Deep Learning-Based Component Detection and Analysis on PCB Cards

Internship Report – ASAP Engineering / HCLTech

SINGARIN-SOLE Livio

Internship Duration: 10 months

Supervisor(s): Muravyeva Ekaterina

May 29, 2025

Contents

1 Introduction **3**

2 Object Detection Pipeline **3**

2.1 Faster R-CNN with ResNet Backbone . . . 3

2.2 Dataset and Classes . . . 4

2.3 Training and Parameter Tuning . . . 4

2.4 Observations . . . 4

3 Multi-Model Fusion and Combination Strategy **4**

3.1 Reasonning . . . 4

3.2 Model Pool and Subclass Grouping . . . 4

3.3 Inference Pipeline . . . 5

3.4 Model Selection and Trade-Offs . . . 5

3.5 Mathematical Formulation of Evaluation Metrics . . . 5

3.6 Experimental Results . . . 6

3.7 Pipeline Overview Example . . . 6

4 Embedding-Based Methods for Component Analysis **7**

4.1 Generating Embeddings with InceptionV3 . . . 7

4.2 Clustering Methods for Unsupervised Classification . . . 7

4.3 Unsupervised Component Classification Pipeline . . . 8

4.4 Parameter Tuning for DBSCAN . . . 8

4.5 KMeans Initialization and Cluster Validation . . . 9

4.6 Use Case 1: Component Classification . . . 9

4.7 Use Case 2: Dataset Cleanup via Duplicate Detection . . . 10

4.8 Use Case 3: Image Matching via Embedding Search . . . 10

5 Discussion and Conclusion **11**

5.1 Technical Contributions Summary . . . 11

5.2 Business Impact and Integration . . . 11

5.3 Future Perspectives and Evolution Paths . . . 11

5.4 Conclusion . . . 12

A Additional Figures and Hyperparameters **13**

1 Introduction

The increasing complexity of modern electronic systems necessitates automated and reliable methods for analyzing printed circuit boards (PCBs). In this context, the PCB-KI project, developed at ASAP Engineering (a subsidiary of HCLTech), explores the use of deep learning techniques for the automatic detection, classification, and analysis of components on PCB images.

A 10-month internship contributed to the development and refinement of several key functionalities within the PCB-KI pipeline. The primary focus areas included:

- Evaluation and tuning of an object detection pipeline based on Faster R-CNN with a ResNet backbone, trained to detect PCB components such as resistors, capacitors, transistors, and integrated circuits.
- Design and implementation of a multi-model fusion approach to enhance detection accuracy by assigning specialized models to subsets of component classes.
- Exploration of image embeddings for use in unsupervised clustering and classification, with applications in error correction, dataset cleaning, and visual matching.

The implementation was conducted using the Python programming language, incorporating major deep learning frameworks such as PyTorch and SAHI (Slicing Aided Hyper Inference). Visualization and analysis of embeddings were facilitated through the use of FiftyOne, while clustering algorithms including DBSCAN and K-Means were applied following dimensionality reduction via UMAP.

This report outlines the technical methodologies employed, the rationale behind design choices, empirical evaluations, and the outcomes of the developed solutions. The proposed approaches were intended to improve the performance of the detection pipeline and to establish more robust and interpretable workflows

2 Object Detection Pipeline

The core of the PCB component detection system is based on the Faster R-CNN architecture [?], which has been widely adopted for object detection tasks due to its balance between accuracy and computational efficiency.

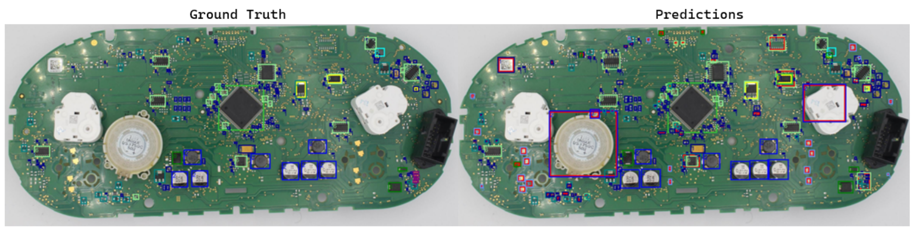

Figure 1: Example of components detections on a pcb-card

2.1 Faster R-CNN with ResNet Backbone

The Faster R-CNN model integrates a Region Proposal Network (RPN) to generate candidate object bounding boxes, followed by a classification and bounding box regression head. A ResNet backbone pretrained on ImageNet is used to extract rich visual features. The backbone choice is critical for achieving good detection performance on diverse PCB components.

