

Harun TABAKOVIC
Noheyla YOUNESSE

Nom de l'application : Zoovision

Méthodologie et choix techniques

L'objectif de Zoovision est d'**identifier un animal présent** dans une **image requête** en la **comparant à une base d'images** à l'aide d'algorithmes de détection et de mise en correspondance de points caractéristiques. Pour cela nous avons utilisé des techniques de vision par ordinateur vues en travaux dirigés.

Elle vise à illustrer le fonctionnement des algorithmes de reconnaissance d'images appliqués à des animaux, un domaine présentant des formes et textures variées.

L'utilisateur fournit une image d'animal. L'application extrait des points caractéristiques à l'aide de l'algorithme ORB, compare ces points avec ceux des images de la base de données, puis affiche les images les plus similaires.

Cycle de fonctionnement de l'application

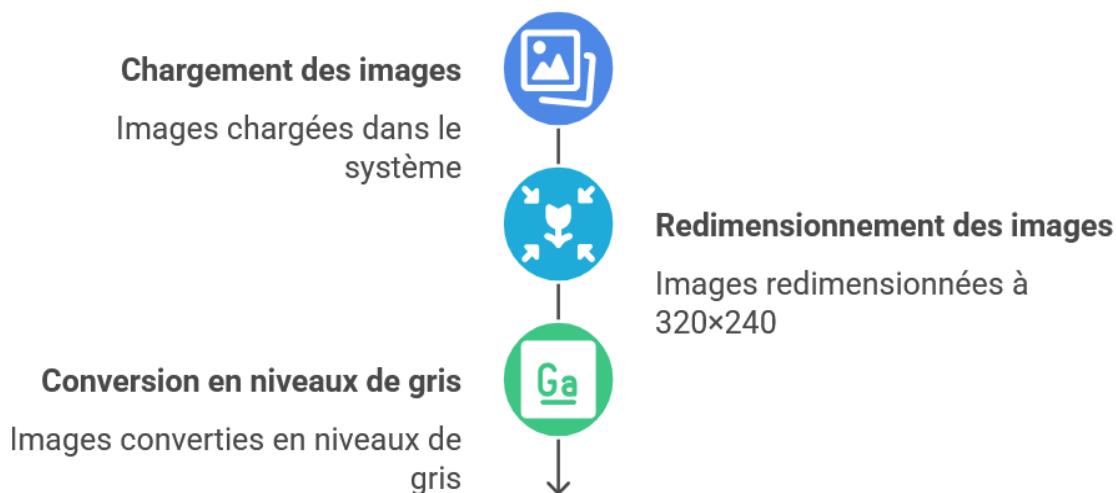


Prétraitement des images

Toutes les images, qu'il s'agisse de l'image requête ou des images de la base de données, sont prétraitées en amont de façon à qu'elle soit tout identique afin de garantir une comparaison cohérente. Chaque image est d'abord **chargée en couleur**, puis **redimensionnée** à une taille fixe (400×300 pixels). Ce redimensionnement permet de limiter les variations dues à la résolution et de réduire le coût de calcul.

Les images sont ensuite **converties en niveaux de gris**, car les algorithmes de détection de points caractéristiques utilisés par la suite fonctionnent sur l'intensité lumineuse et non sur la couleur.

Processus de prétraitement des images pour Zoovision

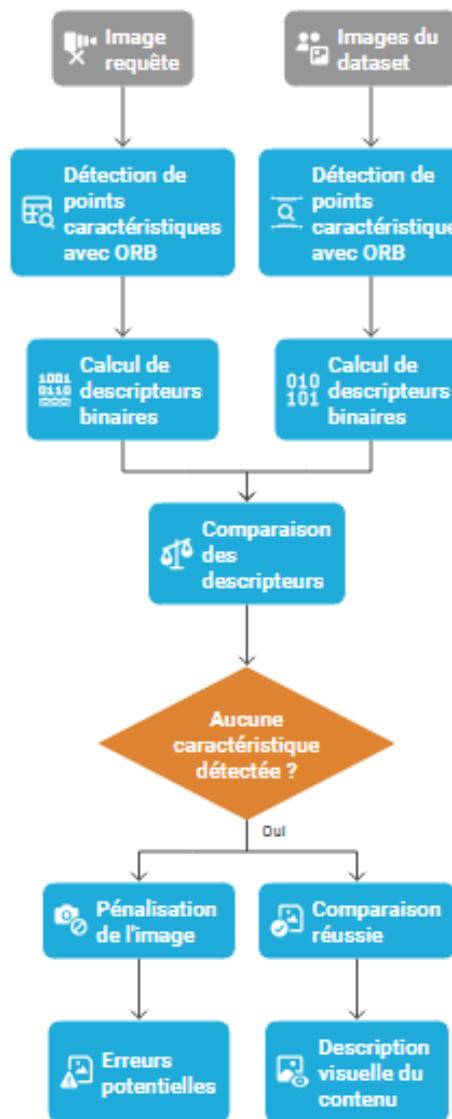


Extraction de points caractéristiques avec ORB

Pour décrire le contenu visuel des images, **l'algorithme ORB** a été utilisé. ORB permet de détecter des points caractéristiques (coins, détails, structures locales) et d'associer à chacun d'eux un descripteur binaire.

