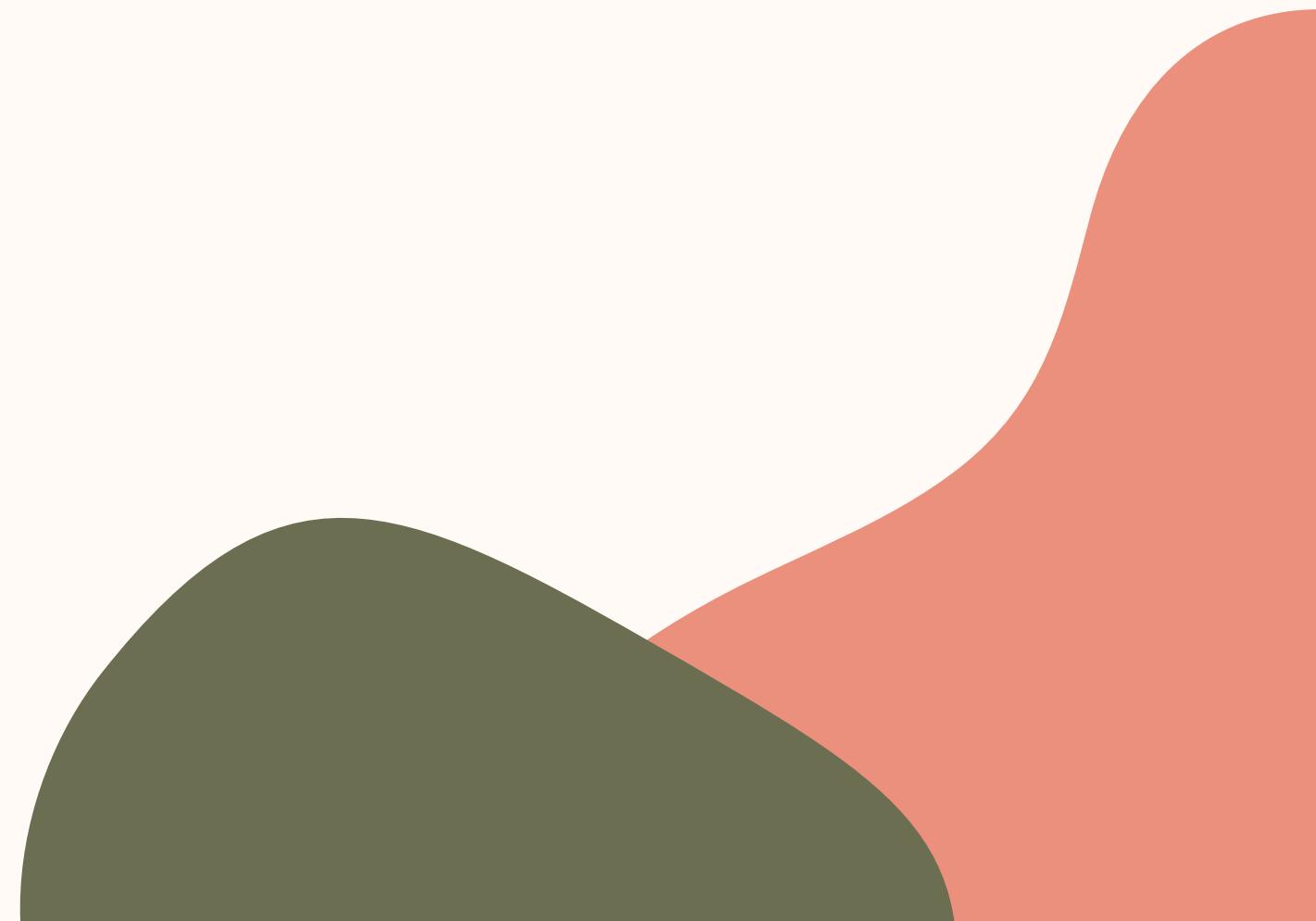

Détection d'anomalies visuelles sur le dataset MVTec AD

JULIAN RABUT &
MARIN JOUFFLINEAU



Contexte et enjeux de la détection d'anomalies

La détection d'anomalies visuelles est un enjeu majeur dans l'industrie :

- Contrôle qualité automatisé (production, emballage, assemblage)
- Réduction des erreurs humaines
- Gain de temps et de fiabilité

Comment détecter automatiquement les défauts visuels sur des produits sans disposer d'exemples d'anomalies pendant l'entraînement ?

