

Mini-Projet :

**Vision Artificielle et Traitement des Images**

Spécialité**: Génie informatique 3**

Thème**:**

**Détection des Changements dans les Pièces d'un Appartement à l'Aide d'un Algorithme de Vision Artificielle**

**Réalisé par : Sous la direction de :**

**TANAFAAT Abdessamad Chaouki Brahim El Khalil**

Année Universitaire 2024-2025

Table des matières

[Introduction](#_Toc167245289)

[1 Contexte et Objectif du Projet :](#_Toc167245271)

[2 Présentation des Technologies Utilisées:](#_Toc167245275)

[Description Générale du Projet](#_Toc167245289)

[1 Problématique et Besoins à Résoudre](#_Toc167245291)

[2 Fonctionnalités Développées](#_Toc167245292)

[Traitement des Images](#_Toc167245298)

[1 Chargement et Prétraitement des Images](#_Toc167245300)

[2 Détection de la Région d’Intérêt (ROI:](#_Toc167245301)

[3 Masquage et Transformation des Images](#_Toc167245301)

[Détection des Objets](#_Toc167245309)

[1 Identification des Différences entre Images](#_Toc167245310)

[2 Application des Seuils et Morphologie](#_Toc167245311)

[3 Extraction des Contours et Visualisation](#_Toc167245311)

[Génération et Sauvegarde des Résultats :](#_Toc167245315)

[1 Création des PDF de Résultats Intermédiaires](#_Toc167245316)

[2 Sauvegarde des Résultats Finaux](#_Toc167245317)

[Démonstration](#_Toc167245320)

**1.** **Introduction**

**1.1 Contexte et Objectif du Projet**

Dans le cadre du développement d’applications de vision par ordinateur, l’identification et la détection d'objets dans des environnements spécifiques jouent un rôle essentiel dans divers domaines, tels que la domotique, la sécurité et la gestion des espaces. L'objectif de ce projet est de concevoir une solution automatisée capable de détecter les objets modifiés ou manquants dans des images d'intérieur, tout en générant des rapports visuels et interactifs.

Ce projet se concentre sur trois pièces spécifiques : Chambre, Cuisine, et Salon, avec une approche adaptée à chaque espace pour optimiser la précision de la détection. En utilisant des algorithmes de traitement d'images avancés, l'application vise à :

Identifier les régions d'intérêt (ROI) pour chaque pièce.

Analyser les différences entre une image de référence et une image capturée.

Détecter les objets ajoutés, supprimés ou déplacés dans l'environnement.

Générer des rapports au format PDF pour une visualisation claire des étapes du processus.

**1.2** **Présentation des Technologies Utilisées**

Ce projet utilise plusieurs bibliothèques Python puissantes pour traiter et analyser des images. Voici un résumé des technologies principales :

**OpenCV** : Bibliothèque de traitement d'images qui permet de lire, manipuler et analyser les images, notamment pour la détection d'objets, le calcul des différences entre les images, et la gestion des régions d'intérêt (ROI).

**NumPy** : Bibliothèque pour la manipulation de données numériques sous forme de tableaux, utilisée ici pour gérer les images et effectuer des calculs mathématiques.

**Pillow (PIL)** : Bibliothèque pour la manipulation d'images, utilisée pour convertir les images traitées en formats compatibles et les sauvegarder sous forme de fichiers JPEG.

**img2pdf** : Bibliothèque pour convertir les images en fichiers PDF de haute qualité, permettant de générer des rapports visuels complets des étapes de traitement.

Ces technologies combinées permettent de créer une solution complète pour la détection d'objets dans des images et la génération de rapports sous forme de PDF.

**2. Description Générale du Projet**

**2.1 Problématique et Besoins à Résoudre**

l’objectif principal est d’automatiser le processus de détection d’objets dans des images issues de différentes pièces d’une maison, tout en générant un rapport détaillé du processus. Ce système doit répondre aux besoins suivants :

Détection d'Objets Précise : La détection des objets dans les images doit être précise et fiable, en se concentrant sur des régions spécifiques des images, appelées régions d’intérêt (ROI). Ces zones doivent être définies en fonction de la pièce à analyser (par exemple, Chambre, Cuisine, Salon) et les objets doivent être détectés au sein de ces zones avec une sensibilité élevée.

Traitement d'Images Complexes : Les images peuvent contenir des variations de lumière, des bruits et des différences de couleurs. Le projet doit permettre de comparer des images de référence avec des images capturées en utilisant des méthodes de traitement d'images avancées, telles que la conversion en espace de couleurs LAB et HSV, pour obtenir des résultats fiables malgré ces variations.

Visualisation des Étapes de Traitement : Le processus de détection doit être transparent, permettant de visualiser les différentes étapes, comme le calcul des différences entre les images, les seuils appliqués, et les contours détectés. Cette transparence est cruciale pour comprendre et valider le fonctionnement du système.

Génération Automatique de Rapports : À chaque étape du traitement des images, un rapport visuel doit être généré. Ce rapport doit inclure les images de référence, les images traitées, les différences, ainsi que les résultats de la détection des objets. Le rapport final doit être sous forme de fichier PDF, incluant toutes les étapes pour une documentation complète du processus.

Flexibilité pour Différentes Pièces : Le système doit être adaptable et capable de traiter plusieurs types de pièces avec des configurations différentes (par exemple, des objets et des arrangements distincts entre une cuisine et un salon). Cela nécessite une capacité à ajuster les zones d'intérêt et les critères de détection en fonction de la pièce analysée.

**2.2 Fonctionnalités Développées**

Le projet a permis de développer plusieurs fonctionnalités principales pour la détection d'objets et la génération de rapports à partir d'images :

1. **Définition Dynamique des ROI** : Des régions d'intérêt spécifiques sont définies pour chaque pièce (Chambre, Cuisine, Salon) afin de se concentrer sur les zones pertinentes des images.
2. **Traitement en Espaces de Couleurs** : Les images sont converties en espaces de couleurs LAB et HSV pour mieux gérer les différences de couleur et faciliter la détection des objets.
3. **Calcul des Différences d'Images** : La différence entre l'image de référence et l'image capturée est calculée pour identifier les changements significatifs, en utilisant les canaux LAB et HSV.
4. **Seuil et Filtrage des Différences** : Un seuil est appliqué pour isoler les différences pertinentes, suivi d’opérations morphologiques pour affiner les résultats et éliminer les bruits.
5. **Détection des Contours** : Les contours des objets détectés sont extraits, filtrés par taille et dessinés sur l'image pour visualiser les objets trouvés.

**3. Traitement des Images**

**3.1 Chargement et Prétraitement des Images :**

Cette étape consiste à préparer les images avant leur analyse. Voici les principales opérations effectuées :

* **Chargement des Images** :  
  Les images d’entrée sont chargées à partir de sources spécifiques, comme une caméra ou un répertoire, en utilisant la bibliothèque OpenCV.
* **Redimensionnement** :  
  Les images sont redimensionnées pour correspondre à une taille standard facilitant les calculs et garantissant une uniformité dans le traitement.
* **Conversion en Échelle de Gris** :  
  Pour certaines analyses, les images sont converties en niveaux de gris pour simplifier les calculs et réduire la complexité de traitement.
* **Filtrage et Réduction de Bruit** :  
  Un flou gaussien ou un filtre médian est appliqué pour réduire le bruit et les artefacts visuels, garantissant une détection plus précise.

**3.2 Détection de la Région d’Intérêt (ROI) :**

La Région d'Intérêt (ROI) permet de limiter le traitement d’image à une zone spécifique contenant les informations pertinentes.

Définition Dynamique de la ROI :

Les coordonnées de la ROI sont définies en fonction de la pièce à analyser (par exemple, Chambre, Cuisine, ou Salon).

Exemple : Une région rectangulaire est définie à partir des points (x, y, largeur, hauteur) spécifiques à chaque pièce.

Isolation de la Zone Pertinente :

Une fois la ROI définie, seule cette région est extraite de l’image pour le traitement ultérieur, économisant ainsi du temps de calcul et augmentant la précision.

**3.3 Masquage et Transformation des Images :**

Pour améliorer la détection des différences et des objets, des transformations supplémentaires sont effectuées sur les images.

Masquage de la Région Non Pertinente :

Un masque est appliqué pour masquer les zones en dehors de la ROI, rendant ces zones noires. Cela garantit que seuls les pixels à l’intérieur de la ROI sont pris en compte lors de la comparaison.

