Génération de commentaires sportifs audio par IA

CentraleSupélec

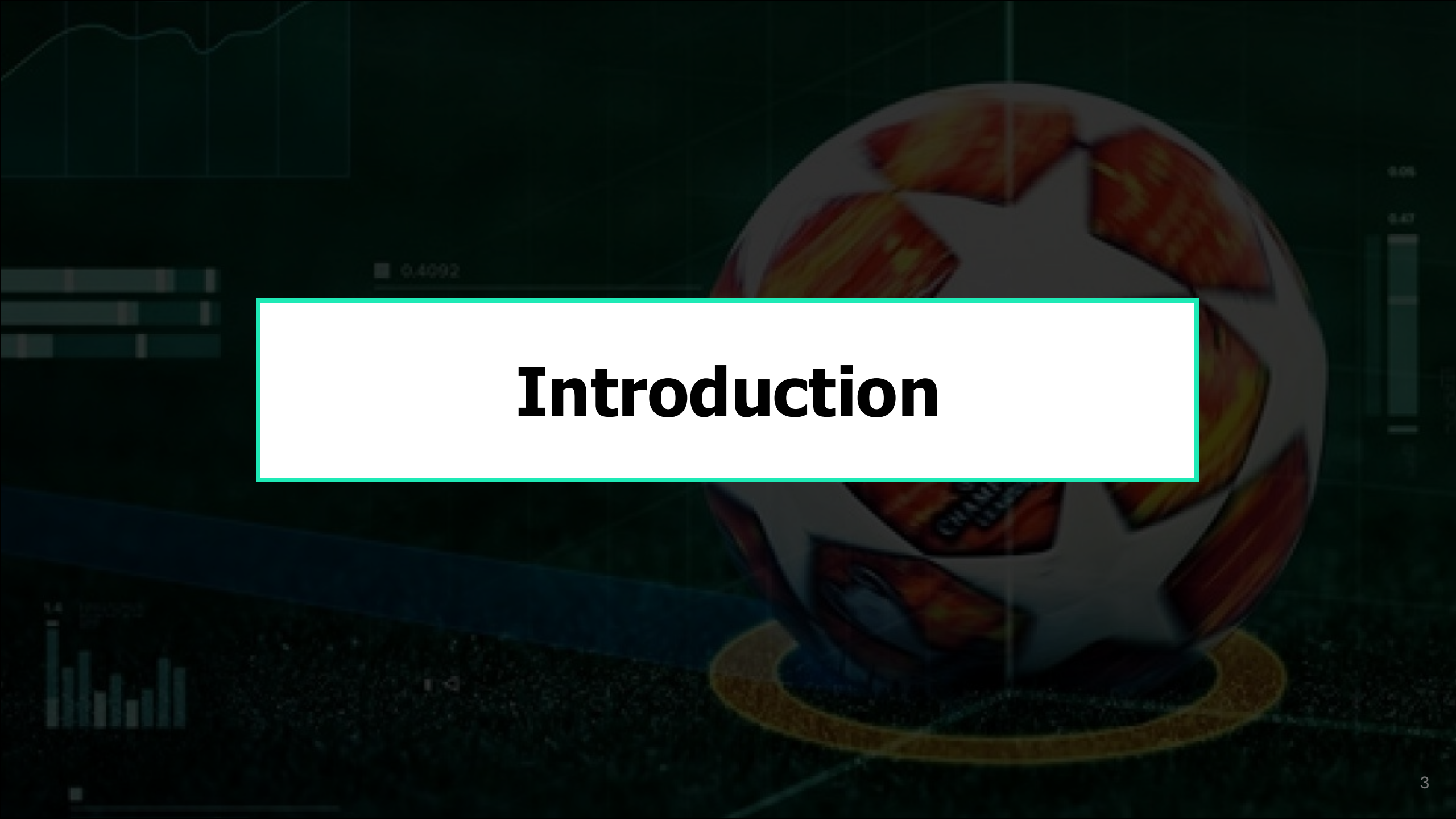
SIAPARTNERS

Soutenance Finale

10 Avril 2025

Antoine Bohin, Mathieu Dujardin,
Ilyess Doragh, Logan Renaud

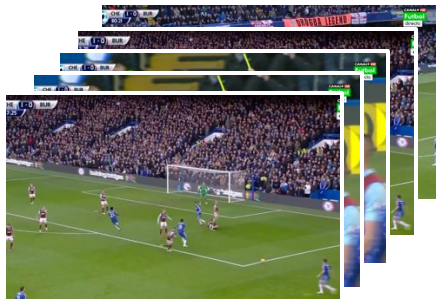
Sommaire



Introduction

Constat: Aucun modèle « tout-en-un » prometteur

Objectifs



Extrait VIDÉO d'un match de football (10 à 40 minutes)

Modèle(s) d'IA

Exploration des modèles d'IA à l'état de l'art, avec une attention particulière pour les « petits » modèles (<10B param.)

Highlights Sportifs commentés du match

Interface

Limites des modèles multimodaux actuels

- **Pas de modèle « video-to-speech » efficace**
- **Traitement vidéo limité** : seulement quelques frames analysables à cause des contraintes de tokens
- **Compréhension partielle du jeu** : intégration difficile des règles ou dynamiques de match
- **Commentaires peu naturels**: commentaires répétitifs, sans storytelling ni émotion
- **Absence de mémoire temporelle**
- **Pas d'enrichissement contextuel** : nom des joueurs, score, enjeux non pris en compte
- **Contraintes matérielles fortes (GPUs)** : recours à des modèles légers, moins performants

➢ **Quelles sont les limites des modèles d'IA générative actuels pour leur utilisation dans le monde du sport?**

Solution : Pipeline de modèles d'IA spécialisés

Extraction visuelle

Génération de commentaires

Extrait VIDÉO
(10 à 40 minutes)

1 CNN

Action Spotting

- Détection et classification des actions clés du match
- Extraction de clips de 10s autour de ces actions clés

2 Tracking

Enrichissement visuel

- Tracking joueurs/balles, détection de numéros pour identification des joueurs…

3 VLM

Description d'action

- Génération d'une description chronologique et précise de chaque clip

Ex: Le joueur blanc tacle sur le bleu. L'arbitre signale une faute. Il brandit un carton jaune.

4 LLM

Commentaire textuel

- Reformulation en commentaire fluide, vivant, avec émotions

Ex: Oh l'énorme faute de Dembélé ! Et c'est un carton jaune, logique il est en retard !

5 TTS

Audio Final

- Synthèse vocale naturelle façon commentateur sportif

Commentaire AUDIO

01:32 - Corner

03:47 - Faute

04:21 - But

Clips d'actions clés (10s)

