and runtime efficiency. To examine this relationship, Figure 4-17 and Figure 4-18 present comparative scatter plots illustrating the trade-offs between F1-score, inference speed, and FLOPs across all five texture groups.

4.4.3.1 Accuracy vs Inference Speed

For each dataset group, Figure 4-17 presents the relationship between pixel-level F1-score and inference speed (FPS). Each bubble represents a model, where its position reflects the trade-off between accuracy and runtime efficiency, and its size corresponds to the number of parameters. Models located in the upper-right quadrant exhibit the most desirable balance — combining high segmentation precision with rapid inference throughput.

Across all five groups, a consistent Pareto-frontier trend emerges:

- PSPNet and FPN dominate the high-speed region, sustaining exceptionally fast inference (exceeding 500 FPS on average) with minimal latency. Despite their slightly lower F1-scores (≈ 0.84 – 0.87), these models demonstrate strong practical efficiency, making them highly suitable for real-time inspection pipelines where throughput is critical.
- U-Net++ and DeepLabV3+ define the efficiency–accuracy equilibrium, balancing precision and speed with F1-scores around 0.87 – 0.88 and inference speeds of approximately 350–400 FPS. Their encoder–decoder refinements and moderate depth enable a good compromise between computation and accuracy.
- SegFormer consistently performs at a favorable trade-off point, achieving accuracy comparable to or better than U-Net++ while maintaining moderate speed (≈ 390 – 400 FPS). Its hierarchical transformer backbone effectively captures global context without the heavy computational penalty observed in UPerNet.
- UPerNet, although achieving the highest accuracy across most groups (often leading or matching SegFormer and U-Net++), records the lowest inference speed (≈ 50 FPS) due to its deep multi-stage architecture and extensive contextual encoding.

Overall, the Accuracy–Speed analysis confirms that SegFormer bridges the gap between lightweight CNNs and high-precision transformer frameworks. While PSPNet and FPN remain the most efficient for real-time deployment, SegFormer offers a compelling middle ground — delivering near state-of-the-art accuracy with moderate computational demand.