Transformation en Espaces de Couleurs (LAB et HSV) :

Les images sont converties en deux espaces de couleurs différents pour mieux distinguer les variations de couleur :

LAB : Permet une meilleure séparation des couleurs et de la luminosité.

HSV : Facilite la détection des teintes et saturations dans l’image.

Fusion des Canaux :

Les différences entre les images de référence et les images capturées sont calculées séparément pour chaque canal de couleur, puis fusionnées pour obtenir une image de différence combinée.

**4. Détection des Objets**

La détection des objets repose sur l’analyse des différences entre les images, l’application de transformations morphologiques et l’extraction des contours. Ces étapes permettent d’identifier les zones d’intérêt et d’afficher les résultats de manière claire.

**4.1 Identification des Différences entre Images**

L’identification des différences repose sur la comparaison de deux images, souvent une image de référence et une image capturée :

* Soustraction d’Arrière-Plan :

La méthode de soustraction pixel par pixel est utilisée pour détecter les zones de changement. Cela génère une image de différence où les variations entre les deux images sont mises en évidence.

Exemple :

* Utilisation de Canaux de Couleur :

Les différences sont analysées sur des canaux spécifiques (comme LAB ou HSV) pour une meilleure précision.

LUM (Luminosité) : Mise en évidence des zones plus claires ou sombres.

Couleur (Teinte/Saturation) : Détection des variations dans les couleurs des objets.

**4.2 Application des Seuils et Morphologie**

Couleur (Teinte/Saturation) : Détection des variations dans les couleurs des objets.

Une fois les différences identifiées, des techniques de binarisation et de traitement morphologique sont appliquées pour améliorer la précision :

**Seuil Binaire** :  
Les pixels de l’image de différence sont convertis en noir et blanc en appliquant un seuil.

* + Les pixels avec des valeurs supérieures au seuil sont définis comme des **zones d’intérêt (blanc)**, et les autres comme **fond (noir)**.

**Opérations Morphologiques** :  
Ces opérations aident à éliminer le bruit et à raffiner les zones d’intérêt :

* **Dilatation** : Agrandit les zones blanches pour combler les petits trous.
* **Érosion** : Réduit les zones blanches pour supprimer les petits artefacts.
* **Ouverture/Fermeture** : Combine dilatation et érosion pour un nettoyage optimal.

**4.3 Extraction des Contours et Visualisation**

L'étape finale consiste à extraire et visualiser les contours des objets détectés :

**Détection des Contours (Canny) :**

L’algorithme de Canny est appliqué pour identifier les contours des objets dans l’image binaire.

Les contours permettent de représenter visuellement les formes et les bords des objets.

**Encadrement des Objets :**

Les contours détectés sont entourés par des boîtes englobantes (bounding boxes) pour une meilleure compréhension visuelle.

Exemple : Dessiner des rectangles autour des objets détectés à l’aide de OpenCV (cv2.rectangle).

**Superposition sur l’Image Originale :**

Les contours ou boîtes englobantes sont superposés sur l’image originale pour fournir une visualisation claire des objets détectés.

Ces étapes permettent de transformer les différences d’image en objets clairement identifiés, facilitant l’analyse et l’interprétation des résultats.

**5. Génération et Sauvegarde des Résultats :**

La génération et la sauvegarde des résultats sont des étapes cruciales pour documenter et conserver les données issues du traitement. Dans ce projet, ces étapes comprennent la création de rapports intermédiaires et la sauvegarde des données finales pour une utilisation ultérieure ou une analyse approfondie.

**5.1 Création des PDF de Résultats Intermédiaires**

Pour chaque étape intermédiaire du traitement des images, des résultats visuels et descriptifs sont générés et regroupés dans des fichiers PDF :

* **Inclusion des Images Prétraitées** :  
  Les images obtenues après chaque étape clé, comme le prétraitement, la détection des régions d’intérêt (ROI), ou l'application des seuils, sont ajoutées au rapport PDF pour illustrer les progrès du traitement.
* **Ajout des Métriques et Observations** :  
  Des informations telles que les seuils utilisés, le nombre d’objets détectés, et les paramètres de transformation sont documentées pour chaque étape. Cela permet une meilleure traçabilité des ajustements effectués au cours du projet.
* **Création Dynamique des Rapports** :  
  Une bibliothèque comme **reportlab** (en Python) ou des outils similaires est utilisée pour générer les PDF automatiquement :
  + Intégration d’images et de légendes.
  + Organisation par sections correspondant aux étapes du traitement.
* **Exemple d’Utilisation** :
  + Section 1 : Image originale.
  + Section 2 : Résultats du masquage et des transformations.
  + Section 3 : Objets détectés, avec une boîte englobante et des contours.

**5.2 Sauvegarde des Résultats Finaux**

Une fois le traitement terminé, les résultats finaux sont sauvegardés sous des formats exploitables et structurés :

