

Final Project Report - Détection de Charlie par la mise en pratique du cours de Reconnaissance Visuelle de troisième année à CentraleSupélec

Fabien Maës

fabien.maes@student-cs.fr

François Petit

francois.petit@student-cs.fr

1. Résumé

Ce projet s'inscrit dans le cadre du cours de Reconnaissance Visuelle de 3A de CentraleSupélec et a pour but de localiser la position de Charlie dans le célèbre jeu "Où est Charlie ?" sans avoir recours à des méthodes issues du Deep Learning. Ce jeu se caractérise par un fort encombrement visuel, des variations d'échelle importantes et la présence d'occultations ou d'éventuels leurre, ce qui nécessite donc des algorithmes spécifiques de Reconnaissance Visuelle.

Pour mener à bien ce projet, nous proposons une architecture de traitement en cascade, du traitement grossier au plus fin. La première étape opère une réduction importante de l'espace de recherche en utilisant un filtrage colorimétrique dans l'espace HSV, ciblant la signature rouge et blanche caractéristique de Charlie. Cette segmentation est affinée par des opérations morphologiques spécifiques permettant de générer des Régions d'Intérêt (ROI) pertinentes. Dans un second temps, chaque candidat est soumis à une validation locale robuste utilisant l'extraction de descripteurs invariants SIFT (Scale-Invariant Feature Transform) comparés à un Bag of words (Bow). L'appariement final est consolidé par une vérification géométrique via l'algorithme RANSAC, assurant la cohérence spatiale des points clés détectés.

Les évaluations menées sur le jeu de données [Hey-Waldo](#) démontrent les capacités de cette approche hybride. Le filtrage chromatique s'avère très utile pour la rapidité du traitement, et la robustesse du système repose essentiellement sur la capacité des descripteurs locaux à discriminer Charlie des personnages lui ressemblant, offrant ainsi une solution pour la recherche d'objets spécifiques sans entraînement neuronal.

2. Introduction/Motivation

Nous avons pris la décision de réaliser ce projet, puisque la détection d'objets spécifiques de manière robuste aux changements d'échelle et aux rotations peut s'avérer être extrêmement utile dans certains cas. Notre projet est certes

mené dans un cadre plutôt ludique, mais nous pouvons facilement lui trouver des applications concrètes dans la vie de tous les jours, que ce soit dans un intérêt public, ou bien encore dans un intérêt industriel. Nous pourrions par exemple citer comme application la reconnaissance de personnes réelles, qui pourrait être une aide précieuse aux forces de l'ordre dans des cas d'urgence, tels que des enlèvements de personnes. Ils pourraient par exemple souhaiter appliquer un tel algorithme sur les images enregistrées par des caméras de surveillance dans des environnements à forte densité de population, tels que des gares ou des aéroports. Dans un contexte plus joyeux et plus orienté vers l'industrie, un tel algorithme pourrait être utilisé pour réaliser de la reconnaissance de pièces parmi un ensemble de pièces en vrac. Cela peut par exemple permettre à l'industriel de réaliser un inventaire de ses stocks et planifier de manière plus précise sa production et ses commandes en fonction des stocks disponibles. Nous pouvons de même très bien imaginer des utilisations dans un domaine médical, comme l'imagerie médicale où un tel traitement des images pourrait permettre de reconnaître des formes particulières, comme des tumeurs.

Nous avons par ailleurs décidé de réaliser ce projet pour mettre en oeuvre les techniques découvertes lors du cours de Reconnaissance Visuelle dispensé en troisième année à CentraleSupélec dans le cursus ingénieur. Il nous semblait pertinent et intéressant d'appliquer ces techniques nouvelles dans un contexte plus général que des sujets ou exercices de TD développés pour une utilisation purement académique.

3. Définition du problème

Ce projet vise à résoudre le problème de la détection d'objets très spécifiques dans des environnements fortement encombrés, sans l'utilisation de méthodes d'apprentissage profond. Nous avons pris la décision d'appliquer ce cas d'étude au jeu "Où est Charlie ?" (*Where's Waldo?*). Ce problème s'avère complexe pour de la vision par ordinateur pure pour diverses raisons. Premièrement, l'échelle de représentation de Charlie n'est pas constante. Le per-