2.2 Dataset and Classes

The dataset consisted of PCB card images annotated with bounding boxes for multiple component classes, including resistors, capacitors, transistors, integrated circuits, and others. Each class presented unique challenges due to variations in size, shape, and appearance.

2.3 Training and Parameter Tuning

In addition, The model is implemented using the PyTorch framework, leveraging GPU acceleration for efficient training. To optimize hyperparameters such as learning rate, batch size, and anchor box sizes, the Optuna library [?] is utilized for automated hyperparameter search.

One challenge identified was the sensitivity of the standard mean Average Precision (mAP) metric to small bounding boxes, which are common among certain PCB components. To address this, the mAP calculation range was adjusted to more accurately reflect detection quality in this specific context.

2.4 Observations

Initial experiments indicated that default parameters frequently underperformed on small components, making custom tuning necessary. The use of Optuna to explore the hyperparameter space resulted in improved mAP scores, thereby validating the optimization approach.

This foundational pipeline served as the basis for subsequent enhancements, including multi-model fusion and embedding-based clustering.

3 Multi-Model Fusion and Combination Strategy

3.1 Reasonning

The heterogeneous nature of PCB components—characterized by variations in size, shape, and visual complexity—motivates the use of a multi-model approach to improve detection performance. Components such as resistors and capacitors, which exhibit small and repetitive features, are better suited to models trained with image slicing to enhance fine-detail detection. In contrast, components like transistors and larger elements benefit from models trained on full images, enabling the capture of broader contextual information.

3.2 Model Pool and Subclass Grouping

Within the PCB-KI project, numerous object detection models were trained under varying conditions, including standard Faster R-CNN models trained on whole images and slicing-based models trained on smaller image patches using the SAHI library. To leverage this model diversity, a strategy was explored to combine different models, each specialized in detecting specific subclasses of components.

Concretely, a model $\mathcal{M}_j$ is associated with a subset of component classes $C_j \subset C$. During inference, only detections belonging to C_j are retained from $\mathcal{M}_j$'s output, effectively filtering out irrelevant classes.

3.3 Inference Pipeline

Given an input PCB image I, the inference procedure is:

1. Run each model $\mathcal{M}_j$ on the full image I. Depending on the model, $\mathcal{M}_j$ may perform inference on the whole image resized (normal model) or on multiple sliced patches (slicing model) internally.
2. Filter predictions from $\mathcal{M}_j$ to keep only bounding boxes of classes in C_j.
3. Aggregate filtered predictions from all models:
$$\mathcal{B} = \bigcup_j \mathcal{B}^j,$$
where $\mathcal{B}^j$ are the retained bounding boxes from model $\mathcal{M}_j$.
4. Apply Non-Maximum Suppression (NMS) with a low IoU threshold on $\mathcal{B}$ to remove duplicate detections while preserving complementary predictions.

This fusion strategy exploits the strengths of each model type. Slicing-based models excel at detecting small, fine components, while normal models better handle larger, global structures. Additionally, restricting each model to a subset of classes reduces conflicting predictions and false positives.

3.4 Model Selection and Trade-Offs

Experiments conducted on a large set of stored models indicated that combining two models is sufficient to achieve near-optimal mean Average Precision (mAP) while maintaining manageable inference time. Incorporating additional models was found to increase computational cost disproportionately.

The evaluation involves pairing a slicing-based model specialized in detecting small components (e.g., resistors, capacitors) with a standard model specialized in larger components (e.g., transistors). This configuration resulted in a significant improvement in mAP with only a modest increase in inference time.

3.5 Mathematical Formulation of Evaluation Metrics

Intersection over Union (IoU)

The Intersection over Union (IoU) quantifies the overlap between a predicted bounding box B_p and the ground truth B_{gt}:

$$\mathrm{IoU}(B_p, B_{gt}) = \frac{\mathrm{Area}(B_p \cap B_{gt})}{\mathrm{Area}(B_p \cup B_{gt})}.$$

A detection is considered correct if its IoU exceeds a threshold θ. Due to the small size of many components, we set $\theta = 0.2$ to allow more lenient matching.

