UNIVERSITÉ DE MONCTON
CAMPUS DE MONCTON

GRENOBLE INP UGA Ensimag

Grenoble INP – ENSIMAG
École Nationale Supérieure d'Informatique et de Mathématiques Appliquées

Rapport de stage Assistant Ingénieur

Effectué à l'université de Moncton

Classification intelligente des vidéos

Sara Bazouane
2ème année – Option MMIS

02 juin 2025 – 22 août 2025

Université de Moncton
18 Antonine-Maillet
E1A 3E9 Moncton, Canada

Responsable de stage
Éric Hervet
Tutrice de l'école
Valérie Bellynck

Résumé

Le département d'informatique de l'Université de Moncton mène plusieurs travaux de recherche en vision par ordinateur, couvrant des domaines tels que la détection d'objets, la segmentation ou encore la reconnaissance d'actions. Dans ce contexte, ce stage porte sur la classification intelligente de vidéos courtes à l'aide de techniques de deep learning.

Une partie du travail a consisté à appliquer la méthode de sélection adaptative des frames au dataset UCF-101, afin de réduire la redondance tout en conservant les informations pertinentes. Les frames ainsi sélectionnées ont servi de base à l'extraction de features via les réseaux ResNet-50 et ResNet-152.

À partir de ces features, plusieurs stratégies de classification ont été évaluées, notamment un MLP et un XGBoost. Les résultats atteignent jusqu'à 97.7 % de précision, proches des performances des solutions de l'état de l'art. Le modèle ConvX-LSTM était également intéressant, en raison de son fort potentiel à capturer la dynamique temporelle et la structure séquentielle des vidéos.

Remerciements

Je tiens à remercier chaleureusement mon encadrant Éric Hervet pour son accompagnement, sa disponibilité et ses conseils tout au long de ce travail. Je souhaite remercier également Mme Michelle Nowlan et Mr Oussema Antri pour leur soutien tout au long de mon expérience à Moncton. Depuis le moment où j'ai reçu l'offre de stage jusqu'à mon installation sur place, ils ont toujours été présents pour m'accompagner, répondre à mes questions et faciliter chaque démarche. Enfin, je remercie plus largement l'ensemble de l'équipe de l'Université de Moncton ainsi que l'équipe Mitacs pour cette expérience à la fois agréable et enrichissante.

Table des matières

1 Introduction

Ces dernières années, la quantité de vidéos produites et stockées sur nos appareils (smartphones, ordinateurs, etc.) et dans le cloud a connu une croissance exponentielle, ce qui soulève la problématique de la recherche et du tri automatique des vidéos, que ce soit à partir d'une description spécifique ou par similarité avec d'autres contenus. Parallèlement, le Deep Learning s'est révélé particulièrement performant dans de nombreux domaines, notamment pour la reconnaissance et la classification automatiques d'images. Toutefois, des défis subsistent lorsqu'il s'agit d'étendre ces approches aux films et aux vidéos, qui comportent une dimension temporelle complexe.

L'objectif principal de notre projet est de concevoir et d'expérimenter un réseau de neurones capable de classifier les vidéos issues du dataset UCF-101, qui regroupe plus de 13 000 vidéos réparties en 101 classes. Cette mission constitue une première étape vers un objectif plus large : réaliser la multi-classification de films et séries, permettant par exemple de déterminer qu'un film (ou une série) est à la fois dramatique et romantique. Une telle approche pourrait être très pertinente pour des entreprises comme Netflix, afin de classer automatiquement leurs contenus, faciliter la recommandation personnalisée et la gestion de vastes catalogues vidéo.

Cependant, comme il s'agit d'un stage de 12 semaines, il n'est pas réaliste de s'attendre à réaliser l'ambitieux objectif de la classification intelligente et multi-genre des films. Ainsi, nos objectifs spécifiques sont :

1. Comprendre l'état de l'art en classification intelligente des vidéos.
2. Développer un modèle capable de classifier les vidéos avec une précision élevée.
3. Évaluer les performances du modèle sur le dataset UCF-101 et comparer différentes architectures.