sonnage peut en effet être représenté en petite ou en plus grande taille selon les planches de jeu. Ensuite, son occultation peut également être différente selon la planche de jeu considérée. Il se peut très bien que Charlie soit caché par des éléments du décor ou de la foule représentée, et que certaines zones caractéristiques de son visage ou de son corps soient obstruées. L'auteur de ce jeu, à savoir Martin Handford, place aussi volontairement des leurre dans ses planches, qui peuvent reprendre les traits caractéristiques de Charlie, comme l'alternance de rayures rouges et blanches de la même couleur que celles du pull de Charlie, et ce pour tromper le joueur. Cela peut entraîner de nombreux faux positifs pour des algorithmes qui se contenteraient de filtres sur la couleur, la corrélation ou la fréquence de répétition des patterns. Ainsi, la résolution de ce problème impose la mise en place d'algorithmes poussés de vision par ordinateur pour détecter les points caractéristiques d'un objet, ainsi que ses descripteurs, pour qu'ils puissent ensuite être comparés à une base de connaissance, pour enfin pouvoir statuer sur la probabilité que le personnage de Charlie soit présent ou non sur un petit échantillon de la planche, à savoir une région d'intérêt.

4. Travaux connexes

Historiquement, l'extraction d'objets reposait sur la corrélation croisée, peu robuste aux changements d'échelle ou aux occultations. Pour y remédier, Swain et Ballard ont introduit en 1991 l'indexation par histogramme de couleur [7], une approche efficace mais dénuée d'informations spatiales. La détection a ensuite été révolutionnée par l'algorithme SIFT [5] de Lowe (2004), qui extrait des descripteurs locaux invariants aux déformations.

Cependant, la simple mise en correspondance point à point génère un bruit considérable dans des environnements visuellement saturés. Csurka et al. (2004) ont donc pallié ce problème avec le modèle *Bag of Visual Words* (BoW) [3], offrant une signature statistique globale robuste. Enfin, les modèles statistiques perdant la structure géométrique, ils sont couramment couplés à des algorithmes de vérification spatiale comme RANSAC [4], et à des détecteurs préservant les contours nets des dessins, tels qu'AKAZE [1]. C'est sur la synergie de ces approches complémentaires que se fonde notre méthodologie.

5. Méthodologie

5.1. Données utilisées

Pour mener à bien ce projet, nous avons dû nous procurer une base de données avec des planches du jeu "Où est Charlie ?" que nous avons trouvée sur le web, au lien suivant <https://github.com/vc1492a/Hey-Waldo>. Cette base de données a été constituée et annotée par un membre de la communauté. Ainsi, nous avons à notre disposition une

vingtaine de planches du jeu, ainsi que la position de Charlie sur ces dernières. Puisque dans la phase d'élaboration de notre méthode nous avons déterminé qu'il pourrait être intéressant d'utiliser une approche avec un Bag of Words (BoW), nous avons rogné une image ne contenant que Charlie ainsi que quelques éléments du décor dans le voisinage très proche de Charlie pour chacune de ces planches. De cette manière, nous avons collecté une vingtaine d'images sur lesquelles seuls ces éléments apparaissaient. Cette base de données, présente dans le dossier */data_bow* de notre dépôt, sera utilisée lors de la troisième étape de notre méthodologie. Notons néanmoins que cette base de données est peu complète, et que cela risque éventuellement de nous poser des difficultés par la suite, puisque le BoW ne sera construit qu'à partir d'un faible nombre d'exemples de Charlie. Nous n'avons malheureusement pas trouvé de base de données publique plus complète. Cela représente sans contestation possible une limite dans les résultats que nous serons en mesure d'atteindre. Afin de détecter Charlie dans les planches de jeu, nous utilisons une approche en cascade divisée en cinq étapes, qui procède d'abord à un filtrage global de l'image pour aboutir sur une analyse plus locale sur l'ensemble des zones déterminées comme étant des zones potentiellement intéressantes. Ce fonctionnement en cascade est décrit dans les parties ci-dessous. Nous illustrons les résultats de chaque étape sur une planche exemple (planche n°19), en figure 1, afin d'aider à la compréhension.

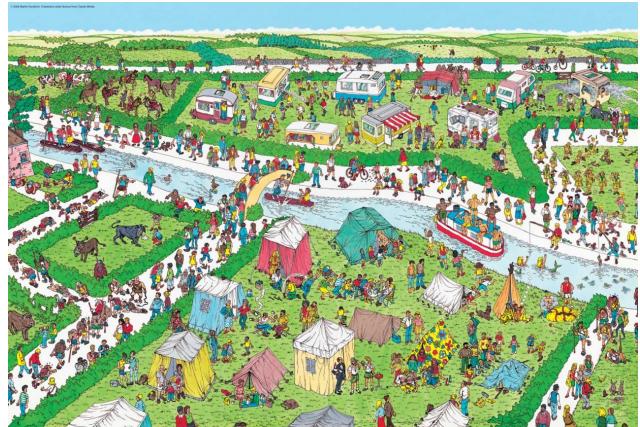


Figure 1. Planche exemple choisie : planche n°19

5.2. Étape 1 : Réduction de l'espace de recherche par filtrage colorimétrique et morphologique