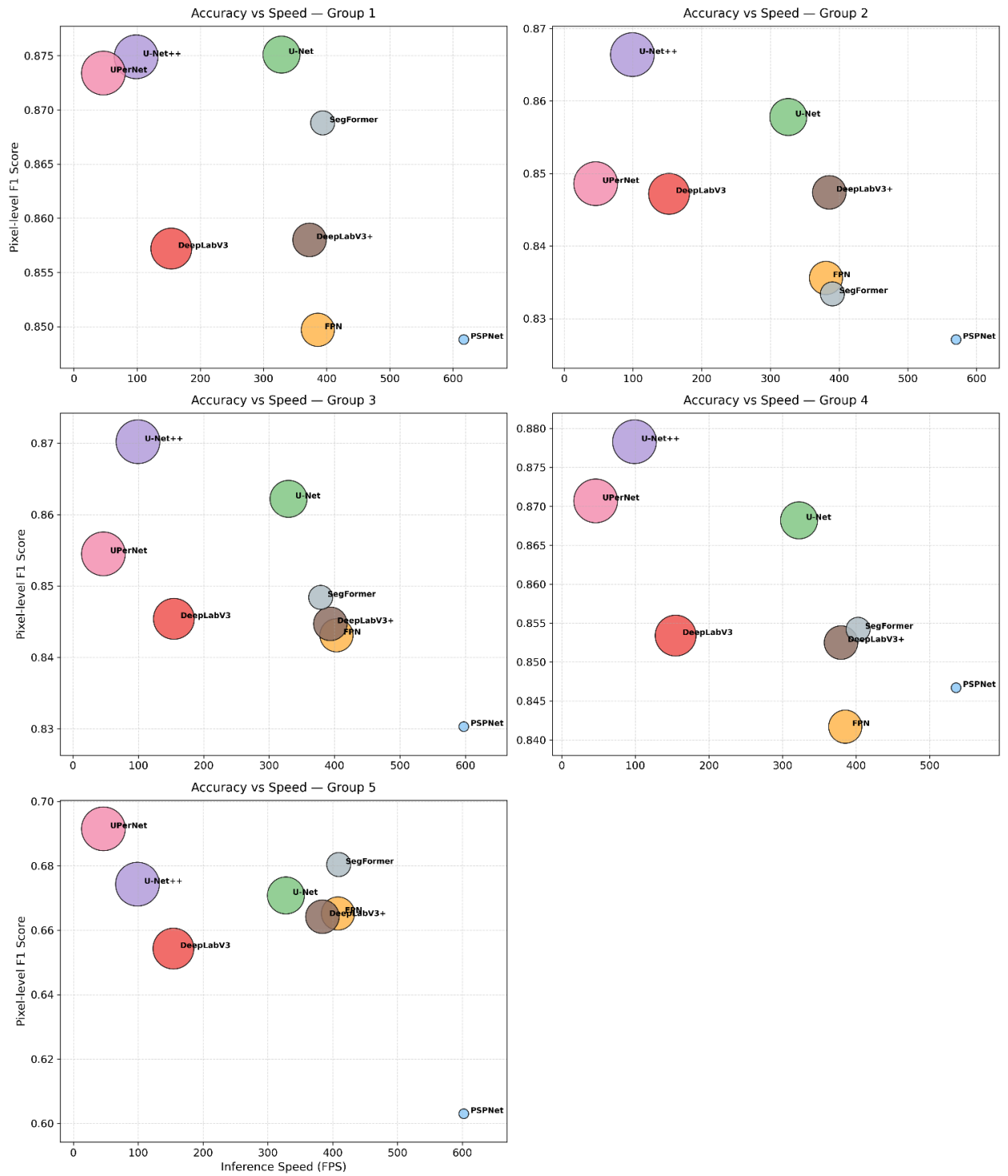


Figure 4-17: Scatter plots illustrating the trade-off between segmentation accuracy (F1-score) and inference speed (FPS) across Groups 1–5. Each bubble represents a model, with its size proportional to the number of parameters, showing how different architectures balance accuracy and runtime efficiency.

4.4.3.2 Accuracy vs FLOPs

A complementary analysis (Figure 4-18) investigates how computational demand (FLOPs) scales with segmentation accuracy. Here, F1-scores are plotted against FLOPs (G), with bubble size again representing parameter count. Models in the upper-left region (high F1, low FLOPs) exhibit optimal computational efficiency, while those drifting toward the upper-right achieve marginally higher accuracy at steep computational cost. Across all groups, CNN-based models—particularly PSPNet, FPN, and U-Net—cluster tightly near the efficient frontier, combining lightweight design with competitive accuracy. In contrast, transformer-enhanced architectures such as UPerNet and SegFormer shift rightward on the FLOPs axis, confirming their higher arithmetic overhead despite superior feature representation. DeepLabV3 and DeepLabV3+ occupy an intermediate zone, maintaining balanced performance between computational load and segmentation precision.

These findings collectively emphasize that, beyond a certain point (≈ 10 GFLOPs), the accuracy gains achieved by heavier architectures yield diminishing returns, making lighter models more cost-effective for industrial deployment.