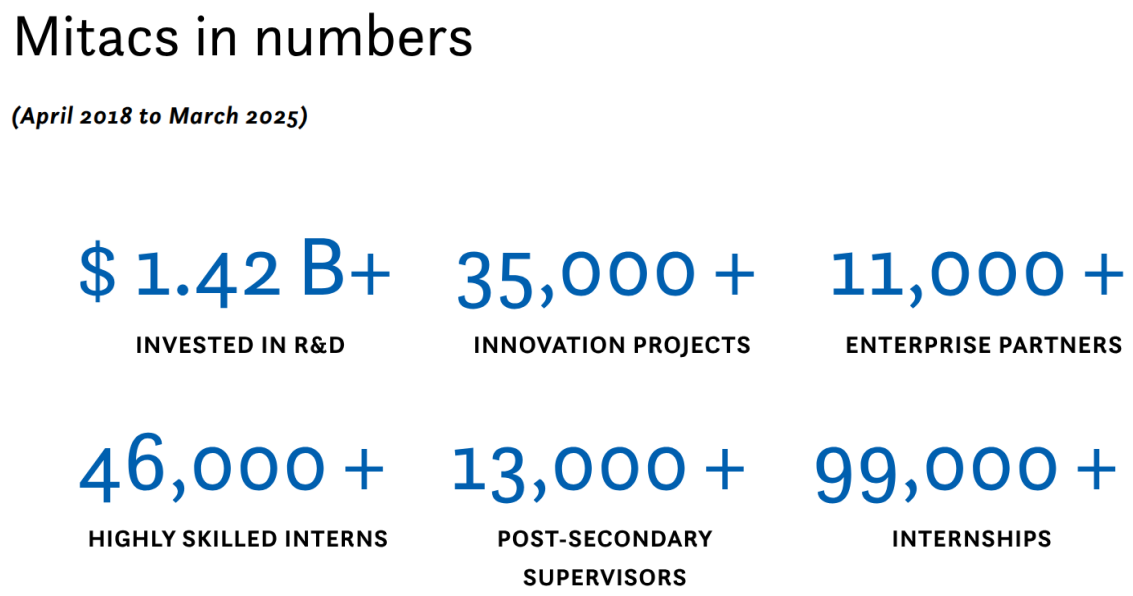
2 Présentation du cadre du stage

2.1 Mitacs Globalink Resarch Internship (GRI)

Ce stage s'inscrit dans le cadre du programme **Mitacs Globalink Research Internship (GRI)**, une initiative canadienne qui permet à des étudiants internationaux d'effectuer un stage de recherche de douze semaines au sein d'universités canadiennes. Ce dispositif favorise les échanges scientifiques et la collaboration entre le Canada et les établissements d'enseignement supérieur à travers le monde.

Mitacs :

Mitacs est un organisme canadien à but non lucratif qui soutient la recherche et l'innovation au Canada depuis plus de vingt ans, en collaboration avec plus de soixante-dix universités canadiennes, six mille entreprises et les différents paliers du gouvernement.



La mission de Mitacs :

La mission de Mitacs est d'*inspirer l'innovation et de concevoir des solutions créatives* aux défis contemporains. L'organisme joue un rôle de catalyseur au sein de l'écosystème canadien de la recherche et de l'innovation, en favorisant la collaboration entre les universités, les entreprises et les institutions publiques. Mitacs s'attache à bâtir une communauté dynamique et diversifiée de chercheurs et d'innovateurs de haut niveau. Pour y parvenir, il attire des talents prometteurs du monde entier, les met en relation avec le secteur privé, et facilite la mise en œuvre de projets ambitieux répondant à des besoins concrets de la société et de l'économie.

2.2 Département de l'informatique de l'université de Moncton

L'Université de Moncton, fondée en 1963, est la plus grande université francophone du Canada à l'extérieur du Québec. Elle joue un rôle central dans la production et la diffusion du savoir au sein de la communauté acadienne et francophone. Dans le cadre du programme **GRI**, mon stage s'est déroulé au sein du **Département d'informatique** de cette université, un pôle reconnu pour la qualité de sa recherche appliquée et de sa formation.

Le département d'informatique de l'Université de Moncton se distingue par ses travaux de recherche dans des domaines de pointe tels que :

- L'intelligence artificielle et l'apprentissage profond.
- L'analyse et la vision par ordinateur.
- La cybersécurité et la protection des données.
- Les systèmes distribués et les réseaux intelligents.
- Le développement logiciel et les technologies éducatives.

Les chercheurs du département participent activement à des projets financés par le CRSNG (Conseil de recherches en sciences naturelles et en génie du Canada), le programme Mitacs, et d'autres organismes provinciaux et fédéraux. Ils publient régulièrement dans des revues internationales et collaborent avec plusieurs universités canadiennes et européennes. Le département est également engagé dans la valorisation de la recherche à travers des applications concrètes dans les domaines de la santé, de l'éducation et des technologies immersives.

Dans ce contexte, le département accorde une attention particulière à la **vision par ordinateur**. C'est dans cette continuité que s'inscrit mon stage, qui vise à développer et expérimenter des modèles capables de reconnaître automatiquement des actions dans des vidéos courtes.

3 Compréhension du problème : Dataset et État de l'art

3.1 Problématique

Comme énoncé dans l'inroduction, la classification intelligente des vidéos est à la croisée de la vision par ordinateur et de l'apprentissage profond. Alors que la classification d'images est aujourd'hui bien maîtrisée grâce aux réseaux de neurones convolutionnels (CNN), les vidéos ajoutent une dimension temporelle complexe : l'action est une succession de frames.

Ainsi, la tâche ne consiste pas seulement à reconnaître des objets présents dans une image, mais à comprendre l'évolution du contenu visuel dans le temps. Cela exige de modéliser à la fois les caractéristiques spatiales (formes, textures, objets) et les caractéristiques temporelles (mouvements, transitions, rythmes).

Dans le cadre de ce projet, la problématique donc peut se formuler ainsi : Quels modèles sont pertinents pour la classification des vidéos, tout en limitant la complexité de calcul et en conservant une bonne précision ?