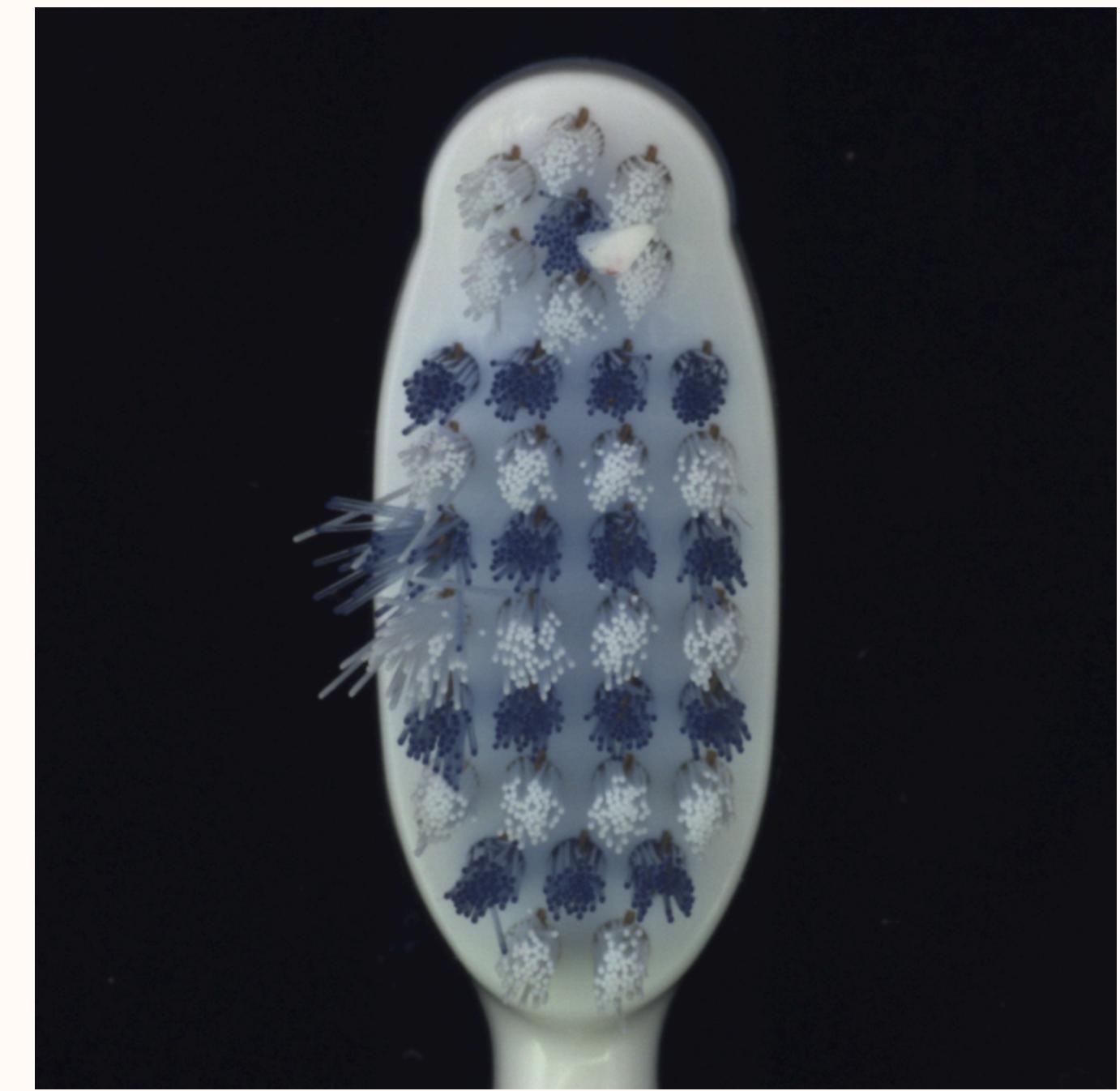