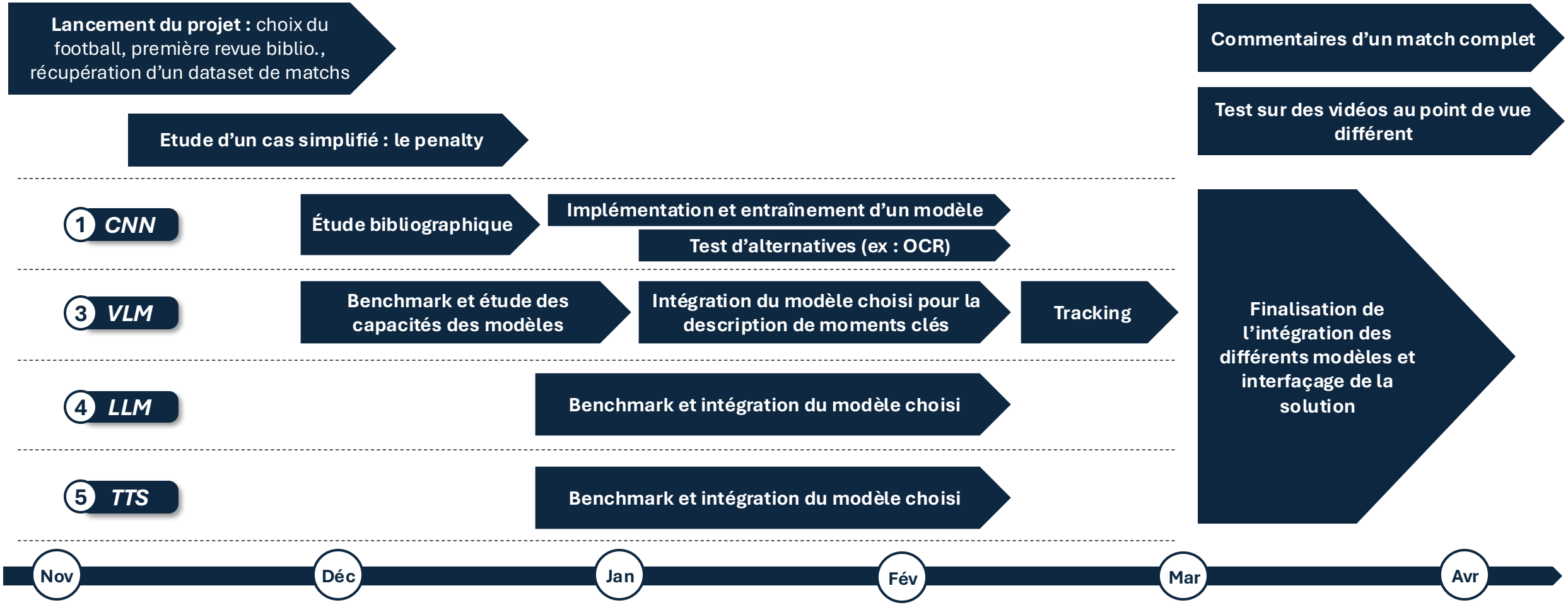
01:32 - Corner

03:47 - Faute

04:21 - But

Highlights commentés

Timeline du projet



1

Action Spotting :
Identification d’actions clés

Pourquoi l'*Action Spotting* ?

Motivations

Filtrer les séquences clés dans des vidéos longues
- Pallier la limite de tokens des modèles multimodaux

Concentrer les modèles sur les extraits informatifs
- Faciliter la compréhension du jeu

Pivot temporel pour la génération de commentaires
- Construire une narration synchronisée avec les moments clés

Défis

Fort déséquilibre des classes (actions rares vs fréquentes)

Variabilité visuelle et temporelle élevée (plans caméras, enchaînements rapides, frontières floues entre actions)

Entraînement de modèles supervisés gourmand en calcul et nécessitant des datasets massifs de vidéos annotées

Formalisation

Identifier précisément le moment (timestamp) et **la nature** (passe, tacle, but, carton, changement…) des actions clés

- Prédiction de couples (timestamp, label)

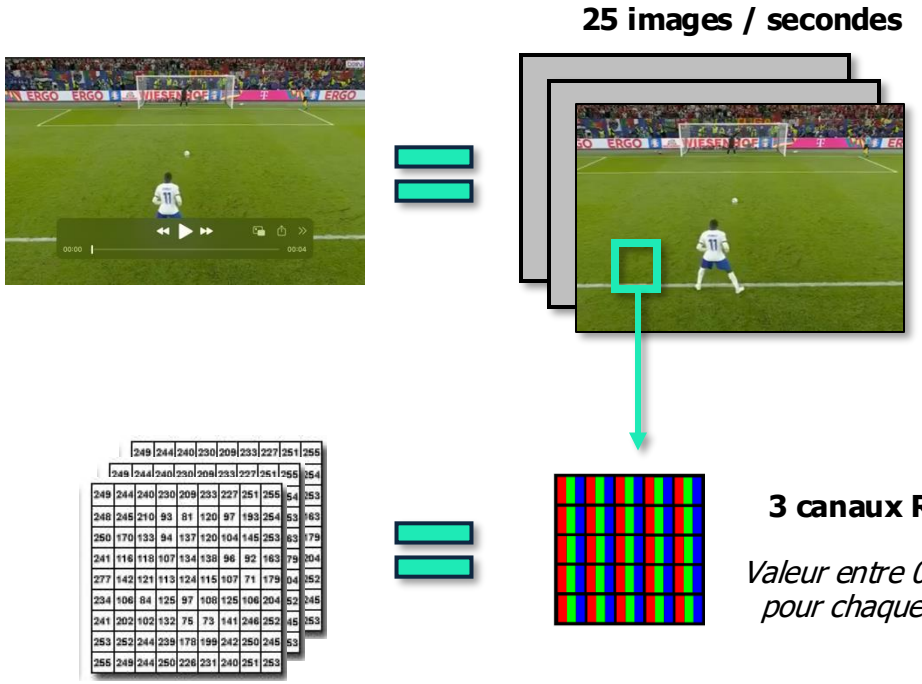
Environnement de recherche dynamique

Benchmarks sur des datasets publics standardisés (e.g. SoccerNet)

Perspectives d'applications concrètes (Highlights automatiques, VAR enrichie, analyse tactique…)

Comment analyser visuellement des vidéos?

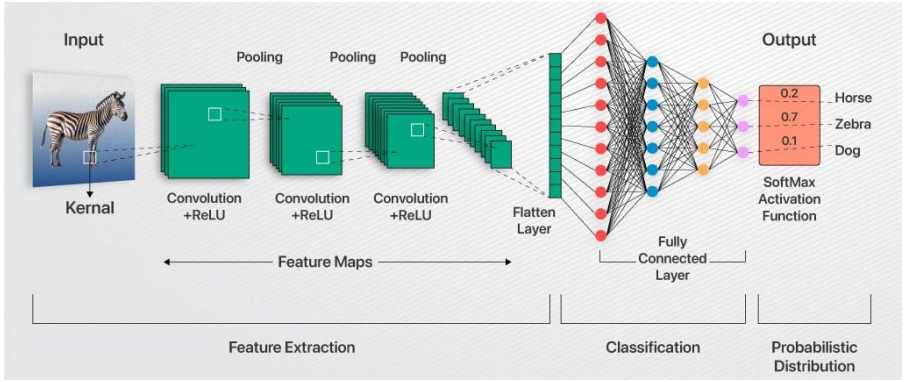
Une vidéo est un ensemble d'images, donc de matrices de pixels



- La Computer Vision permet d'analyser le contenu visuels, à des fins de classification d'image, détection d'objets, segmentation, reconnaissance d'action…
- Modèles supervisés = données annotées !

Comment les réseaux de neurones traitent les images?

Briques de base: les CNNs



Filtres glissés sur l'image pour extraire des features de plus en plus abstraites…

… utilisées ensuite pour la classification

Adapter la vision à la vidéo : traitement temporel

Nécessité de capturer la dynamique temporelle (mouvement, continuité des actions) :

- **Late fusion = concaténation des features extraites sur chaque frame**
- **CNNs 3D = convolution aussi dans le temps**

Autres architectures : RNN / LSTM, Transformers…

Dataset SoccerNet

SoccerNet: la référence de la Computer Vision dans le football

- Fournir un **dataset annoté standardisé pour l'entraînement de modèles** de Computer Vision spécialisés pour le football
- Créer un **écosystème de recherche open-source avec benchmarks et challenges annuels** (CVPR, ECCV…): segmentation, captioning, analyse tactique…

- 4 à 8 challenges avec cashprize organisés chaque année, grâce à 10 datasets annotés
- **+5 articles récompensés « Best Paper »** CVPR
- Déjà +**60 méthodes** publiées pour l'Action Spotting
- Applications concrètes : analyse de performance, diffusion TV, recrutement…

Le Dataset SoccerNet-v2: conçu pour l'Action Spotting

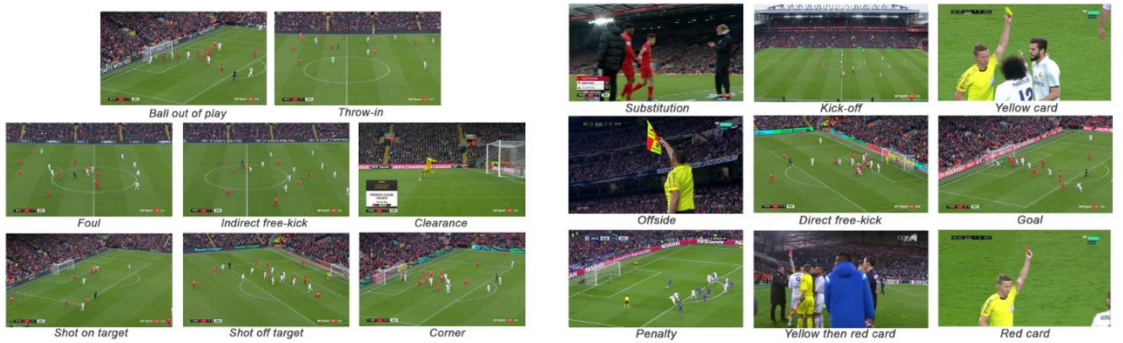
+550 Matchs (764h)

