Object Clustering using Graphs

Bachelor Thesis Presentation

Kerem Aslan

Student Number: 300206013

Advisor: Dr.-Ing. Aylin Taştan

Department of Electrical and Electronics Engineering
Izmir Institute of Technology

8 January 2026

Outline

Outline

Introduction and Motivation

- **Global Waste Challenge:**
  - Waste generation: **3.40 billion tonnes** by 2050 [1].
  - Recycling success depends on accurate separation [2].
- **Limitations of Manual Sorting:**
  - Labor-intensive, slow, expensive.
  - Prone to human error; hazardous conditions.
- **The Automation Advantage:**
  - ML & CV offer scalable, consistent sorting [3].

Manual Sorting (Ignácio Costa, CC BY 3.0)

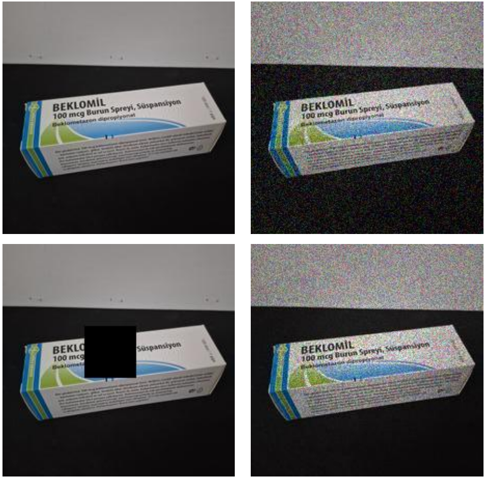
Rise of the Recycling Robots (Forbes, 2020)

State-of-the-Art

- **Visual Representation:**
  - Vision Transformers (ViT) [4]
  - Self-Supervised Learning (DINOv3) [5]

- **Clustering Paradigms:**
  - **Deep Clustering** (DEC, Contrastive Clustering) [6]
    - + Powerful representations
    - − Resource heavy, "black box"
  - **Sparse Graph Learning** (SSC, BDR) [7] ← *This Project*
    - + Interpretable, robust to noise
    - − High computational complexity

1. **High-Dimensional Feature Spaces:**
   - Modern features (e.g., ViT) are high-dimensional.
   - "Curse of Dimensionality" affects distance metrics.
2. **Noise and Domain Shift:**
   - Varying lighting, occlusions, and clutter.
   - Gap between controlled and real environments.
3. **Unsupervised Operation:**
   - No labeled data during training.



Real-World Challenges:
Clean vs. Noisy

Creative Solution Development

Alternative 1: Descriptor + Standard Graph Clustering (Baseline)

- **Method:** Handcrafted features (e.g., HOG) + K-NN Graph + Spectral Clustering.
- **Evaluation:** Low computational cost, poor robustness to noise and variations.

Alternative 2: Graph Neural Networks (GNNs)

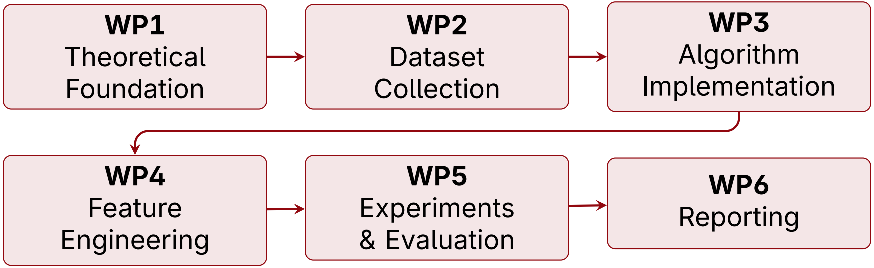
- **Method:** End-to-end deep learning on graph structures.
- **Evaluation:** High accuracy, requires large training data and significant computational resources.

Creative Solution Development

Alternative 3: Sparse Graph Learning (Adopted)

- **Method:** Learn graph structure via **Sparse Graph Representation** algorithms.
- **Justification:**
  - **Technical Feasibility:** Mathematically robust to noise and outliers.
  - **Efficiency:** More computationally-efficient than deep learning.
  - **Sustainability:** Aligns with green computing goals.

