

Recherche de Solutions Techniques Alternatives

Dans cette analyse, nous examinons pour chaque **module** du projet les différentes solutions techniques envisageables, en privilégiant la **précision maximale** lorsque c'est nécessaire. Pour chaque module, nous identifions les meilleures technologies disponibles en 2025, qu'elles soient open source ou propriétaires, en précisant leur statut et les modalités d'utilisation le cas échéant. Enfin, un **tableau comparatif** rapide récapitule ces solutions par module.

Détection d'Objets (Joueurs, Ballon, etc.)

Alternatives open source : Les modèles de détection d'objets de dernière génération s'appuient sur des réseaux de neurones profonds **monoflôts** (type YOLO) ou **Transformers** (type DETR). Trois références émergent pour le contexte football :

- **YOLOv10** – Variante récente de la famille YOLO (You Only Look Once). Elle est optimisée pour la vitesse en éliminant l'étape de Non-Maximum Suppression (NMS) classique, ce qui lui permet d'être plus rapide tout en maintenant une bonne précision ¹. Des rapports indiquent que YOLOv10 améliore encore le compromis précision/efficacité par rapport aux versions précédentes (p.ex. ~94% mAP) ². YOLOv10 est open source (dérivé du projet Ultralytics/Tsinghua) et peut être utilisé librement. Son principal atout est le **temps réel** : il peut surpasser RT-DETR en vitesse pour une précision comparable ³.
- **RT-DETR** – Variante "Real-Time DETR" basée sur l'architecture Transformer DETR optimisée temps réel. RT-DETR élimine le besoin de NMS et améliore la latence tout en offrant une excellente précision (environ 96% mAP) ⁴. C'est un compromis **précision/vitesse** très intéressant pour du 60 FPS en football. Le modèle est publié en open source (par Baidu Research) et sous licence permissive. RT-DETR offre une détection **end-to-end** robuste, notamment efficace pour des classes multiples sans post-traitement lourd.
- **DINO-DETR** – Modèle DETR de pointe (DETR with Improved DeNoising) atteignant l'état de l'art en précision. DINO-DETR a repoussé la précision SOTA sur COCO (par ex ~63.3 AP avec ResNet50) en 2023 ⁵, ce qui correspond dans le contexte à ~97% mAP mentionné pour le mode précision maximale ⁶. Entièrement open source ⁷, il présente la **meilleure précision** parmi les modèles cités, au prix d'une **inférence plus lente**. Pour des analyses hors temps réel où la qualité prime, DINO-DETR est un excellent choix.

Toutes ces solutions étant open source, elles peuvent être intégrées directement (implémentations PyTorch/TensorRT disponibles). Dans un mode **précision maximale**, on privilégiera DINO-DETR pour son accuracy, tout en sachant qu'un modèle comme RT-DETR-X offre presque autant de précision avec un bien meilleur débit ⁴.

Alternatives propriétaires : Il existe peu de modèles de détection propriétaires surpassant l'open source aujourd'hui. Les grands fournisseurs cloud (Google Vision, AWS Rekognition) proposent des API de détection d'objets, mais elles sont génériques et non spécialisées foot. Une solution spécialisée foot serait **Opta Vision** (StatsPerform) qui combine reconnaissance d'entités terrain et tracking, mais elle n'est pas disponible en tant que modèle autonome. En pratique, les meilleurs algorithmes de détection

sont issus de la recherche open source, ce qui fait que les solutions ouvertes citées couvrent déjà l'état de l'art.

En résumé, pour la détection de joueurs/ballon, on recommandera un détecteur multi-modèle configuré avec YOLOv10, RT-DETR ou DINO-DETR selon le besoin (vitesse vs précision), ces trois modèles étant open source

1 8 .

Suivi Multi-Objets (Tracking)

Alternatives open source : Le suivi multi-objets (MOT) repose souvent sur un algorithme associant les détections successives en trajectoires. Actuellement, **ByteTrack** s'impose comme une référence **SOTA**. ByteTrack (proposé en 2021) améliore les trackers existants en incluant *toutes* les détections (même à faible score) dans la phase d'association, ce qui accroît nettement le taux de détection d'objets persistants ⁹. Sur le benchmark MOT17, ByteTrack atteint par exemple **80.3 MOTA** et **77.3 IDF1** (identité) à 30 FPS, surclassant les trackers précédents ¹⁰. Le code ByteTrack est open source (licence MIT) et largement adopté. Il s'intègre parfaitement avec YOLOX/YOLOR et autres détecteurs modernes ¹¹.