Pour répondre à cette question, nous avons décidé de travailler sur le dataset UCF101.

3.2 Analyse exploratoire du dataset UCF-101

Le dataset **UCF101** est un jeu de données réaliste et exigeant, utilisé pour la recherche en vision par ordinateur. Il constitue une extension de UCF50 et contient **13 320 vidéos** issues de YouTube et réparties en **101 classes** d'actions.

Ces 13 320 vidéos sont organisées en **25 groupes** (g01–g25), chaque groupe comprenant 4 à 7 vidéos d'une même classe. Les clips d'un même groupe partagent souvent des caractéristiques visuelles (même décor, même angle de prise de vue, etc.).

Les 101 classes se répartissent en 5 grandes familles d'actions :

1. **Body motion.**
2. **Human–human interactions.**
3. **Human–object interactions.**
4. **Playing musical instruments.**
5. **Sports.**

Durant la première semaine du stage, notre mission a consisté d'analyser la répartition des classes dans le dataset. L'objectif était de détecter les éventuelles classes sur- ou sous-représentées, susceptibles d'introduire un biais lors de l'entraînement du modèle de classification. Pour réaliser cette analyse, je me suis intéressée au nombre total de frames par

classe et la répartition des durées des vidéos par classe. Les deux figures ci-dessous résument les résultats obtenus :

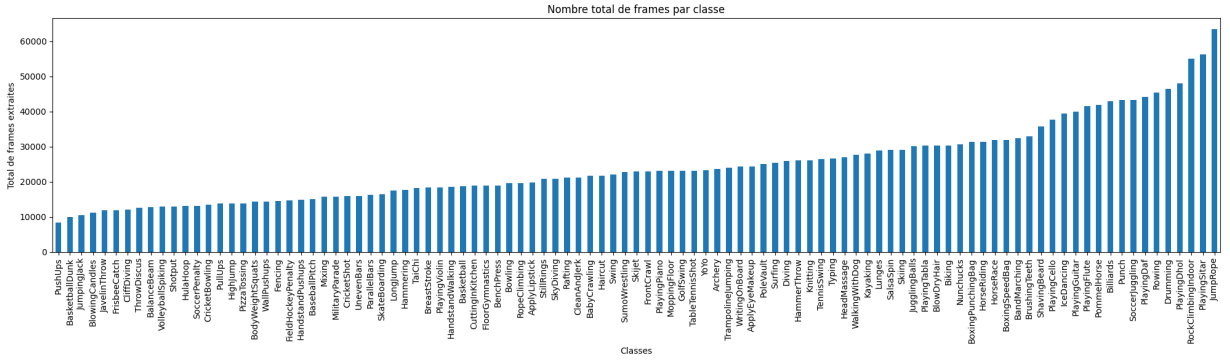


FIGURE 1 – Nombre total de frames par classe dans UCF-101.

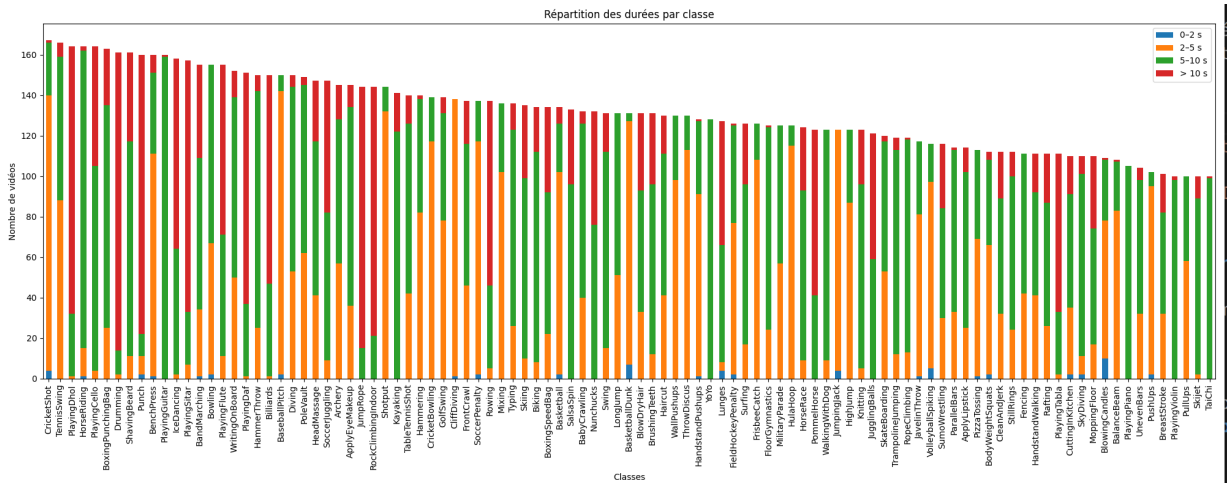


FIGURE 2 – Répartition des durées des vidéos par classe.