+200 évènements par match

Vidéos en 720p ou 224p

Accessible via package Python

Annotations (timestamp, label) en .json



17 Classes d'évènements (tir, but, corner, carton…)

Etat de l'art de l'Action Spotting

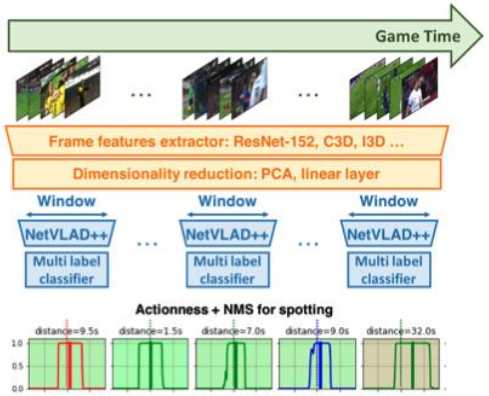
1 Modèles basés sur des features spatiales

NetVLAD, NetVLAD++

- **Features visuelles extraites frame par frame via ResNet / ImageNet (1-2 fps)**
- **Pooling temporel simple (local/statistique) pour prédire des scores sur des fenêtres**

+ Simples, rapides, peu coûteux

- Pas de modélisation temporelle profonde (pas séquentiel ni dynamique)



NetVLAD++

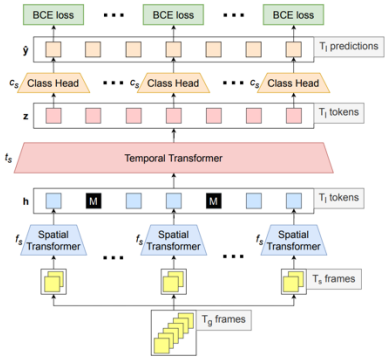
2 Modèles end-to-end contextuels

E2E-Spot, T-DEED, COMEDIAN…

- **Apprentissage conjoint des dynamiques visuelles et temporelles**
- **Intègrent GRU, LSTM, Transformers temporels, Conv3D…**

+ Modélisation précise du contexte, robuste aux ambiguïtés

- Coûts en calcul et mémoire élevés



COMEDIAN

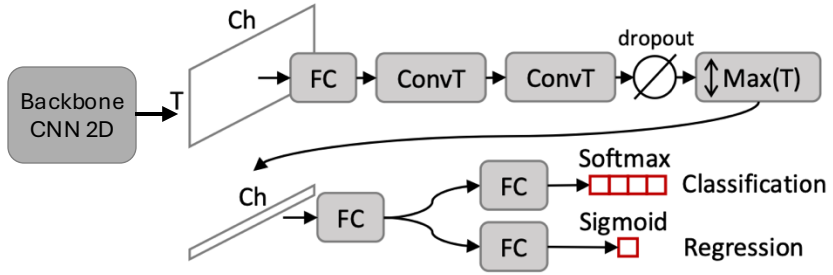
3 Modèles hybrides et légers pour l'inférence

RMS-Net, Ruslan Baikulov…

- **Backbone 2D pré-entraîné pour l'apprentissage de la dynamique locale**
- **Puis des couches convolutives temporelles pour l'apprentissage de motifs temporels**

+ Bon compromis entre performance, robustesse et efficacité

Adapté aux GPUs modestes, pour des solutions déployables



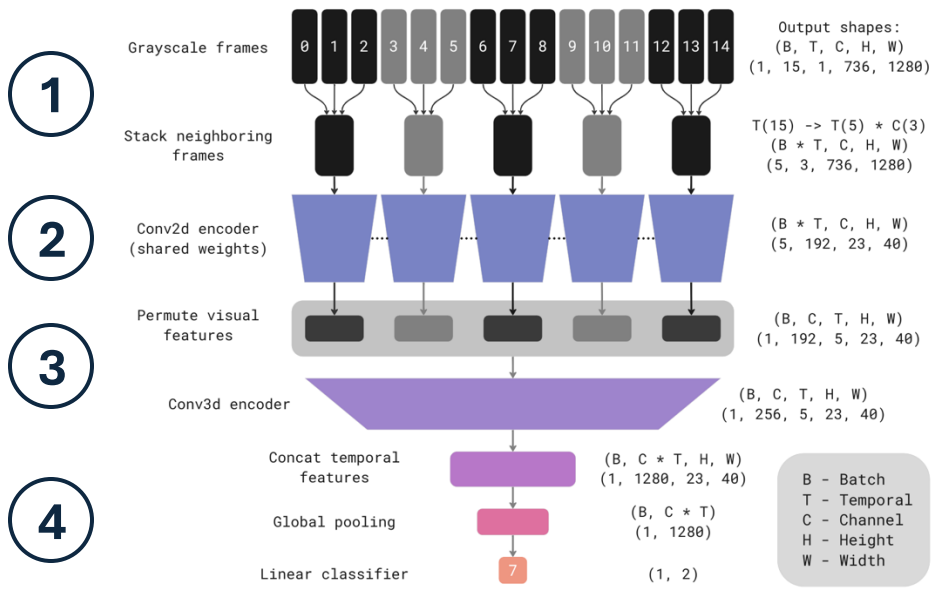
RMS-Net

Présentation du modèle supervisé sélectionné

Points clés

- **#1ère Place Challenge SoccerNet Ball Action Spotting 2023 (87.47% mAP)**
- **Robustesse : bonne généralisation sur des vidéos variées**
- **Chargement optimisé des vidéos pour maximiser l'utilisation des GPUs**
- **Code source publié comme baseline pour les challenges 2024/2025**
- **Modèles pré-entrainés disponibles pour de l'inférence directe**

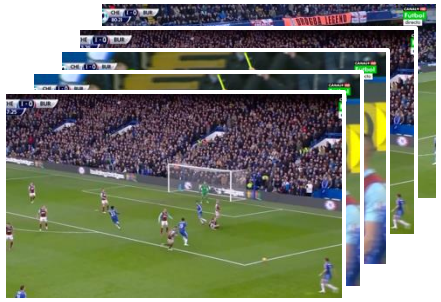
Architecture



1. **Empilement de 3 frames en niveaux de gris :** capture locale du mouvement, réduction de la dépendance aux couleurs
2. **Backbone 2D EfficientNetV2 :** extraction de features par block de frames, léger et efficace
3. **Neck 3D convolutionnel léger :** fusion temporelle pour capturer le mouvement
4. **Head Classifier Linéaire :** pour prédire type d'action par fenêtre

Test sur un extrait de match

Extrait initial



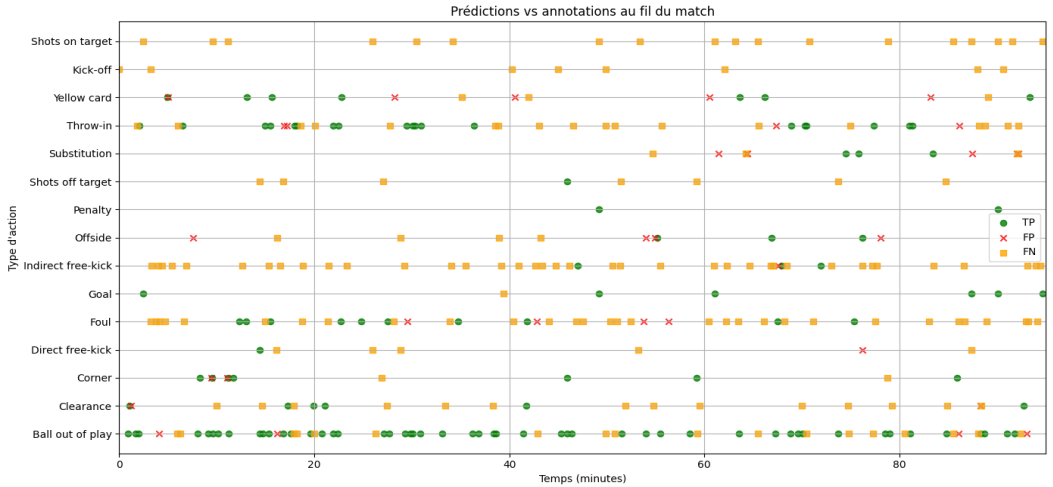
90 min de match Champions League
Barça 6-1 PSG

Prédiction

Confiance > 0.9	Confiance 0.8-0.9	Confiance 0.7-0.8
✓ 54 actions	✓ 96 actions	✓ 101 actions