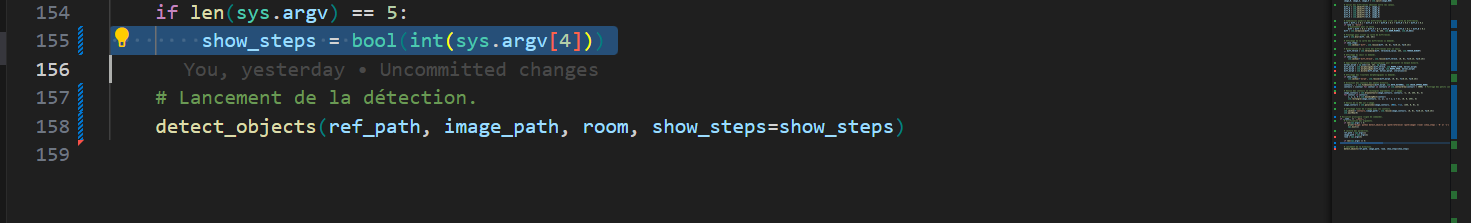
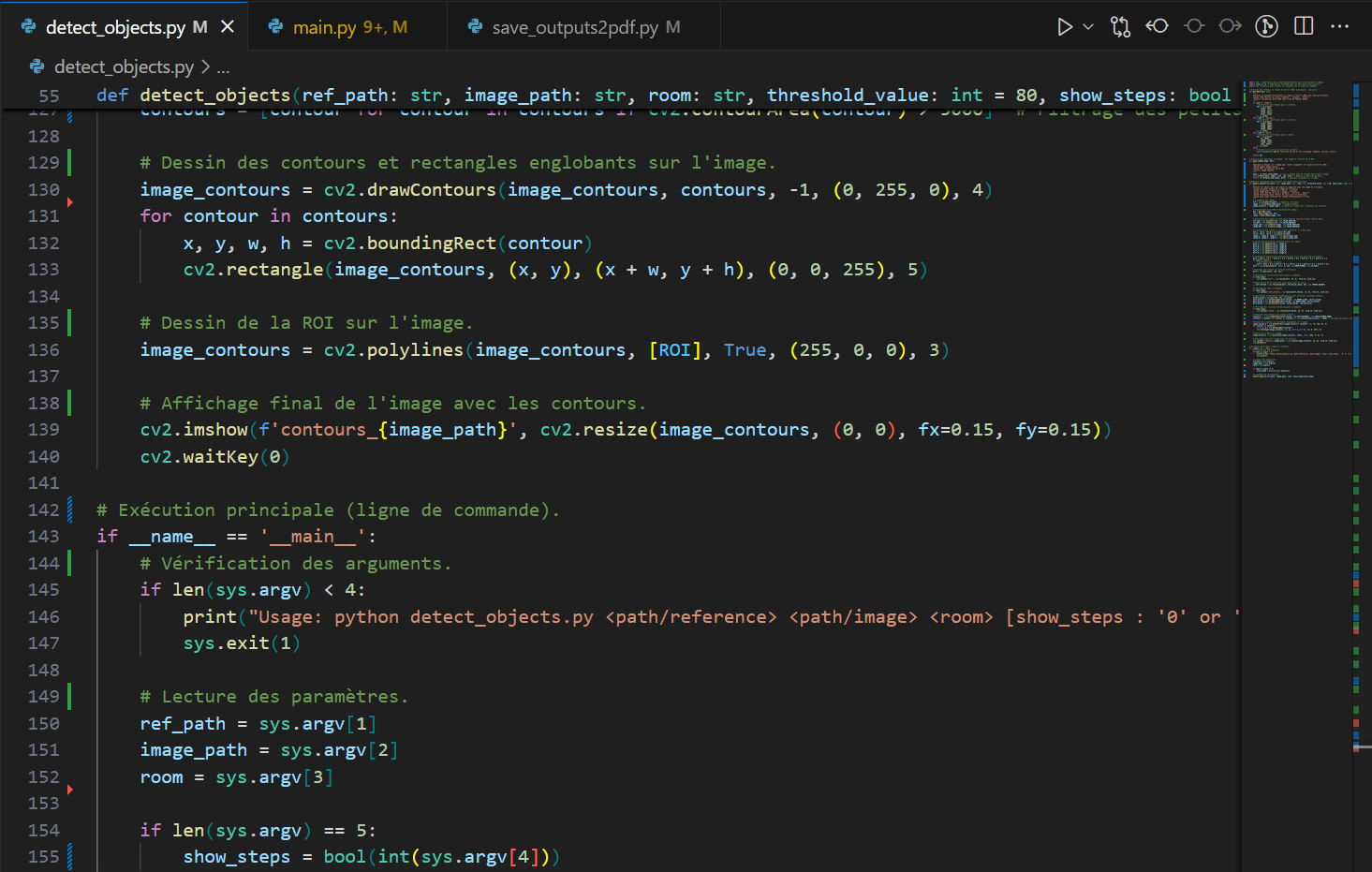
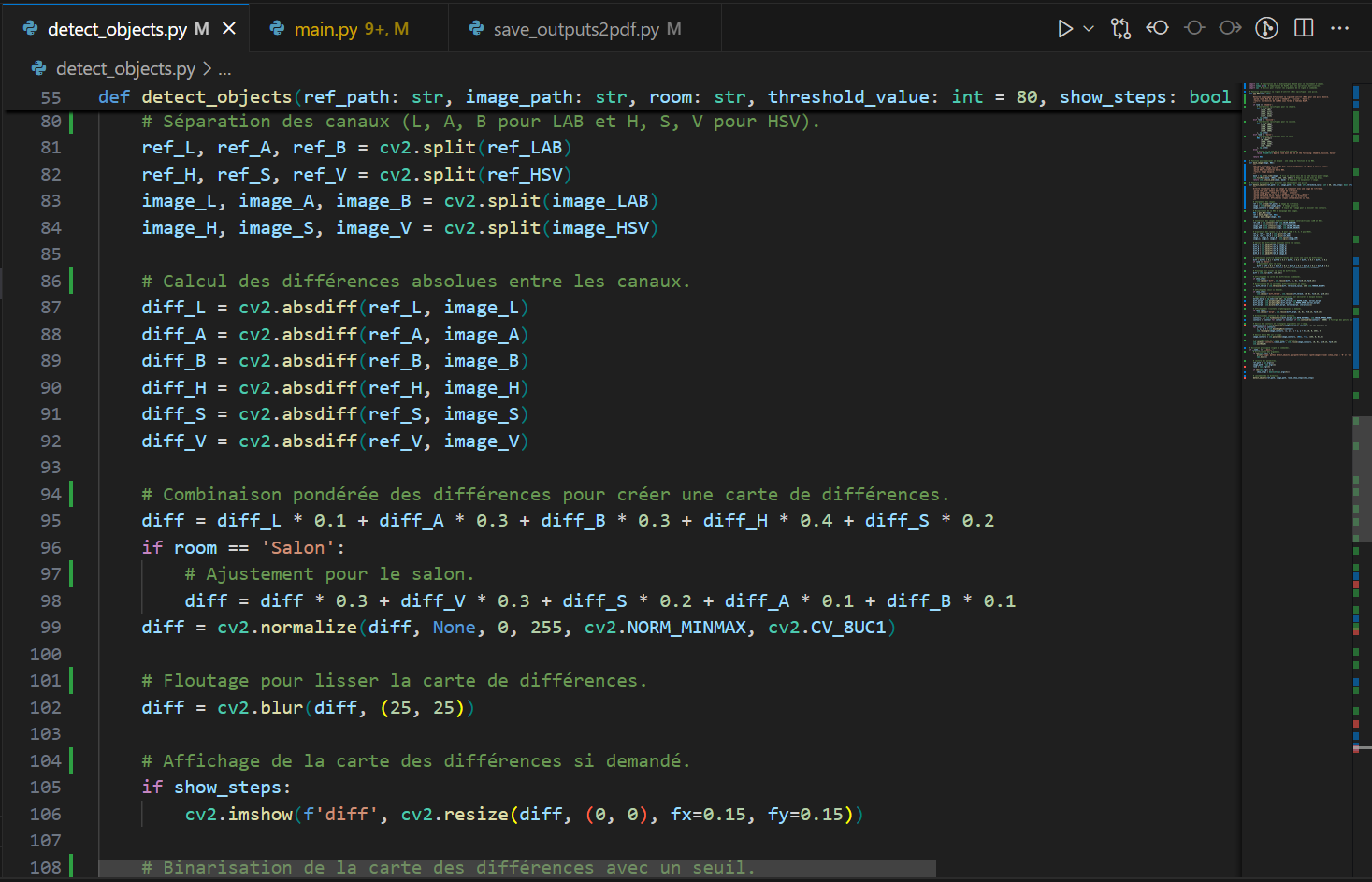
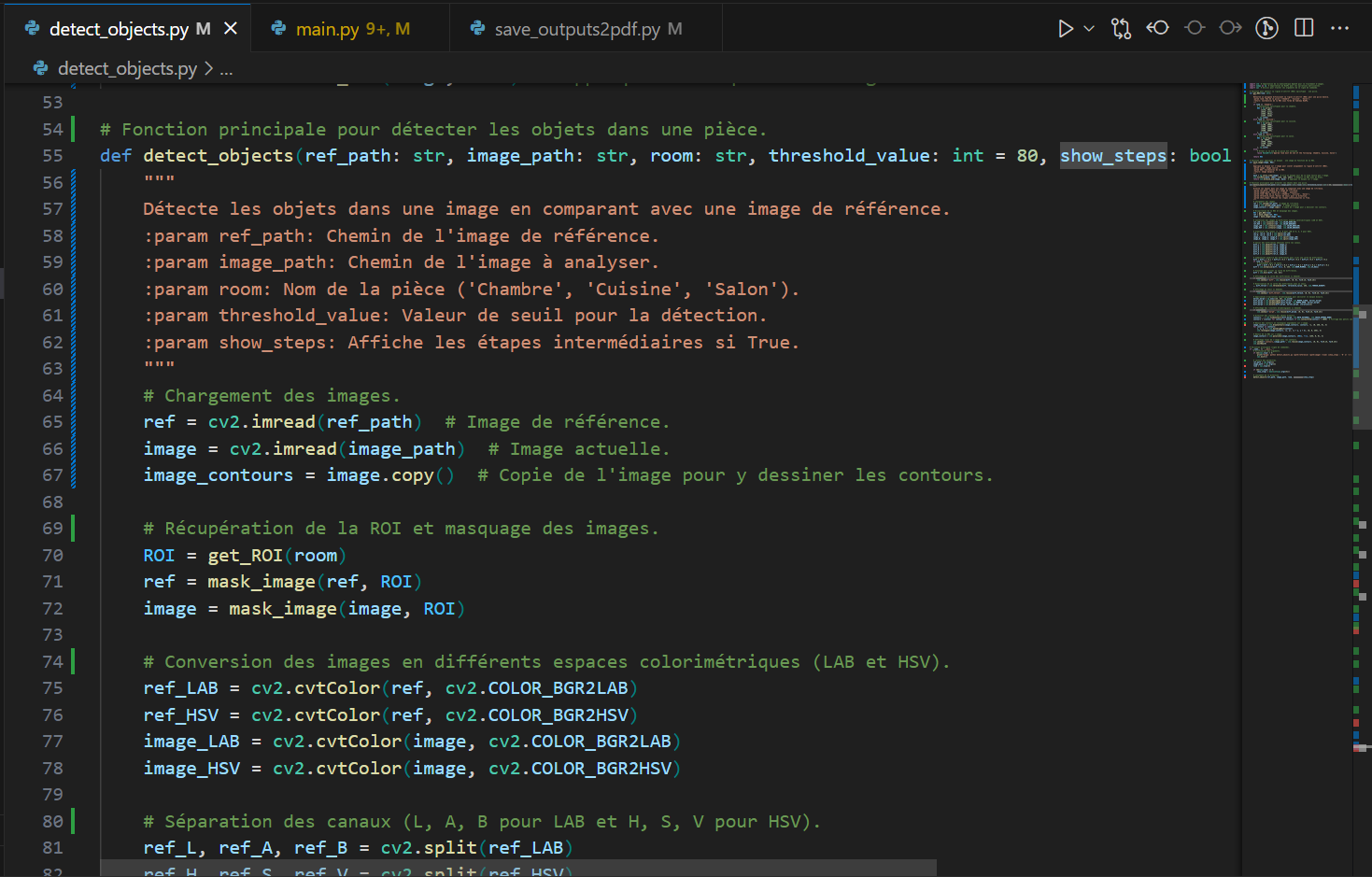
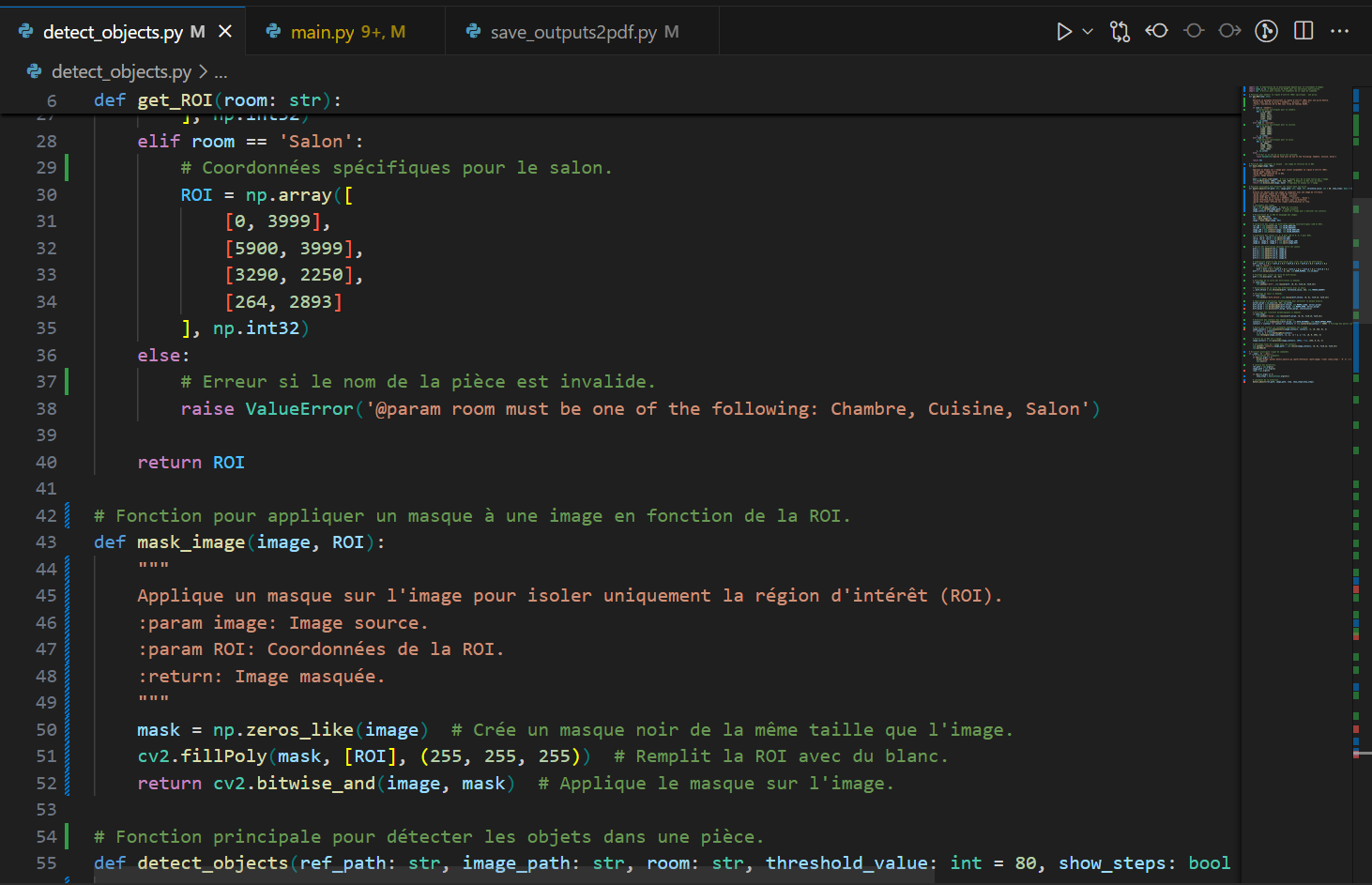
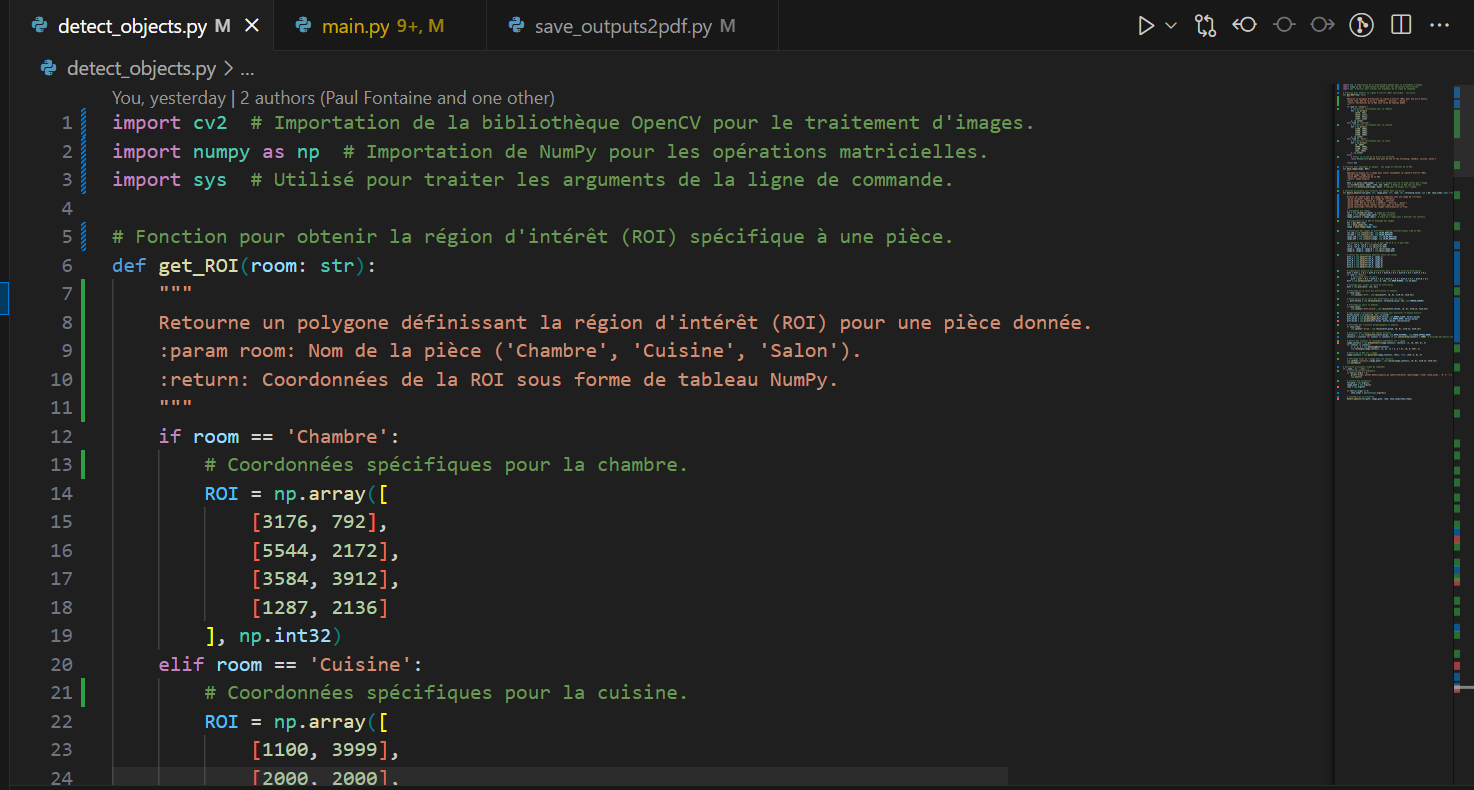
* **Images Résultantes** :  
  Les images finales avec les objets détectés (boîtes englobantes et contours) sont exportées au format **PNG** ou **JPEG** pour une visualisation rapide.
* **Fichiers Structurés** :  
  Les données liées aux objets détectés, comme leurs coordonnées, tailles, et autres métriques, sont sauvegardées au format **CSV** ou **JSON** pour faciliter une analyse quantitative.
* **Dépôt des Fichiers** :  
  Les fichiers générés sont organisés dans une hiérarchie claire :
  + **Rapports Intermédiaires** : Dans un répertoire dédié (e.g., results/intermediate\_reports/).
  + **Données Finales** : Dans un répertoire séparé (e.g., results/final\_outputs/).
* **Automatisation de la Sauvegarde** :  
  Un script Python est mis en place pour gérer automatiquement la sauvegarde des résultats avec des noms de fichiers pertinents et des horodatages :