Les résultats obtenus montrent une forte variation entre les durées de vidéos par classe, entraînant un volume d'images très inégal. La majorité des classes montrent un pic dans la tranche 5–10 s, indiquant que les vidéos sont cadrées pour capturer l'action complète sans excès. La seconde catégorie la plus représentée est 2–5 s, surtout pour les gestes rapides ou isolés. Les clips < 2 s sont très rares, car trop courts pour décrire un mouvement entier. Certaines actions cycliques (> 10 s) comme le Jumprope ou le Drumming sont capturées par des clips plus longs.

En somme, la répartition des durées des vidéos par classe varie fortement et n'est donc pas homogène. Cette hétérogénéité nous incite à réordonner les données et à les organiser par

groupes avant le pré-traitement, car un entraînement aléatoire du modèle sur des classes très disparates risquerait de créer des biais.

C'est pourquoi, le dataset UCF-101 a été préparé en 25 groupes (g01–g25). Cette structuration permet de réduire l'impact des classes très longues ou très courtes et de lisser les déséquilibres observés entre les classes. La figure ci-dessous illusre cette répartition :

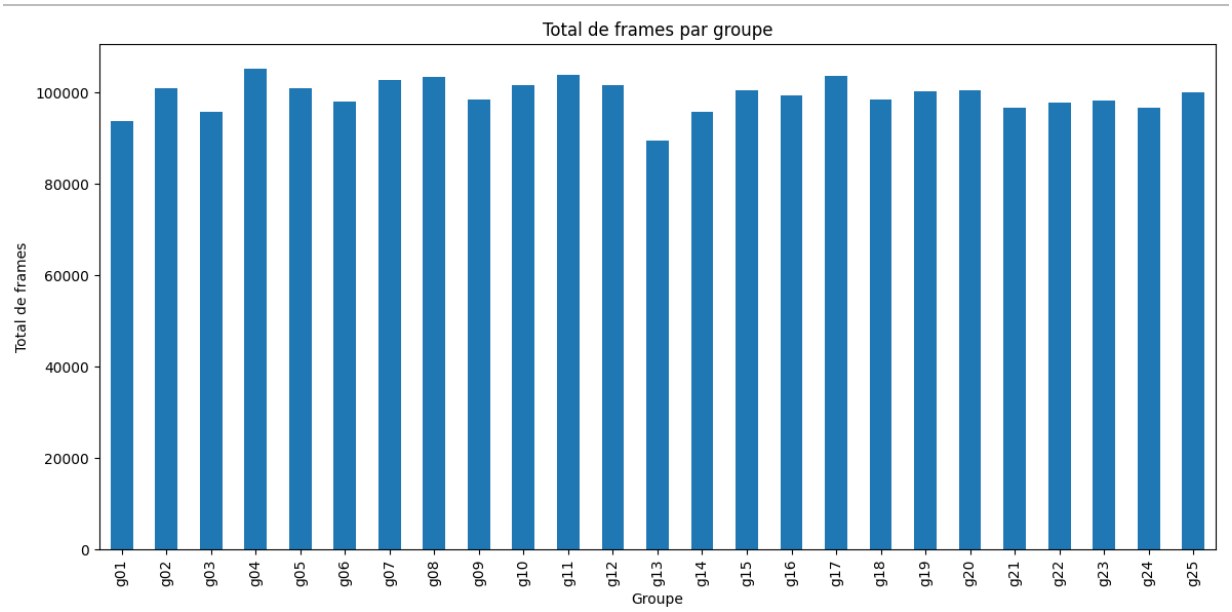


FIGURE 3 – Nombre total de frames par groupe dans UCF-101.

Le volume total de frames par groupe varie entre 89.000 frames et 105.000 frames. Donc, nous remarquons un écart de près de 16 % autour de la moyenne (99.000 frames), reflétant des durées moyennes de vidéos plus longues ou plus courtes selon le groupe. Par conséquent, la répartition homogène des vidéos et des frames entre les groupes permet d'appliquer directement les pré-traitements standard (fenêtrage, augmentation, normalisation) et de déployer des architectures telles que les CNN ou les LSTM sans ajustements spécifiques par groupe.

3.3 L'état de l'art des solutions de classification de UCF-101

Les caractéristiques du dataset influencent fortement le choix du modèle et sa performance. Dans cette séction, l'analyse de l'état de l'art se concentre spécifiquement sur les méthodes développées pour la classification des vidéos du dataset UCF-101 plutôt que la classification vidéo générale.

Bien que de nombreuses approches existent pour l'action recognition sur UCF-101, les solutions présentées dans le tableau ci-dessous sont les plus récentes et les plus pertinentes en termes de performance.

Dans le domaine de la classification des vidéos, la précision (accuracy) est la métrique de référence pour évaluer la performance des modèles. Cependant, la pertinence d'un modèle ne se limite pas uniquement à sa précision : elle dépend également de la quantité de ressources nécessaires à son exécution (mémoire, puissance de calcul) et du temps requis pour effectuer la classification.

TABLE 1 – Synthèse des méthodes de pointe en reconnaissance d'actions sur le dataset UCF-101.