D'autres trackers open source existent, par exemple **DeepSORT** (qui ajoute de la re-identification visuelle à SORT) ou ses améliorations **StrongSORT**, **OC-SORT**, etc. DeepSORT est fiable mais un peu dépassé en précision par ByteTrack. **OC-SORT** améliore l'association en cas d'occlusion. Néanmoins, ByteTrack demeure l'option privilégiée pour la précision maximale aujourd'hui, étant donné son **score SOTA** sur plusieurs datasets MOT ⁹ ¹⁰.

Alternatives propriétaires : NVIDIA propose dans son SDK **DeepStream** un tracker optimisé GPU appelé **NvDCF** (Discriminative Correlation Filter) qui intègre à la fois association par IoU et suivi visuel par filtre de corrélation ¹² ¹³. NvDCF est performant en exploitant le GPU, cependant il n'est pas "open source" (fourni en binaire dans DeepStream). Il offre une intégration facile sur plateforme NVIDIA, mais en termes de précision, il est considéré équivalent à DeepSORT d'après NVIDIA ¹⁴. À ce jour, aucune solution propriétaire n'a démontré une avance claire sur ByteTrack sur les métriques publiques.

ByteTrack étant gratuit et hautement performant, on le retiendra comme **meilleure technologie** pour le tracking multi-objets. Son utilisation est directe (implémentations Python/C++ disponibles). Les solutions fermées comme NvDCF peuvent être utilisées via le SDK NVIDIA si l'on travaille dans cet écosystème, mais elles n'apportent pas de gain significatif de précision sur des données standard ¹⁴.

Estimation de Pose (Keypoints 2D/3D) et Analyse Biomécanique

Pour l'analyse technique des gestes des joueurs, il faut extraire la **pose du corps** (positions des articulations) idéalement en 3D, puis en déduire des mesures biomécaniques (angles, vitesses, symétries, etc.).

Alternatives open source (Pose): Deux cadres dominent la pose humaine :

- **MediaPipe Pose** (Google) – Solution open source (Apache 2.0) très populaire. Elle offre un modèle **léger en temps réel** pouvant détecter **33 points clés** du corps (y compris mains et visage simplifiés) à partir d'une seule caméra ¹⁵. MediaPipe privilégie une approche *top-down* efficace (détection de la personne puis pose). Avantages : **rapide**, fonctionne sur CPU/mobile, facile à intégrer (API en Python, C++, etc.), et **3D approchée** (retourne des coordonnées relatives

3D normalisées). C'est idéal pour avoir un pipeline temps réel ou multi-personnes sur du foot. Toutefois, en contrepartie de sa vitesse, la précision des points est un peu moindre que les méthodes lourdes ¹⁶.

- **OpenPose** (Carnegie Mellon Univ.) – Bibliothèque référence pour la pose **multi-personnes** précise. Elle suit une approche *bottom-up* (détection des points puis assemblage en squelettes) et fournit jusqu'à **25 points clés** par personne (et extension main/visage). Son atout majeur est la **haute précision** de détection des articulations, ce qui la rend adaptée aux analyses fines (sport de haut niveau, médical) ¹⁷ ¹⁸. En revanche, OpenPose est très exigeant : chaque image requiert ~160 milliards d'opérations ¹⁹ et le temps de traitement peut se compter en secondes par image sans GPU puissant. De plus, la licence OpenPose n'est **gratuite que pour usage non commercial** – une licence commerciale doit être négociée pour un produit propriétaire ²⁰. On peut toutefois utiliser OpenPose librement pour R&D ou projets open source.
- **Autres** : Des modèles récents combinent détection et pose, p. ex. **YOLOv8-Pose** d'Ultralytics (licence AGPL 3.0, donc open source *virale*) ou **AlphaPose** (open source, par Shanghai Jiao Tong University). **HRNet** (Microsoft) couplé à **DeepLabCut** est aussi utilisé pour une pose 2D très précise. Ces alternatives peuvent surpasser MediaPipe en précision, mais aucune n'a la simplicité et l'optimisation extrême de MediaPipe ou l'exhaustivité d'OpenPose. Pour la 3D, on peut citer **VNect** ou **MediaPipe Holistic** (pose 3D + mains + visage), open source également.

Étant donné l'objectif de **précision maximale**, on pourrait envisager une combinaison : utiliser **OpenPose** pour extraire des points 2D très précis (éventuellement sur quelques images clés si temps réel impossible) puis reconstruire une pose 3D (par triangulation multi-vues si plusieurs caméras, ou via modèle apprentissage profond monoculaire si disponible). **MediaPipe** pourrait servir de *fallback* rapide en cas d'échec ou pour le temps réel, vu sa robustesse et sa vitesse ²¹ ¹⁶.