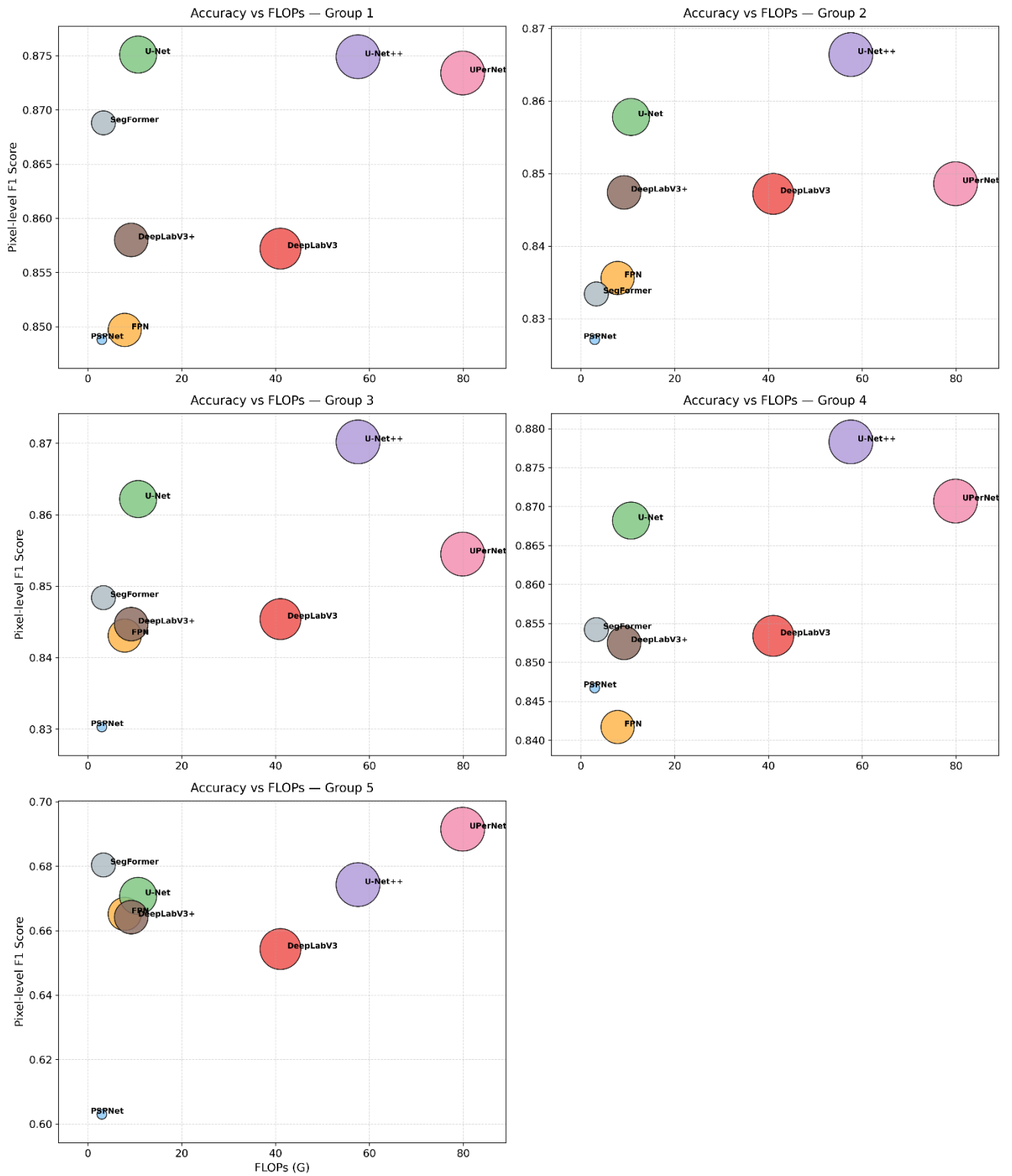


Figure 4-18: Scatter plots showing the trade-off between segmentation accuracy (F1-score) and computational complexity (FLOPs) across Groups 1–5. Bubble sizes denote parameter counts, highlighting how model complexity influences segmentation performance.

4.4.5 Discussion

The comparative evaluation of segmentation accuracy and computational efficiency provides a comprehensive view of each model’s practical suitability for fabric defect inspection. Up to this point, the analyses across sample-level, pixel-level, and region-level metrics have established that UPerNet, U-Net++, and SegFormer consistently achieve the highest segmentation accuracy, while PSPNet and FPN dominate in computational efficiency. The integration of runtime and complexity metrics in Section 4.4 now contextualizes these observations in terms of deployability and real-time feasibility.

From the FLOPs and parameter analysis, it is evident that computational complexity varies substantially among architectures. UPerNet and U-Net++ exhibit the largest parameter counts (exceeding 100M and 45M, respectively) and high FLOP requirements, reflecting their multi-scale fusion and dense skip-connection designs. These characteristics, while enhancing representational richness, also increase inference latency, limiting their use in high-throughput inspection systems. Conversely, PSPNet, FPN, and SegFormer maintain relatively low to moderate FLOPs (3–8 G for PSPNet/FPN and ~3.3 G for SegFormer), signifying compact and efficient computation without severe degradation in accuracy. DeepLabV3 and DeepLabV3+ occupy the middle ground, balancing architectural sophistication with manageable resource demand.

Runtime performance aligns closely with these trends. Models like PSPNet and FPN achieve outstanding inference speeds (above 500 FPS) and extremely low latency (below 3 ms), positioning them as prime candidates for real-time industrial inspection environments. U-Net and DeepLabV3+ provide a balanced compromise, achieving competitive accuracy while maintaining speeds around 350–400 FPS. SegFormer demonstrates remarkable efficiency among transformer-based models—sustaining speeds close to CNN-level performance while delivering near state-of-the-art accuracy—largely owing to its hierarchical attention and token-

reduction mechanisms. UPerNet, in contrast, despite its consistently top accuracy, operates at a substantially lower inference rate (~50 FPS), making it better suited for offline or high-precision inspection where throughput is secondary.