2628 prédictions à différents niveaux de confiance

Confiance > 0.8

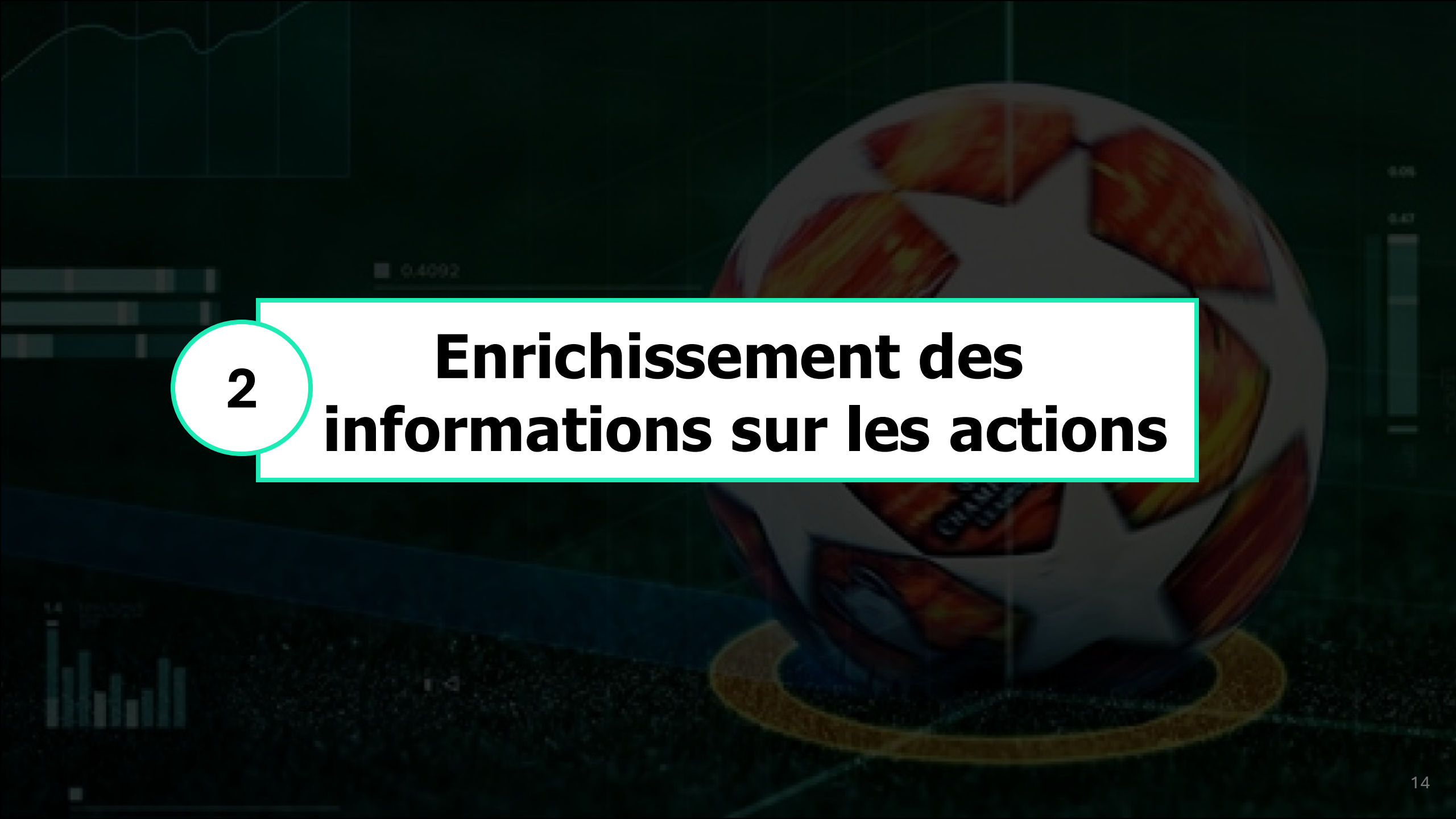


+

- Bonne généralisation sur des matchs récents
- Bonne reconnaissance de certaines actions principales (buts, corners, touches…)
- Précision de la localisation temporelle (timestamp) des actions
- Bonne densité d'actions reconnues
- Possibilité d'adapter les seuils de prédiction en fonction des classes, ou du niveau de précision souhaité

-

- Sensibilité aux changements de plan caméra (ex: gros plans sur les joueurs)
- Performances / niveaux de confiance variables en fonction de la classe (ex: frappes difficilement reconnues)
- Informations à enrichir (seulement timestamp et label)
- Temps d'inférence assez élevé (environ 2x la durée de la vidéo), qui peut être réduit avec de meilleurs GPUs



2 Enrichissement des informations sur les actions

Pourquoi enrichir les informations sur les actions?

Faciliter l'analyse du VLM

Aider à suivre la temporalité image par image

- Identifier les joueurs impliqués plusieurs fois dans une action
- Comprendre le mouvement de la balle

Réduire les confusions

- Exemple : Le joueur bleu tire sur le gardien (alors qu'il n'a pas tiré)
- Ajout d'information sur qui est le gardien, quel joueur est dans quel équipe

Enrichir les commentaires

Ajouter les noms des joueurs

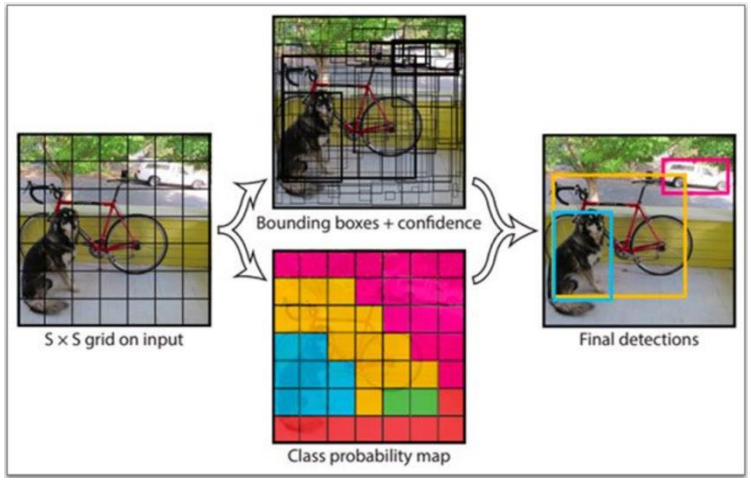
- Nommer les joueurs au cours des actions
- Savoir qui concernent les remplacements, coup francs…

Tracer le mouvement de la balle

- Mieux comprendre la trajectoire du ballon et sa vitesse (tir, passe…)
- Identifier le ballon comme élément clé pour le VLM

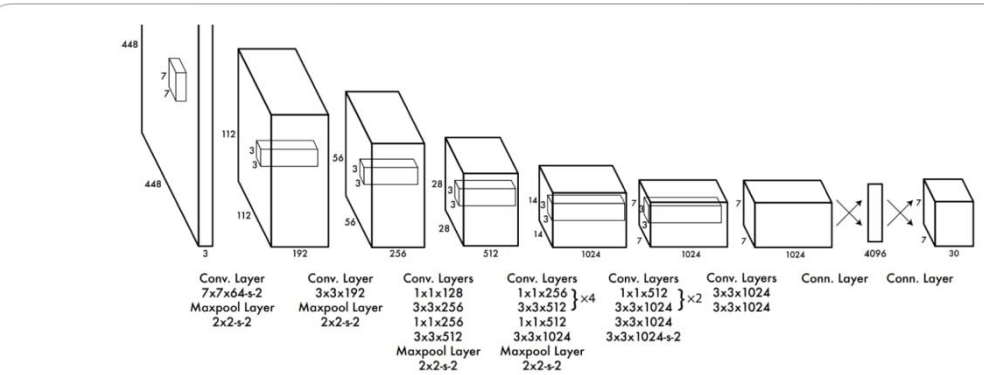
Detection des joueurs et du ballon

Principe : classification par région de l'image



- Objectif : détecter des éléments prédéfinis sur une image
- Ex: Détecter les élément parmis *(vélo, chien, voiture)*