Project Management: Work Packages



Risk Analysis & Economic Feasibility

Risk $\rightarrow$ Mitigation

- Overfitting $\rightarrow$ Real-world dataset
- Feature insufficiency $\rightarrow$ SIFT-based descriptors
- Complexity $\rightarrow$ ADMM solver
- Hyperparameters $\rightarrow$ Grid search

Economic Feasibility

- ✓ Low cost (software-centric)
- ✓ Standard hardware
- ✓ Scalable automation

Engineering Standards & Design Constraints

Engineering Standards

- **ISO 9001** [8]
  → Quality & Reproducibility
- **ISO 14001** [9]
  → Energy-efficient methods

Design Constraints

- Hardware → Standard resources
- Scalability → Time monitoring
- Modularity → Flexible architecture

Proposed Methodology Overview

Unsupervised Image Clustering Pipeline

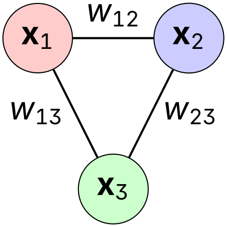
Dataset Input → Feature Extraction → Affinity Learning → Spectral Clustering → Validation

Details will be explained in the following sections.

Outline

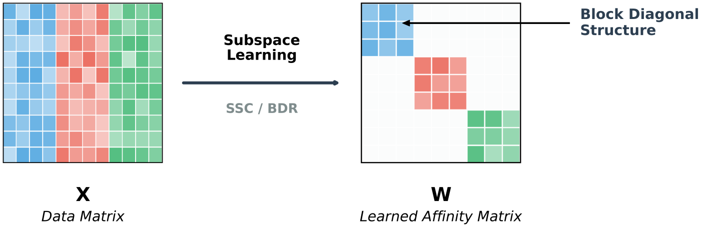
Graph-Based Clustering Basics

- **Notation:**
  - $\mathbf{x}_i \in \mathbb{R}^d$: Feature vector of image i
  - $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$: Data matrix
- **Graph:** $G = (V, E)$ [10]
  - V: Data points (Images)
  - E: Edges with weights w_{ij}



Graph: Nodes = feature vectors
Edges = similarity weights

Affinity Matrix Construction



Goal:

- Learn an affinity matrix **W** that reveals the underlying relationships.
- **Subspace Clustering** uses self-expressiveness to find global structure.

Ideal Structure:

- We aim for a **Block-Diagonal Matrix.**
- $\mathbf{W}_{ij} \neq 0$ only if $\mathbf{x}_i$ and $\mathbf{x}_j$ form the same subspace.
- Ensures perfect cluster separation in the subsequent spectral clustering step.

Sparse Graph Learning: SSC & BDR

SSC [7]

How it works:

- Uses ℓ_1-norm $\rightarrow$ **sparsity**
- Few neighbors per point
- Solved via **ADMM**

Key property:

- Sparse connections
- Block diagonal *indirectly*

BDR [11]

How it works:

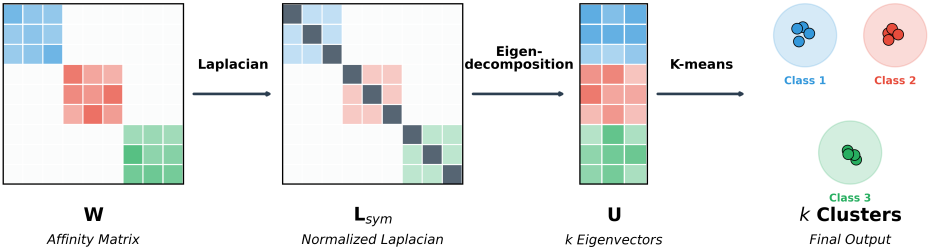
- Enforces k-**block diagonal**
- Block diagonal rank regularizer
- Solved via **Alt. Minimization**

Key property:

- Explicit cluster constraint
- Block diagonal *directly*

Main Difference: SSC achieves block diagonal *implicitly* via sparsity; BDR enforces it *explicitly* via rank constraint.