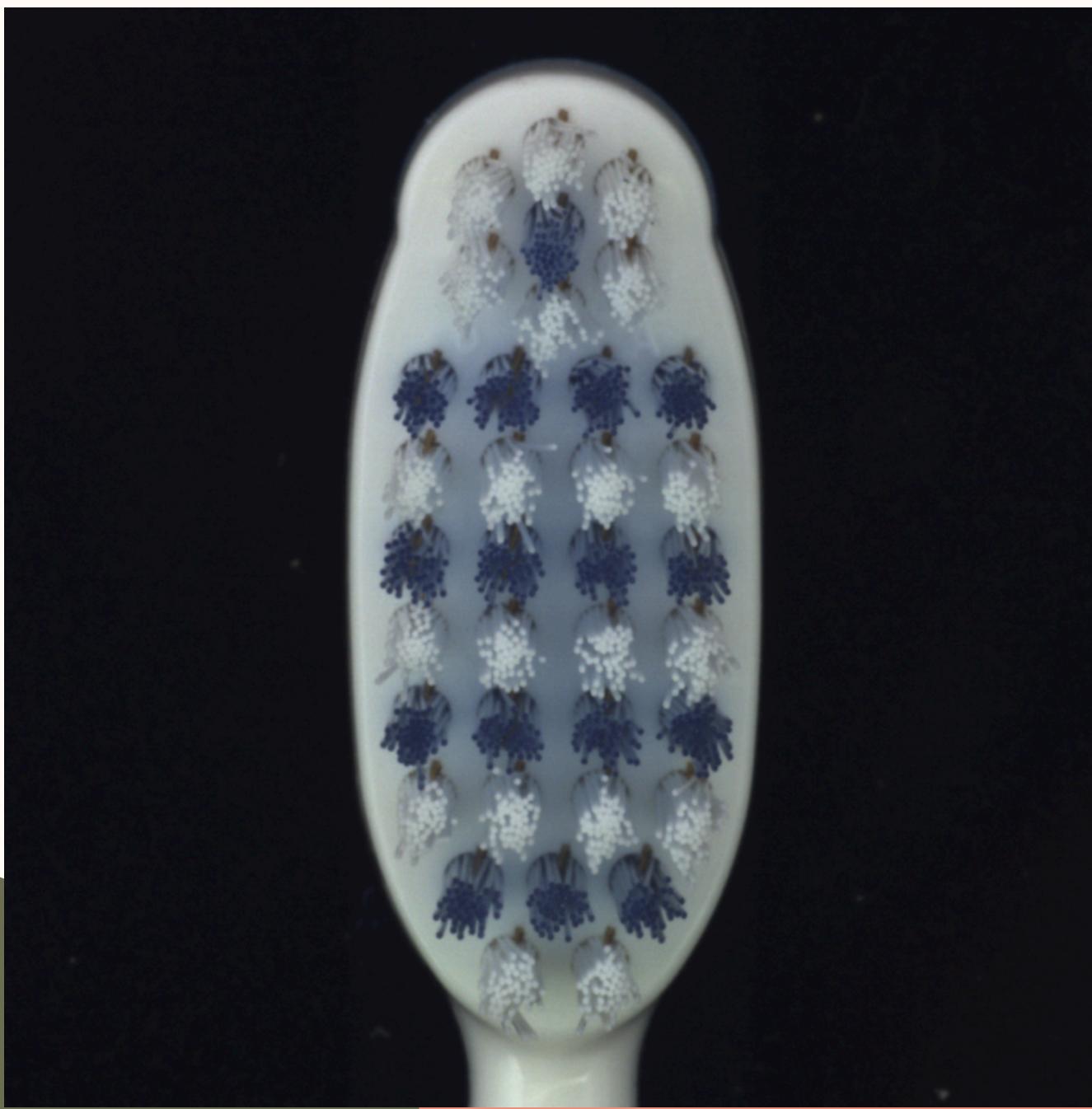
État de l'art