Mean Average Precision (mAP)

For each class c, Average Precision (AP) is defined as the area under the precision-recall curve:

$$\mathrm{AP}_c = \int_0^1 p_c(r)\, dr,$$

where $p_c(r)$ is precision at recall r.

The mean Average Precision (mAP) over all C classes is:

$$\mathrm{mAP} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{AP}_c.$$

3.6 Experimental Results

Table 1 summarizes the performance and inference time of the different configurations:

Model Type	**mAP (IoU=0.2)**	**Inference Time (s)**
Normal (single monolithic model)	0.746	2.2
Slicing-based model (SAHI)	0.794	53.4
Combined model (one slicing + one normal)	**0.820**	55.9

Table 1: Performance comparison between different model configurations.

The combined model notably improves mAP compared to the best single slicing model, with only a slight increase in inference time. This validates the efficacy of combining complementary models specialized for different component subclasses.

3.7 Pipeline Overview Example

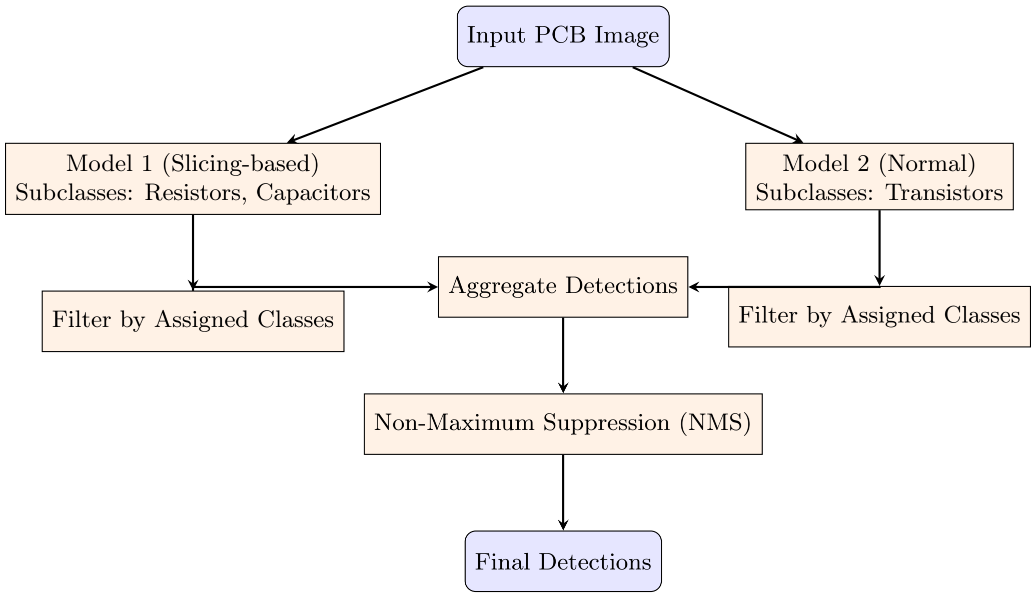
Input PCB Image
Model 1 (Slicing-based)
Subclasses: Resistors, Capacitors
Model 2 (Normal)
Subclasses: Transistors
Filter by Assigned Classes
Aggregate Detections
Filter by Assigned Classes
Non-Maximum Suppression (NMS)
Final Detections

Figure 2: Overview of the multi-model detection pipeline combining slicing-based and normal models specialized for different subclasses.

4 Embedding-Based Methods for Component Analysis

In addition to object detection, visual embeddings were explored for tasks such as component classification, dataset cleanup, and image-based retrieval. These embeddings offer a compact representation of image content that remains robust to noise and geometric variations.

4.1 Generating Embeddings with InceptionV3

Embeddings are computed using the InceptionV3 convolutional neural network [1], pretrained on ImageNet. For performance considerations, the output of the classification layer (a 1000-dimensional vector) was extracted, although the penultimate layer (2048-dimensional) is known to produce more descriptive embeddings. All component images were resized to the required InceptionV3 input dimensions (299×299) and normalized using ImageNet mean and standard deviation.

To enhance embedding consistency under geometric transformations, horizontal or vertical flipping is applied during preprocessing.