Notre première étape consiste à réduire l'espace de recherche à une liste restreinte de régions d'intérêt (ROI) en utilisant un filtrage colorimétrique ciblé. Bien que notre cible possède des caractéristiques complexes (lunettes, rayures), sa signature visuelle la plus robuste à faible résolution reste la prédominance de rouge et blanc sur son bonnet et son pull. Notons dès à présent que, dans beaucoup

de planches, seule la tête de Charlie est représentée, son corps étant caché. Pour ce faire, nous avons implémenté et comparé deux approches distinctes basées sur la signature visuelle de Charlie. La première consiste en un filtrage relativement permissif concentré sur la couleur rouge, et tandis que la seconde tente un filtrage plus agressif basé sur les rayures rouges et blanches.

5.2.1. Approche 1 : Filtrage permissif par dominance chromatique

Nous faisons l'hypothèse que la caractéristique la plus robuste de Charlie, même à faible résolution ou partiellement occulté, reste la prédominance du rouge sur son bonnet et son pull. Nous convertissons d'abord l'image de l'espace RGB vers l'espace HSV (*Hue, Saturation, Value*) afin de séparer l'information chromatique de l'intensité lumineuse, rendant la détection moins sensible aux ombres. Nous appliquons ensuite un seuillage sur la composante *Hue* pour isoler les pixels rouges. Afin de couvrir l'intégralité du spectre rouge, nous combinons deux masques correspondants aux plages [0, 10] et [170, 180], avec une saturation minimale de 60 pour éviter les faux positifs grisâtres.

Le traitement morphologique est le suivant :

- Une ouverture (3×3) élimine le bruit granulaire ;
- Une fermeture verticale avec un noyau anisotrope (18, 8) est appliquée et permet de fusionner verticalement le bonnet et le pull de Charlie (souvent séparés par son visage) afin de ne former qu'une seule composante connexe.

Enfin, nous appliquons une étape de padding où l'on agrandit les marges ; les boîtes englobantes sont élargies de 20% en largeur et 40% en hauteur. Nous faisons cela à but préventif, car nous voulons garantir l'inclusion du visage, qui n'est pas détecté par le masque rouge (du fait de sa couleur chaire), et fournir un contexte suffisant pour les descripteurs SIFT de l'étape suivante.

5.2.2. Approche 2 : Filtrage strict par texture implicite

Nous tentons ici de capturer la structure "rayée" spécifique de Charlie en détectant la contiguïté spatiale entre le rouge et le blanc, à la fois sur son pull et sur son bonnet. En plus du masque rouge, nous générions un masque blanc (faible saturation, haute valeur). Nous appliquons ensuite une dilatation importante sur les deux masques, puis effectuons une opération d'intersection entre les deux. L'objectif est de ne conserver que les zones où le rouge et le blanc sont voisins immédiats. Ensuite, nous appliquons deux filtres géométriques :

- Ratio d'aspect : Conservation uniquement des formes verticales.
- Densité : Rejet des ROI contenant moins de 35% de pixels actifs (texture rayée). Ce seuil a été choisi après expérimentations.

5.2.3. Comparaison

Tout d'abord, notons que, dans l'esprit, la première méthode priviliege le rappel alors que la seconde priviliege plus la précision.

A première vue, l'approche 2 semble plus attractive car elle élimine efficacement les objets rouges uniformes (parasols, voitures). Cependant, elle s'est avérée trop agressive pour la plupart des planches. En effet, sur la majorité des planches, seule la tête de Charlie est visible (pas son corps), et donc peu de rayures blanches et rouges sont présentes. De plus, la résolution peut être trop faible, engendrant une fusion des rayures, ou encore la balance des blancs de l'image est parfois différente (le blanc du pull et du bonnet apparaît jaune/gris). Pour le pipeline final, nous avons opté pour une approche en cascade conditionnelle : nous appliquons d'abord l'approche 2 (plus stricte) et, si celle-ci ne trouvait pas Charlie, nous basculions sur l'Approche 1.

Voici les résultats de l'étape 1 avec l'approche 1 et avec l'approche 2 sur notre planche exemple.