Alternatives propriétaires (Pose): Une solution commerciale notable était **Wrnch AI**, un SDK de pose 2D/3D temps réel autrefois proposé par la société Wrnch. Wrnch vantait une précision proche d'OpenPose mais optimisée pour des applications industrielles (SDK C++ payant) ²². Nvidia a racheté Wrnch et intégré son moteur dans Omniverse (extension **wrnch AI Pose Estimator**) ²³. Pour utiliser Wrnch, il faut obtenir une licence (souvent via Nvidia Omniverse ou des accords directs). D'autres offres existent (par ex. **DeepMotion**, **Vicon** pour la capture de mouvement markerless haut de gamme), mais elles nécessitent du matériel ou des licences coûteuses, ce qui les rend peu accessibles.

En pratique, **MediaPipe** constitue la solution par défaut (open source, aucune restriction) pour un usage commercial agile ²⁴. **OpenPose** demeure l'option pour maximiser la précision (en traitant potentiellement image par image hors ligne du fait de sa lenteur), à condition d'obtenir une licence commerciale si le projet n'est pas open source ²⁰.

Analyse biomécanique : Une fois la pose obtenue, l'analyse des gestes (angles articulaires, amplitude, équilibre, etc.) se fait via un **code custom**. On calcule par exemple les angles de genou, hanche, etc., on détecte des déséquilibres, on établit un score technique. Il n'existe **pas de bibliothèque open source "clé en main"** pour évaluer la technique gestuelle en football – c'est généralement un ensemble de règles expertes et de calculs à implémenter. Le projet spécifie par exemple un *technique_scorer* évaluant timing, surface de pied, suivi du geste, etc. ²⁵ ²⁶. Ces règles sont à coder manuellement ou à entraîner via ML supervisé si un dataset de gestes notés est disponible.

Aucun produit propriétaire standard ne fait cela non plus, mis à part des solutions sur mesure utilisées par certains clubs (souvent basées sur la capture de mouvement Vicon ou sur des services d'analyse

vidéo). Dans notre cas, l'approche recommandée est d'utiliser les données de pose open source (MediaPipe/OpenPose) et de **développer un module interne** d'analyse biomécanique (calcul d'angles, détection d'anomalies, scoring). Cette partie étant spécifique, la précision dépendra de la finesse des règles ou du modèle ML entraîné (si l'on suit une approche data-driven).

Reconnaissance d'Actions (Classification d'Actions Football)

Ce module vise à classifier automatiquement les actions de jeu (passe, tir, dribble, tackle, etc.) à partir de séquences vidéo. Les méthodes actuelles les plus précises s'appuient sur des modèles de **vision vidéo à base de deep learning**, en particulier des réseaux de neurones 3D ou des Transformers spatio-temporels.

Alternatives open source :

- **TimeSformer** (Facebook, 2021) – Premier modèle purement Transformer appliqué à la vidéo. Il traite la séquence vidéo comme des patches spatiaux-temporels et utilise des mécanismes d'attention pour capturer la dynamique. **TimeSformer a atteint l'état de l'art** sur des benchmarks d'action recognition comme Kinetics-400/Kinetics-600 ²⁷, surpassant les anciens CNN 3D. Il offre en outre un entraînement plus rapide et peut traiter de longues vidéos (plus d'une minute) ce qui est idéal pour analyser des actions complexes ²⁷. Le code est open source (MIT/Apache) et disponible sur GitHub avec des modèles pré-entraînés ²⁸.
- **VideoMAE** (Masked Autoencoder for Video, 2022) – Modèle Transformers autoencodeur auto-supervisé. VideoMAE se distingue par son pré-entraînement auto-supervisé très efficace (masquage de ~90% des images) qui lui permet d'apprendre des représentations vidéo robustes avec moins de données ²⁹ ³⁰. Fine-tuné sur des tâches de classification, il a atteint des performances *state-of-the-art* sur plusieurs benchmarks (par ex. ~84% sur Kinetics-400 sans data externe) ³¹. Le code est open source (MIT) et intégré dans HuggingFace/Transformers ³². VideoMAE est particulièrement pertinent si l'on dispose de peu de données annotées spécifiques (matchs de football), car son pré-entraînement comble ce manque.
- **Autres** : *SlowFast* (Facebook, 2019) est un réseau bi-flux (un branch rapide, un lent) qui est encore un solide baseline pour la vidéo. *I3D* (Inflated 3D ConvNet, Google, 2017) a été beaucoup utilisé initialement pour la reconnaissance d'actions sportives. Plus récemment, *VideoSwin Transformer* (Microsoft) ou *MViT* (Multiscale ViT) ont aussi excellé. Cependant, TimeSformer et VideoMAE figurent parmi les meilleures **architectures disponibles en open source en 2025**, spécialement pour de la **précision maximale**.