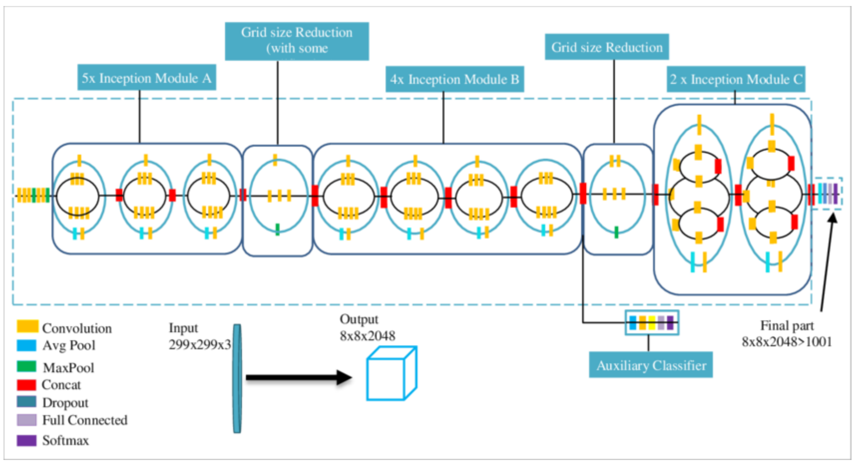

Figure 3: Simplified InceptionV3 architecture. The 1000D output vector is used for embedding; the 2048D vector before classification offers higher performance at increased cost.

4.2 Clustering Methods for Unsupervised Classification

To classify components based on visual similarity, unsupervised clustering was applied to the embeddings. Two main approaches were employed:

- **DBSCAN (Density-Based Spatial Clustering)** [3], which groups components based on density in a lower-dimensional embedding space (after UMAP projection to 3D).
- **KMeans**, a centroid-based clustering method applied directly to the full 1000-dimensional embeddings.

For DBSCAN, dimensionality reduction using UMAP (Uniform Manifold Approximation and Projection) [2] is essential to preserve the local structure and facilitate the detection of arbitrary cluster shapes. KMeans performed better with high-dimensional data when the number of clusters was known or estimated.

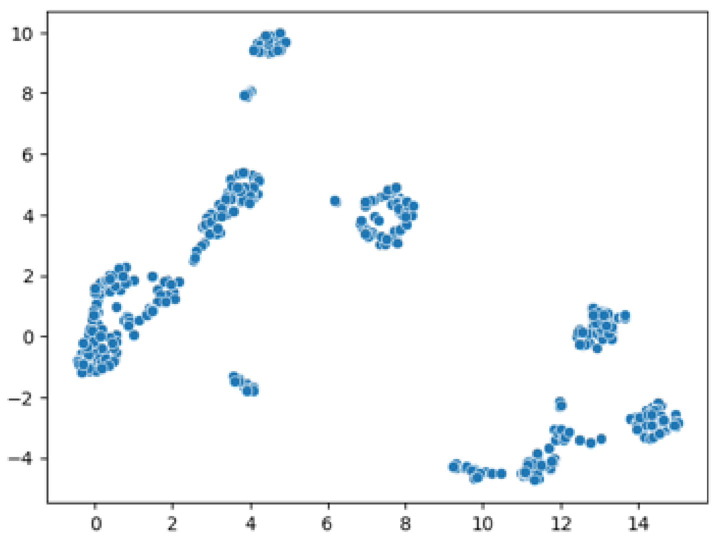

Figure 4: UMAP embeddings for around 500 components

4.3 Unsupervised Component Classification Pipeline

The pipeline for clustering-based component classification is illustrated below:

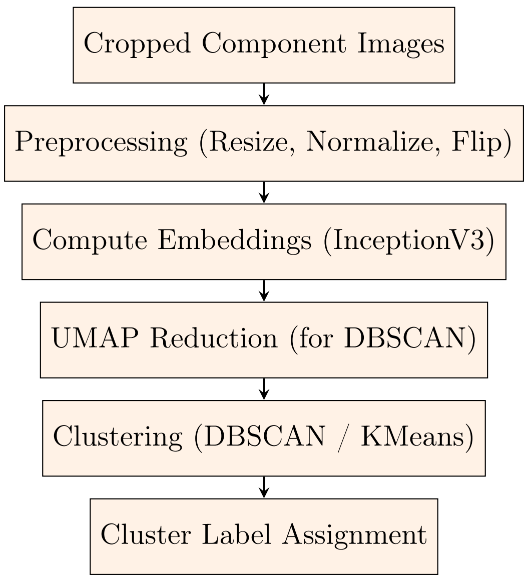
Cropped Component Images
Preprocessing (Resize, Normalize, Flip)
Compute Embeddings (InceptionV3)
UMAP Reduction (for DBSCAN)
Clustering (DBSCAN / KMeans)
Cluster Label Assignment

Figure 5: Pipeline for component classification using embeddings and clustering. UMAP is applied only for DBSCAN.