Figure 2. ROI obtenues avec l'approche 1

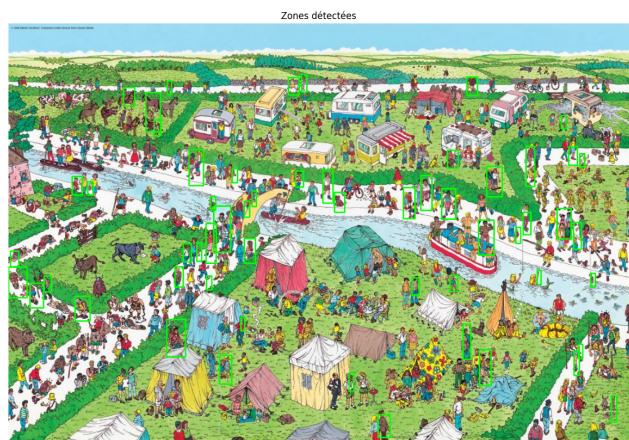


Figure 3. ROI obtenues avec l'approche 2

5.3. Étape 2 : Extraction de primitives locales

Suite à la détermination des régions d'intérêt potentiellement intéressantes isolées par nos filtres colorimétriques appliqués à l'ensemble de la planche, nous devons valider ces candidats. Pour ce faire, nous souhaitons extraire des points clés, accompagnés de leurs descripteurs, pour caractériser la structure interne de chaque ROI, conformément à ce qui a été vu en classe.

Nous avons donc choisi d'appliquer l'algorithme SIFT (Scale-Invariant Feature Transform) proposé par David Lowe [5], qui possède une caractéristique importante dans notre contexte : l'invariance à l'échelle (comme indiqué par son nom). En effet, la taille apparente de Charlie peut varier légèrement selon la perspective de la planche ou simplement selon la façon dont l'auteur a choisi de le représenter. SIFT construit un espace échelle (scale-space) en convoluant l'image avec des Gaussiennes de variances progressives, puis en calculant les Différences de Gaussiennes (DoG).

Les points clés sont détectés en utilisant les extrema locaux de ces différences de gaussiennes. Cela nous permet donc de repérer des caractéristiques propres à Charlie (comme son pompon, ses lunettes, ses rayures...) qui sont quant à elles répétitives d'une planche à l'autre, et définissent tout simplement Charlie.

L'unique implémentation de l'algorithme SIFT ne suffit pas toujours pour des cibles de petite taille et potentiellement floues comme Charlie. Nous avons donc mis en place une suite de prétraitements spécifiques dans la fonction *extraire_sift_roi*. Premièrement une amélioration du contraste local (CLAHE [8]) est appliquée puisque les planches du jeu sont souvent visuellement encombrées, et comportent aussi potentiellement des zones d'ombres ou de faible contraste qui peuvent masquer les gradients nécessaires au bon fonctionnement de l'algorithme SIFT. Ensuite, nous effectuons un sur-échantillonnage (upscaleing) accompagné d'une interpolation bicubique. En effet, les ROI détectées sont souvent de très petite taille. Or, l'algorithme SIFT commence par appliquer un flou gaussien initial, et sur une image d'une trop petite dimension, ce flou risque de supprimer certains détails, empêchant ainsi leur détection. Pour contrer ce problème, nous effectuons donc un upscaleing d'un facteur 2 en utilisant une interpolation bicubique, ce qui permet d'augmenter artificiellement la résolution de la ROI.

Lorsque l'image a subi ces pré-traitements, la fonction *sift.detectAndCompute* peut alors être exécutée. Elle permet de récupérer, comme expliqué précédemment, les points clés accompagnés de leurs descripteurs spatiaux. Nous finissons cette étape par un post-traitement

pour assurer la cohérence spatiale. Les ROI ayant subi des retraitements, les coordonnées des points clés sont faussées, nous effectuons donc simplement les traitements inverses de sorte à obtenir la bonne localisation sur la planche d'origine. Pour illustrer, voici ci-dessous l'image de Charlie de notre planche exemple sans traitements :



Figure 4. Charlie sans traitements

Et l'image de Charlie après les traitements et utilisation de l'algorithme SIFT (on remarque que tout son corps est repéré par des points caractéristiques) :