Spectral Clustering Steps



Pipeline:

1. Compute Laplacian: $\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
2. Find k smallest eigenvectors $\rightarrow$ Embedding $\mathbf{U}$
3. Apply K-means on rows of $\mathbf{U}$ [12]

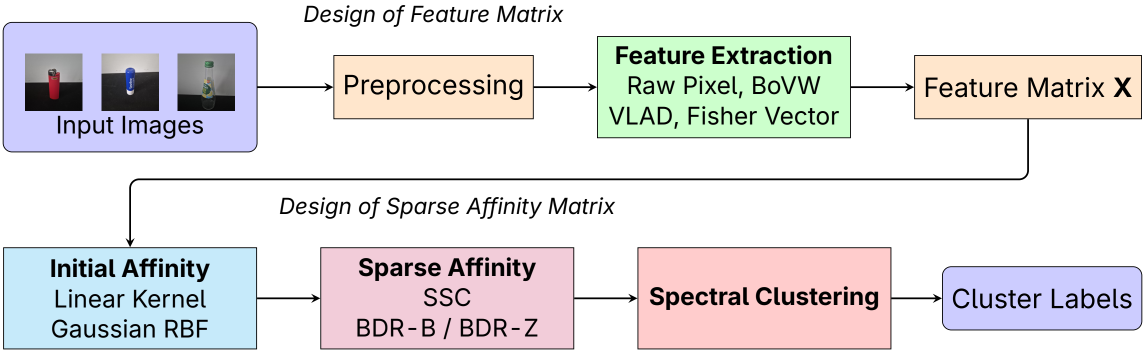
Why Spectral Clustering?

- Separates non-convex clusters
- Exploits graph structure
- Works with any similarity metric

Outline

1. Introduction and Motivation
2. Theoretical Information
3. **Proposed Method**
4. Experimental Results
5. Conclusion

Proposed Pipeline



Design of Feature Matrix:

- Transform images into feature vectors
- 4 extraction methods compared

Design of Sparse Affinity Matrix:

- 2 similarity measures
- 2 clustering algorithms (SSC, BDR)

Dataset Design

Primary Dataset (Controlled):

- 533 images, 16 object categories
- Clean white/black background
- Consistent lighting

Real-World Dataset:

- 419 images, same 16 categories
- Variable backgrounds & lighting
- Used to test robustness

Controlled

Real-World

Image Preprocessing & Raw Pixel Baseline

Preprocessing Steps

1. **Resize:** 224×224 (Lanczos-3)
2. **Grayscale:** $Y = 0.299R + 0.587G + 0.114B$
3. **Normalize:** Scale to $[0, 1]$

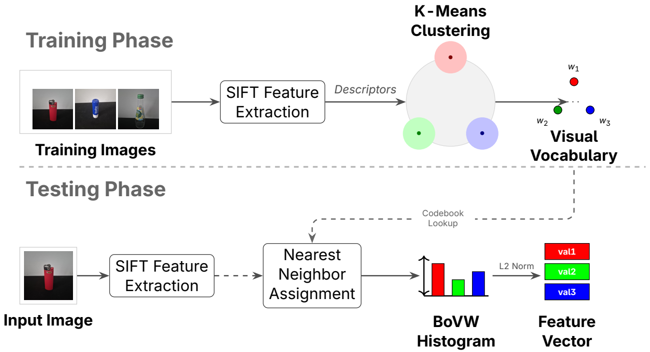
Raw Pixel (Baseline)

- Direct vectorization of preprocessed image
- **Dimension:** 50176 (224×224)

Original (RGB)

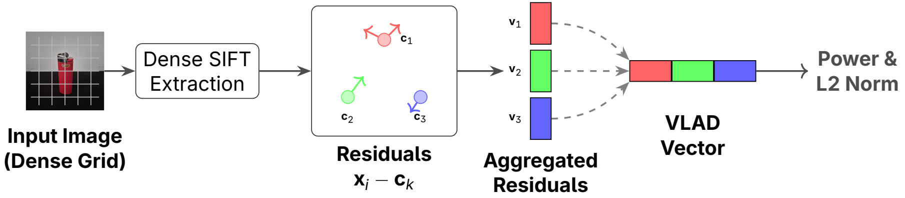
Preprocessed (224×224, Gray)