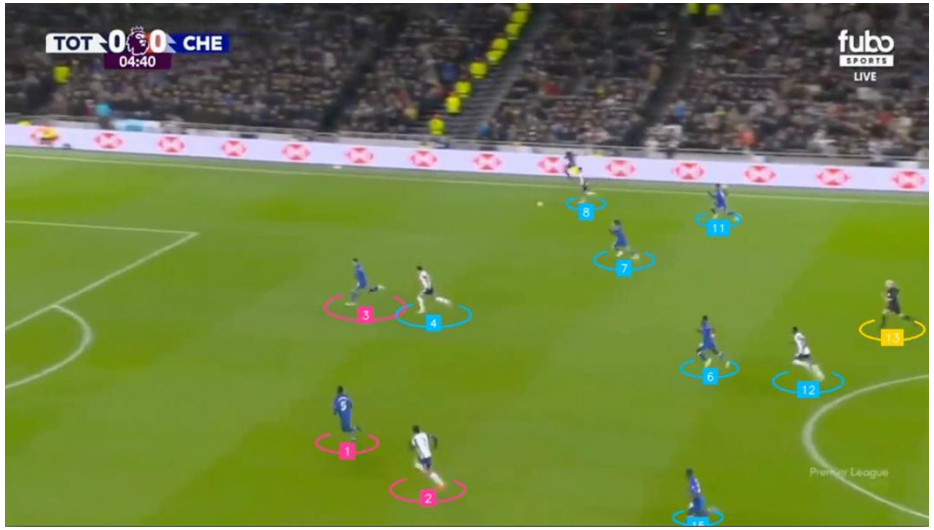
Architecture : Réseau Convolutif



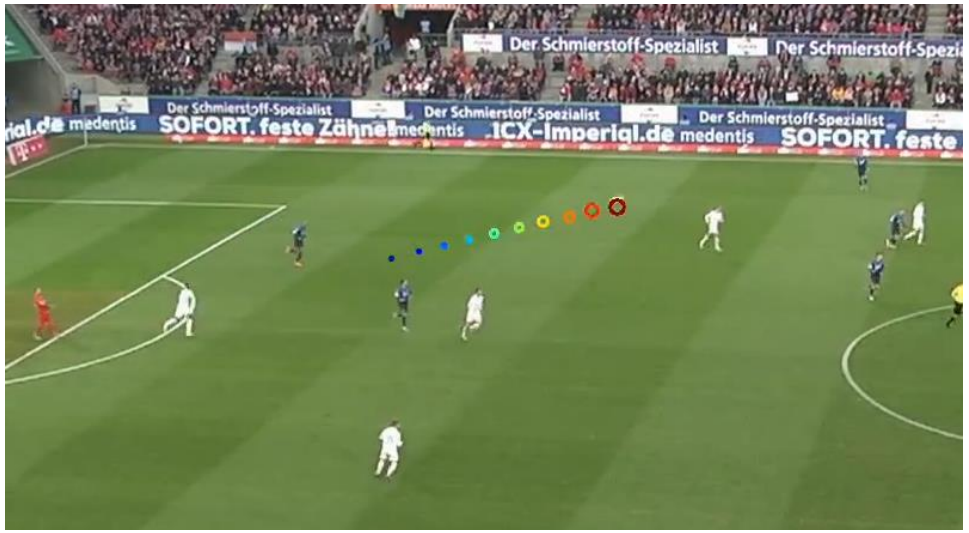
The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

- La **dernière couche linéraire** prédit la classe de chaque maille
- Entrainement par **fine-tuning** en remplacant la **dernière couche**

Tracking des joueurs et du ballon



Detection de joueur et arbitre
Modèle YOLO
+
ByteTracker



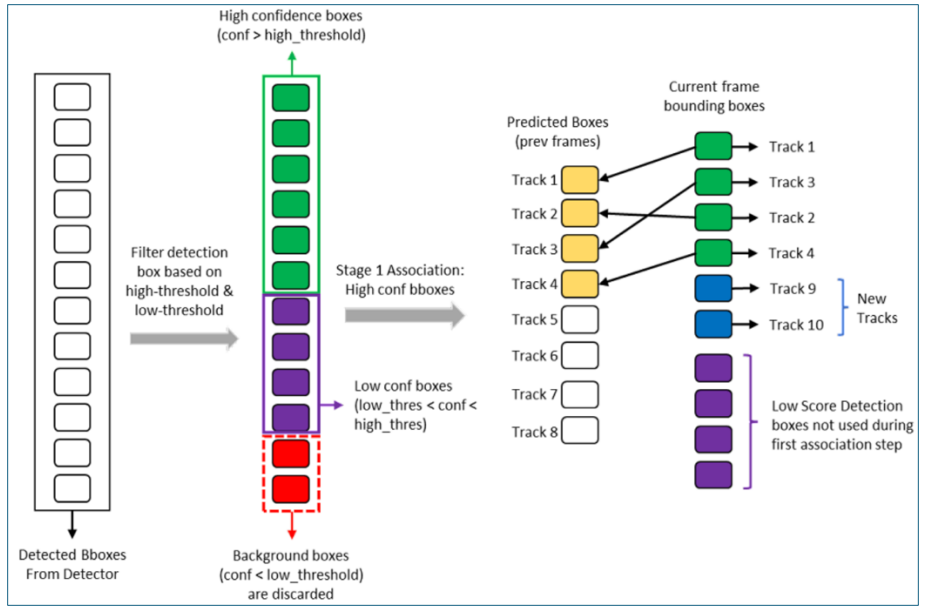
Tracking de ballon
Modèle YOLO
+
Tracking Average

- Modèles pré-entrainés sur des datasets publics
- Permet d'enrichir les images pour le VLM
- Reconnaissance des joueurs

Tracking des joueurs et du ballon

ByteTracker

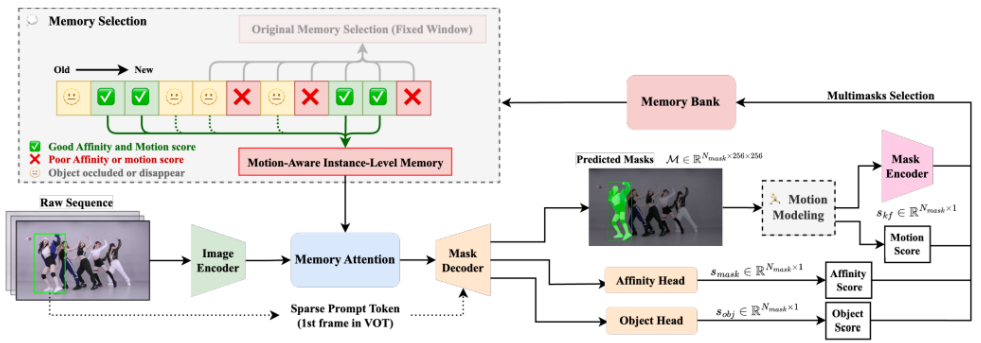
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



- IoU et Filtres de Kalman
- Très rapide (detection + association)

Samurai (Segmentation)

Tracking par segmentation (plutôt que bounding box)



Architecture du modèle **SAMURAI** : Segmentation sur vidéo

- Spécialisé dans le tracking dans des environnements dynamiques
- Sélection de mémoire sensible au mouvement
- Beaucoup + lent (segmentation + Association par DeepLearning)

Reconnaissance des joueurs

Enjeux

Identifier et nommer les joueurs par Vision

- Il faut pouvoir lire le numéro ou le nom du joueur
- Les joueurs peuvent être face à la caméra
- Les changements de plan (ralentis, changements) interrompent la séquence et donc le tracking des joueurs

Idée developpée

Détection des joueurs (YOLO)

↓

Extraction des bounding boxes contenant les joueurs

↓

Modèle **d'OCR** Florence-2 Large (VLM Microsoft)

Florence - 2

Méthodologie de **validation** (Lecture du même nombre plusieurs fois par seconde)

Améliore la robustesse

Résultats

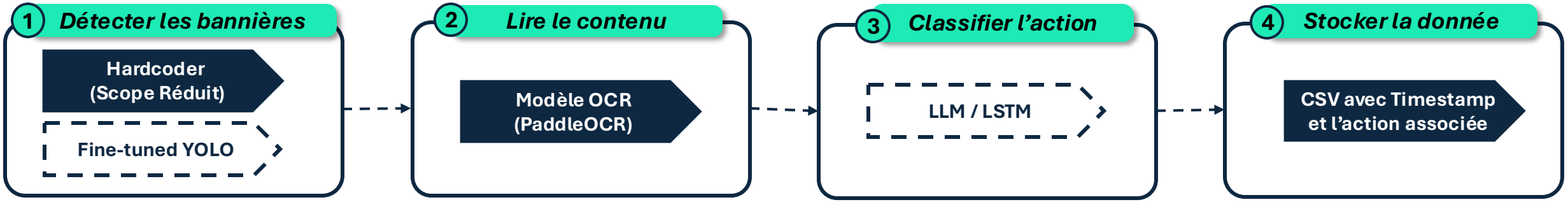
Conclusion

- Beaucoup de detection du meme numéro (#1), en particulier pour les joueurs de face ou profil
- **Complexité importante pour un résultat encore trop peu robuste**

Approche supplémentaire: OCR du tableau d'affichage

Idée

Utiliser les **bannières** de match pour détecter les moment importants



14:38
« Changement de joueur »

04:17
« Score 0-0 »

05:58
« Nom du Buteur »

3 Description des actions par Visual Language Model

Bases du NLP

Prompt

Texte d'entrée : « Quelle est la capitale de l'Italie"

Tokenisation

["Quelle", "est", "la", "capitale", "de", "l", " ' ", "Italie", "?"]