4.4 Parameter Tuning for DBSCAN

The DBSCAN algorithm requires the choice of an ϵ neighborhood radius, which critically affects clustering results. We propose an empirical heuristic for selecting ϵ, based on the Shannon entropy of pairwise distances between embeddings (after UMAP projection to 3D):

$$H = -\sum_i p_i \log p_i, \quad \epsilon = k \cdot e^{-2H}$$

Here, p_i represents the normalized frequency of distances in histogram bin i, and k is a calibration factor. This approach is based on the intuition that higher entropy (more spread-out data) should lead to smaller ϵ, resulting in tighter clusters. Conversely, lower entropy corresponds to denser data, warranting a larger ϵ.

While this formulation is not perfect, it has shown consistent performance in practice. The constant k is calibrated from known ϵ values on reference PCB boards. An important feature of DBSCAN is that it labels noisy or outlier points with -1, allowing automatic detection of components that do not fit any known cluster.

4.5 KMeans Initialization and Cluster Validation

For KMeans clustering, the number of clusters was estimated using both the elbow method and the silhouette score.[4]:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))},$$

where $a(i)$ is the mean intra-cluster distance for sample i, and $b(i)$ is the mean nearest-cluster distance.

4.6 Use Case 1: Component Classification

Clustering is used to automatically group visually similar components and aid classification. Cluster labels are optionally mapped to known classes or used to identify annotation inconsistencies. This method helped uncover duplicate labels and missing categories in unlabeled PCB boards.

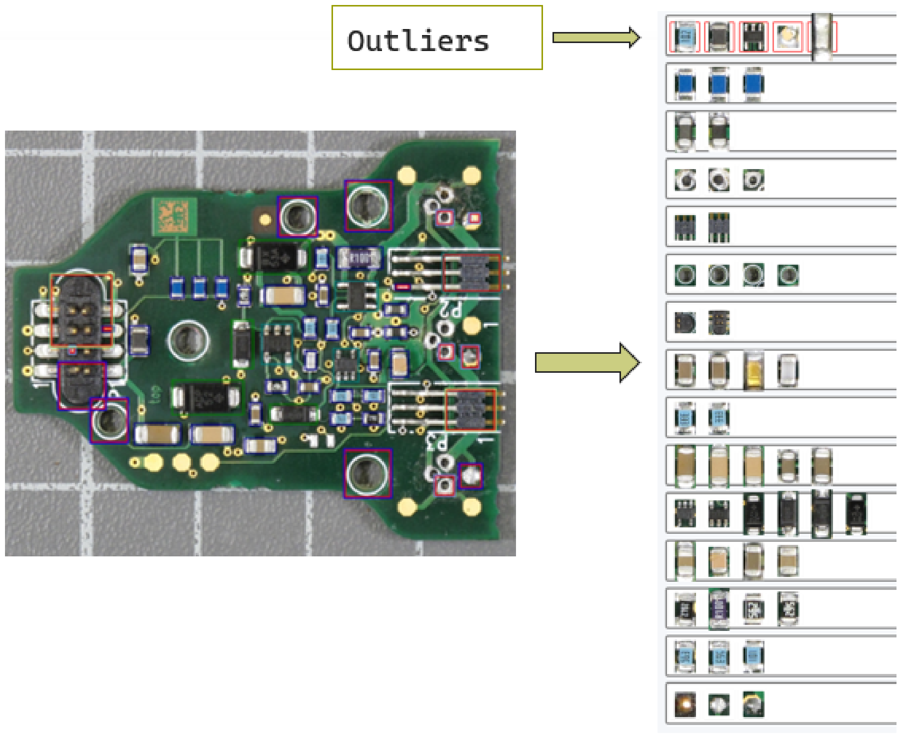

Figure 6: Example of pcb-card components embeddings clustering

4.7 Use Case 2: Dataset Cleanup via Duplicate Detection

A fuzzy variant of DBSCAN is applied [5] to detect duplicate images. Unlike classical DBSCAN, this algorithm allows each image to belong to multiple clusters with a membership score, reflecting uncertainty and allowing overlap. This is effective for identifying near-identical components appearing across boards or in misannotated files.