Ces modèles nécessitent un fine-tuning sur le dataset spécifique (ici des clips d'actions football annotés). Cela demande des ressources GPU importantes, mais la **précision** obtenue peut dépasser 90% par classe comme visé ³³. Notons que des frameworks comme **MMAction2** (OpenMMLab) offrent des implémentations prêtes de ces modèles, ce qui facilite leur entraînement.

Alternatives propriétaires : Il n'y a pas de solution commerciale universelle pour la reconnaissance d'actions sportives – généralement, chaque acteur (ex: IBM, Microsoft) propose des services de vision génériques qui ne couvrent pas finement les actions football. Des entreprises spécialisées sport (StatsPerform, Second Spectrum) ont développé en interne des modèles pour reconnaître certaines actions (tirs, passes clés, etc.), mais ce n'est pas accessible librement. Ainsi, la voie pragmatique est de s'appuyer sur les modèles open source SOTA.

Si toutefois une solution non-open devait être envisagée, on pourrait citer l'API **AWS Rekognition Custom Labels** ou **Google Video AI** où l'on entraîne un classifieur vidéo custom sur le cloud. Mais cela coûte cher, et les performances sont tributaires du modèle sous-jacent qui est souvent moins optimisé pour ce cas précis qu'un TimeSformer fine-tuné.

En conclusion, on privilégiera une **architecture open source SOTA** (TimeSformer, VideoMAE, ou équivalent) fine-tunée sur les actions de football. Ces modèles étant publiés sous licence permissive, il n'y a pas de contraintes d'utilisation commerciale. L'effort réside surtout dans la constitution du dataset d'entraînement et la puissance de calcul nécessaire à l'entraînement puis à l'inférence (qu'on pourra optimiser via export ONNX/TensorRT, quantification INT8, etc. ³⁴).

Évaluation Technique des Gestes (Scoring Technique)

Ce module doit attribuer un **score technique** à une action individuelle d'un joueur (qualité d'une passe, d'un tir, d'un dribble) et fournir des feedbacks. C'est une tâche complexe qui peut être abordée par des règles expertes, de l'apprentissage supervisé, ou une combinaison des deux.

Approche manuelle (open source) : Comme esquissé dans la configuration, on peut implémenter un évaluateur basé sur des **critères experts pondérés** ²⁶ . Par exemple, pour une passe on évalue timing du contact, surface du pied, regard, suivi du geste, précision... chaque aspect recevant une note pondérée ²⁵ . Le score final est une agrégation. Cette approche nécessite de définir *à la main* les règles et seuils. C'est open source par nature (puisque c'est notre propre code) et transparent. L'inconvénient est que c'est potentiellement biaisé ou incomplet, et nécessite un expert humain pour calibrer les pondérations. Néanmoins, pour avoir des feedbacks explicables (ex: "améliorer ton équilibre après la frappe"), ce système **expert** est utile. On peut s'appuyer sur la littérature sportive pour définir les bonnes pratiques (par ex. angle d'approche idéal ~20°, etc.). Aucun paquet logiciel ne le fournit tel quel, c'est vraiment du développement spécifique.

Approche apprentissage supervisé : Alternativement, si l'on dispose de données annotées (par ex. 1000 vidéos de gestes notés par des coaches), on peut entraîner un modèle ML qui prédira un score. Cela pourrait être un modèle de **classification/régression** prenant en entrée les caractéristiques du geste (extraits de la pose : angles, vitesse segmentaire, etc., ou même directement la séquence vidéo) et fournissant un score 0-100. Par exemple, un réseau de neurones prenant les coordonnées du squelette sur chaque frame (peut-être modélisé via un **GCN spatio-temporel** ou un Transformer) pour estimer la "qualité" du geste. Des recherches récentes en *skill assessment* vont dans ce sens, avec des architectures de graphes ou de LSTM qui évaluent la performance à partir du mouvement ³⁵ ³⁶ . Cette approche data-driven peut potentiellement repérer des subtilités que les règles figées manquent, mais elle requiert un dataset conséquent et bien étiqueté, et le modèle résultant est une "boîte noire" moins explicable.

Solutions disponibles : Il n'y a pas sur étagère de solution prête pour scorer un geste sportif précis comme un tir au foot. Toutefois, quelques startups ont des produits approchants dans d'autres sports (par ex. **HomeCourt** pour le basketball évalue le tir au panier via smartphone). Pour le football, des entreprises comme **ShotTracker** ou **Coach AI** ont exploré des notations automatiques, sans offre grand public. Donc concrètement, ce module sera du développement sur mesure. On utilisera des outils open source (les données de pose, bibliothèques ML comme PyTorch/TensorFlow) pour implémenter soit les règles, soit entraîner un modèle.