Feature Extraction: Bag-of-Visual-Words

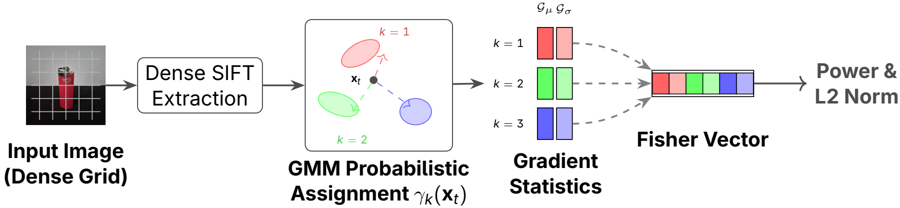


- **Sparse SIFT** features detected and extracted [16].
- **Visual Vocabulary (Dimension):** ($K = 800$) learned via K-Means [13].
- **Histogram Encoding:** Hard assignment of descriptors to nearest words.

Feature Extraction: VLAD

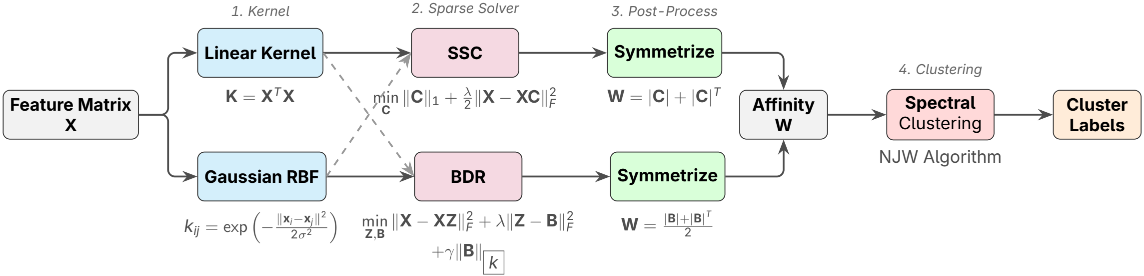


- **Vector of Locally Aggregated Descriptors (VLAD)** [14]
- **Dense SIFT** extraction on a regular grid.
- Accumulates **residuals** $(\mathbf{x}_i - \mathbf{c}_k)$ for $K = 64$ clusters.
- Captures first-order statistics.
- Dimension: 8192 (64×128).



- **Fisher Vector (FV)** [17]
- Models feature distribution with **Gaussian Mixture Model (GMM)** ($K = 64$).
- Encodes gradients w.r.t mean & variance.
- Captures first & second-order statistics.
- Highest dimension: 16384.

Sparse Affinity & Spectral Clustering



Method Summary

16 Combinations Systematically Evaluated

4 Features $\times$ 2 Kernels $\times$ 2 Algorithms

Features

- Raw Pixel
- BoVW
- VLAD
- Fisher Vector

Kernels

- Linear (Cosine)
- Gaussian RBF

Algorithms

- SSC
- BDR

Outline

Experimental Overview

Objective

Systematically evaluate SSC and BDR algorithms under varying data conditions.

Setting 1: Real-World Replacement

- Replace controlled with real-world images
- Tests domain shift robustness
- *(See thesis for details)*

Setting 2: Noise Robustness *Focus*

- Apply synthetic noise types
- Gaussian, Speckle, Occlusion
- Tests algorithmic resilience

*This presentation focuses on **Setting 2: Noise Robustness**.*

Setting 2: Noise Robustness

Clean

Gaussian

Speckle

Occlusion

Gaussian

- Additive ($\sigma = 0.1$)
- Sensor/thermal noise

Speckle

- Multiplicative (var=0.03)
- Imaging artifacts

Occlusion

- 50×50 black block
- Structural loss

Setting 2: Experimental Setup

Parameter Search Space

Algo	Param	Values
SSC	λ	$\{115, 120, 125\}$
	Affine	{F, T}
	ρ	$\{0.55, 0.65, 0.75\}$
BDR	λ	$\{15, 17, 19\}$
	γ	$\{0.95, 1.0\}$
	ρ	$\{0.35, 0.4, 0.45\}$
Euc.	σ	$\{0.5, 1.0\}$

