



Apprentissage profond

Reconnaissance d'un personnage de League of Legends

Saboureau Elio - Abounaceur Younes - Dang Alexandre - Rivière Alexandre

Département Sciences du Numérique - Systèmes Logiciels
2020-2021

Table des matières

1	Description du projet	3
2	Récupérer et utiliser les images	4
2.1	Récupérer les images	4
2.2	Définir les ensembles	4
3	Complexité du problème et les résultats attendus	5
3.1	Complexité du problème	5
3.2	Résultats attendus	5

Table des figures

1	Partie de League of Legends avec les personnages encadrés	3
2	Format d'image utilisé	3
3	3 exemples de personnages	4
4	Caitlyn avec le skin Battle Academia	5

1 Description du projet

Notre projet s'articule autour du jeu vidéo [League of Legends](#) développé par Riot Games. L'une des problématiques principales est l'analyse des parties de très haut niveau. En effet, cela demande l'analyse de longues heures de jeux avec énormément d'informations à prendre en compte chaque seconde. Ne serait-il pas possible d'automatiser ça ?

De nombreux outils utilisent déjà l'intelligence artificielle afin d'analyser une partie à postériori. Qui a le mieux réussi sa partie ? Contre qui ? Avec quel style de jeu ? Voici beaucoup de questions auxquelles ce genre d'outil apporte une réponse. Cependant, l'analyse en temps réel d'une partie n'est pas possible et cette analyse à postériori manque de finesse.



FIGURE 1 – Partie de League of Legends avec les personnages encadrés

Un outil efficace analyserait la partie en temps réel à partir d'un flux vidéo et permettrait de faire ressortir les erreurs et les bonnes décisions tout le long de la partie. L'un des points les plus importants pour obtenir ce genre de résultat et d'être capable de **reconnaitre les personnages sur chaque image de notre flux vidéo**. C'est ici que vient se placer notre projet.

Ainsi, notre projet ne consiste qu'à la reconnaissance d'un personnage sur une image fixe. Afin, de faire correspondre notre sujet aux attentes de nos professeurs, nous nous abstrairons de toutes interfaces graphiques et de tout ce qui pourrait polluer notre image. Chaque personnage est isolé sur notre image (voir l'exemple ci-dessous).



FIGURE 2 – Format d'image utilisé

L'ensemble de nos sources ainsi que de nos données se trouvent [ici](#) (lien GitHub).

2 Récupérer et utiliser les images

2.1 Récupérer les images

League of Legends plus d'une centaine de personnages, afin de simplifier la construction de la base de données, nous n'en traiterons qu'une dizaine.

Pour obtenir des images répondant à toutes nos attentes, nous devons les créer nous-mêmes. Pour cela, nous avons accès au jeu et nous enregistrons des vidéos de nous en train d'utiliser chaque personnage. Ensuite, un script Python vient en extraire les frames pour constituer notre base d'images.

Lors de l'enregistrement de chaque vidéo, nous nous assurons de couvrir différents environnements et avec notre personnage dans différentes positions. Les images n'ont pas d'annotation particulière si ce n'est le nom du personnage.

Voici quelques exemples figurants dans notre base de données.



FIGURE 3 – 3 exemples de personnages

Il pourrait être intéressant de venir annoter chaque image en décrivant son environnement ainsi que sa position afin de pouvoir tirer des conclusions plus facilement quant à la détection d'un personnage ou pas. Cela nécessitant beaucoup de travail, nous avons décidé de le faire qu'une fois que cela sera nécessaire et/ou approuvé par nos professeurs.

2.2 Définir les ensembles

Il est aussi nécessaire de choisir si certaines images serviront pour l'entraînement, la validation ou pour les tests.

Affecter chaque image à un de nos ensembles est arbitraire, dans notre cas cela n'a pas d'importance. Parmi nos images générées automatiquement, nous associons deux-cinquième de nos images pour l'entraînement, deux-cinquième pour la validation et un cinquième pour les tests.

Aussi, pour les tests, nous prendrons à la main des images représentants des cas un peu plus compliqués (avec des effets visuels présents en jeu par exemple).

3 Complexité du problème et les résultats attendus

3.1 Complexité du problème

Notre problème n'est pas simple, ce qui le rend compliqué, c'est le fait d'avoir plusieurs objets de classes différentes sur la même image, mais pour que notre modèle soit capable de détecter plusieurs personnages à la fois dans une partie, il doit d'abord être capable de détecter un seul personnage dans une image. Pour cette raison, nous allons commencer par entraîner le modèle à détecter un seul personnage, ce qui est demandé dans le cadre de ce projet d'apprentissage profond. Par la suite, nous utiliserons ce modèle pour détecter tous les personnages dans une image.

La deuxième difficulté est la ressemblance entre certains personnages du jeu, par exemple entre Yasuo et Yone, Nautilus et Blitzcrank ou encore Xayah et Rakan. Cette ressemblance peut être à la fois physique ou part les compétences du personnage.

La troisième difficulté sera les "skins" des personnages. Un skin de champion fait référence à une apparence alternative pour un personnage donné. Il peut s'agir d'un modèle différent ou simplement d'un changement de couleurs. Ainsi, certains skins ont une apparence très différente de la forme originale d'un champion et peuvent ressembler à d'autres personnages.

Par exemple Caitlyn avec son skin *Battle Academia* (vue ci-dessous), est difficilement différenciable d'autres champions comme Irelia ou Fiora. Nous allons donc travailler dans un premier temps sur l'apparence de base des champions sans skins et ajouterons éventuellement la reconnaissance avec les skins par la suite.

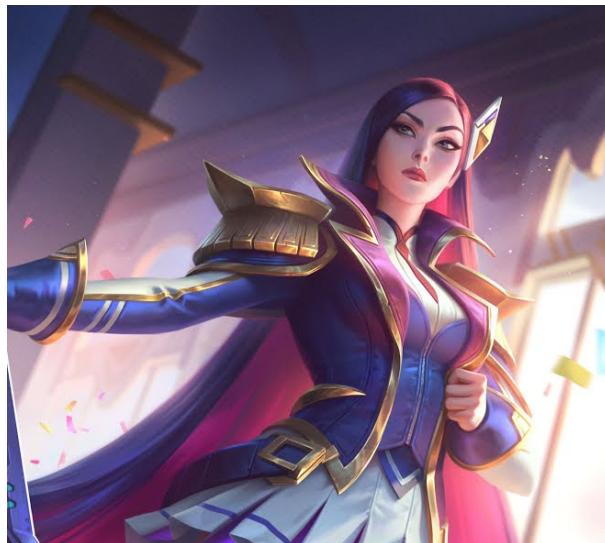


FIGURE 4 – Caitlyn avec le skin Battle Academia

3.2 Résultats attendus

Dans un premier temps, nous souhaitons que le modèle arrive à reconnaître le personnage dans l'image dans une posture simple. Puis, reconnaître le personnage dans une position de combat et/ou en utilisant une compétence.

Dans un second temps, nous souhaitons que le modèle soit capable de reconnaître plusieurs champions présents dans une seule image mais cela demande de dépasser le sujet du projet.