Apprentissage non supervisé

Deux méthodes comparées :

- kNN : basé sur les distances entre caractéristiques d'images
- Autoencoder (AE) : basé sur la reconstruction d'images

Différenciations entre un image normale et défectueuse



Présentation du dataset MVTec Anomaly Detection (MVTec AD)

- Jeu de données industriel de référence pour la détection d'anomalies visuelles.
- 15 catégories d'objets (bottle, cable, hazelnut, pill, metal_nut, screw, etc.).

Chaque catégorie contient :

- Des images normales (données d'entraînement)
- Des images défectueuses (types d'anomalies variés pour le test)

Caractéristiques clés :

- Environ 5 000 images haute résolution.
- Annotations précises des zones anormales.
- Répartition :
 - train/good → uniquement images normales
 - test/defect_type → anomalies variées

Présentation du dataset MVTec Anomaly Detection (MVTec AD)



Présentation sur hazelnut d'une image bonne contre une image défectueuse



Architecture générale du projet et pipeline d'analyse

Structure du projet :

- data/ → dataset MVTec AD (images normales et défectueuses)
- src/ → code source (modèles, méthodes, utilitaires)
- scripts/ → scripts exécutables pour chaque étape
- artifacts/ → résultats générés automatiquement

Pipeline global :

1. Chargement et visualisation du dataset
2. Extraction de caractéristiques (features) avec un modèle pré-entraîné
3. Entraînement :
 - Méthode 1 → kNN sur les features
 - Méthode 2 → Autoencoder (AE) sur les images
4. Évaluation (métriques : AUROC, F1, Accuracy, AP)
5. Comparaison globale et visualisation finale



Méthode 1 : Détection d'anomalies par kNN sur des features pré-entraînées

Principe général :

- Chaque image est transformée en un vecteur de caractéristiques à l'aide d'un modèle pré-entraîné (ex. ResNet).
- On compare ensuite les distances entre ces vecteurs :
 - Si une image est très éloignée des images normales, elle est considérée comme anormale.

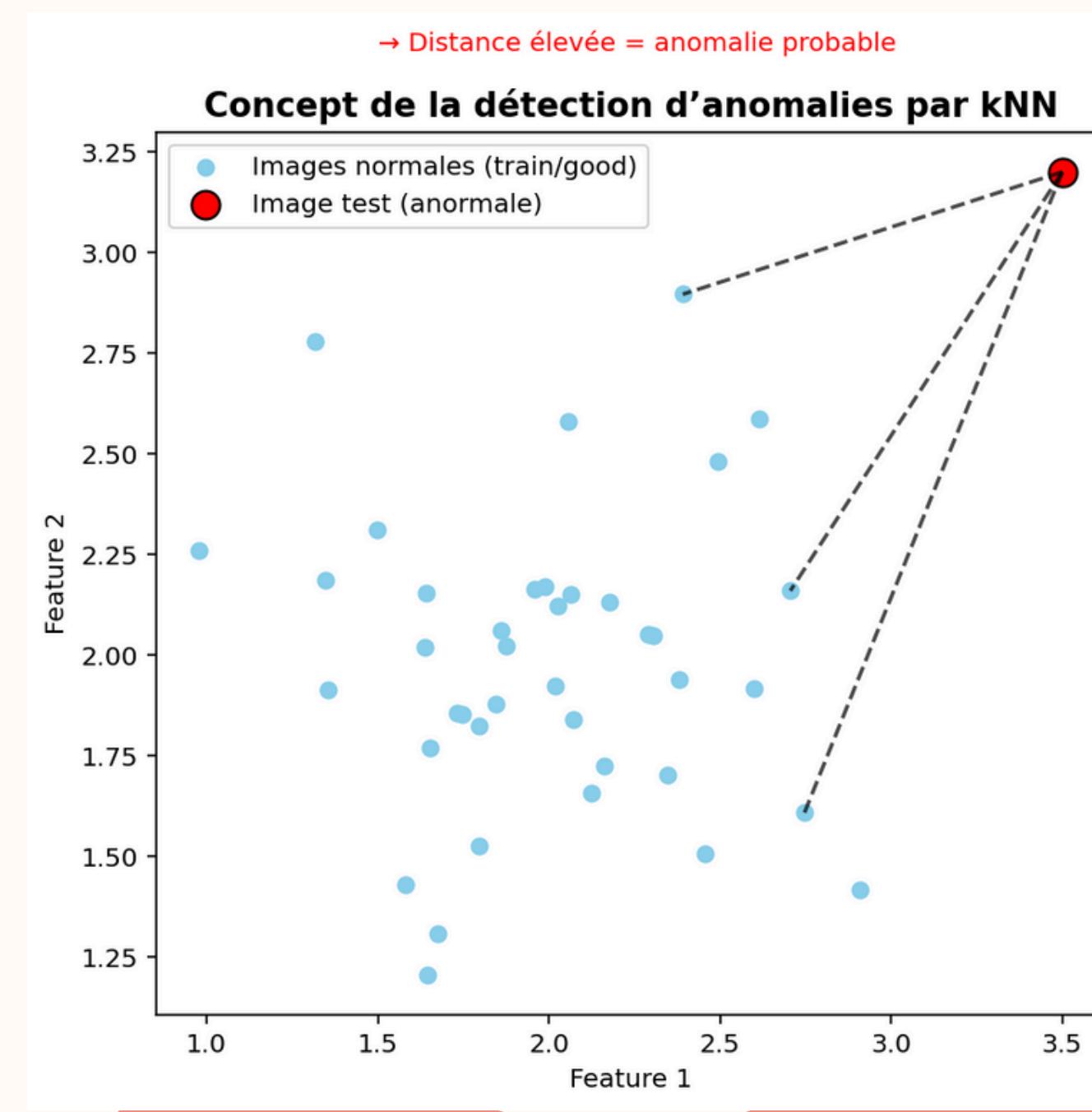
Mise en œuvre :