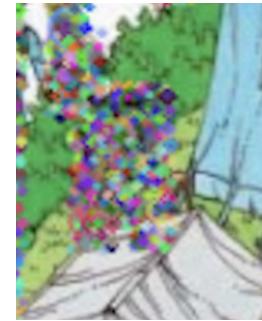


Figure 5. Charlie après traitements

Cependant, l'algorithme SIFT opère sur des images en niveaux de gris, il écarte donc l'information chromatique. Or, la signature visuelle de Charlie repose en grande partie sur l'alternance de rouge et de blanc. Ainsi, pour combler cette lacune, nous avons enrichi la description de chaque ROI par une analyse colorimétrique. Pour cela, nous calculons des histogrammes de couleurs dans l'espace HSV, ce qui offre une meilleure robustesse aux variations d'éclairage. Plus précisément, nous calculons deux histogrammes distincts sur les canaux Teinte (H) et Saturation (S). Afin de limiter la dimensionnalité et d'assurer une certaine tolérance aux légères variations de teinte, nous quantifions ces histogrammes sur 16 bins. Ces histogrammes sont ensuite normalisés (par min-max) pour rendre le descripteur invariant à la taille de la ROI, car le nombre de pixels ne doit pas influencer la signature couleur. Finalement, les histogrammes sont concaténés pour former un vecteur

de caractéristiques colorimétriques qui viendra compléter les descripteurs SIFT.

5.4. Étape 3 : Modélisation de la cible Charlie (approche Bag of Visual Words)

Charlie pouvant adopter plusieurs postures selon la planche (il peut par exemple être de profil, courbé, en train de marcher...), une comparaison directe avec une unique image de référence se montrera probablement peu efficace. Nous avons donc implémenté une approche statistique inspirée de la recherche textuelle, à savoir le modèle Bag of Visual Words (Bow)[3]. L'idée en utilisant cette approche est de ne plus considérer Charlie comme une image rigide, mais comme une collection non ordonnée de motifs locaux caractéristiques.

La première étape, implémentée dans la fonction *construire_vocabulaire*, consiste à définir ce que sont nos mots. Nous avons constitué une base d'apprentissage contenant plusieurs vignettes de Charlie (dossier */data_bow*, comme expliqué en préambule de la partie 5). Pour chaque image, un sur-échantillonnage ($x2$) est appliqué pour garantir la détection des détails fins (même raisonnement que précédemment), puis les descripteurs SIFT sont ensuite extraits.

L'ensemble de ces descripteurs est ensuite projeté dans un espace vectoriel. Nous utilisons l'algorithme de clustering K-Means pour partitionner cet espace en régions. Les centroïdes de ces clusters définissent alors les mots du vocabulaire. Plus concrètement, un "mot" de ce dictionnaire peut représenter une caractéristique de l'image, comme un coin de lunette, une transition rouge-blanc verticale ou encore une courbure de bonnet, dont la définition est faite en utilisant l'ensemble de la base de données. Comme expliqué précédemment, notre base de données n'est pas suffisamment complète pour avoir des mots suffisamment précis, ce qui a été un frein dans notre avancée et limite la qualité de nos résultats.

Maintenant que le dictionnaire de mot est établi, nous devons décrire chaque ROI de la planche. Notre algorithme, détaillé dans la fonction *extraire_hists*, ne se contente pas du Bow classique mais opère une fusion selon deux critères : la ressemblance aux mots du dictionnaires et la ressemblance colorimétrique.

Pour une ROI donnée, chaque descripteur SIFT est associé à son "mot" le plus proche dans le dictionnaire via l'algorithme FLANN [6].

Ensuite, effectuons un post-traitement : les histogrammes résultants sont normalisés (norme L1), puis nous appliquons une racine carrée élément par élément (Hellinger Kernel). Cette transformation (appelée "RootSIFT" dans la

littérature [2]), permet d'atténuer l'effet des motifs visuels répétitifs et améliore la comparaison de distributions. En parallèle, nous calculons un histogramme de couleurs dans l'espace HSV.

La dernière étape (implémentée dans la fonction *selectionner_suspects*) consiste à comparer la signature de chaque ROI candidate avec notre base de référence. Puisque Charlie change de posture d'une image à l'autre, il n'est pas pertinent de comparer la ROI avec une "moyenne" des références. Nous avons opté pour une stratégie de Max-Pooling : le score de la ROI est défini par sa similarité maximale avec l'une des 21 images de la base (la ROI est comparée à chaque référence individuellement, et seul le meilleur score est conservé). Pour comparer deux histogrammes H_{roi} et H_{ref} , nous n'utilisons pas la corrélation simple, mais l'Intersection d'Histogrammes :

$$Score(H_{roi}, H_{ref}) = \sum_{i=1}^n \min(H_{roi}[i], H_{ref}[i]) \quad (1)$$

Cette mesure a l'avantage d'être robuste aux occultations, donc si Charlie est caché derrière un personnage, l'intersection ne comptabilisera que les caractéristiques communes sans pénaliser lourdement les parties manquantes.