Embeddings

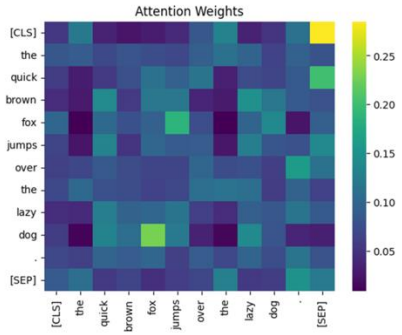
Quelle	[0.20, 0.60, 0.15, 0.10, 0.05]
est	[0.12, 0.88, 0.55, 0.11, 0.03]
la	[0.05, 0.02, 0.07, 0.04, 0.01]
capitale	[0.80, 0.09, 0.40, 0.90, 0.59]
de	[0.03, 0.01, 0.05, 0.02, 0.01]
l'Italie	[0.85, 0.15, 0.42, 0.92, 0.63]
?	[0.00, 0.00, 0.00, 0.00, 0.00]

Dépend du modèle utilisé:

- Word2Vec (Google, 2013) : embeddings statiques (pas de contexte)
- BERT (Google, 2018) : embeddings contextuels (générés en fonction des autres mots)

Mécanisme d'attention

- Chaque mot "regarde" les autres mots via des poids d'attention.
- Ces poids indiquent à quel point chaque mot est important pour comprendre le mot en cours.

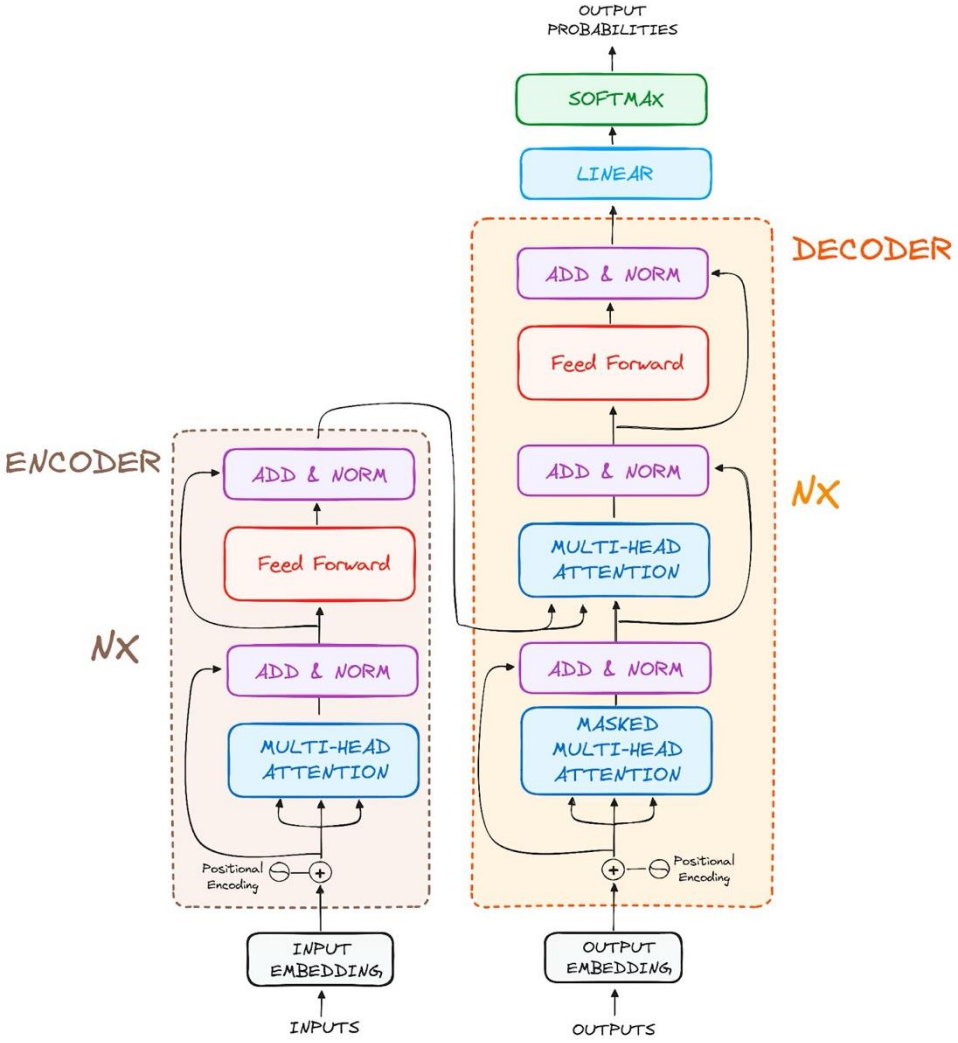


Sortie

Le modèle transforme les scores en **probabilités** sur tout le vocabulaire

- "Rome" : 0.85
- "Milan" : 0.10
- "Paris" : 0.03 …

Architecture d’un Large Language Model (LLM)



Encoder: Comprendre l’entrée

Attention Multi-Tête : Analyse les relations entre les mots de l'entrée pour comprendre le contexte.
Feed-Forward : Traite chaque position individuellement après l'attention.
Résultat : Crée une représentation riche de chaque mot, tenant compte des autres mots de la phrase d'entrée.

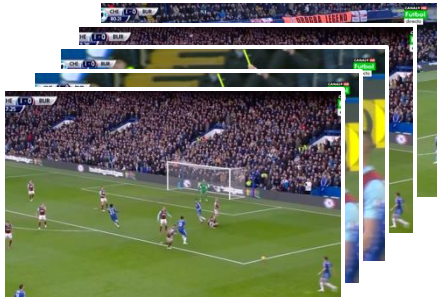
Decoder: générer la sortie

Attention Masquée: Lors de la génération, ne regarde que les mots déjà produits.
Attention à l'Encodeur : Se concentre sur les parties pertinentes de la représentation de l'entrée fournie par l'encodeur.
Processus Répété : Génère la séquence de sortie un mot à la fois, en s'appuyant sur les prédictions précédentes et l'entrée encodée.

Prédiction finale: choisir le mot suivant

Couche Linéaire : Transforme la représentation finale du décodeur en scores pour chaque mot possible dans le vocabulaire de sortie.
Softmax : Convertit ces scores en probabilités, indiquant la probabilité que chaque mot soit le mot suivant correct. Le mot avec la plus haute probabilité est sélectionnée comme sortie.

Architecture d’un Visual Language Model (VLM)



Encodeur visuel: extrait des représentations denses de l’image / des frames vidéo.

“Décris brièvement l’action.”

Encodeur textuel: extrait des représentations denses de l’invite en entrée.

Fusion multimodale: lie les representations textuelles et visuelles.

Décodeur: génère le texte final en sortie.

“Le joueur de Arsenal centre, puis le numéro 9 mets une tête et c’est un but.”

Choix du VLM

Contraintes

GPUs du cluster: Maximum 24GB de RAM GPU

→ **Utilisation de modèles petits en taille**

→ **Optimisation de la taille mémoire (Flash Attention, poids et inférence en Float16).**

→ **Utilisation de modèles open-source**

Modèles testés:

- Phi-3.5 Vision
- Gemma-3-4B
- IBM Granite
- Apollo-7B
- SmolVLM2-2.2B
- **QWEN2.5-VL-7B: Modèle retenu**

Comment a-t-on réalisé les benchmarks ?

→ **Tests qualitatifs** effectués en inférence locale

→ Utilisation de **benchmarks en ligne**

→ Benchmark plus poussé: **Classification d'action** de foot entre SmolVLM2.2B et QWEN2.5-VL-7B avec une vidéo annotée du dataset SoccerNet

Résultat sans appel: **51%** d'accuracy pour **QWEN2.5-VL-7B** qui surpasse largement les autres modèles.