1. Extraction des features sur les images normales (fit_knn.py)
2. Construction de la “feature bank” → knn_bank.npz
3. Évaluation des images test (eval_knn.py)
4. Calcul des scores d'anomalie et des métriques (AUROC, etc.)

Avantages :

- Simple et rapide à mettre en place
- Peu de paramètres à ajuster
- Limité si les anomalies sont subtiles ou non linéaires

Méthode 1 : Détection d'anomalies par kNN sur des features pré-entraînées



Méthode 2 : Détection d'anomalies par reconstruction avec Autoencoder

Principe général :

L'Autoencoder apprend à reconstruire les images normales.

Il se compose de deux parties :

- Encodeur → compresse l'image en un vecteur latent
- Décodeur → reconstruit l'image à partir de ce vecteur

Lorsqu'on lui présente une image anormale, la reconstruction est moins précise.

- → La différence entre l'image originale et reconstruite révèle les anomalies.

Mise en œuvre :

1. Entraînement sur les images normales (`repeat_ae.py`)
2. Évaluation sur les images défectueuses (`eval_ae.py`)
3. Mesure de la reconstruction et calcul du score d'anomalie (erreur pixel par pixel)
4. Comparaison multi-seed pour robustesse (5 répétitions)

Avantages :

Permet d'apprendre la structure interne des images
Peut détecter des anomalies subtiles
Entraînement plus long et résultats variables selon les catégories

Méthode 2 : Détection d'anomalies par reconstruction avec Autoencoder

Schéma simplifié de l'auto encodeur



Comparaison des performances : kNN vs Autoencoder

Objectif :

Comparer les performances des deux méthodes sur toutes les catégories du dataset MVTec AD à l'aide de la métrique principale AUROC.