Le score final est une moyenne pondérée (dont les coefficients ont été déterminés manuellement selon des critères visuels de performance) :

$$Score_{Final} = 0.9 \times Score_{BoW} + 0.1 \times Score_{Couleur} \quad (2)$$

Cette stratégie hybride permet de récupérer des candidats que le SIFT seul aurait manqués (par manque de texture) ou que la couleur seule aurait confondu (faux positifs rouges et blancs), offrant ainsi une bonne robustesse face aux déformations et aux occultations partielles.

En ordonnant les ROI par score, on obtient un premier classement de nos suspects. Les cinq premiers suspects dans notre planche exemple sont représentés sur la figure ci-dessous. On observe que Charlie se situe premier, tandis que Félicie, son amie lui ressemblant, est classée troisième.



Figure 6. Top 5 des suspects à l'issue de l'étape 3

5.5. Étape 4 : Mise en correspondance et vérification géométrique (RANSAC)

L'étape précédente (BoW) permet de filtrer les zones selon leur texture globale, mais elle ne garantit pas la cohérence spatiale des motifs. Une zone rayée rouge et blanche désordonnée (comme par exemple un parasol) peut encore réussir à tromper notre modèle. Cette nouvelle étape vise à vérifier si les points d'intérêt détectés s'agencent géométriquement pour former la structure caractéristique de Charlie, donc si nous pouvons trouver une homographie pour coller notre candidat à un des modèles de notre base de données.

Contrairement aux étapes précédentes, nous utilisons ici le détecteur AKAZE [1], plus performant que SIFT sur les contours nets des dessins. Avant l'extraction, une étape de pré-traitement est appliquée sur notre candidat : un upscaling d'un facteur trois par interpolation bicubique, comme cela a déjà été expliqué. Cela permet de révéler des détails haute fréquence invisibles à l'échelle originale. Le contraste local est aussi fortement accentué (CLAHE) pour tenter d'augmenter le nombre de keypoints détectés.

Pour chaque candidat issu du BoW, nous tentons de l'apparier avec nos modèles de référence. Le filtrage s'opère en deux temps. Nous ne conservons d'abord que les paires de points validant le test du ratio (*Lowe's ratio test* avec un seuil relâché à 0.8), éliminant ainsi les ambiguïtés répétitives propres aux foules denses. Puis nous cherchons ensuite à calculer une matrice d'homographie H reliant le modèle au candidat. L'algorithme RANSAC [4] sélectionne aléatoirement des sous-ensembles de points pour estimer H avec un seuil de tolérance de 5.0 pixels, et ne garde que le modèle qui maximise le nombre d'inliers (points conformes au modèle géométrique).

Trouver une homographie n'est pas forcément suffisant pour s'assurer de la présence du modèle chez le candidat sélectionné, donc nous intégrons à notre algorithme une vérification de vraisemblance de taille. À partir de la matrice H , nous extrayons les facteurs d'échelle locaux s_x et s_y :

$$s_x \approx \sqrt{H_{0,0}^2 + H_{1,0}^2} \quad \text{et} \quad s_y \approx \sqrt{H_{0,1}^2 + H_{1,1}^2} \quad (3)$$

Nous imposons la contrainte stricte $0.2 < s < 4.0$ (valeurs déterminées empiriquement), ce qui signifie que nous rejetons tout candidat qui nécessiterait d'étirer ou de compresser le modèle de Charlie de manière irréaliste pour coller à l'image. Enfin, les candidats validés sont classés selon un score de confiance géométrique (*inliers*²/*matches*) pour isoler les meilleurs suspects.

Le résultat de cette étape sur notre planche exemple est illustrée ci-dessous. On observe que le véritable Charlie reste le suspect numéro 1, tandis que Félicie est maintenant passée numéro 2. Cela fait sens car ils sont très similaires en terme de géométrie.

Suspect n°1 : suspect classé n°1 dans l'étape 3.
Suspect n°2 : suspect classé n°3 dans l'étape 3.
Suspect n°3 : suspect classé n°8 dans l'étape 3.
Suspect n°4 : suspect classé n°19 dans l'étape 3.
Suspect n°5 : suspect classé n°14 dans l'étape 3.