Action	SmolVLM2-2.2B	Qwen2.5-7B	Gemma3-4B
Kick-off	0%	16.67%	33,33%
Foul	0%	78.57%	28,57%
Shots on target	0%	0%	0%
Goal	60%	100%	40%
Clearance	33.33%	33.33%	33,33%
Overall	**11.43%**	**51.43%**	**25,7%**

Génération des descriptions d'actions

Prétraitement video pour le VLM

Clip de 10s à 25 fps = trop lourd pour nos GPUs 24Go

Deux leviers d'optimisation testés :

- **Réduction de la résolution** des frames
- **Réduction du fps** traité par le VLM

➢ **Conserver un fps > 3 est plus crucial que la résolution**.

➢ **Tracking pour compenser la faible résolution**

Prompting stratégique pour guider le VLM

Prompt incluant : label de l'action, durée de clips, demande explicite de traitement chronologique et timestamps indicatifs dans l'output

Génération par « chunks » temporels pour une narration précise et chronologique

Permet au LLM de réécrire avec cohérence et de mieux se synchroniser

Le modèle fonctionne mieux avec une **variété de plans, plus instructifs pour comprendre l'action**

Le tracking aide le modèle à faible résolution

Même à basse résolution, lecture possible des dossards en gros plan ou des changements de tableau d'affichage

Parfois, bonne description malgré un mauvais label

Eviter des prompts trop complexes (ex : reformuler directement comme un commentateur)

4 Ecriture des commentaires sportifs par LLM

Choix du LLM

Pourquoi ?

→**Utiliser un modèle plus gros (**puisqu'il n'a pas à traiter des vidéos en inferences**)** et **plus performant pour la generation de texte**.

→Séparer la partie **extraction d'information** de la video de la partie **generation de commentaire.**

Les défis:

- **Optimiser les prompts** du VLM et du LLM de manière simultané

Choix du LLM ?

- GPT-2 (modèle léger basique d'OpenAI)
- Mistral-7B (LLM de Mistral)
- Qwen2.5-7B-Instruct (plus récent et fine-tuned sur les instructions humaines)
- **Gemma 3 4B (dernier modèle multimodal de Google)**

Modèle	Gemma 3-4B Instruct	Qwen 2.5-7B Instruct
Score d'inventivité	0.75	0.70
Taux d'hallucination	2%	30%

Génération des commentaires sportifs

3 VLM

Action Description (avec Qwen 2.5 VL 7B)

- *Exemple: "Messi took the penalty and scored for Barcelona, making it 1-0."*

Description textuelle des actions clés

4 LLM

Gemma 3 4B avec ce prompt: "Rephrase this sentence as a commentator: VLM output" exemples:

- *"What an incredible finish from the superstar! Absolutely world-class!"*
- *"The crowd erupts in cheers. Lionel Messi steps up to take the penalty and finds the back of the net for Barcelona, giving them a 1-0 lead."*
- *"The crowd goes wild as they celebrate this crucial goal. Messi steps up to take the penalty and scores for Barcelona, making it 1-0."*

Commentaires sportifs enrichis

En pratique:

- Le prompt VLM demande une génération d'une **description très détaillée au VLM** sur une video de 10 secondes.
- Le LLM concatène en 30 mots toute l'interpretation du VLM

- - - - - → gomme les **hallucinations** du VLM

5 Génération des commentaires audios

Base du TTS

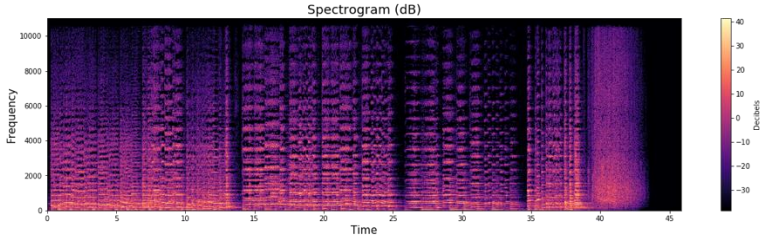
Texte

Texte d'entrée : "Bonjour, comment vas-tu ?"

Text-to-phonétique (Transformer/LSTM)

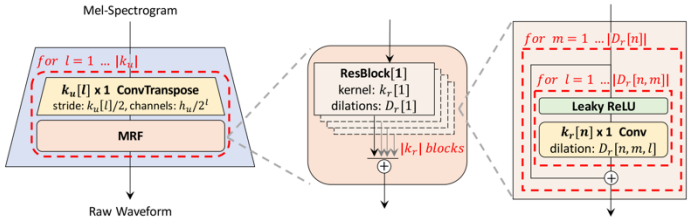
1. **Tokenisation** : ["Bonjour", ",", "comment", "vas", "-", "tu", "?"]
2. **Phonétisation** (IPA) : /bɔ̃.ʒuʁ/ /ko.mɑ̃/ /va/ /ty/
3. **Ajout de prosodie** : /bɔ̃ʒ̩uʁ/ (pause courte) /koˈmɑ̃/ /vaˈty/ (intonation montante)

Représentation spectrale (Transformer/CNN/LSTM)



Tacotron
FastSpeech

Vocoder (GAN)

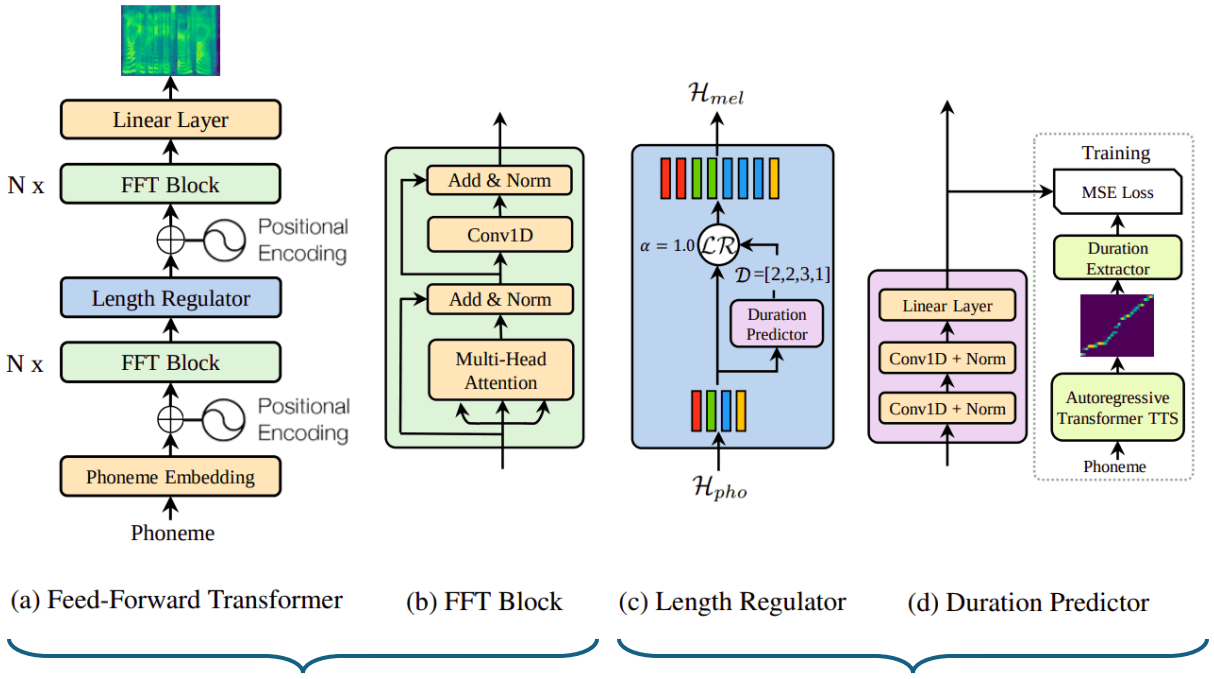


WaveGlow
HiFi-GAN

Onde Sonore

Output audio

Architecture d'un modèle Text-to-Speech (TTS)



(a) Feed-Forward Transformer (b) FFT Block (c) Length Regulator (d) Duration Predictor