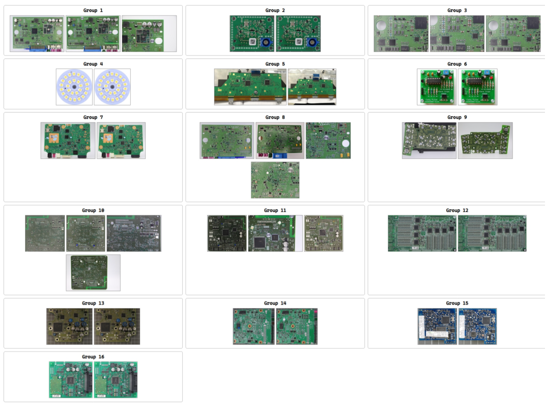

Figure 7: Duplicates found in the current PCB-CARDS Dataset

4.8 Use Case 3: Image Matching via Embedding Search

To match a query image against a set of component embeddings, The simple Euclidean distance to all references is computed:

$$\text{match} = \arg\min_{\mathbf{e}_i \in \mathcal{E}} \|\mathbf{e}_{\text{query}} - \mathbf{e}_i\|_2$$

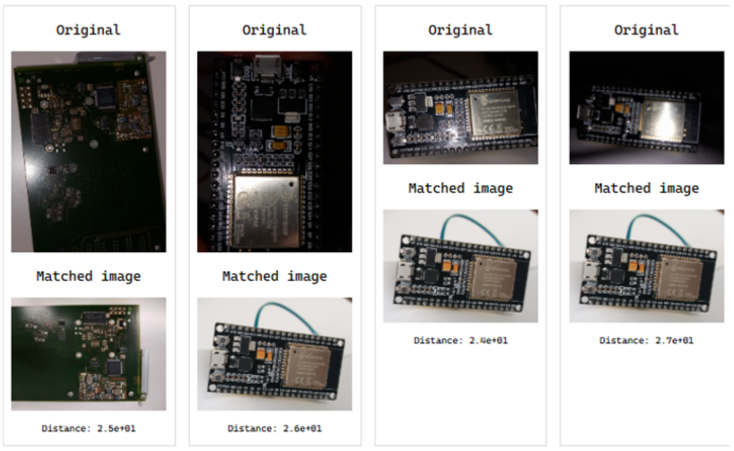

Figure 8: Exemple of Image Matching via Embedding Search

Furthemore, an enhancement is implemented that performs data augmentation (e.g., Gaussian noise, cropping, rotation) on the query image, then selects the closest embedding to any reference:

$$\mathbf{e}^* = \arg\min_j \min_i \|\mathbf{e}_{\text{aug},j} - \mathbf{e}_i^{\text{ref}}\|_2$$

This method mitigates small geometric discrepancies and lighting variations, especially under perspective shifts. It provided a small but consistent improvement in real PCB search scenarios.

5 Discussion and Conclusion

5.1 Technical Contributions Summary

Two main approaches were developed to enhance PCB component detection and classification within the PCB-KI project at ASAP Engineering.

A multi-model fusion strategy was implemented, combining Faster R-CNN models specialized for different component classes. This approach pairs slicing-based models for small components with standard models for larger ones, achieving a mAP of 0.820 at IoU=0.2 compared to 0.746 for the base model.

An embedding-based pipeline using InceptionV3 was developed for unsupervised component analysis. The system implements DBSCAN and KMeans clustering with automatic parameter selection, enabling component classification, dataset cleanup via duplicate detection, and image matching through nearest-neighbor search.

5.2 Business Impact and Integration

The methods developed provided tangible value to ASAP Engineering and its parent company HCL Technologies. The quantifiable improvements achieved include:

- **Detection accuracy enhancement**: 9.9% improvement in mAP (from 0.746 to 0.820)
- **Production system integration**: The embedding-based clustering pipeline for component classification and the image matching module were fully implemented and deployed in the backend production system

These developments strengthen ASAP Engineering's competitive position in the automated PCB inspection market by providing more reliable and efficient solutions to industrial clients. The successful deployment of these systems in production demonstrates their robustness and industrial viability.