En conclusion, **la meilleure "technologie" ici est l'expertise métier combinée à la vision par ordi** : on recommande d'exploiter les données de pose (obtenues via MediaPipe/OpenPose) et d'appliquer soit

un **scoring analytique** (règles pondérées validées par des coaches), soit de l'**apprentissage** (réseau supervisé) si les données le permettent. Dans les deux cas, aucune restriction de licence ne s'applique puisque c'est notre propre logique. Si l'on utilisait un modèle pré-entraîné propriétaire pour cela, il faudrait par ex. faire appel à un service en ligne, ce qui n'existe pas vraiment pour ce besoin précis. Donc le module sera nécessairement **open source ou interne**.

Analyse Tactique – Formation et Positionnement

Ce module doit détecter la **formation tactique** d'une équipe sur le terrain (p.ex 4-3-3, 4-4-2 losange, etc.), possiblement suivre son évolution en phase offensive/défensive, et fournir des métriques (distances entre lignes, compacité, etc.) ³⁷ ³⁸ .

Alternatives open source : Le défi ici est avant tout algorithmique, sur des données de position des joueurs (souvent issues de tracking). Deux grandes approches existent :

- **Clustering non supervisé + classification** : On peut utiliser des méthodes de **clustering** (K-means, DBSCAN, etc.) pour regrouper les joueurs par lignes sur le terrain en se basant sur leurs positions moyennes ³⁹ . Par exemple, un K-means avec $k=4$ clusters pourrait correspondre à défense, milieu, attaque, etc., ou des algos sans nombre fixé comme **FOREL** peuvent trouver un nombre de clusters optimal automatiquement ⁴⁰ . Ensuite, on peut **classifier** la configuration résultante en comparant aux schémas connus : par ex. calculer la distance entre les positions moyennes obtenues et des "templates" de formations (4-4-2, 4-3-3, 3-5-2...). Cette comparaison peut se faire via un algorithme de plus proches voisins ou un petit réseau de classification entraîné sur des données simulées. Tout ceci peut être implémenté avec des bibliothèques open source standards (scikit-learn pour le clustering, par exemple). Il n'y a pas de *package* tout prêt qui, en input les coordonnées x,y des 11 joueurs, donne "formation 4-3-3" : cela nécessite de coder la logique, mais les briques ML nécessaires sont disponibles librement.
- **Méthodes à base de règles et heuristiques** : Certaines recherches proposent de segmenter le terrain en zones et de voir le nombre de joueurs par zone pour inférer le schéma. Par exemple, identifier la **ligne de défense** via les 4 joueurs les plus proches de la surface, etc. Ce genre de règle peut compléter un clustering pour fiabiliser le résultat (ex: assurer qu'il y a bien 4 défenseurs alignés pour dire "4-3-3"). Cela reste du développement custom, mais relativement faisable.

En termes de **précision**, le défi est de gérer les variations en cours de jeu (les formations sont dynamiques). Une approche open source innovante est présentée par Decroos et al. 2020 sur données simulées RoboCup, où un algorithme reconnaît la formation en temps réel en adaptant les positions d'attentes des agents ⁴¹ . Mais ce n'est pas un code réutilisable directement.

Alternatives propriétaires : La société **Stats Perform (Opta)** a lancé en 2022 un produit nommé *Shape Analysis* qui automatiquement **identifie la formation d'une équipe sur chaque phase de jeu** ⁴² . Leur approche, décrite publiquement, consiste à : 1) segmenter le match en phases d'offensive et défensive, 2) appliquer un clustering non supervisé sur les positions des joueurs durant ces phases pour déterminer la "shape" (disposition) la plus stable, 3) attribuer un label de formation en comparant avec un set de templates pré-défini (17 types en possession, 13 types hors possession) ⁴² ⁴³ . C'est très proche de ce qu'on décrit en open source, sauf qu'eux disposent de **données de tracking précises** et ont calibré leur algorithme sur un grand nombre de matchs. Shape Analysis est une solution **fermée** intégrée à la plateforme Opta Vision, destinée aux clubs pro. Pour l'utiliser, il faut être client de Stats Perform – ce n'est pas un outil qu'on peut intégrer à notre logiciel sans accord commercial.