Les descripteurs sont calculés pour l'image requête ainsi que pour chaque image de la base de données. Si aucune caractéristique n'est détectée sur une image (image floue ou peu texturée), celle-ci est pénalisée lors de la comparaison et cela peut créer des erreurs.

Processus d'extraction de points caractéristiques avec ORB



Mise en correspondance des descripteurs

La comparaison entre l'image requête et les images de la base de données est réalisée à l'aide de l'algorithme **BFMatcher** (Brute Force Matcher). Étant donné que les descripteurs ORB sont binaires, la distance de Hamming est utilisée pour mesurer la similarité entre deux descripteurs. Pour chaque descripteur de l'image requête, la méthode **knnMatch** est appliquée afin de récupérer les k meilleurs correspondants dans l'image comparée.

Les correspondances obtenues sont ensuite triées par distance croissante, les plus petites distances représentant les correspondances les plus pertinentes. Cette étape permet d'évaluer la qualité de la similarité entre deux images.

Définition du score de similarité

Un score de similarité est attribué à chaque image de la base afin de permettre leur classement par ordre de pertinence par rapport à l'image requête. Dans un premier temps, ce score repose sur le nombre de correspondances trouvées entre l'image requête et chaque image de la base de données. Plus ce nombre est élevé, plus les deux images sont considérées comme similaires.

Les correspondances obtenues sont triées par distance croissante, les plus faibles distances correspondant aux correspondances les plus pertinentes. Parmi l'ensemble des correspondances, seules les **25 meilleures** sont conservées afin de limiter l'influence des correspondances moins fiables.

Le score de similarité est alors calculé en comptant le nombre de correspondances, parmi ces 25 meilleures, dont la distance est inférieure à un **seuil fixé expérimentalement à 50**. Ce nombre constitue le score final associé à l'image de la base de données.

L'image de la base obtenant le score de similarité le plus élevé est considérée comme la meilleure correspondance avec l'image requête.

La classe prédite correspond alors au **label extrait du nom du fichier** de cette image (par exemple : *chat_1.jpg* → *chat*).