Figure 7. Top 5 des suspects à l'issue de l'étape 4

5.6. Étape 5 : Raffinement sémantique et décision finale

La dernière étape de notre pipeline vise à faire un choix final parmi la liste des candidats dressée dans l'étape précédente, dont l'analyse des résultats a montré que la validation géométrique pouvait accorder un score élevé à d'autres personnages possédant une structure de pull rayé identique (tels que Félicie). Pour désigner un suspect principal, nous avons implémenté une fonction d'arbitrage qui combine le classement de l'étape 4 et des critères sémantiques. Cette dernière étape repose sur trois approches :

1. Analyse de la signature chromatique globale : Nous mesurons ici la présence brute des trois couleurs fondamentales de Charlie (rouge, blanc et teinte chaire) sur l'ensemble de la région d'intérêt via l'espace HSV. Cela nous permet d'analyser des régions englobantes potentiellement décentrées ou incluant des éléments du décor.
2. Filtre anti-sosie : Charlie possède une morphologie visuelle spécifique où la surface de sa peau est généralement proportionnelle à celle du bonnet. À l'inverse, des personnages ayant les mêmes vêtements exposent généralement une surface de peau supérieure (Félicie a par exemple un visage dégagé et des jambes nues, alors que Charlie a des lunettes et un pantalon). Nous calculons donc un ratio $R = \text{pixels}_{\text{peau}} / \text{pixels}_{\text{rouge}}$. Si ce ratio excède un seuil de 1,4, le candidat subit une pénalité sévère (multiplicateur 0,4), permettant d'écartier ces sosies.
3. Prise en compte de la confiance statistique issue du BoW : Nous réintroduisons le rang initial obtenu lors de l'étape du BoW en appliquant un bonus de rang défini par $1/\sqrt{\text{rang}_BOW}$. Pour deux candidats à géométries égales, cela permet de privilégier le candidat qui présente la plus forte ressemblance avec notre dictionnaire de mots visuels de référence.

Finalement, le score final est le produit de la confiance géométrique RANSAC, de la densité de présence des couleurs (calculée logarithmiquement pour éviter la saturation sur les grands objets) et du filtre anti-sosie cités

précédemment :

$$Sc_{Final} = Sc_{Geo} \times \left(\sum \log(1 + p_{xl_c}) \right) \times Pénalité_{Sosie} \times Bonus_{Rang} \quad (4)$$

Le candidat maximisant ce score composite est alors officiellement désigné comme étant notre suspect n°1. Dans le cas de notre planche exemple, le suspect final obtenu est illustré en figure 8. Il s'agit effectivement de Charlie.



Figure 8. Suspect final identifié

6. Évaluation

Notre méthodologie en cascade a été testée sur plusieurs planches du jeu "Où est Charlie?", toujours trouvées sur le même git. Ces planches présentaient chacune des niveaux de difficultés variables, Charlie était plus ou moins caché, plus ou moins grand... Nous pouvons donc mesurer, selon des critères plutôt qualitatifs et visuels, la performance de chaque étape de la cascade, et ce sur chaque planche de notre base de données.

6.1. Efficacité du Pré-traitement et Filtrage des ROI

La première étape, à savoir le filtrage colorimétrique et morphologique, avait pour vocation de diminuer l'espace de recherche des correspondances possibles. Cette étape nous a donné beaucoup de fil à retordre, puisque beaucoup d'approches différentes ont été tentées. D'abord les filtres colorimétriques seuls, qui conservaient trop de parties de la planche par nature du jeu (dont le but est de leurrer le joueur) et s'avéraient donc insuffisamment performants puisque trop de descripteurs devaient alors être analysés. Donc en plus de ces filtres colorimétriques, nous avons également filtré en fonction des motifs du pull de Charlie, à savoir un empilement de rayures blanches et rouges. Nous avons aussi ajouté la forme approximative des zones

d'intérêts, Charlie étant souvent debout, les ROI sont très souvent, si ce n'est tout le temps, des rectangles verticaux. En menant ces étapes nous arrivons à une réduction de l'espace de recherche relativement satisfaisante, avec environ une soixantaine de zones restantes à comparer et avec Charlie qui y apparaît systématiquement.

6.2. Evaluation de la validation géométrique et robustesse du modèle

Cette dernière étape vise à valider l'existence d'homographies entre les modèles de notre base de données et le candidat identifié sur la planche de jeu. L'analyse des performances de cette phase révèle des nuances selon la composition de notre base de références.