1. **Démonstration:**

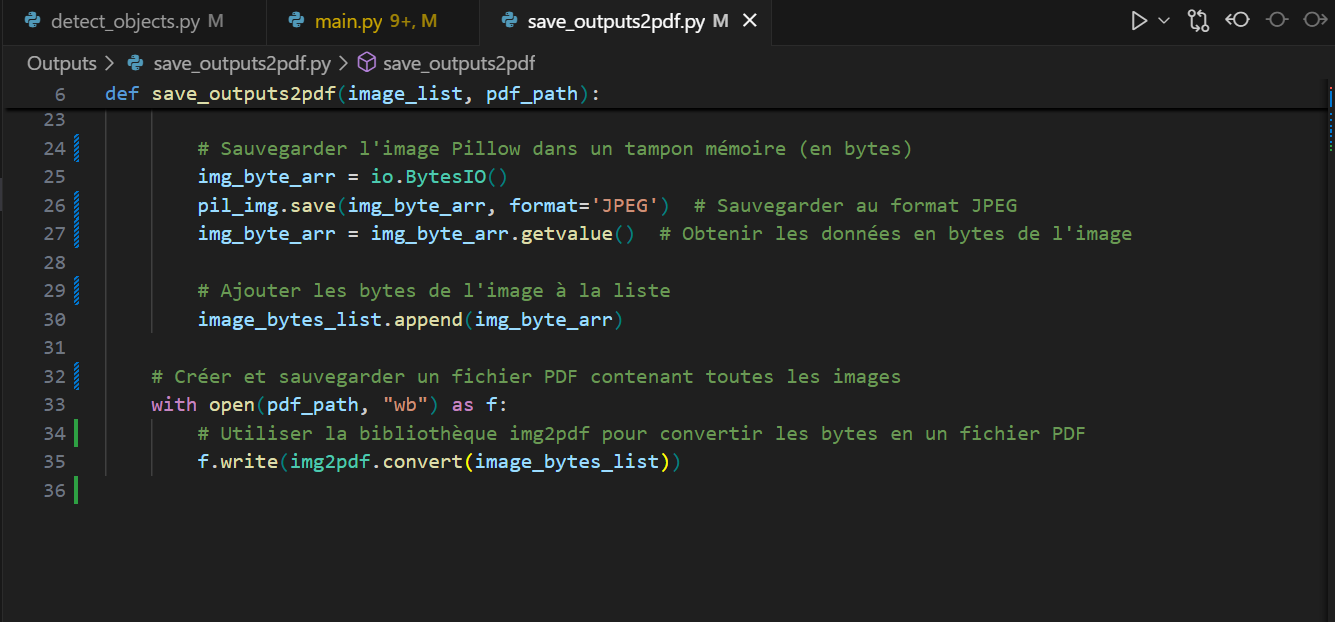
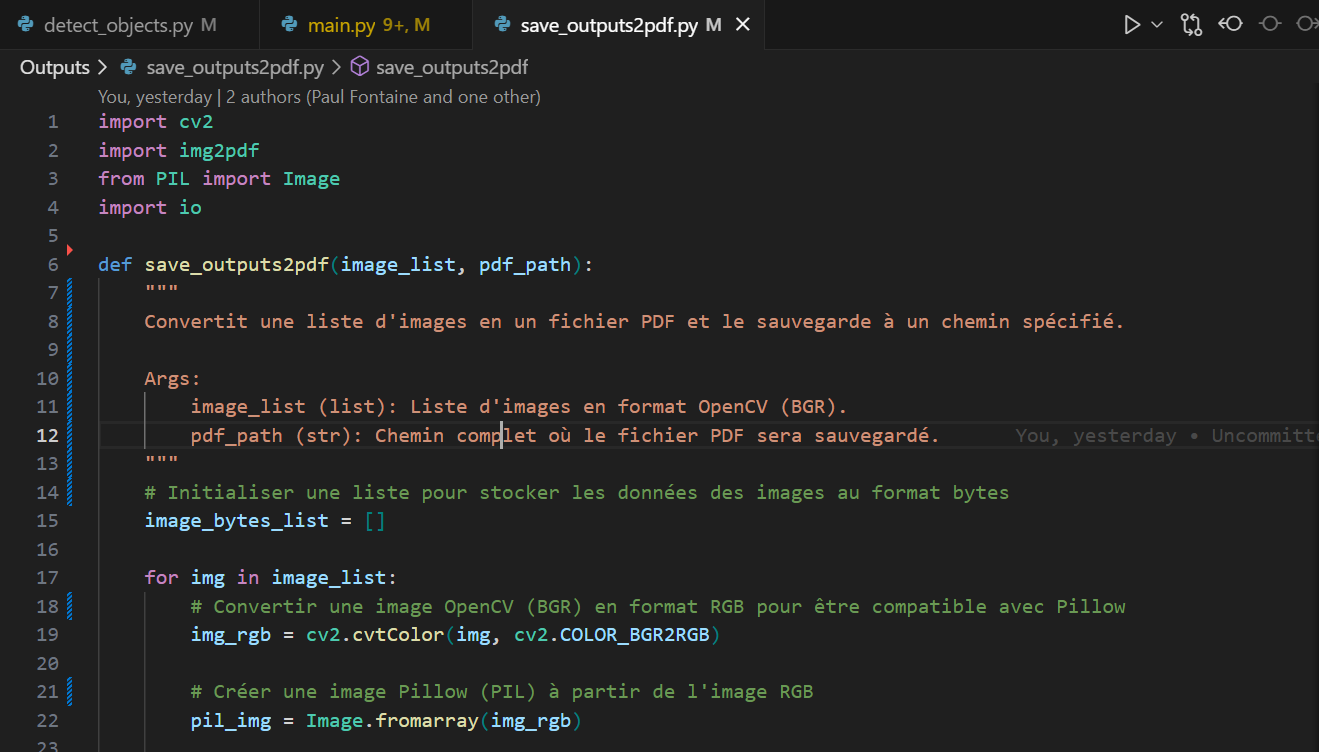
La démonstration est l’étape finale qui illustre la mise en œuvre complète des fonctionnalités développées dans le cadre du projet. Elle permet de valider la pertinence des traitements réalisés et de montrer les résultats obtenus de manière visuelle et interactive .