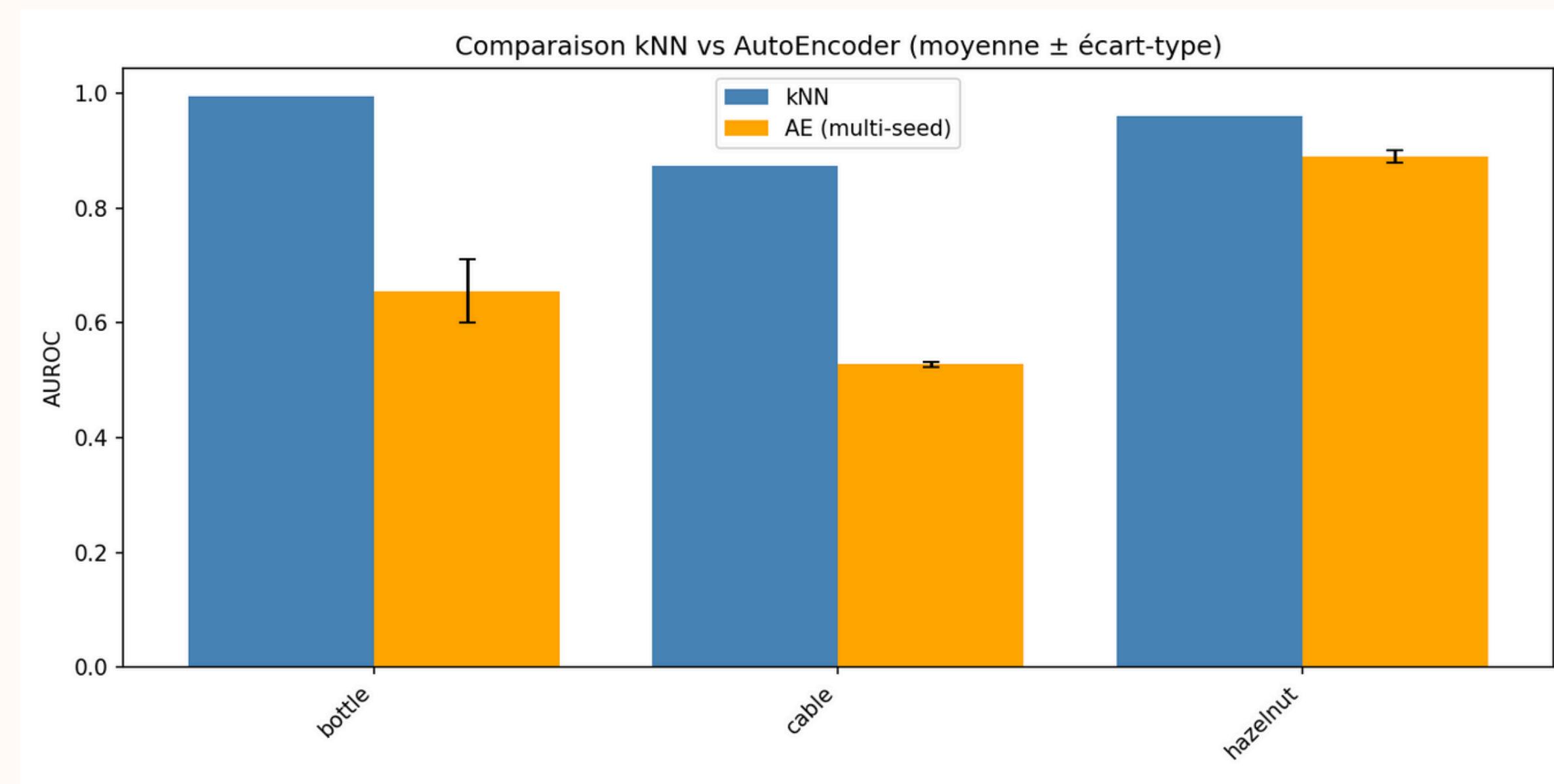
Métrique utilisée :

- AUROC (Area Under ROC Curve) : mesure la capacité du modèle à séparer les images normales et anormales
- (1 = parfait, 0.5 = aléatoire)

Résultats synthétiques :

Catégorie	kNN	AE (multi-seed)	Commentaire
Bottle	0.99	0.89	Très bon résultat pour kNN
Cable	0.87	0.53	AE moins stable
Hazelnut	0.96	0.89	Bon équilibre

Comparaison des performances : kNN vs Autoencoder



Visualisation des anomalies détectées (Heatmaps et Reconstructions AE)

But :

