

Réalisation personnelle :

Analyse des Stratégies de Jeu dans Valorant à l'Aide de la Vision par Ordinateur

Ce document décrit un projet personnel, développé en parallèle de mes études, dans lequel j'essaie d'appliquer mes connaissances en Intelligence artificielle à une situation concrète en lien avec ma passion pour les jeux vidéo. Il constitue un premier challenge réalisé en autonomie complète.

Le code en l'état actuel est disponible sur le dépôt github du projet : https://github.com/trk376/valo_vision.git

1- Présentation du projet :

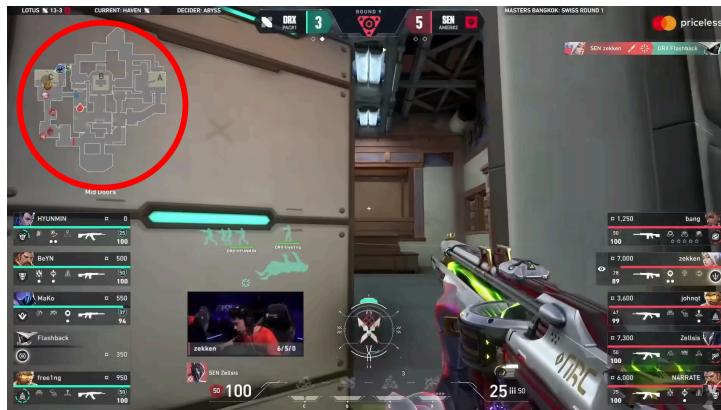
Valorant est un jeu de tir tactique développé par Riot Games. Deux équipes de cinq joueurs s'affrontent avec des objectifs bien définis : l'une doit poser un dispositif explosif sur un site précis, tandis que l'autre doit l'empêcher ou le désamorcer. Chaque joueur contrôle un Agent, un personnage ayant des compétences spéciales (comme des fumigènes pour bloquer la vision ou des drones pour repérer les adversaires). Pour gagner, une équipe doit coordonner ses actions, utiliser intelligemment ses ressources et anticiper les stratégies de l'adversaire.

Objectif du projet :

L'objectif de ce projet est d'utiliser la vision par ordinateur pour analyser les déplacements des joueurs sur la *minimap* à partir de vidéos de matchs compétitifs. En identifiant automatiquement la position des joueurs, nous pouvons :

- Repérer les tendances stratégiques des équipes
- Visualiser les trajectoires et schémas de déplacement
- Aider les joueurs, coachs et analystes à mieux comprendre les stratégies adverses

Ce projet s'adresse aux joueurs compétitifs, aux analystes d'eSports et aux entraîneurs, mais aussi à toute personne curieuse d'explorer l'utilisation de l'IA dans l'analyse de jeux vidéo.



Capture d'écran d'un match de Valorant, avec la minimap située en haut à gauche entourée en rouge. Cette zone est utilisée pour analyser les déplacements des joueurs.

2- Problématique et motivation

L'analyse des stratégies en *Valorant* repose sur la compréhension des déplacements des joueurs. Cependant, les outils actuellement disponibles, comme l'API officielle de *Riot Games*, sont limités : ils ne fournissent la position des joueurs qu'à certains moments clés (éliminations, pose et désamorçage du Spike). Ces données sont utiles, mais insuffisantes pour observer les mouvements en continu et repérer les schémas stratégiques des équipes.

Or, dans un contexte compétitif, comprendre comment une équipe se positionne, effectue ses rotations et contrôle la carte est essentiel pour anticiper ses décisions.

Ce projet répond à ce besoin en utilisant un algorithme de vision par ordinateur pour extraire et analyser les déplacements directement depuis les vidéos de match (VODs). Cela permet de :

- Obtenir une cartographie complète des mouvements des joueurs
- Détecter des patterns de jeu et adaptations stratégiques
- Générer des outils de visualisation clairs et interactifs pour faciliter l'analyse

En offrant une vision plus détaillée du jeu, cet outil devient une ressource précieuse pour les analystes et les équipes compétitives.

3- Méthodologie

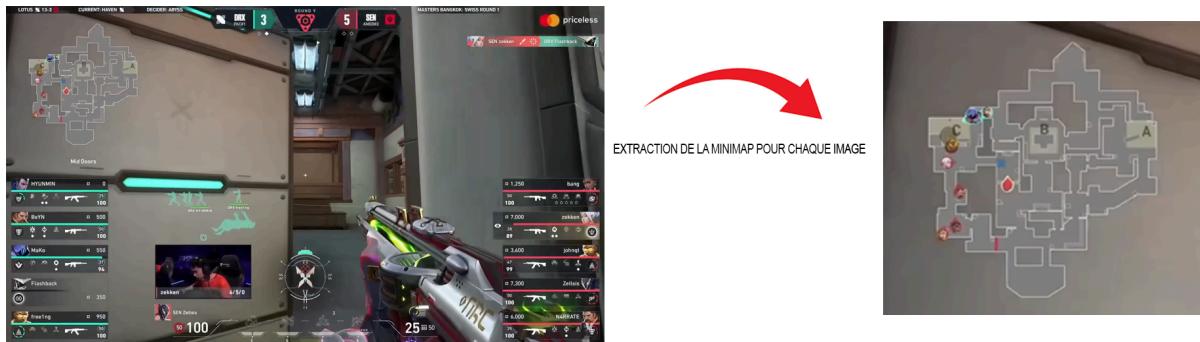
Mon approche repose sur plusieurs étapes clés : l'extraction des minimaps depuis les vidéos, la détection des joueurs avec un algorithme de vision par ordinateur, l'analyse de leurs déplacements et la création de visualisations adaptées à l'étude des stratégies d'équipe.