Méthode	**Année**	**Précision (%)**
OmniSource	2020	98.6
SMART	2020	98.64
BIKE	2023	98.8
VideoMAE	2023	99.6
OmniVec2	2024	99.6
FTP-UniFormerV2-L	2024	**99.7**

Chaque méthode repose sur une idée clé. Nous présentons brièvement ci-dessous les principes essentiels de chaque modèle :

- **OmniSource**[1] : Cette approche unifie les formats image et vidéo pour l'apprentissage faiblement supervisé sur le web (Cet apprentissage permet d'exploiter la quantité massive des données disponibles sur Internet).
- **SMART**[2] : Cette méthode sélectionne les frames les plus informatives grâce à un sélecteur Monotrame et à un sélecteur Global. Chaque frame reçoit un score de pertinence, et les frames les plus significatives sont ensuite utilisées dans la classification.
- **BIKE**[3] : Ce modèle crée une connexion bidirectionnelle entre la vidéo et le texte. Par exemple, la description "Football" alerte le modèle pour qu'il se concentre sur les pieds et le ballon dans la vidéo (texte vers vidéo), tandis que, lors de l'analyse de la vidéo, le modèle va repérer des objets comme l'objet "filet" et les associer au mot "football"(vidéo vers texte).
- **VideoMAE V2-g**[4] : VideoMAE V2-g est un modèle qui apprend le contenu des vidéos en créant des résumés numériques très détaillés appelés embeddings. Pour UCF-101, ces résumés sont utilisés par un petit modèle de classification qui identifie l'action montrée dans chaque vidéo. VideoMAE emploie un dual masking et un entraînement progressif pour construire des modèles de fondation vidéo efficaces avec

des milliards de paramètres. L'utilisation de ces modèles nécessite des GPU puissants et un temps d'entraînement élevé.

- **OmniVec2**[5] : OmniVec2 utilise des tokenizers spécialisés par modalité et un transformeur partagé avec attention croisée, permettant un apprentissage multitâche unifié. Cette approche est plus coûteuse en ressources mais améliore la généralisation multimodale.
- **FTP-UniFormerV2-L**[6] : FTP combine des Vision Transformers et des Video Language Models avec quatre processeurs pour obtenir une représentation enrichie des actions. Cette méthode atteint la meilleure précision mais au prix d'une consommation de ressources et d'un temps d'inférence plus élevés.

Une autre approche qui mérite l'attention, malgré sa précision - 98,05% légèrement inférieure à celle des solutions précédemment présentées, est la sélection adaptative des frames, proposée par Rahnama, Esfahani et Mansouri [7]. Elle met en évidence l'importance de fournir au modèle un ensemble de frames représentatif pour que l'identification des actions soit précise. Plutôt que de traiter toutes les frames de la vidéo ou d'en extraire un nombre fixe, la sélection adaptative des frames choisit celles qui apportent des informations nouvelles et complémentaires en utilisant le coefficient de Dice.

Les frames retenues sont ensuite traitées par un réseau de neurones convolutionnel préentraîné, le ResNet-50, qui extrait les caractéristiques visuelles pertinentes de chaque image. Ces caractéristiques sont ensuite regroupées et utilisées comme entrée d'un perceptron multi-couches (MLP) qui effectue la classification finale de l'action. Cette combinaison réduit le nombre de frames à traiter tout en conservant suffisamment d'information pour obtenir des performances élevées.

3.4 Vers notre solution

Comme constaté dans la séction précédente, plusieurs approches pertinentes et performantes existent pour la reconnaissance d'actions dans le dataset UCF-101. Certaines atteignent presque la perfection en précision. Devant de tels réalisations, nous nous sommes alors demandés s'il n'était pas possible d'optimiser davantage le compromis entre précision et consommation de ressources, ou encore de développer une nouvelle architecture tout aussi pertinente que celles présentées dans l'état de l'art.

3.4.1 La sélection adaptative des frames

Dans un premier temps, nous avons choisi de reproduire la méthode de sélection adaptative des frames, dans le but de valider expérimentalement les résultats annoncés et d'atteindre la précision de référence de 98,05 %. Cette approche nous a semblé particulièrement intéressante, car elle repose sur une architecture relativement simple. De plus, nous ne disposions

que de la description conceptuelle de la méthode étant donné que les auteurs n'ont publié ni code source ni implémentation, ce qui a rendu la reproduction du modèle à la fois plus exigeante et plus formatrice. Par la suite, nous avons également envisagé d'optimiser cette approche en expérimentant différentes configurations, notamment en remplaçant le ResNet-50 initialement utilisé par un ResNet-152. Ce choix se justifie par le fait que ResNet-152, grâce à sa profondeur accrue et à ses blocs supplémentaires, permet d'extraire des caractéristiques visuelles plus riches et plus discriminantes, ce qui pouvait potentiellement améliorer la qualité des embeddings utilisés pour la classification des actions. L'idée était ainsi de vérifier si une meilleure extraction de caractéristiques spatiales pouvait compenser la relative simplicité de l'architecture temporelle, et ainsi conduire à une amélioration mesurable des performances globales.