Pipeline de génération de Spectrogramme

- Permet d'embedder le texte et émotions en spectrogramme

Durée et Vitesse

- Permet de réguler le débit de parole et la longueur de l'audio final

Choix du modèle TTS

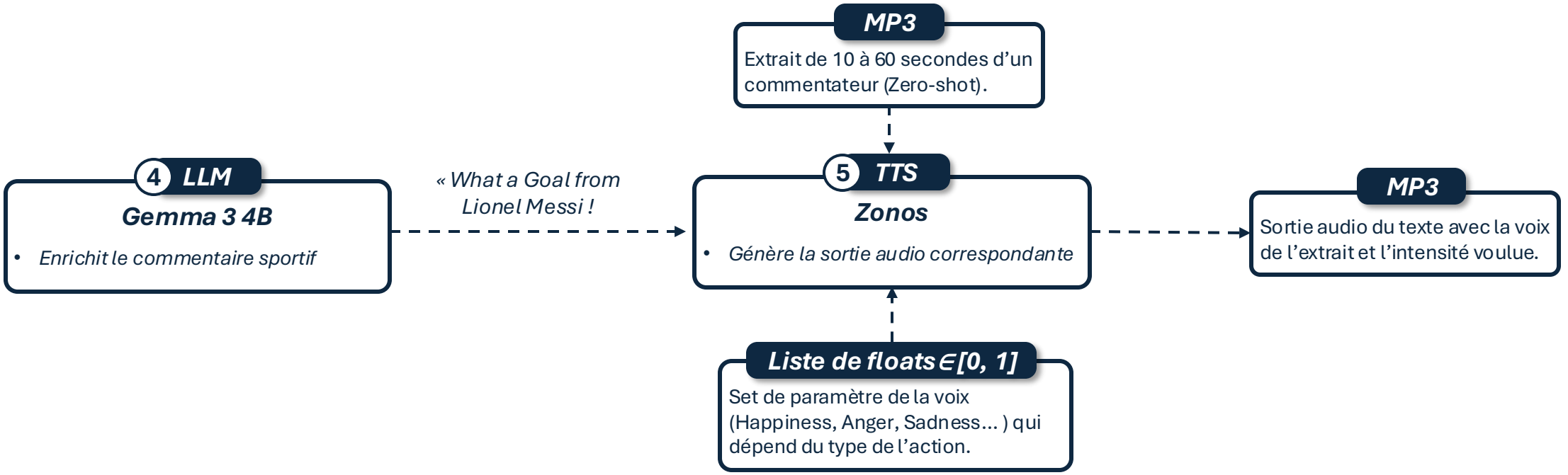
Points importants pour choisir le modèle de TTS sont : la **langue**, la possibilité de spécifier des **émotions** et de **fine-tuning,** le temps d'**inférence**, l'**open-source**.

Python Library

TTS (Coqui-AI) is open-source, simplifies fine-tuning, efficient and flexible.
Eleven Labs requires a license but is complete and has a lot of features.

TTS Model	Zonos	F5-TTS	Parler-TTS	Bark	XTTS-v2
Emotions/Style	Emotions (Fear, Anger, Happiness..)	Emotions, style, Background noise	Emotions, style, Background noise	Emotions, style, Background noise	Only in english and not robust
Fine Tuning	Fine-Tuning UI	Fine-Tuning UI	CLI Fine Tuning	Fine-Tuning class Methods	
Inference Time	< 10s for processing + generating (on ZERO)	10s for processing + generating (on ZERO)	10s (on ZERO)	Quite long (80s on T4)	Streaming Inference <200ms
Open Source ?	MIT License	MIT License	MIT License	MIT License	Non-commercial use

Génération des commentaires audios

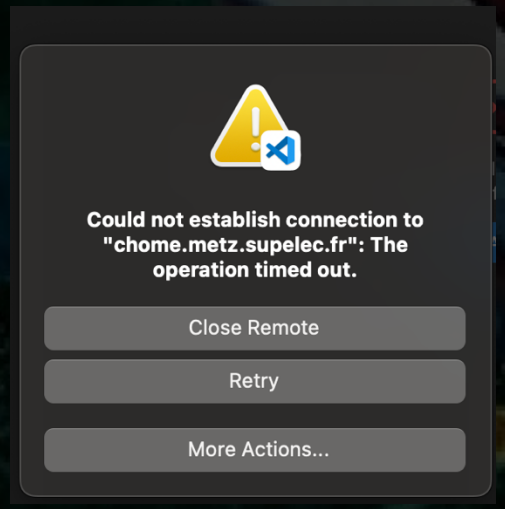


Exemples

The ball's in motion, midfield clash between TOTT and CHE! A pass forward... a quick dribble, dodging the blue shirt! A clever pass to a teammate... he shoots! GOOAAAL!!! The keeper dives, but it's in! TOTT celebrates, a joyous eruption from the crowd!

Right, there he is – a Tottenham player down! Another Chelsea player passes, showing concern. Medical staff rush onto the field immediately. He's being carefully attended to now. A coach assists him off the pitch slowly. The player is being carried off – a tough moment for Spurs!

Démonstration



Conclusion et Discussion

Objectif rempli, avec une valeur ajoutée pour Sia Partners

Cas d'usage pertinent pour les highlights

- Résumés YouTube, commentaires automatisés post-match, reporting visuel ou textuel (type L'Equipe)

Projet techniquement ambitieux mais faisable

- Pipeline complète combinant vision, langage et audio
- Limites actuelles liées au hardware (vidéos en entrée de VLM), pas à la faisabilité conceptuelle

Architecture modulaire et évolutive

- Pipeline en blocs facilement adaptable selon les besoins (prompts), l'évolution de l'état de l'art ou les ressources hardware

Veille technique approfondie

- Revue de l'état de l'art sur l'Action Spotting et le tracking sportif
- Nouveaux modèles GenAI testés au long du projet

Perspectives d'adaptabilité

- Méthodologies réutilisables pour d'autres sports ou contextes (basket, rugby, e-sport…)

Limites & Discussion

Difficulté d'inférence en temps réel

- Enchaînement Action Spotting + Tracking + VLM + LLM + TTS beaucoup plus long que la vidéo initiale (environ 5h pour 1h30 de match sur GPU 24Go VRAM). Impossible d'atteindre un streaming fluide sans GPU très performant.
- Pipeline conçue pour l'inférence post-match, pas en temps réelle

Limites de synchronisation temporelle hors highlights

- Longueur des clips limitées (10s max pour les VLMs)
- Commentaire sportif souvent plus long que l'action
  - Couper les commentaires = perte de fluidité
  - Laisser les commentaires se finir = décalage avec l'action suivante

Impact environnemental à considérer

- Inférence répétée de modèles lourds (VLMs, LLMs) malgré un coût économique faible (env. 3 euros sur un A100 Collab), engendre une consommation énergétique très importante
- Gain en valeur ajoutée justifie-t-il le **coût écologique** ?

Notes Annexes

Statistiques des outputs du model d'action spotting

- Taux de TP très élevé pour les actions rares, qui ont des changements de caméra marquants (goal, penalty, touche...)
- Taux de FN élevé élevé pour les actions moins marquantes. Cela met en lumière les limites du modèle pour certaines action (Tirs hors cadre, coup franc indirect)

Ball out of play	TP: 50	FP: 4	FN: 18	Precision: 0.93	Recall: 0.74
Clearance	TP: 6	FP: 2	FN: 14	Precision: 0.75	Recall: 0.30
Corner	TP: 7	FP: 2	FN: 2	Precision: 0.78	Recall: 0.78
Direct free-kick	TP: 1	FP: 1	FN: 5	Precision: 0.50	Recall: 0.17
Foul	TP: 10	FP: 4	FN: 31	Precision: 0.71	Recall: 0.24
Goal	TP: 6	FP: 0	FN: 1	Precision: 1.00	Recall: 0.86
Indirect free-kick	TP: 3	FP: 1	FN: 38	Precision: 0.75	Recall: 0.07
Kick-off	TP: 0	FP: 0	FN: 8	Precision: 0.00	Recall: 0.00
Offside	TP: 3	FP: 4	FN: 4	Precision: 0.43	Recall: 0.43
Penalty	TP: 2	FP: 0	FN: 0	Precision: 1.00	Recall: 1.00
Shots off target	TP: 1	FP: 0	FN: 7	Precision: 1.00	Recall: 0.12
Shots on target	TP: 0	FP: 0	FN: 18	Precision: 0.00	Recall: 0.00
Substitution	TP: 3	FP: 6	FN: 3	Precision: 0.33	Recall: 0.50
Throw-in	TP: 19	FP: 4	FN: 18	Precision: 0.83	Recall: 0.51
Yellow card	TP: 7	FP: 5	FN: 3	Precision: 0.58	Recall: 0.70