5.3 Future Perspectives and Evolution Paths

The successful implementation of the developed methods opens several avenues for further advancement in PCB component analysis:

Technical enhancements:

- Migration to more recent architectures such as Vision Transformers or EfficientNet backbones could yield additional performance gains
- Implementation of contrastive learning methods may improve embedding discriminability for challenging component classes
- Semi-supervised learning approaches could leverage unlabeled PCB data to enhance model generalization

System scalability:

- Extension of the multi-model fusion approach to encompass additional component types and PCB form factors

- Development of adaptive clustering algorithms that automatically adjust to new component categories
- Integration of continuous learning mechanisms to maintain model performance as new PCB designs emerge

Industrial applications:

- Adaptation of the framework to related domains such as flexible PCBs and 3D electronic assemblies
- Development of real-time processing capabilities for inline production inspection
- Creation of standardized APIs for integration with existing quality control systems

5.4 Conclusion

The work conducted represents a significant advancement in automated PCB component detection and analysis. The combination of optimized object detection models with embedding-based classification techniques provides a robust foundation for industrial-scale PCB inspection systems. The successful integration of these methods into ASAP Engineering's production environment demonstrates their practical viability and commercial value.

The multi-faceted approach developed—encompassing model optimization, multi-model fusion, and embedding-based analysis—establishes a comprehensive framework that balances detection accuracy with computational efficiency. This framework positions ASAP Engineering to address evolving challenges in electronic component inspection while maintaining scalability for diverse industrial applications.

References

[1] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

[2] McInnes, L., Healy, J., and Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426.

[3] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*.

[4] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics.*

[5] Fuzzy DBSCAN implementation, GitHub: `https://github.com/yourrepo/fuzzy-dbscan`

Acknowledgments

I would like to sincerely thank the entire R&D team at ASAP Engineering and HCLTech for their support throughout this internship. I am especially grateful to my supervisor Marwitz Andreas for their insightful guidance, regular feedback, and the freedom to explore innovative approaches during this project. Their mentorship was instrumental in shaping both the technical depth and the practical impact of this work.

I would also like to thank Muravyeva Ekaterina for the helpful discussions on embedding-based techniques, and the infrastructure team for maintaining the computing environment that made large-scale experiments feasible.

A Additional Figures and Hyperparameters

RÉSUMÉ

Ce rapport présente les travaux menés dans le cadre du projet PCB-KI d'ASAP Engineering pour l'amélioration de la détection et classification automatique de composants sur cartes de circuits imprimés (PCB) par techniques d'apprentissage profond.

Objectifs: Améliorer la précision du système de détection de composants électroniques (résistances, condensateurs, transistors, circuits intégrés) tout en maintenant des performances d'inférence compatibles avec un usage industriel.

Méthodes développées:

- *Optimisation du modèle de détection:* Mise au point d'un pipeline basé sur Faster R-CNN avec backbone ResNet, optimisé via Optuna pour la détection de composants de petite taille
- *Fusion multi-modèles:* Stratégie combinant modèles spécialisés par classes de composants avec post-traitement par suppression non-maximale (NMS)
- *Analyse par embeddings:* Pipeline complet utilisant InceptionV3 pour la génération d'embeddings visuels, avec clustering non-supervisé (DBSCAN, K-Means) et réduction dimensionnelle (UMAP)

Résultats obtenus:

- Amélioration de 9.9% de la précision moyenne (mAP) par des techniques de combinaison de modèles: de 0.746 à 0.820 à IoU=0.2
- Développement de trois applications pratiques : classification automatique, nettoyage de jeux de données, recherche par similarité visuelle
- Développement de trois applications pratiques : classification automatique à partir d'images de composants électroniques, nettoyage de jeux de données en détectant les doublons et recherche par similarité visuelle

Impact industriel: Les méthodes développées ont été intégrées et déployées dans le système de production d'ASAP Engineering. Le pipeline de clustering et le module de matching d'images sont opérationnels dans l'environnement backend, attestant ainsi de la scalabilité de ces méthodes.

Perspectives: Les travaux ouvrent des voies d'amélioration vers des architectures plus récentes (Vision Transformers), l'apprentissage contrastif, et l'extension à d'autres domaines d'inspection électronique. Le framework développé constitue une base robuste pour l'évolution vers des solutions d'inspection industrielle plus performantes et polyvalentes.

Mots-clés: Apprentissage profond, détection d'objets, Faster R-CNN, embeddings, clustering, PCB, inspection automatisée, vision par ordinateur