**Code** : Utilisation de Python.

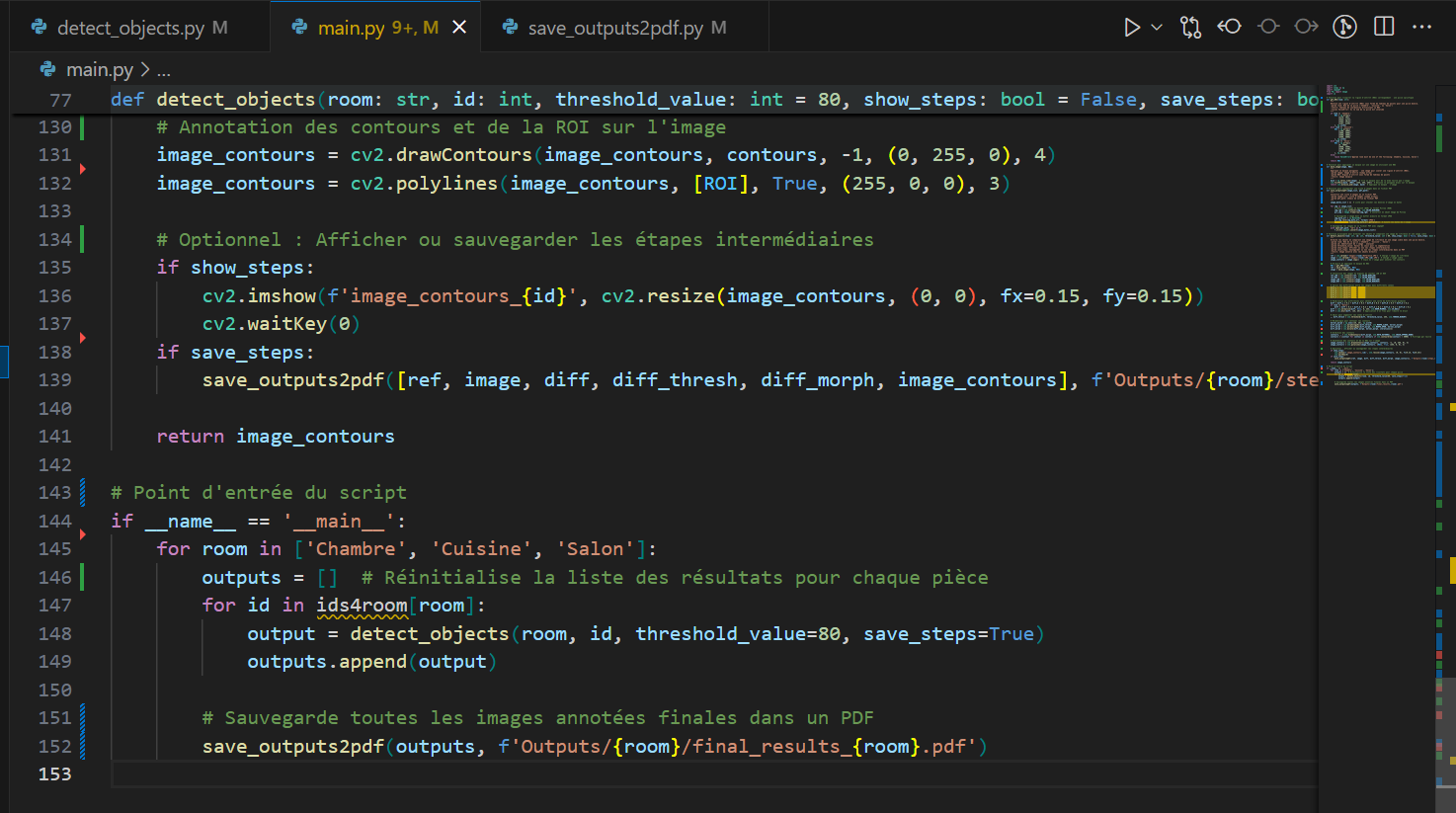
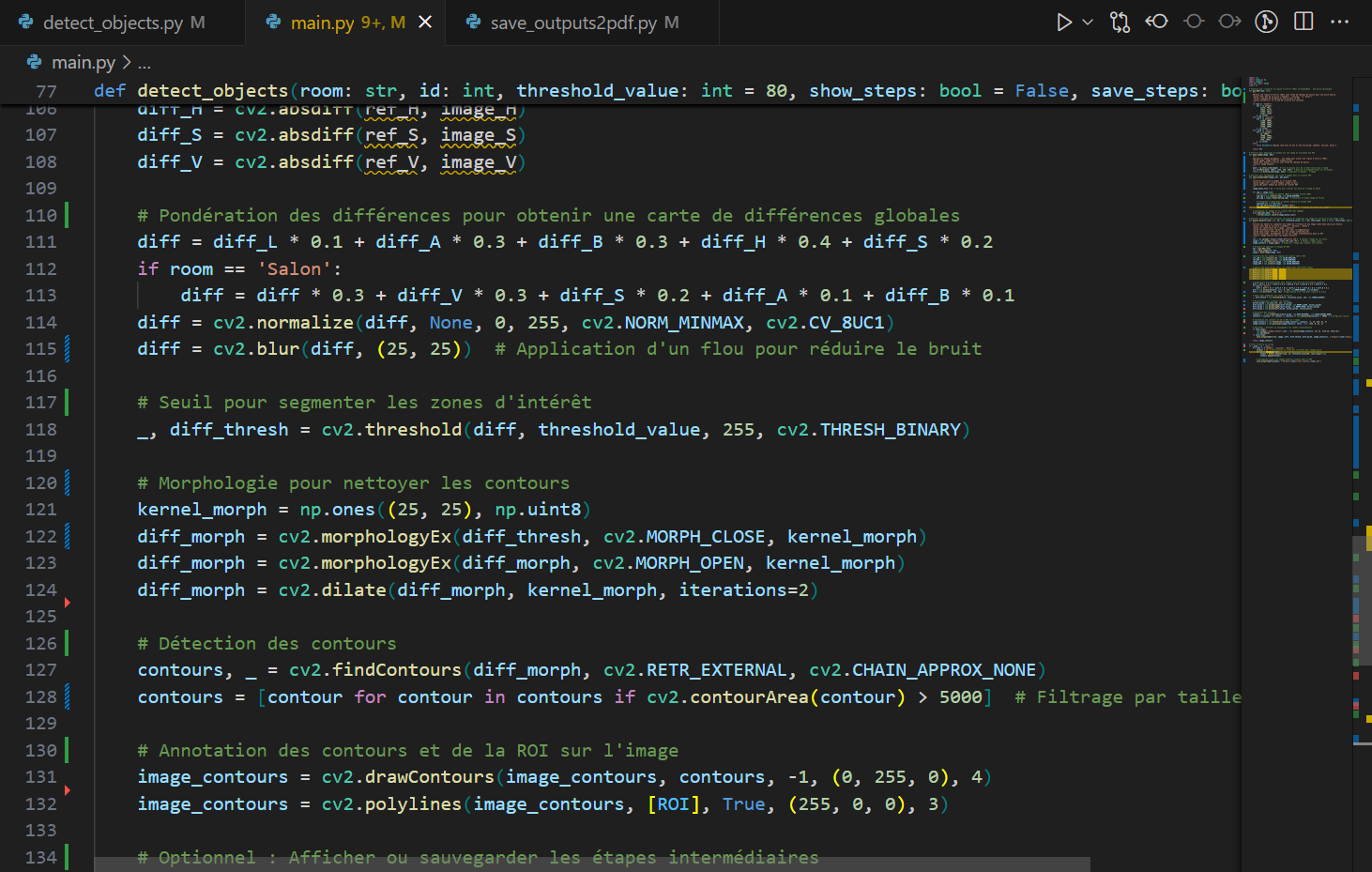
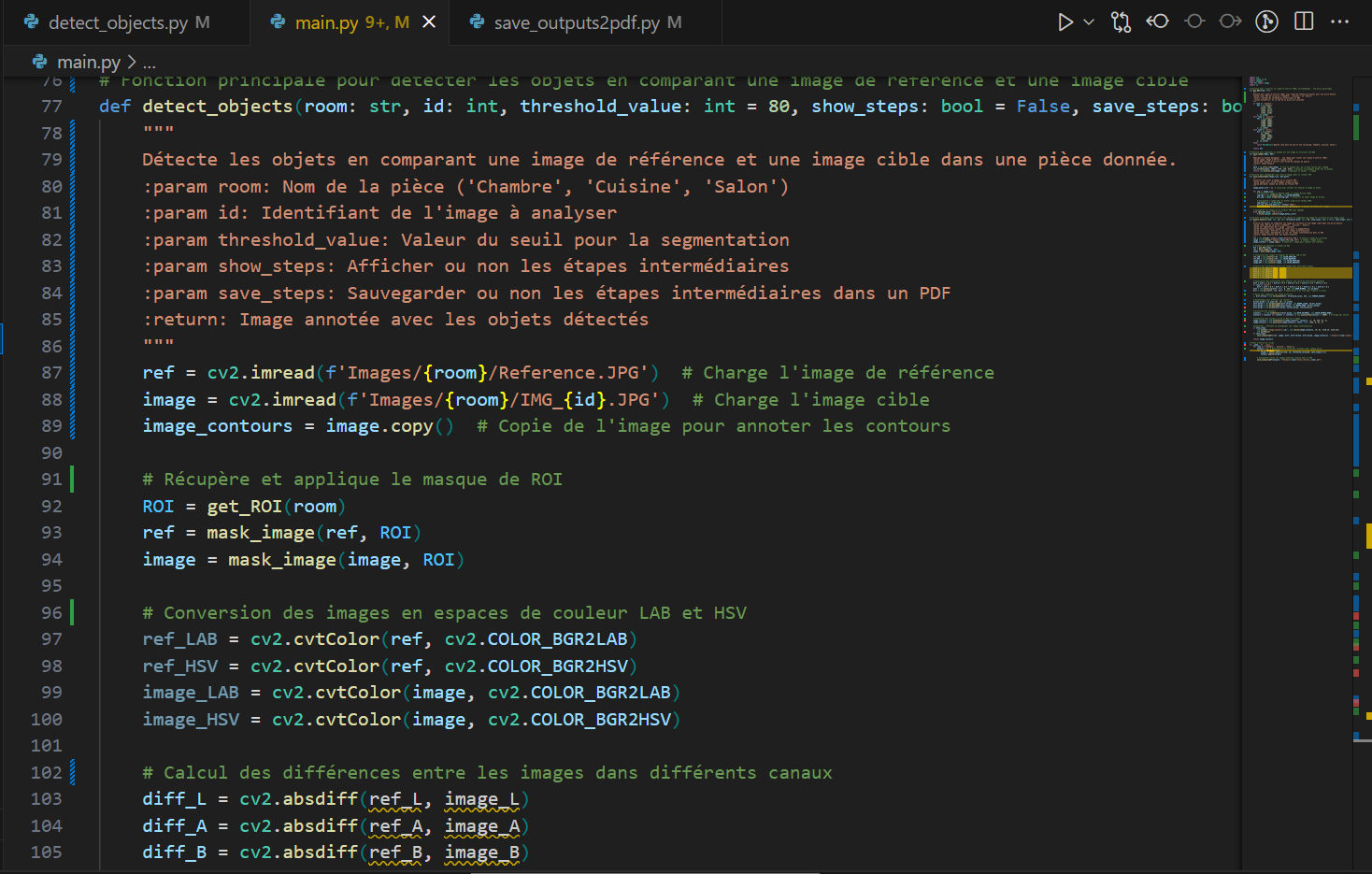