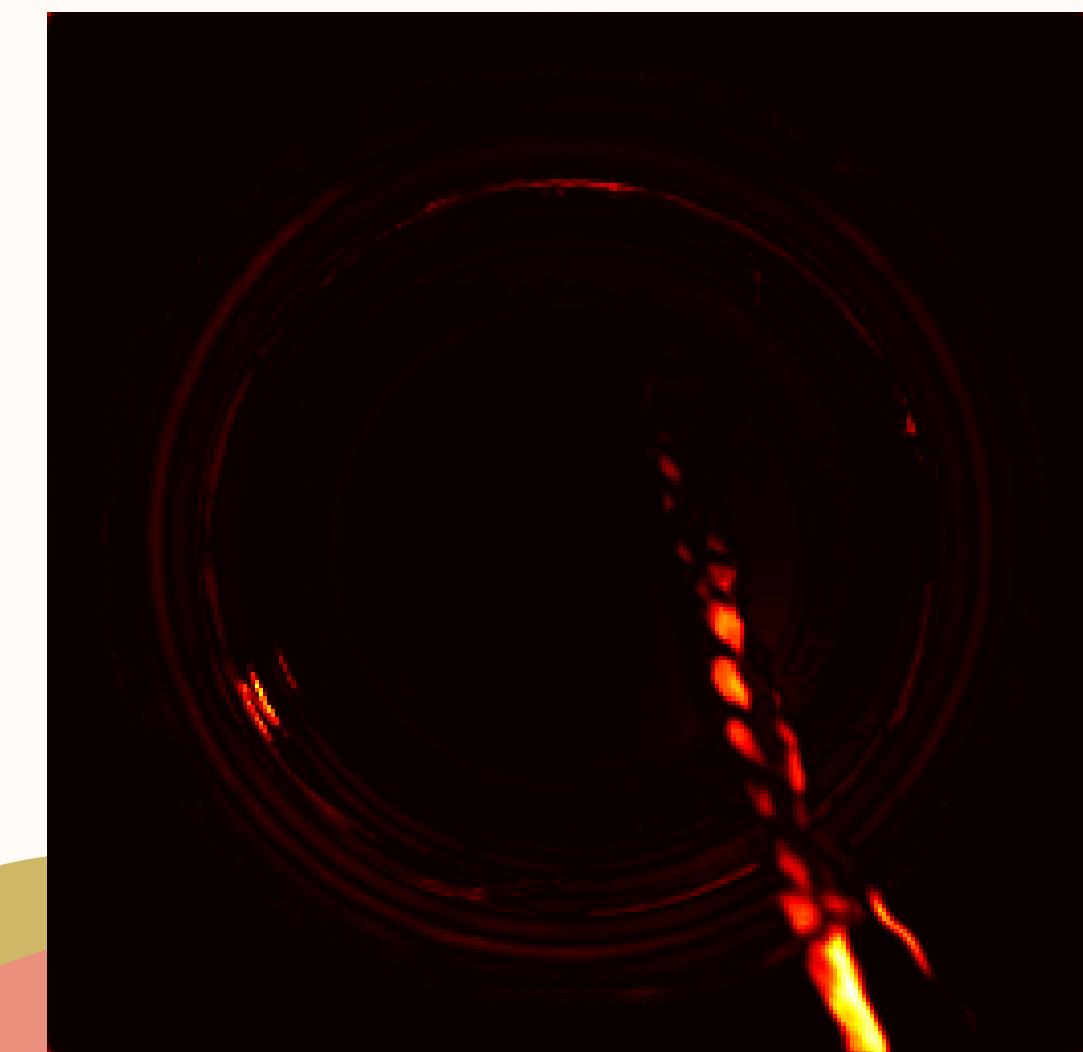
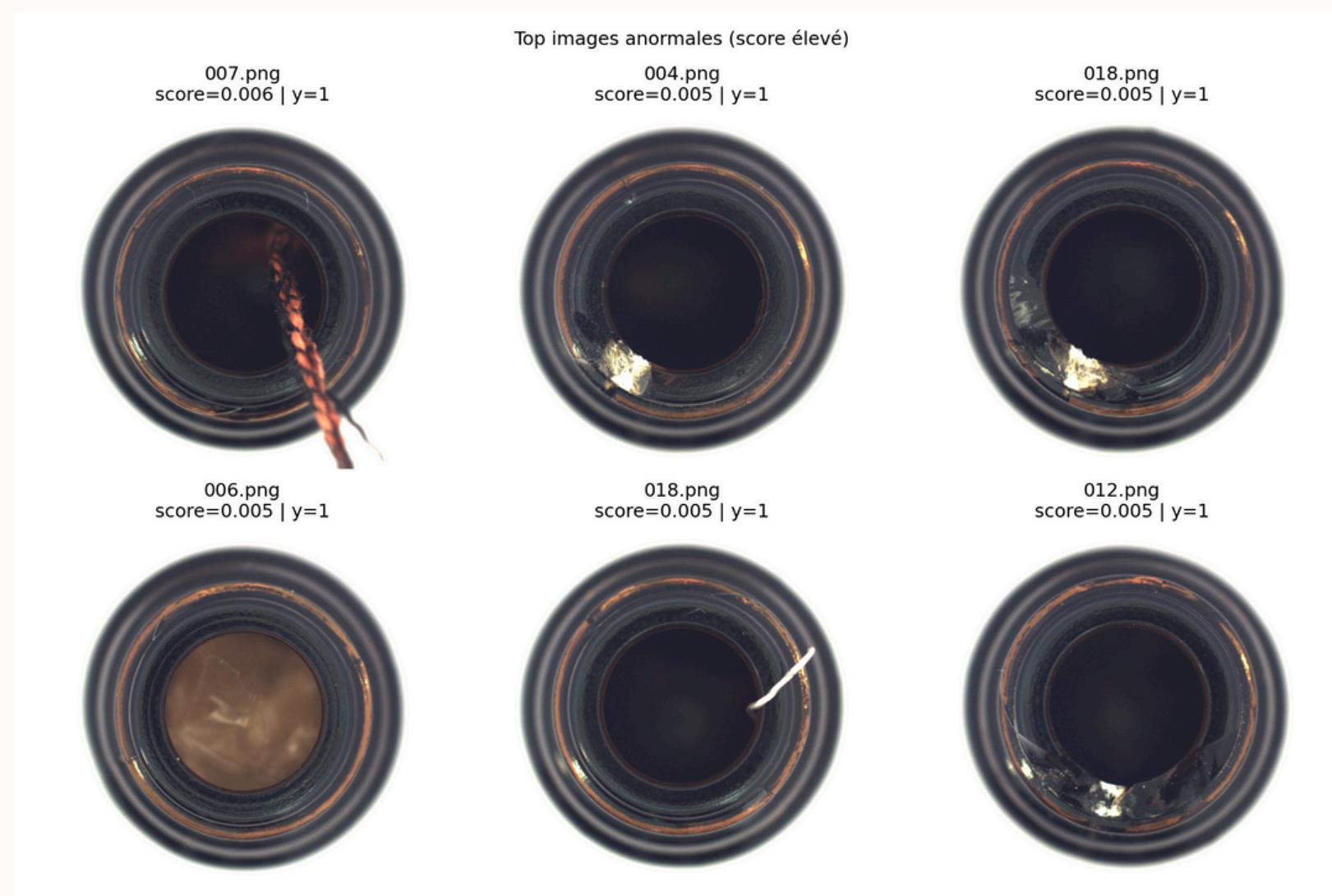
Visualiser où et comment les modèles détectent les anomalies dans les images test.