En résumé, pour l'analyse de formation, la **meilleure technologie open source** est d'utiliser les données de tracking disponibles et d'implémenter un algorithme combinant *clustering* des positions et *classification* par comparaison à des formations de référence. Les algorithmes de clustering (k-means, etc.) et de classification (distance de Hausdorff aux positions types, SVM, etc.) sont tous libres d'utilisation (scikit-learn, numpy). Cela permettra de détecter par exemple "l'équipe est en 4-2-3-1 en phase défensive et bascule en 3-4-3 en phase offensive". La précision dépendra de la qualité des données et du raffinement de l'algo, mais s'il est bien conçu il approchera l'efficacité des solutions propriétaires. **StatsPerform Shape Analysis** montre que cette approche est viable à l'échelle industrielle ⁴⁴ ; cependant leur solution étant fermée, on ne pourrait l'utiliser qu'indirectement (via des API ou rapports Opta si on était client). Donc, on s'orientera vers du développement maison open source.

Analyse Décisionnelle Tactique

Ce module vise à évaluer la **qualité des décisions tactiques** prises par un joueur ayant la balle : passer, dribbler, tirer, etc., compte tenu du contexte (position des joueurs, espaces libres, risque/récompense) ⁴⁵ ⁴⁶ . En sortie, on veut un **score de décision** et possiblement quelles alternatives auraient été meilleures ⁴⁷ .

Alternatives open source : C'est un domaine très pointu de l'analyse foot qui recoupe les métriques de *valeur d'action*. Une référence académique est le concept de **VAEP (Valuing Actions by Estimating Probabilities)** développé par l'université KU Leuven et implémenté dans la librairie open source `socceraction` ⁴⁸ . VAEP entraîne un modèle (par ex. XGBoost ou Réseau) sur un large historique de matchs afin de donner à chaque action un score correspondant à l'augmentation de probabilité de marquer ou d'encaisser un but suite à cette action ⁴⁹ . En gros, VAEP estime la "valeur" d'une passe, d'un dribble, etc., en termes d'impact sur le score futur. Ce genre de modèle **data-driven** est très pertinent pour évaluer objectivement une décision : par exemple, une passe latérale dans ses 30 mètres aura une valeur faible (ne crée ni danger ni risque), tandis qu'une passe en profondeur casse-ligne aura une valeur plus élevée si elle augmente la chance de but, mais aussi un risque si elle est manquée.

Avec `socceraction` (Python, open source MIT), on peut calculer de telles métriques à partir de données d'événements ou de tracking. Il intègre les frameworks VAEP et aussi l'approche **Expected Threat (xT)** de Karun Singh, qui attribue à chaque zone du terrain une valeur de "danger" et calcule comment une action déplace la balle vers des zones plus dangereuses ⁵⁰ ⁵¹ . **xT** est plus simple (tableau de valeurs du terrain) et peut être implémenté open source également. Ces métriques fournissent un **score quantitatif** à chaque action. Par exemple, un dribble réussi peut avoir +0.05 xT (a créé un léger danger en plus), un tir de loin non cadré -0.02 (a mis fin à une possession prometteuse inutilement).

Pour notre besoin de scoring décisionnel, on peut tout à fait utiliser ces approches : entraîner un modèle sur des données pros pour estimer la probabilité de marquer après chaque type d'action dans le contexte, et s'en servir pour **noter la décision choisie vs les options**. Par exemple, dans un contexte donné, si le joueur a tiré alors qu'une passe aurait eu une valeur attendue plus élevée, le module pourrait signaler que le tir n'était pas optimal.

L'avantage, c'est que VAEP et xT sont documentés et open source (articles IJCAI 2019 ⁵² , code disponible). On pourrait s'appuyer sur `socceraction` pour ne pas repartir de zéro. La précision sera fonction de la qualité du modèle et des données d'entraînement, mais ces méthodes ont montré de bons résultats pour *évaluer les décisions a posteriori* dans la littérature.

Alternatives propriétaires : Des entreprises comme **SciSports** (NL) ont commercialisé des analyses de valeur d'action (leur produit **Insight** intégrait VAEP). **StatsPerform** a son modèle propriétaire nommé **Expected Possession Value (EPV)** qui évalue le rendement des décisions en temps réel. Ces solutions ne sont pas accessibles librement. Néanmoins, SciSports avait publié son travail, et StatsPerform communique sur EPV dans des conférences. Si on voulait utiliser directement ces outils, il faudrait passer par leurs plateformes (contrats B2B). Autant dire que ce n'est pas envisageable dans une solution indépendante.

Une autre approche propriétaire pourrait être d'utiliser un système de **renforcement learning** entraîné sur des données de matchs (DeepMind a par ex. travaillé sur des agents simulant le foot). Mais c'est de la R&D pure et pas offert en tant que service.