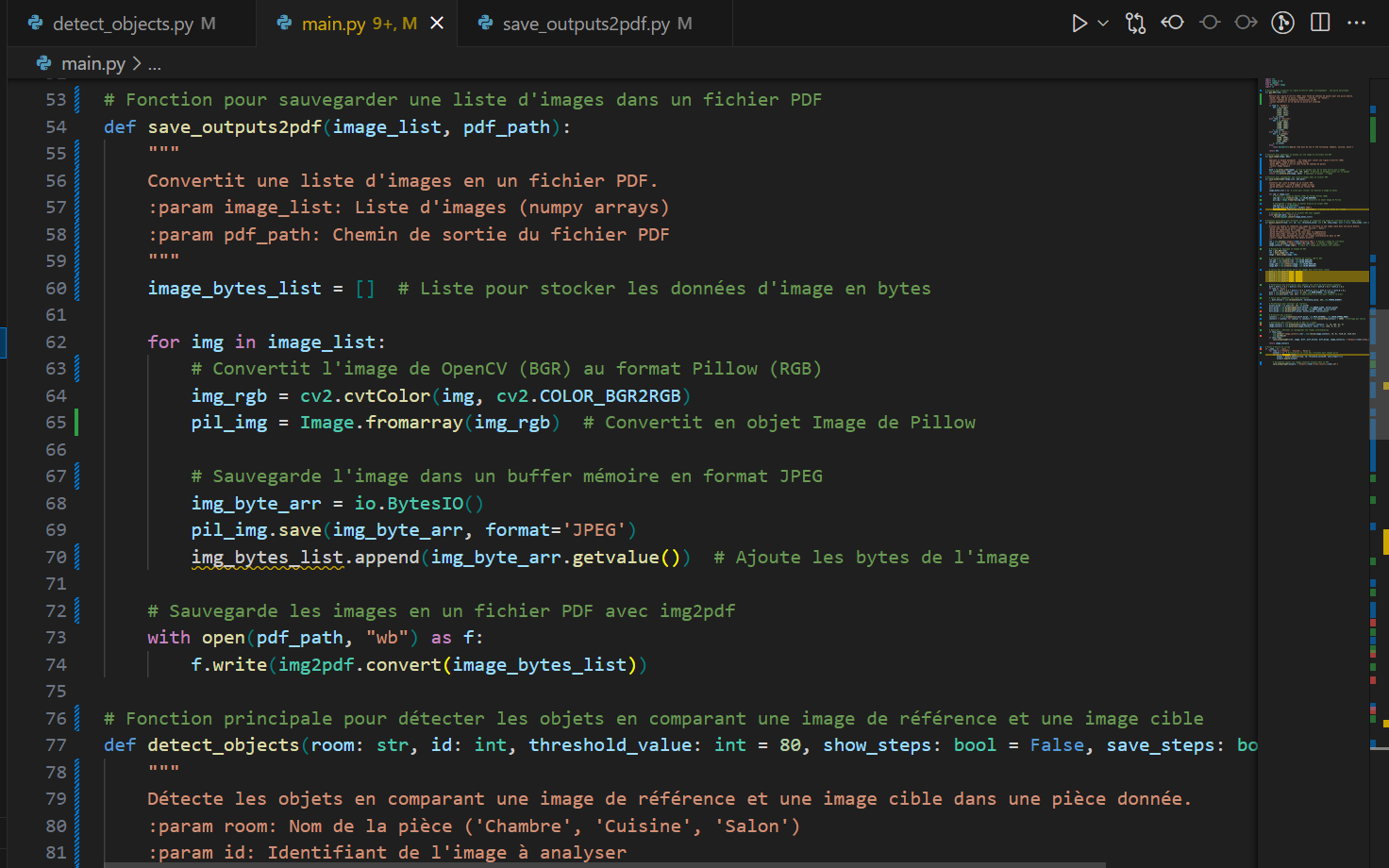
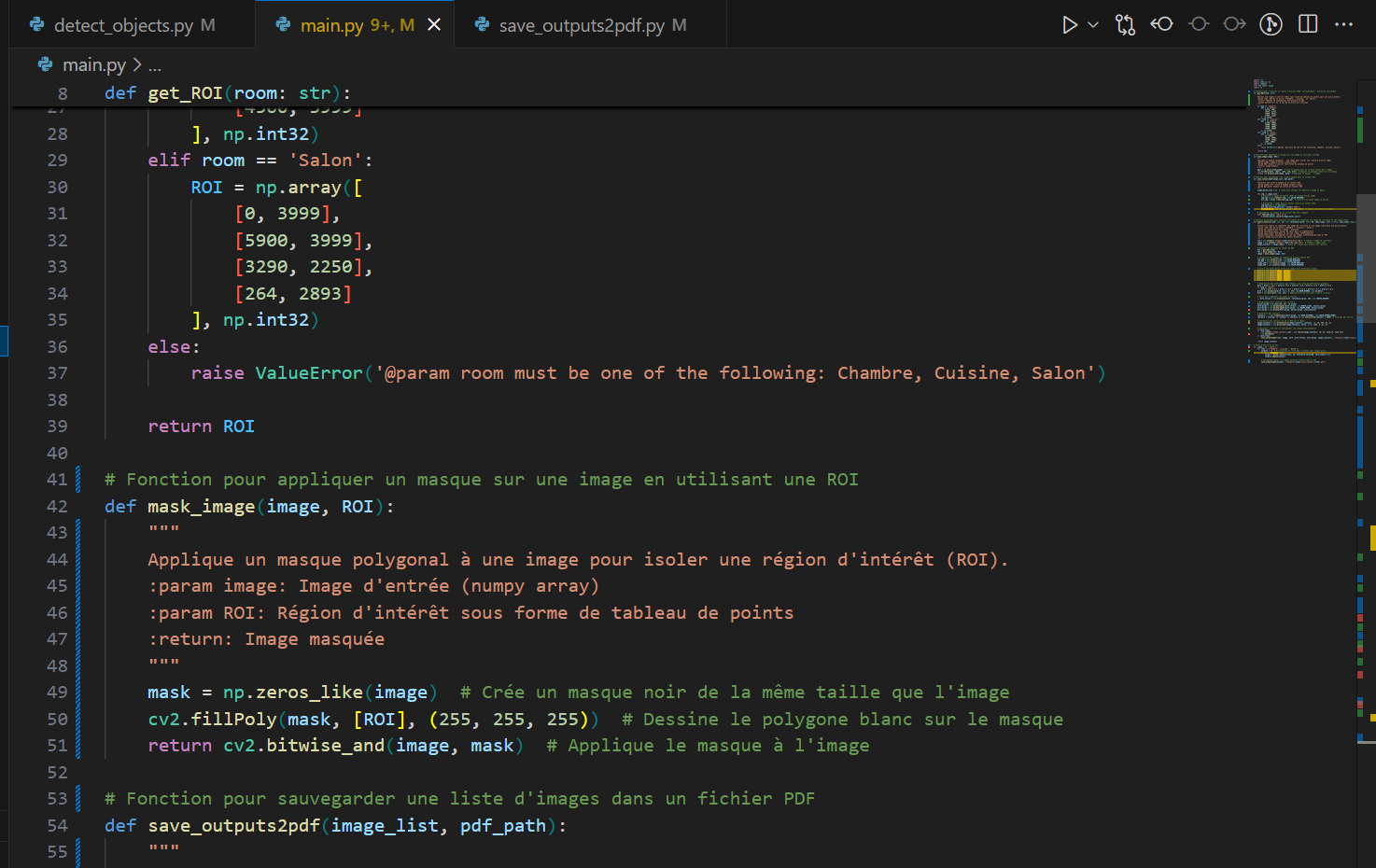
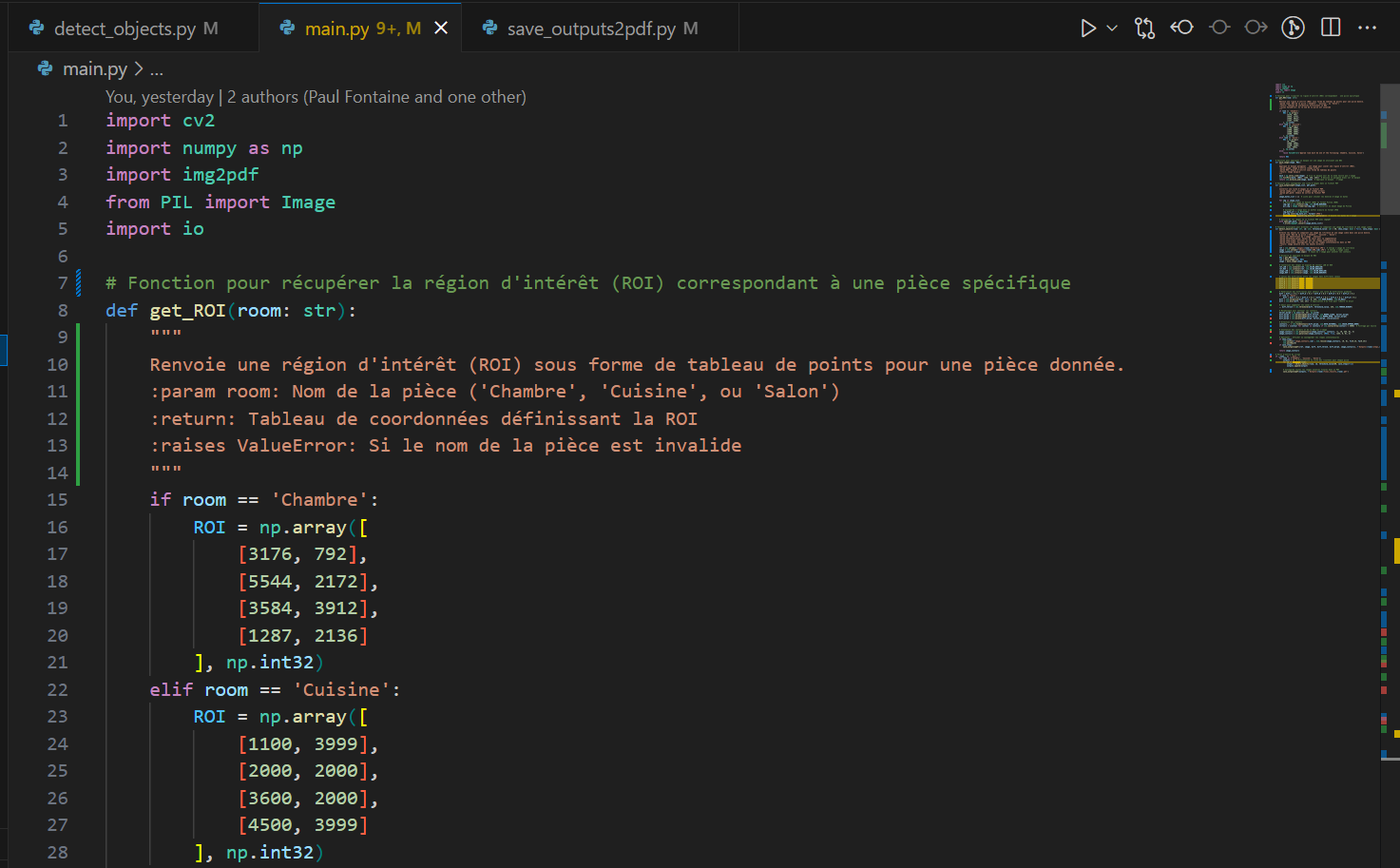
1. **La fonction DetectObjects.py :**