Principe :

- Les Autoencoders produisent une carte d'erreur entre l'image originale et sa reconstruction.
- Cette carte est affichée sous forme de heatmap (zones rouges = anomalie probable).
- Permet une interprétation visuelle du modèle.

Visualisation des anomalies détectées (Heatmaps et Reconstructions AE)

Exemple sur Bottle



Analyse et interprétation des performances

Résumé des observations :

Méthode	Forces principales	Limites
kNN	Très stable, rapide à exécuter, peu de paramètres à régler	Peu interprétable, dépend fortement des features pré-entraînées
Autoencoder (AE)	Détection plus fine, visualisation des anomalies, flexible	Entraînement long, sensible aux paramètres et à la complexité des images

Analyse globale :

- Le kNN sur features pré-entraînées reste la méthode la plus performante et robuste sur la majorité des catégories.
- L'Autoencoder offre des résultats plus variables, mais apporte une meilleure compréhension visuelle des anomalies.
- Certaines catégories comme hazelnut ou bottle montrent une bonne complémentarité entre les deux approches.

Conclusion générale et perspectives d'amélioration

Bilan global du projet :

- Le projet a permis de détecter efficacement les anomalies visuelles sur le dataset MVTec AD.
- Les deux méthodes (kNN et AE) ont été implémentées, testées et comparées sur toutes les catégories.
- Les résultats sont cohérents avec la littérature :
 - kNN : simple, rapide, très performant.
 - Autoencoder : plus complexe, mais offre une visualisation qualitative.

Principaux enseignements :

- La qualité des features est cruciale pour la détection d'anomalies.
- L'Autoencoder nécessite un paramétrage précis (taille du latent space, epochs, learning rate).
- Les métriques comme l'AUROC et l'AP permettent une évaluation fiable entre catégories.
- Le projet met en avant l'importance de comparer des approches complémentaires.

Merci pour votre écoute !