3.4.2 ConvXLSTM : une architecture avancée pour la classification vidéo

Dans un deuxième temps, nous nous sommes intéressés à l'architecture ConvXLSTM, d'une part parce qu'il s'agit d'un modèle relativement récent, encore peu exploré dans la littérature scientifique, et d'autre part parce qu'il propose une évolution pertinente du ConvLSTM classique, largement utilisé pour la reconnaissance d'actions dans les vidéos. En effet, si les ConvLSTM ont permis d'importants progrès en combinant l'extraction spatiale des caractéristiques via des convolutions et la modélisation temporelle grâce aux LSTM, ils présentent néanmoins plusieurs limitations. Parmi celles-ci, on retrouve :

- Problème de dépendances à long terme : les LSTM traditionnels ont tendance à perdre des informations contextuelles importantes lorsqu'ils traitent de longues séquences vidéo.
- Capacité d'attention limitée : le ConvLSTM ne possède pas de mécanisme explicite lui permettant de se concentrer sur les instants les plus pertinents de la séquence.
- Inefficacité computationnelle : son architecture récurrente et séquentielle est difficile à paralléliser, ce qui la rend peu adaptée au traitement de séquences vidéo longues.
- Problème du vanishing gradient : malgré les améliorations apportées par rapport aux RNN classiques, le modèle devient difficile à optimiser sur de grandes profondeurs temporelles.
- Manque de granularité : le ConvLSTM traite toutes les caractéristiques extraites de manière uniforme, sans différencier leur importance relative dans la séquence temporelle.

Le modèle ConvXLSTM vise à surmonter ces limites en intégrant des mécanismes d'état améliorés et des techniques d'attention inspirées des Transformers, qui permettent une meilleure modélisation des dépendances temporelles complexes.

Au fil du stage, nous avons également exploré plusieurs autres pistes afin d'affiner notre compréhension du problème. Diverses architectures et algorithmes ont ainsi été testées, notamment des modèles et des méthodes comme XGBoost ou des combinaisons hybrides entre réseaux convolutionnels et classifieurs plus légers, ou encore des techniques différentes de pooling et de batching pour un même modèle. Toutefois, dans le cadre de ce rapport, nous nous concentrerons uniquement sur les deux approches principales — la sélection adaptative des frames et le modèle ConvXLSTM — afin de mettre en avant les contributions les plus significatives du travail réalisé.

4 Modèles et Résultats

Dans cette section, nous allons présenter pour chaque approche son principe de fonctionnement, les choix d'implémentation, ainsi que les résultats obtenus et les métriques utilisées pour évaluer la pertinence et l'efficacité.

4.1 Première approche : Adaptative Frame Selection et MLP

4.1.1 Architecture du modèle

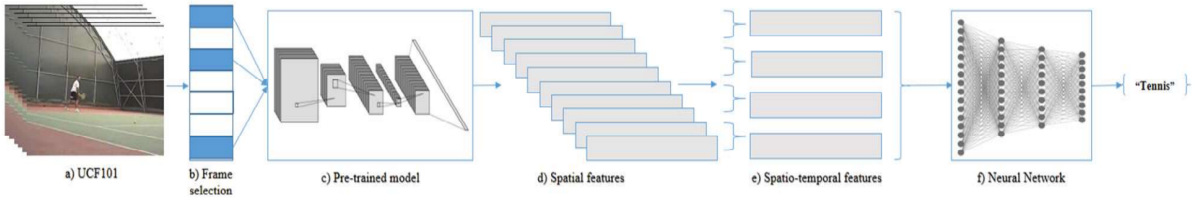


FIGURE 4 – Schéma illustrant le principe de la méthode de sélection adaptative des frames.

Supposons que nous disposons d'une courte vidéo issue du dataset UCF-101 montrant un joueur de tennis en train de frapper la balle. Le processus de classification repose alors sur trois étapes :

A. Sélection des frames les plus pertinentes :

Cette méthode débute par l'extraction des deux premières frames de la vidéo. Ensuite **l'indice de Dice** est calculé pour ces deux frames, selon la formule suivante :

$$\mathrm{D(F(i,j),SF(i,j))} = \frac{2 \times F(i,j) \times SF(i,j)}{F(i,j)^2 + SF(i,j)^2}$$

où $F(i,j)$ représente le pixel situé à la position i,j dans la première frame F, et $SF(i,j)$ désigne le pixel occupant la même position dans la seconde frame. Le calcul est réalisé pour l'ensemble des pixels de la frame, puis la moyenne des valeurs obtenues est prise afin d'obtenir le **score de similarité global** entre les deux images.