3.1- Extraction des données

La première étape consiste à récupérer les informations des matchs en traitant directement les VODs (enregistrements vidéo). Pour cela :

- Un script Python utilisant OpenCV extrait une image chaque seconde (fréquence ajustable).
- L'image est ensuite rognée pour ne conserver que la minimap, qui est toujours située en haut à gauche de l'écran.

Cette méthode permet de disposer d'une séquence continue de positions des joueurs, indépendamment des limites de l'API officielle de *Riot Games*.



Exemple d'une image extraite d'une VOD, avec un focus sur la minimap. L'image est traitée pour isoler cette zone et permettre l'analyse des déplacements.

3.2- Annotation et Entraînement du modèle

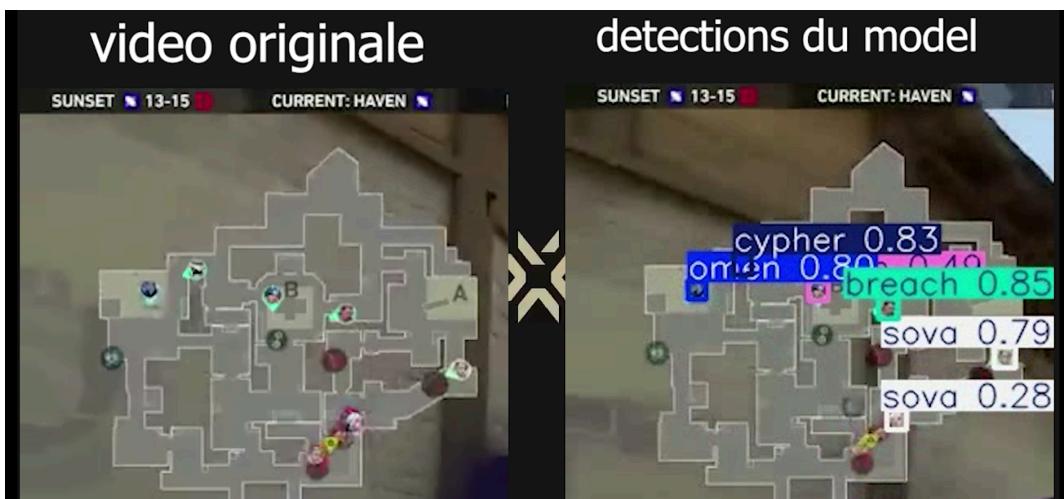
Avant de pouvoir détecter les icônes des joueurs sur la minimap, il est nécessaire d'entraîner le modèle en lui fournissant un ensemble de données annotées manuellement. Cette étape consiste à annoter les images extraites à l'aide d'un outil d'annotation (ex. [cvat.ai](#)) : chaque minimap extraite est analysée image par image pour identifier et délimiter les icônes et associer une étiquette à chaque type d'agent détecté.

Une fois un nombre suffisant d'images annotées (500 à ce jour), on génère un fichier de labels au format YOLO (.txt) contenant les coordonnées de chaque icône. Ces données sont utilisées pour entraîner le modèle afin qu'il puisse reconnaître automatiquement les icônes.

L'entraînement du modèle se fait à l'aide du framework [Ultralytics YOLOv8](#) sous [PyTorch](#), il est réalisé en ajustant certains hyperparamètres (ex. nombre d'époques, taux d'apprentissage, taille du batch) afin d'optimiser la précision du modèle et minimiser les erreurs de détection.

3.3- Détection des joueurs sur la minimap

Pour identifier la position des joueurs sur la minimap, nous utilisons le modèle YOLO, un algorithme de détection d'objets performant.



Détection des icônes des agents sur la minimap grâce à l'algorithme YOLO. Chaque joueur est identifié par son icône. video complète : <https://youtu.be/acM1gPvnjqg>

3.4- Suivi et analyse des déplacements

Contrairement à d'autres approches utilisant un algorithme de tracking, ici nous effectuons une détection image par image. Cette méthode est possible car :

- Un même agent ne peut pas apparaître plusieurs fois dans la même équipe, ce qui évite toute ambiguïté dans l'attribution des positions.
- Si un agent disparaît temporairement (ex. icône cachée par un autre joueur), une solution en cours d'étude consiste à interpoler sa position entre deux instants où il est détecté, en supposant un mouvement uniforme.