Total Configurations

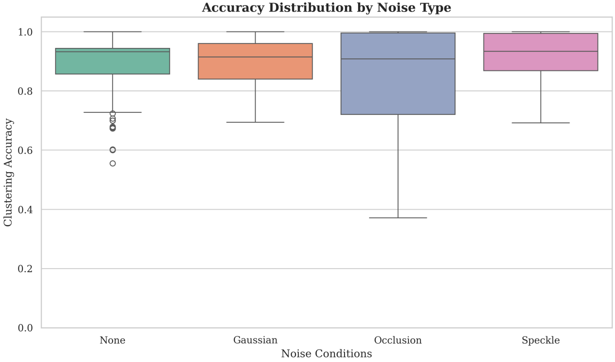
- **Features:** 4 (Raw, BoVW, VLAD, FV)
- **Algorithms:** 4 (SSC, BDR-B, BDR-Z $\times$ 2 sim.)
- **Params per algo:** 18–36
- **Noise types:** 4 (Clean + 3)

Total runs: $\sim$2,300

Success Rate Definition

% of configs achieving
perfect accuracy (1.0)

Setting 2: Accuracy Distribution Under Noise

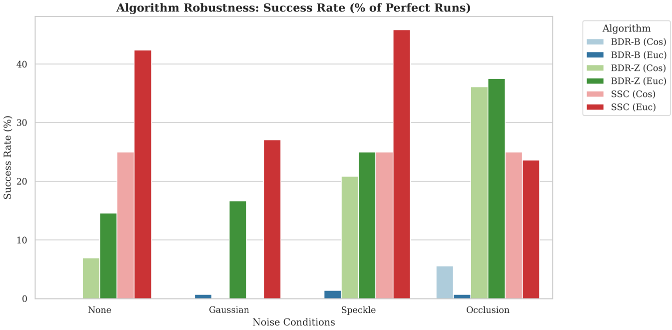


Noise	Acc.	Success
Clean	0.90	16.2%
Gaussian	0.90	9.9%
Speckle	0.92	21.1%
Occlusion	0.82	21.1%

Observations:

- Gaussian: lowest success (9.9%)
- Speckle: highest success (21.1%)
- Occlusion: high variance (std=0.21)

Setting 2: Algorithm Performance Under Noise



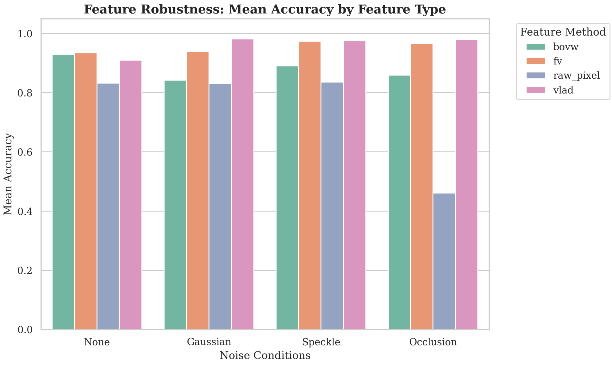
Best per Condition:

- Clean: SSC (Euc) 42%
- Gaussian: SSC (Euc) 27%
- Speckle: SSC (Euc) 46%
- Occlusion: BDR-Z 38%

Key Insights:

- Euclidean dominates
- SSC (Cos) 0% Gaussian
- BDR-B $<6\%$ everywhere

Setting 2: Feature Performance Under Noise



Noise	Best	Acc.
Clean	FV	0.935
Gaussian	**VLAD**	**0.982**
Speckle	**VLAD**	**0.975**
Occlusion	**VLAD**	**0.979**

VLAD: best under all noise

Setting 2: Summary

Algorithm Findings

- **SSC (Euc)**: Best for general noise
- **BDR-Z (Euc)**: Best for occlusion
- Gaussian RBF outperforms Cos
- Avoid BDR-B

Feature Findings

- **VLAD**: Most robust (0.98)
- **FV**: Best on clean data
- **BoVW**: Moderate (0.84-0.93)
- **Raw Pixel**: Fails under occlusion

Recommendation

VLAD + SSC (Euc) for noise-robust clustering
VLAD + BDR-Z (Euc) for occlusion scenarios

Key Experimental Findings

Feature Extraction

- **VLAD** is most robust
- Near-perfect under noise (0.98)
- Best for unconstrained environments