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

haruntabakovic@Haruns-MacBook projetIA % python3 main2.py
chat_17.jpg => score: 2
girafe_10.jpg => score: 25
chien_7.jpg => score: 10
chien_14.jpg => score: 12
chien_15.jpg => score: 3
chien_6.jpg => score: 25
chien_17.jpg => score: 19
girafe_2.jpg => score: 7
girafe_3.jpg => score: 18
chat_18.jpg => score: 16
chien_16.jpg => score: 13
chien_5.jpg => score: 5
chien_1.jpg => score: 10
chien_12.jpg => score: 10
girafe_7.jpg => score: 7
girafe_6.jpg => score: 2
chien_13.jpg => score: 14
chien_2.jpg => score: 5
chien_11.jpg => score: 13
girafe_4.jpg => score: 11
girafe_5.jpg => score: 5
chien_10.jpg => score: 9
chien_3.jpg => score: 10
mouton_3.jpg => score: 15
cheval_5.jpg => score: 6
vache_04.jpg => score: 25
vache_10.jpg => score: 25
chat_1.jpg => score: 3
cheval_4.jpg => score: 8
mouton_2.jpg => score: 20
cheval_6.jpg => score: 12
vache_06.jpg => score: 25
chat_2.jpg => score: 21
cheval_7.jpg => score: 3
mouton_5.jpg => score: 7
cheval_3.jpg => score: 19
chat_6.jpg => score: 10
vache_02.jpg => score: 15
vache_03.jpg => score: 17
chat_7.jpg => score: 4
cheval_2.jpg => score: 17
mouton_4.jpg => score: 17
mouton_6.jpg => score: 10
chat_5.jpg => score: 25
chat_4.jpg => score: 24
cheval_1.jpg => score: 5
mouton_7.jpg => score: 21
Prediction : cheval
Meilleure image : cheval_17.jpg
Score : 25
[]
```

Exemple de comparaison d'images et de prédition finale basée sur le score de similarité.

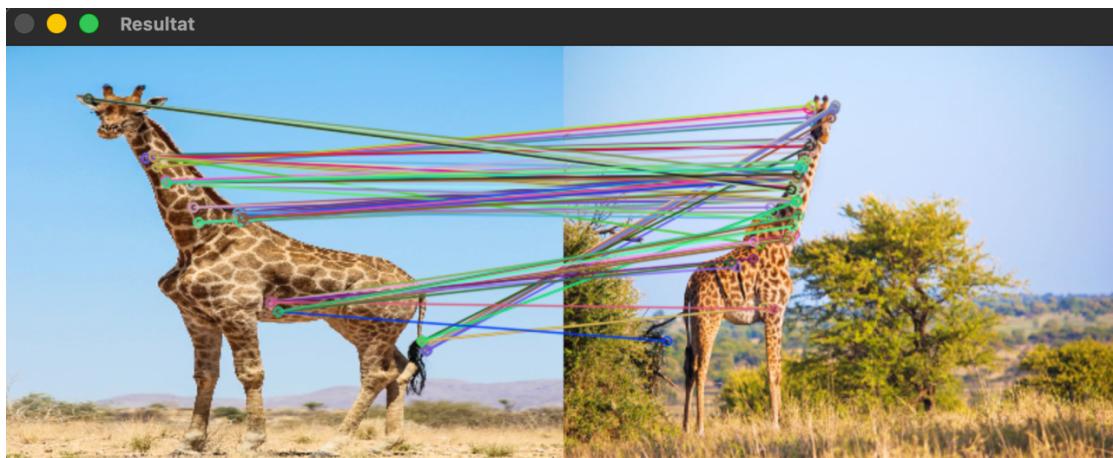
Résultats obtenus

Cas de reconnaissance correcte (résultats positifs)

Plusieurs tests montrent que l'algorithme fonctionne correctement lorsque les animaux présentent des caractéristiques visuelles distinctives.

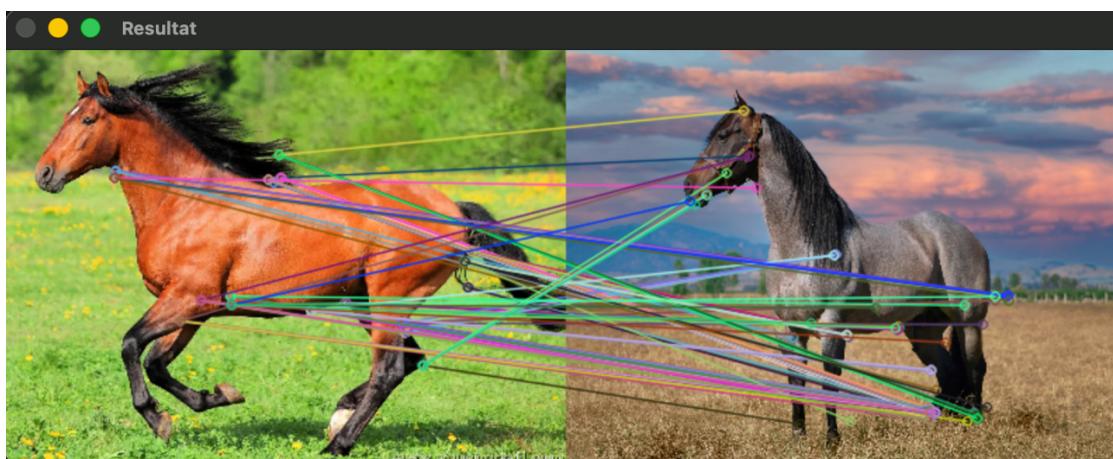
Girafe

La reconnaissance de la girafe est généralement correcte. Les motifs très spécifiques de la robe ainsi que la forme du cou génèrent de nombreux points d'intérêt pertinents. Les correspondances observées sont majoritairement concentrées sur l'animal, ce qui conduit à un score élevé pour la bonne classe.



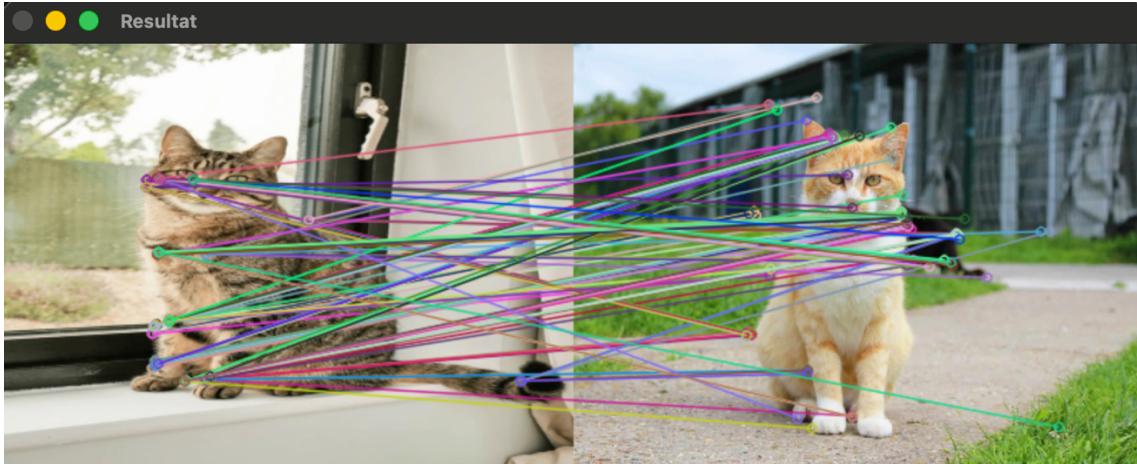
Cheval

Les images de chevaux sont également bien reconnues dans la majorité des cas. La crinière, la tête et les contours du corps fournissent suffisamment de points clés robustes, même lorsque l'arrière-plan varie.