Recommandation : Pour doter notre module décisionnel de la **meilleure technologie**, nous pouvons exploiter les travaux open source existants : par exemple intégrer la métrique VAEP via `socceraction` pour obtenir un score quantitatif de chaque action ⁴⁹. Cela donnerait une base objective. Ensuite, on peut combiner avec des règles plus simples pour l'aspect *compréhensible* : ex. "il y avait 2 coéquipiers démarqués mieux placés pour une passe (option plus sûre)". Le prompt initial mentionne un "modèle ML entraîné sur pros" pour le scoring ⁴⁷ – c'est exactement l'idée de VAEP/xT. Ce modèle étant le résultat de notre entraînement, il est **sous notre contrôle** et donc utilisable sans restriction (les données de match pourraient être soumises à licence, mais en open source il y a des datasets publics comme Wyscout, StatsBomb que l'on peut utiliser légalement).

En somme, on dispose de méthodes **open source de pointe** pour quantifier la qualité des décisions (valeur attendue). Il faudra les implémenter/adapter, mais on n'est pas bloqué par une technologie fermée. Les meilleures solutions propriétaires font certes la même chose, mais comme elles ne sont pas accessibles, l'open source est la voie à suivre.

Enfin, voici un **tableau comparatif** synthétisant pour chaque module les principales solutions techniques envisagées, avec leur statut open source ou non, et leurs points saillants :

Module	Solutions (open source)	Solutions (propriétaires)	Remarques Clés
Détection d'objets	YOLOv10, RT-DETR, DINO-DETR (modèles SOTA, code ouvert)	APIs vision génériques (Google, AWS) – pas spécialisées foot	DINO-DETR = précision max (~97% mAP) ⁴ ; YOLOv10 très rapide (NMS-free). Open source privilégié.
Tracking multi-objets	ByteTrack (SOTA MOT, open source), DeepSORT/OC-SORT (open)	NvDCF Tracker (NVIDIA DeepStream SDK, binaire propriétaire)	ByteTrack SOTA (80+ MOTA sur MOT17) ¹⁰ , performant en temps réel. Solutions NVIDIA utilisables si écosystème GPU, sinon intérêt limité.

Module	Solutions (open source)	Solutions (propriétaires)	Remarques Clés
Pose 2D/3D & Skeleton	MediaPipe Pose (Apache 2.0), OpenPose (code libre non-com *), AlphaPose	Wrnch AI (SDK Nvidia Omniverse), systèmes mocap (Vicon, etc.)	OpenPose plus précis mais lourd ¹⁶ et licence commerciale requise en prod ²⁰ . MediaPipe ultra-rapide, multiplateforme. Solutions propriétaires coûteuses si utilisées (Wrnch, etc.).
Analyse biomécanique	Calculs custom à partir de la pose (bibliothèques maths open, ex. NumPy)	Aucune solution packagée (systèmes sur-mesure en clubs pro)	Doit être développé sur mesure (règles expertes sur angles, etc.). Briques logicielles open (calcul d'angles) disponibles, mais pas de "moteur" prêt à l'emploi.
Reconnaissance d'actions	Transformers vidéo SOTA (TimeSformer ²⁷ , VideoMAE ³²), CNN3D (SlowFast) – tous open source	Services ML cloud (AWS/GCP Custom Video ML) – génériques	TimeSformer/VideoMAE offrent haute précision sur actions (~90%+) avec fine-tuning. Pas de restriction licence. Solutions cloud payantes et moins optimisées.
Scoring technique gestuel	Règles expertes codées (open), ou modèle ML entraîné (TensorFlow/PyTorch)	- (pas de produit dispo, sauf consultation spécialisée)	Approche open nécessaire. Expert system pour explication + ML pour objectivité. Aucune licence tierce requise.
Analyse formation tactique	Clustering + classification via libs Python (scikit-learn, etc.)	Opta Vision Shape Analysis (StatsPerform) – fermé	Algorithme open possible reproduisant approche StatsPerform ⁴² . Pas de solution libre clef en main, mais faisable avec outils standard.
Analyse décisionnelle	Modèle valeur d'action type VAEP/xT (lib socceraction, open source)	Metrics propriétaires (EPV de StatsPerform, SciSports) – fermés	VAEP disponible open source pour évaluer actions ⁴⁹ . Permet scoring quantitatif des choix. Solutions fermées non utilisables directement, d'où choix open.

(* OpenPose est open source pour R&D, mais nécessite une licence pour usage commercial ²⁰.)