The scatter plots of Accuracy vs Inference Speed and Accuracy vs FLOPs further reinforce these conclusions. A visible Pareto frontier emerges across all groups, where lightweight CNNs such as PSPNet and FPN define one extreme (high speed, moderate accuracy) and transformer-heavy architectures like UPerNet define the other (high accuracy, low speed). SegFormer and U-Net++ occupy the optimal middle ground on this frontier, balancing predictive precision with manageable computational overhead. Importantly, beyond approximately 10 GFLOPs, accuracy improvements diminish significantly, indicating diminishing returns for further architectural complexity. This observation underlines the critical importance of selecting models based not only on accuracy but also on computational scalability and deployment context.

In summary, the efficiency analysis demonstrates that the most practically viable segmentation models for fabric defect localization depend on the target application. For real-time production systems, PSPNet, FPN, and SegFormer provide the best trade-off between accuracy and speed. For offline quality control or high-precision defect mapping, UPerNet, U-Net++, and DeepLabV3+ remain the most effective due to their superior segmentation fidelity. These findings collectively emphasize that no single architecture universally dominates—instead, an application-aware balance between accuracy, speed, and complexity is essential for effective industrial deployment.

Chapter 5 Conclusion and Future Direction

5.1 Conclusion

This thesis presented a comprehensive comparative analysis of supervised semantic segmentation and unsupervised anomaly detection models for fabric defect localization on the ZJU-Leaper dataset. A total of nine models were examined—eight supervised architectures (PSPNet, FPN, U-Net, U-Net++, DeepLabV3, DeepLabV3+, UPerNet, and SegFormer) and one unsupervised model (EfficientAD). The evaluation was performed across multiple levels, including sample-level, pixel-level, and region-level analyses (further categorized by defect size). Quantitative and qualitative assessments were conducted to determine the relative performance, robustness, and efficiency of each model.

The results demonstrated that supervised segmentation models consistently outperform the unsupervised EfficientAD in terms of accuracy and localization precision. Among the supervised models, UPerNet and U-Net++ achieved the most balanced overall performance, maintaining strong F1-scores across pixel and region levels while preserving reasonable computational efficiency. When comparing CNN-based and Transformer-based architectures, distinct performance patterns were observed. CNN models such as U-Net, FPN, and DeepLabV3+ delivered faster inference speeds and lower computational costs, making them well-suited for real-time inspection scenarios. In contrast, Transformer-based architectures like UPerNet and SegFormer demonstrated superior feature representation and achieved higher accuracy, particularly for complex textures and irregular defect boundaries. This indicates that Transformers possess enhanced contextual understanding at the expense of computational overhead, whereas CNNs offer practical efficiency advantages in industrial deployment.

Region-level evaluation provided deeper insights into the detection behavior of models across

different defect sizes. The findings revealed that most models perform exceptionally well on large defects, moderately well on medium-sized defects, and comparatively weaker on small defect regions. This decline in accuracy for smaller defects highlights a fundamental limitation of existing architectures in capturing fine-grained spatial irregularities and subtle defect patterns. Therefore, enhancing model sensitivity to small-scale anomalies remains an important research direction.

In addition to accuracy, efficiency metrics such as FLOPs, parameter count, and inference time were analyzed to assess the practicality of each model. PSPNet and FPN were found to be the most computationally efficient, while Transformer-based models exhibited higher accuracy at increased cost. The comparative analysis illustrated that U-Net++ and DeepLabV3+ achieved an optimal balance between segmentation quality and efficiency, indicating their suitability for real-time industrial applications. Scatter plot analyses further reinforced this finding, with these models occupying the ideal trade-off region between inference speed and predictive precision.

Overall, this thesis establishes a robust benchmarking framework for evaluating segmentation and anomaly detection models for fabric defect localization. It demonstrates that while unsupervised methods such as EfficientAD provide significant label efficiency, their accuracy and localization reliability remain inferior to supervised models. Supervised semantic segmentation, particularly CNN and hybrid Transformer-based approaches, continues to represent the most effective strategy for high-precision automated fabric inspection in industrial settings.