Chat

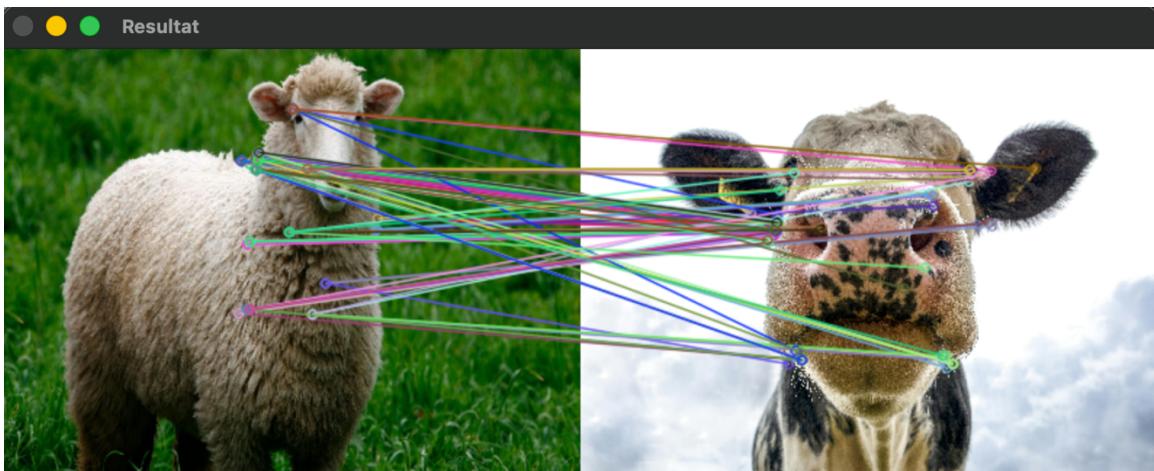
Les images de chats sont généralement bien reconnues par l'algorithme. Les traits distinctifs du visage (yeux, museau, oreilles), ainsi que les motifs du pelage, génèrent de nombreux points caractéristiques pertinents.



Cas de reconnaissance incorrecte (résultats négatifs)

Malgré de bons résultats globaux, l'algorithme présente aussi des erreurs.

Confusion mouton / vache



Certaines images de moutons sont mal classées comme des vaches. Cette confusion s'explique par :

- Une forme globale assez similaire,
- Des zones de pelage peu texturées ou similaire,
- Des correspondances influencées par l'arrière-plan (herbe, ciel).

Les visualisations montrent que plusieurs correspondances relient des zones non pertinentes ou appartenant au fond plutôt qu'à l'animal lui-même.

Analyse critique, limites et difficultés

Limites de l'approche

L'approche basée sur les descripteurs locaux ORB présente plusieurs limites. Tout d'abord, l'algorithme ne distingue pas l'objet principal de l'arrière-plan : les points caractéristiques peuvent être détectés aussi bien sur l'animal que sur le fond, ce qui peut générer des correspondances non pertinentes.

De plus, le score de similarité repose sur un critère simple basé sur la distance ou le nombre de correspondances, sans garantir une cohérence géométrique globale entre les images.

Enfin, les résultats sont sensibles aux variations de luminosité, de cadrage et d'échelle, qui influencent fortement le nombre et la qualité des points détectés.

Difficultés rencontrées

Le choix d'un seuil de distance adapté à l'ensemble des images s'est révélé complexe, car un seuil trop strict ou trop permissif impacte fortement les résultats. Certaines classes d'animaux présentent également des caractéristiques visuelles proches, limitant la discrimination par descripteurs locaux. Enfin, le nombre de points détectés varie fortement selon les images, ce qui complique la définition d'un score de similarité stable.

Nous aurions également souhaité développer une application plus interactive permettant à l'utilisateur de sélectionner directement l'animal qu'il souhaite rechercher. Cependant, la mise en place d'une interface graphique complète dépassait le cadre des outils abordés en travaux dirigés et les contraintes de temps du projet.