Une valeur du coefficient D proche de 1 indique que les deux frames sont très similaires et apportent la même information, tandis qu'une valeur proche de 0 indique qu'elles diffèrent fortement et apportent des informations complémentaires.
Nous présentons ci-dessous notre implémentation optimisée du calcul du coefficient de Dice :

```python
# Frame selection algorithm :
# frameOne & frameTwo sont des tenseurs d'ordre 3.
# frameOne[y, x, 0] = valeur du rouge au pixel (x,y).
def Dice(frameOne, frameTwo):
    """ Calcule le taux de similarité """
    # Pour plus de précision dans les calculs :
    f1 = frameOne.astype(np.float32)
    f2 = frameTwo.astype(np.float32)

    # Opérations vectorisées pour une efficacité maximale :
    cMul = 2.0 * f1 * f2
    cPow = f1 * f1 + f2 * f2

    cDivid = np.divide(cMul, cPow, out=np.zeros_like(cMul), where=(cPow != 0))
    return float(np.mean(cDivid))
```

Par la suite, les deux premières frames sont retenues par défaut, car elles servent à initialiser le processus de sélection. À partir de la troisième frame, le coefficient de Dice entre la frame courante et la dernière frame conservée est calculé. Si la similarité obtenue est inférieure à la moyenne des coefficients de Dice des frames déjà retenues, cela indique que la nouvelle frame apporte une nouvelle information et elle est donc ajoutée à l'ensemble des frames sélectionnées. Dans le cas contraire, elle est jugée redondante et ignorée. Ce processus se répète sur l'ensemble de la séquence vidéo, de manière à ne conserver que les frames les plus représentatives et informatives.

Notre implémentation de cette logique de sélection est présentée ci-dessous :

```python
def FrameCheck(frames, flag_resize=True):
    """ Sélectionne les frames informatives et complémentaires """
    if len(frames) < 4:
        return frames

    # Optimisation: Utiliser des frames de plus petite taille.
    if flag_resize:
        dice_frames = [cv2.resize(f, (160, 120)) for f in frames]
    else:
        dice_frames = frames

    finalFrame_indices = [0]
    tempFrame = dice_frames[0]
    tempResult = []

    if len(dice_frames) > 1:
        tempResult.append(Dice(tempFrame, dice_frames[1]))
```

```python
    for frameIndex in range(2, len(dice_frames)-1):
        algoritmResult = Dice(tempFrame, dice_frames[frameIndex])
        if len(tempResult) > 0 and np.mean(tempResult) > algoritmResult:
            finalFrame_indices.append(frameIndex)
            tempFrame = dice_frames[frameIndex]
            tempResult.append(algoritmResult)

    #  Retourner à la résoluion originale :
    return [frames[i] for i in finalFrame_indices]
```

B. Extraction des caractéristiques (features) :

Les frames sélectionnées sont transmises au ResNet-50 préentraîné sur ImageNet, un dataset de plus d'un million d'images réparties sur 1000 classes. Ce modèle est utilisé uniquement comme extracteur de caractéristiques : sa dernière couche de classification est retirée pour ne conserver que la partie convolutionnelle, responsable de l'extraction des motifs visuels (formes, textures, contours, etc.). Chaque frame est ainsi convertie en un vecteur de 2048 caractéristiques. Nous avons choisi de ne pas effectuer de fine-tuning du ResNet-50, car notre dataset, composé d'environ 700 000 frames après la sélection adaptative, reste bien plus restreint qu'ImageNet.

C. Classification :

Les vecteurs de features obtenus sont regroupés en quatre sous-ensembles sur lesquels un max pooling est appliqué. Cette opération permet à la fois de réduire la taille des données et de conserver les informations les plus pertinentes. On se retrouve ainsi avec

quatre vecteurs de taille 2048, ensuite concaténés pour former un unique vecteur de taille 8192 représentant la vidéo. Ce vecteur est transmis à un Perceptron Multi-Couche (MLP) chargé de la classification finale. Sa couche de sortie produit les probabilités associées aux 101 classes du dataset UCF-101.

4.1.2 Métriques d'évaluation et résultats

Métriques d'évaluation :

Dans le domaine de la *reconnaissance d'actions*, plusieurs métriques peuvent être utilisées pour évaluer les performances d'un modèle, telles que la précision (*accuracy*), la précision moyenne (*mean average precision*) ou encore le rappel (*recall*). Dans le cadre de ce projet, nous avons choisi d'utiliser **la précision (accuracy)** comme principale métrique d'évaluation.
Elle est définie par la formule suivante :

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100$$

où N_{correct} représente le nombre de vidéos dont la classe prédite correspond à la classe réelle, et N_{total} désigne le nombre total de vidéos testées. Cette métrique convient bien au dataset UCF-101, dont les classes sont globalement équilibrées.

En complément de la précision, nous considérons également le temps d'exécution et les ressources matérielles utilisées comme des indicateurs clés de performance et d'efficacité computationnelle.

Résultats et tentatives d'optimisation :

Bien que nous ayons suivi la méthode décrite dans l'article "Adaptive Frame Selection in Two Dimensional Convolutional Neural Network", certains éléments essentiels pour une reproduction exacte des résultats ne sont pas précisés. En particulier, le nombre d'époques d'entraînement et de batches, le type de matériel utilisé (par exemple l'usage éventuel de GPU A100), ainsi que la fonction d'activation de la dernière couche du classifieur MLP ne sont pas mentionnés. Nous avons donc dû compléter ces informations manquantes en nous basant sur des choix raisonnables et conformes aux pratiques courantes.

Dans le tableau ci-dessous, nous présentons les résultats obtenus suite à la reproduction du modèle ResNet50 & MLP, ainsi que ceux issus de nos propres variantes et tentatives d'améliorations.