5.2 Future Work

Our future work will focus on several key directions to further enhance model accuracy, efficiency, and practical deployability for automated fabric inspection systems.

First, our future work will aim to develop hybrid or semi-supervised learning frameworks that integrate the strengths of both supervised and unsupervised paradigms. By combining feature extraction capabilities from pre-trained anomaly detection modules with segmentation-based fine-tuning, we intend to minimize dependence on large annotated datasets while retaining localization accuracy.

Our future work will also address the persistent challenge of detecting small-scale and subtle defects, which remain difficult for both CNN and Transformer architectures. We plan to explore multi-scale feature fusion strategies, adaptive attention mechanisms, and high-resolution training pipelines that emphasize fine structural details to improve sensitivity to small defect regions.

Furthermore, our future work will include the design and evaluation of CNN–Transformer hybrid architectures that combine convolutional inductive bias with the long-range contextual reasoning of Transformers. Such models may achieve superior trade-offs between computational cost and accuracy, especially for complex, high-texture fabrics. These architectures will be optimized to exploit the localized detail-capturing ability of CNNs while leveraging the global attention mechanisms of Transformer encoders.

Another aspect of our future work will focus on multi-criteria decision-making (MCDM) techniques such as the Analytic Hierarchy Process (AHP) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to support systematic model selection. This will enable evaluation based on a combination of accuracy, inference time, and hardware constraints, thereby facilitating deployment-oriented optimization.

We also plan to extend our future work toward cross-dataset and real-world industrial validation by testing the trained models on production-line imagery. Such validation will assess the models’ adaptability to variations in lighting, resolution, and texture complexity, ensuring

their robustness beyond benchmark datasets.

Finally, our future work will investigate deployment optimization techniques such as model pruning, quantization, and low-bit computation to enable efficient real-time inference on edge devices. These strategies will help integrate defect detection systems into existing industrial inspection pipelines with minimal latency and resource consumption.

References

- [1] Deageon Kim and Dongoun Lee, “Fabric Defect Detection Based on Artificial Intelligence,” *International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING*, vol. 11, no. 5s, pp. 1–7, 2023.
- [2] Y. Xu, R. Quan, W. Xu, Y. Huang, X. Chen, and F. Liu, “Advances in Medical Image Segmentation: A Comprehensive Review of Traditional, Deep Learning and Hybrid Approaches,” *Bioengineering*, vol. 11, no. 10, p. 1034, Oct. 2024, doi: 10.3390/bioengineering11101034.
- [3] R. Usamentiaga, D. G. Lema, O. D. Pedrayes, and D. F. Garcia, “Automated surface defect detection in metals: a comparative review of object detection and semantic segmentation using deep learning,” in *2021 IEEE Industry Applications Society Annual Meeting (IAS)*, Vancouver, BC, Canada: IEEE, Oct. 2021, pp. 1–8. doi: 10.1109/IAS48185.2021.9677231.
- [4] C. Zhang, S. Feng, X. Wang, and Y. Wang, “ZJU-Leaper: A Benchmark Dataset for Fabric Defect Detection and a Comparative Study,” *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 3, pp. 219–232, Dec. 2020, doi: 10.1109/TAI.2021.3057027.
- [5] G. Menghani, “Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–37, Dec. 2023, doi: 10.1145/3578938.
- [6] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

- [7] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, July 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.
- [8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid Scene Parsing Network,” 2017, pp. 2881–2890. Accessed: Oct. 23, 2025. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html
- [9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.
- [10] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv, June 2021. doi: 10.48550/arXiv.2010.11929.
- [11] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified Perceptual Parsing for Scene Understanding,” 2018, pp. 418–434. Accessed: Oct. 23, 2025. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Tete_Xiao_Unified_Perceptual_Parsing_ECCV_2018_paper.html
- [12] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 12077–12090. Accessed: Oct. 23, 2025. [Online]. Available: <https://proceedings.neurips.cc/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html>