Afin d'évaluer la capacité de généralisation de notre algorithme, nous avons mené des tests croisés sur l'ensemble des planches du jeu de données *Hey-Waldo*. Un point important doit être soulevé : notre base de données comporte nativement les vignettes extraites de presque toutes les planches cibles (à l'exception de la planche n°19). Pour assurer l'honnêteté de nos mesures, nous avons donc testé le modèle selon deux scénarios distincts :

- **Scénario de correspondance exacte** : Nous laissons la vignette de la planche actuelle dans la base de données. Dans ce scénario, l'algorithme RANSAC trouve une correspondance exacte et notre modèle trouve systématiquement Charlie.
- **Scénario de généralisation (Leave-one-out)** : Nous retirons de la base de données la vignette correspondant à la planche analysée. La correspondance géométrique est donc plus complexe à établir, et les résultats ne sont pas aussi bons que dans le premier scénario. En effet, l'algorithme RANSAC ne permet plus de bien identifier Charlie. Ainsi, en moyenne, il ne fait pas monter le ROI contenant Charlie dans le classement des suspects et l'étape de raffinement finale, quant à elle, ne le fait monter que de quelques places. Résultats : le classement final du vrai Charlie oscille entre 1er (Charlie est trouvé sur 4 des 21 planches) et environ 30ème (sur 60 ROI au total), mais il est généralement trouvé dans le top 10.

Cependant, les planches n°3, n°14, n°18 et n°19 viennent confirmer la pertinence de notre approche. En effet, Charlie y est bien identifié malgré son absence dans la base de données (second scénario). Sur les planches 14 et 19, la raison pour laquelle la détection est plus efficace est directement liée à la représentation de Charlie, dont le corps apparaît presque entièrement et est légèrement détaché du décor. Il est donc plus visible sur ces planches, et son corps offre une richesse de points d'intérêt AKAZE supérieure. Cela nous pousse à penser que notre modèle peut s'avérer beaucoup plus efficace, et systématiquement indiquer la ROI où se trouve Charlie comme celle présentant le plus

de correspondance avec les éléments de la base de données, si cette dernière était plus complète et comportait par exemple une centaine d'images de Charlie plutôt qu'une petite vingtaine. Néanmoins, nous n'avons pas trouvé de base de données avec plus d'images exploitable, il faudrait donc probablement en construire une à la main pour s'assurer des performances du modèle.

7. Conclusion

Dans le cadre de ce projet, nous avons conçu une chaîne algorithmique en cascade visant à isoler le personnage de Charlie au sein d'environnements saturés et complexes. Notre méthodologie a mobilisé une succession de concepts clés du cours de Reconnaissance Visuelle : un filtrage colorimétrique et morphologique pour la réduction de l'espace de recherche, une modélisation statistique par Bag of Visual Words enrichie d'une signature chromatique HSV, et enfin une validation structurelle par estimation d'homographie via l'algorithme RANSAC.

Si cette architecture a prouvé sa capacité à réduire notablement l'espace de recherche, l'évaluation de l'étape RANSAC a mis en lumière l'influence cruciale de la base de référence sur le classement final. Nos tests croisés sur le jeu de données *Hey-Waldo* ont révélé que si la présence du modèle exact dans la base garantit une identification immédiate, le véritable défi réside dans la capacité de généralisation du modèle.

En adoptant une approche de test plus exigeante (scénario Leave-one-out), nous avons constaté qu'en l'absence du modèle identique, la cible se maintient de manière régulière dans le top 10 des régions d'intérêt les plus probables, oscillant parfois jusqu'à la première place. Bien que cette configuration soit complexe en raison de la grande variabilité des postures et des occultations, ce résultat atteste de la pertinence de notre filtrage : même sans correspondance parfaite, le pipeline parvient à extraire Charlie du "bruit" visuel de la foule.

Les performances obtenues sur les planches n°3, n°14, n°18 et n°19 viennent confirmer cette analyse. Sur ces planches, Charlie est correctement identifié en première position malgré son absence de la base de données. Pour les planches 14 et 19 notamment, cette efficacité s'explique par une représentation plus dégagée du personnage, dont le corps offre une richesse de points d'intérêt AKAZE supérieure. Ces succès démontrent que notre architecture algorithmique est théoriquement saine et performante, mais qu'elle est actuellement limitée par la dimension de notre échantillonnage de référence.

En conclusion, ce projet prouve qu'une approche hy-

bride mêlant statistiques locales (BoW) et géométrie rigide (RANSAC) est une stratégie efficace pour la recherche d'instance en milieu encombré. Pour atteindre une précision systématique au premier rang, l'étape suivante consisterait à enrichir significativement la base de modèles de bonnes résolutions afin de couvrir l'intégralité du domaine de variation du personnage.

References

- [1] Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2013. [2](#) [6](#)
- [2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3415. IEEE, 2012. [5](#)
- [3] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*. Prague, 2004. [2](#) [5](#)
- [4] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 1981. [2](#) [6](#)
- [5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004. [2](#) [4](#)
- [6] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 331–340, 2009. [5](#)
- [7] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1), 1991. [2](#)
- [8] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems IV*, pages 474–485, 1994. [4](#)