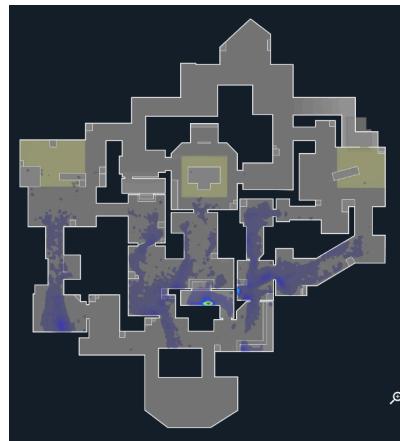
L'objectif est de reconstituer en continu la trajectoire de chaque joueur sans perte d'information critique.

3.5- Visualisation et exploitation des données de déplacements

Les données collectées sont ensuite utilisées pour générer des dashboards analytiques avec différentes représentations visuelles :

- Heatmaps : visualisation des zones les plus fréquentées en fonction de paramètres comme le rôle des agents, l'économie du round ou encore l'état du Spike (posé ou non).
- Trajectoires des joueurs : affichage des déplacements sur plusieurs rounds pour détecter des patterns récurrents (ex. un joueur qui emprunte systématiquement le même itinéraire pour attaquer un site).
- Comparaison entre rounds : superposition des trajectoires à différents moments du round, afin d'identifier des tendances stratégiques.

Ces outils permettent aux analystes et aux équipes compétitives d'extraire des informations clés sur les habitudes des adversaires et d'adapter leurs décisions en conséquence.



*Exemple de heatmap illustrant les zones où les joueurs attaquants sont éliminés dans les 10 premières secondes. Ce type de visualisation permet d'identifier les zones de danger et les tendances des engagements précoce*s. (image issue de [VStats.gg](#))

5- Etat d'avancement , Limites et Améliorations

État d'avancement actuel

À ce stade, le modèle de détection des icônes sur la minimap est fonctionnel mais encore limité. Actuellement :

- Le modèle reconnaît 6 agents parmi ceux disponibles dans Valorant.
- Le taux de détection est globalement bon, mais il subsiste encore quelques erreurs, notamment sur certaines icônes qui se chevauchent ou sont partiellement visibles.
- Le modèle ne reconnaît pas encore la couleur des icônes, ce qui signifie qu'il ne peut pas différencier les équipes (alliés vs adversaires).

Améliorations envisagées

Étendre l'entraînement à tous les agents du jeu

- Ajouter des images annotées pour tous les agents existants dans Valorant.
- Ré-entraîner le modèle avec plus de données pour améliorer la robustesse des détections.

Améliorer la précision de détection

- Tester d'autres architectures YOLO (par ex. YOLOv8m ou YOLOv8l au lieu de YOLOv8s) pour une meilleure performance.
- Affiner les hyperparamètres et augmenter le nombre d'époques d'entraînement.

Intégrer la reconnaissance des couleurs

- Une solution possible est d'ajouter un second modèle basé sur l'analyse des couleurs, qui détectera si une icône est entourée de bleu ou de rouge.
- Une autre approche pourrait être d'entraîner YOLO sur des images où la couleur fait partie des annotations.

Détection automatique des débuts et fins de round

- Actuellement, les rounds doivent être découpés manuellement pour analyser les trajectoires.
- Une amélioration serait d'automatiser cette détection, en s'appuyant par exemple sur des indicateurs visuels comme le timer de la partie.

Ces améliorations permettront d'obtenir un outil plus précis, plus automatisé et plus facilement adaptable aux évolutions du jeu.

6- Conclusion et Perspectives

Ce projet explore l'utilisation de la vision par ordinateur pour analyser les déplacements des joueurs dans *Valorant* et fournir des outils de visualisation stratégiques. En contournant les limitations de l'API officielle de *Riot Games*, nous

avons développé un système capable de suivre en continu les trajectoires des agents directement depuis les vidéos de match (*VODs*).

Grâce à la détection des icônes des joueurs sur la minimap et à l'analyse de leurs mouvements, ce projet offre un nouvel angle d'approche pour les analystes et joueurs compétitifs, leur permettant de mieux comprendre les stratégies adverses et d'adapter leur gameplay en conséquence.

Cependant, plusieurs améliorations restent à apporter :

- Amélioration du modèle YOLO pour détecter tous les agents et s'adapter aux mises à jour du jeu.
- Automatisation du découpage des rounds, afin d'éviter la manipulation manuelle des vidéos.
- Affinage des méthodes d'interpolation des déplacements, pour mieux gérer les cas où les icônes des joueurs disparaissent temporairement.
- Ajout d'outils interactifs permettant une analyse dynamique et personnalisable des matchs.

À terme, ce système pourrait être intégré dans des plateformes d'analyse esport ou être utilisé par des équipes professionnelles pour affiner leur préparation stratégique. L'objectif est de rendre l'analyse tactique plus accessible, plus rapide et plus précise, en exploitant pleinement le potentiel de la vision par ordinateur dans l'univers du jeu compétitif.