1. **La fonction save\_outputs2pdf.py :**

****

1. **La fonction main.py :**

****

**Exécution** : Exemple : On choisit par exemple Chambre :

**Commande :**

****

**Étapes d'Exécution**

1. **Chargement des Images**
   * **Entrée :**
     + Reference.JPG : Image de référence.
     + IMG\_1.JPG : Image à analyser.
   * **Action :**
     + Les deux images sont chargées en mémoire et redimensionnées si nécessaire pour s’assurer de leur compatibilité.



1. **Prétraitement des Images**

* **Action :**
  + Conversion des images en niveaux de gris pour simplifier l’analyse.
  + Floutage des images (Gaussian Blur) pour réduire le bruit.
* **Résultat :**
  + Les images sont prêtes pour la comparaison.

1. **Détection des Différences**

* **Action :**
  + Soustraction des deux images pour identifier les zones où des changements ont eu lieu.
  + Application d'un seuil binaire pour mettre en évidence les régions d’intérêt.
* **Résultat :**
  + Une image en noir et blanc montrant les zones différentes.

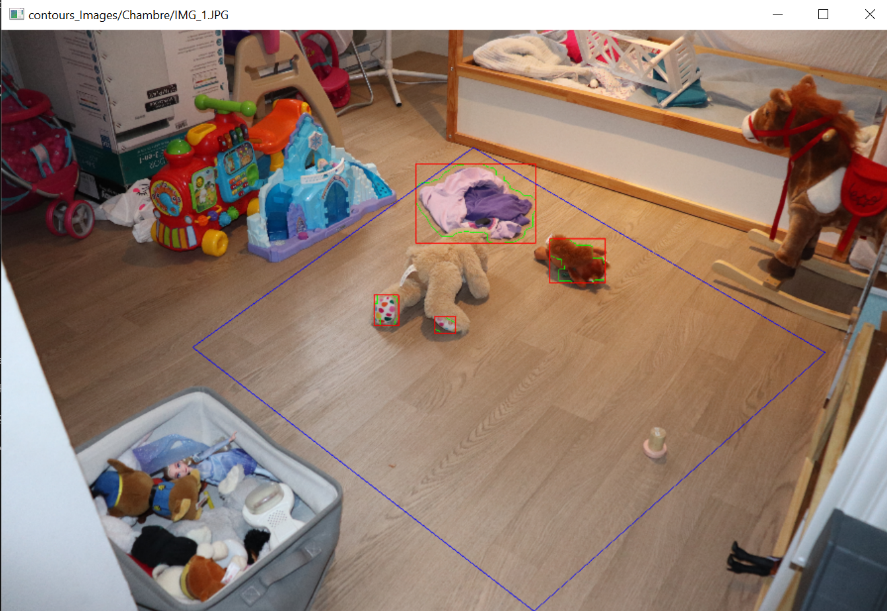
1. **Traitement Morphologique**

* **Action :**
  + Techniques de dilatation/érosion pour nettoyer les zones détectées (réduction des petits artefacts et amélioration des contours).
* **Résultat :**
  + Une image binaire avec des contours nets.

1. **Extraction des Contours**

* **Action :**
  + Détection des contours (algorithme de Canny ou similaire).
  + Dessin des contours détectés sur l’image d’origine.
* **Résultat :**
  + Une image annotée mettant en évidence les objets ou les zones différentes.

1. **Génération de Résultats**

****