- [13] Y. Cui, Z. Liu, and S. Lian, “A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images,” *IEEE Access*, vol. 11, pp. 55297–55315, 2023, doi: 10.1109/ACCESS.2023.3282993.
- [14] D. Gong *et al.*, “Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 1705–1714. doi: 10.1109/ICCV.2019.00179.
- [15] T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization,” in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., Cham: Springer International Publishing, 2021, pp. 475–489. doi: 10.1007/978-3-030-68799-1_35.
- [16] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 9584–9592. doi: 10.1109/CVPR.2019.00982.
- [17] J. Santos, T. Tran, and O. Rippel, “Optimizing PatchCore for Few/many-shot Anomaly Detection.” arXiv, July 2023. doi: 10.48550/arXiv.2307.10792.
- [18] D. Gudovskiy, S. Ishizaka, and K. Kozuka, “CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows.” arXiv, July 2021. doi: 10.48550/arXiv.2107.12571.
- [19] K. Batzner, L. Heckler, and R. König, “EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies.” arXiv, Feb. 2024. doi: 10.48550/arXiv.2303.14535.

- [20] S. Sugawara and R. Imamura, “PUAD: Frustratingly Simple Method for Robust Anomaly Detection.” arXiv, Feb. 2024. doi: 10.48550/arXiv.2402.15143.
- [21] “Pyramid Pooling Modules - an overview \textbar ScienceDirect Topics.” Accessed: Oct. 25, 2025. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/pyramid-pooling-modules>
- [22] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. S. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds., Cham: Springer International Publishing, 2018, pp. 3–11. doi: 10.1007/978-3-030-00889-5_1.
- [23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” 2018, pp. 801–818. Accessed: Oct. 23, 2025. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.html
- [24] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, “Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies,” in *Product-Focused Software Process Improvement*, M. Morisio, M. Torchiano, and A. Jedlitschka, Eds., Cham: Springer International Publishing, 2020, pp. 202–216. doi: 10.1007/978-3-030-64148-1_13.
- [25] X. Zhao, H. Zhang, C. Song, H. Li, and H. Guo, “Lightweight intelligent detection algorithm for surface defects in printed circuit board,” *Computers & Industrial Engineering*, vol. 203, p. 111030, May 2025, doi: 10.1016/j.cie.2025.111030.

- [26] M. Capra, B. Bussolino, A. Marchisio, M. Shafique, G. Masera, and M. Martina, “An Updated Survey of Efficient Hardware Architectures for Accelerating Deep Convolutional Neural Networks,” *Future Internet*, vol. 12, no. 7, p. 113, July 2020, doi: 10.3390/fi12070113.
- [27] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 2015. doi: 10.48550/arXiv.1409.1556.
- [28] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 2015. doi: 10.48550/arXiv.1409.1556.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 2015. doi: 10.48550/arXiv.1512.03385.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks.” arXiv, July 2016. doi: 10.48550/arXiv.1603.05027.
- [31] Z. Liu *et al.*, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.” arXiv, Aug. 2021. doi: 10.48550/arXiv.2103.14030.
- [32] I. Tolstikhin *et al.*, “MLP-Mixer: An all-MLP Architecture for Vision.” arXiv, June 2021. doi: 10.48550/arXiv.2105.01601.
- [33]. Usha Ruby Dr.A, “Binary cross entropy with deep learning technique for Image classification,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020, doi: 10.30534/ijatcse/2020/175942020.
- [34] S. Jadon, “A survey of loss functions for semantic segmentation,” in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Oct. 2020, pp. 1–7. doi: 10.1109/CIBCB48159.2020.9277638.
- [35] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 2017. doi: 10.48550/arXiv.1412.6980.

- [36] H. Mostafa and X. Wang, “Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization,” in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 4646–4655. Accessed: Oct. 25, 2025. [Online]. Available: <https://proceedings.mlr.press/v97/mostafa19a.html>
- [37] J. Hestness, N. Ardalani, and G. Diamos, “Beyond human-level accuracy: computational challenges in deep learning,” in *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*, Washington District of Columbia: ACM, Feb. 2019, pp. 1–14. doi: 10.1145/3293883.3295710.
- [38] Md. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, “Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review,” *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, Jan. 2023, doi: 10.1109/JPROC.2022.3226481.
- [39] D. Velasco-Montero, J. Fernández-Bemi, R. Carmona-Gálán, and A. Rodríguez-Vázquez, “On the Correlation of CNN Performance and Hardware Metrics for Visual Inference on a Low-Cost CPU-based Platform,” in *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, June 2019, pp. 249–254. doi: 10.1109/IWSSIP.2019.8787329.
- [40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.