En conclusion, l'ensemble des modules peut être couvert par des technologies majoritairement **open source** sans sacrifier la précision : les meilleurs algorithmes actuels de vision (détection, pose, reconnaissance d'actions) sont disponibles librement et souvent dominent les offres propriétaires. Dans

les rares cas où la solution optimale n'est pas open (par ex. l'analyse de formation Opta, ou un SDK de pose commercial), il existe des alternatives ouvertes crédibles (reproduire l'approche Opta avec du clustering, utiliser MediaPipe/OpenPose à la place de Wrnch...). Il faudra simplement investir du temps de développement pour intégrer ces briques et éventuellement entraîner les modèles sur nos données. Cette stratégie garantit une **précision maximale** tout en évitant les verrous propriétaires, ce qui correspond aux objectifs du projet.

1 8 25 26 33 34 37 38 39 45 46 47 PROMPTS.md

<file:///file-RDSrbQZMKVCJnSrpS1ENJ3>

2 4 6 CONFIGURATION_PRECISION_MAX.md

<file:///file-XJ88MfYFtzM8Yjio3toZv6>

3 Object Detection Models: Comparing YOLOv10, DETR, and Top Models of 2024 - DFRobot

<https://www.dfrobot.com/blog-13914.html?srltid=AfmBOoofvZXJfRY3XWjfdW3v9z8e0ps0shGHmCQNAjbPdJJ-P7Y2vbRF>

5 DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End ...

<https://medium.com/@gagatsis94/dino-detr-with-improved-denoising-anchor-boxes-for-end-to-end-object-detection-4f32b9389baa>

7 DINO: DETR with Improved DeNoising Anchor Boxes for End ... - ar5iv

<https://ar5iv.labs.arxiv.org/html/2203.03605>

9 10 11 12 13 14 Object Tracking State of the Art 2022 | by Pedro Azevedo | Medium

<https://medium.com/@pedroazevedo6/object-tracking-state-of-the-art-2022-fe9457b77382>

15 16 17 18 21 OpenPose Vs MediaPipe: | Sushant Sarswat

https://www.linkedin.com/posts/sushantsarswat_openpose-vs-mediapipe-openpose-and-mediapipe-activity-7044924397874278400-EyT6

19 OpenPose vs. MediaPipe | In-Depth Comparison for Human Pose Estimation | by Saiwa | Medium

<https://medium.com/@saiwadotai/openpose-vs-mediapipe-in-depth-comparison-for-human-pose-estimation-402c5a07b022>

20 OpenPose License - Adaptive Support - AMD

https://adaptivesupport.amd.com/s/question/0D52E00006hpWXeSAM/openpose-license?language=en_US

22 What You Need to Know About Pose Estimation and OpenPose?

<https://medium.com/@augmentedstartups/what-you-need-to-know-about-pose-estimation-4ac189a5e07e>

23 [PDF] wrnch AI Pose Estimation extension for NVIDIA Omniverse

<https://s3.amazonaws.com/cdn.wrnch.ai/wp-content/uploads/2021/04/11213550/partners-nvidia-documentation.pdf>

24 Can MediaPipe be used commercially? - QuickPose.ai

<https://quickpose.ai/faqs/can-mediapipe-be-used-commercially/>

27 28 TimeSformer

https://huggingface.co/docs/transformers/en/model_doc/timesformer

29 30 31 32 VideoMAE

https://huggingface.co/docs/transformers/en/model_doc/videomae

35 36 Pose Estimation in Sports | Enhancing Performance

<https://saiwa.ai/blog/pose-estimation-in-sports/>

40 41 Formation Recognition by Clustering-Based Method in Virtual Soccer

https://ceur-ws.org/Vol-2893/paper_19.pdf

42 43 44 Shape Analysis: Automatically Detecting Formations - Stats Perform

<https://www.statsperform.com/resource/shape-analysis-automatically-detecting-formations/>

48 ML-KULeuven/socceraction: Convert soccer event stream data to ...

<https://github.com/ML-KULeuven/socceraction>

49 VAEP: combining 36 actions into one performance indicator | Footovision

<https://www.footovision.com/VAEP-combining-36-actions-into-one-performance-indicator>

50 [PDF] Valuing On-the-Ball Actions in Soccer: A Critical Comparison of xT ...

https://tomdecroos.github.io/reports/xt_vs_vaep.pdf

51 Valuing On-the-Ball Actions in Soccer: A Critical Comparison of xT ...

<https://dtai.cs.kuleuven.be/sports/blog/valuing-on-the-ball-actions-in-soccer-a-critical-comparison-of-xt-and-vaep/>

52 [PDF] VAEP: An Objective Approach to Valuing On-the-Ball Actions ... - IJCAI

<https://www.ijcai.org/proceedings/2020/0648.pdf>