Algorithm Selection

- **SSC (Euc)**: General purpose
- **BDR-Z (Euc)**: Occlusion
- Avoid BDR-B ($<6\%$ success)

Similarity Measure

- **Euclidean (Gaussian RBF)** dominates
- SSC (Cos) fails under Gaussian noise
- Non-linear mapping aids robustness

Performance vs. Speed

- BDR: <1 sec ($10\times$ faster)
- SSC (Euc): ~ 2 sec
- SSC (Cos): ~ 10 sec

Outline

Alignment with UN SDGs [18]

SDG 9: Industry & Innovation

- Industrial automation for recycling
- Innovation for recycling facilities
- Upgrading waste management tech

SDG 12: Responsible Consumption

- Improving waste sorting efficiency
- Enhancing recycling accuracy
- Sustainable production patterns

Summary of Findings

Best Configuration

- **VLAD** + **SSC** (Gaussian RBF)
- Near-perfect accuracy on clean data
- Lowest degradation under domain shift (42%)

Feature Comparison

- VLAD outperforms BoVW, FV, Raw Pixels
- Residual aggregation preserves discriminative info

Algorithm Selection

- **SSC** (Gaussian RBF): General noise
- **BDR-Z** (Gaussian RBF): Occlusion
- Gaussian RBF outperforms Linear kernel

Key Insight

Selecting the right feature, similarity measure, and hyperparameters is key to robust clustering.

Future Work

Deep Feature Integration

- Modern CNNs (ResNet [19])
- Vision Transformers (ViT [4])
- Self-supervised (DINOv3 [5])

Multi-Object Scenes

- Handle cluttered images
- Integration with detection/segmentation
- Real-world scenarios

Adaptive Parameters

- Auto-tuning parameters
- Meta-learning from data statistics
- Online adaptation mechanisms

Scalability

- GPU/NPU acceleration
- Approximate nearest neighbor methods

Source Code

The source code for this project will be publicly available at:

`github.com/Kerem-Aslan/object-clustering-graphs/`

Thank you for your attention.

Any questions?

References I

[1] S. Kaza, L. C. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050.* World Bank Publications, Aug. 2018.

[2] N. Ferronato, V. Torretta, N. Ferronato, and V. Torretta, “Waste Mismanagement in Developing Countries: A Review of Global Issues,” *International Journal of Environmental Research and Public Health*, vol. 16, Mar. 2019.

[3] M. Abdallah, M. Abu Talib, S. Feroz, Q. Nasir, H. Abdalla, and B. Mahfood, “Artificial intelligence applications in solid waste management: A systematic research review,” *Waste Management*, vol. 109, pp. 231–246, May 2020.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” June 2021.

[5] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski, “DINOv3,” Aug. 2025.

References II

[6] H. Hou, S. Ding, C. Gui Cao, X. Xu, L. Guo, and X. Li, “A comprehensive survey of image clustering based on deep learning,” *Pattern Recognition*, vol. 172, p. 112590, Apr. 2026.

[7] E. Elhamifar and R. Vidal, “Sparse Subspace Clustering: Algorithm, Theory, and Applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2765–2781, Nov. 2013.

[8] “ISO 9001:2015 — Quality management systems — Requirements,” 2015.

[9] “ISO 14001:2015 — Environmental management systems — Requirements with guidance for use,” 2015.

[10] F. R. K. Chung, *Spectral Graph Theory*.
American Mathematical Soc., 1997.

[11] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, “Subspace Clustering by Block Diagonal Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 487–501, Feb. 2019.

[12] A. Ng, M. Jordan, and Y. Weiss, “On Spectral Clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems*, vol. 14, MIT Press, 2001.

[13] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” *Work Stat Learn Comput Vision, ECCV*, vol. Vol. 1, Jan. 2004.

References III

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, June 2010.

[15] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image Classification with the Fisher Vector: Theory and Practice," *International Journal of Computer Vision*, vol. 105, pp. 222–245, Dec. 2013.

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004.

[17] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[18] United Nations Department of Economic and Social Affairs, *The Sustainable Development Goals Report 2025*.
The Sustainable Development Goals Report, United Nations, July 2025.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015.