Au cours de ce travail, nous avons également procédé à plusieurs optimisations du code — notamment pour réduire la consommation mémoire, améliorer la précision des calculs et

accélérer l'exécution par parallélisation. Cependant, nous concentrons ici notre analyse sur les idées d'optimisation structurelles appliquées.

TABLE 2 – Résultats obtenus pour ResNet-50 & MLP et nos variantes.

Architecture	**Précision (%)**	**Temps (min)**	**Matériel**
ResNet-50 + MLP	97.7	5.1	2 x A100 (40 GB)
ResNet-50 + XGBoost	91.5	98	CPU - 48 coeurs
ResNet-152 + MLP	96.8	9	A100, 12-cœurs

Le temps d'exécution indiqué correspond au temps global, incluant la phase de sélection des frames, l'entraînement du classifieur et l'évaluation sur le jeu de test. L'ensemble du dataset UCF-101 a été réparti de manière équilibrée en 72% pour l'entraînement, 18% pour la validation et 10% pour le test.

Commentaire :

Les résultats présentés dans la table 2 mettent en évidence l'impact des choix d'architecture et des ressources matérielles sur la performance et le temps d'exécution. Le modèle **ResNet-50 associé à un MLP** offre le meilleur compromis entre rapidité et précision, atteignant une exactitude de **97.7 %** en seulement **5 minutes** grâce à l'utilisation de deux GPU A100.

À l'inverse, le **ResNet-152**, bien que plus profond, n'a atteint que **96.8 %** de précision. Cette baisse s'explique par un surapprentissage et une optimisation moins efficace du classifieur pour des descripteurs plus complexes. De plus, le modèle n'a pu être exécuté que sur un seul GPU A100, l'allocation des ressources étant gérée automatiquement par **Compute Canada**.

Malgré ces contraintes, le modèle ResNet-152 conserve une excellente cohérence de prédiction, avec une **Top-5 accuracy de 99.3 %**, indiquant que la classe correcte figure presque toujours parmi les cinq prédictions les plus probables. Ces résultats démontrent que la profondeur du réseau ne garantit pas nécessairement de meilleures performances.

Enfin, le **XGBoost** a été testé comme alternative non neuronale pour la classification, afin d'évaluer si un modèle plus léger, fondé sur des arbres de décision, pouvait rivaliser avec le MLP sans recours au GPU. Cependant, ses performances (**91.5 %**) sont restées inférieures, car les descripteurs visuels extraits par le ResNet présentent une forte non-linéarité que les modèles à base d'arbres captent moins efficacement que les réseaux neuronaux profonds.

4.1.3 Perspectives d'amélioration

Les résultats obtenus sont **encourageants**, car ils sont proches des performances de l'état de l'art. Cependant, la **contrainte de temps**, ainsi que la performance décevante du **ResNet-152**, qui n'a pas apporté de gain par rapport au **ResNet-50**, nous ont amenés à repenser la direction de notre recherche.

Nous avons décidé de nous concentrer sur la façon dont les images sont traitées dans le temps, plutôt que de nous focaliser uniquement sur l'extraction des features pertinentes. En effet, dans notre modèle actuel, toutes les frames d'une vidéo sont analysées simultanément, ce qui revient à ignorer la chronologie des mouvements. Or, un être humain reconnaît une action en regardant les images **les unes après les autres**, et en reliant les gestes observés entre eux.

Néanmoins, plusieurs **perspectives d'amélioration** restent envisageables, notamment l'ajout d'une régularisation plus fine, l'expérimentation de fonctions d'activation récentes comme *GELU* ou *Swish*

4.2 Deuxième approche : ConvXLSTM

4.2.1 Architecture du modèle

Le modèle ConvXLSTM reprend la structure du ConvLSTM, mais intègre le module xLSTM avancé à la place du LSTM, afin d'améliorer la modélisation temporelle. Son architecture complète se compose :

A. Encodeur CNN :

Un réseau ResNet-152 préentraîné est utilisé pour extraire les charactéristiques pertinentes à partir de chaque trame vidéo. Les trames utilisées à cette étape sont sélectionnées à l'aide de la sélection adaptative basée sur le coefficient de Dice, afin de garantir un ensemble d'informations représentatif et non redondant.

B. Module xLSTM :

Le module xLSTM est une séquence de couches sLSTM et mLSTM, chargées de traiter les caractéristiques extraites par le ResNet-152.

- *sLSTM (State LSTM)* : une version qui améliore la gestion d'état par :
  - Normalisation par couches pour stabiliser l'entraînement.
  - Convolutions causales pour capturer les dépendances temporelles locales.
  - Mécanismes d'oubli et d'entrée optimisés grâce à la normalisation.

    - Intégration d'un réseau feed-forward avec activation GELU.
- *mLSTM (Mixture LSTM)* : une variante du LSTM intégrant des mécanismes d'attention :
    - Calcul de scores d'attention query–key–value similaire aux transformers.
    - Normalisation par groupes pour chaque tête d'attention.
    - Connexions résiduelles facilitant la propagation du gradient.
    - Projection des représentations latentes dans un espace de dimension supérieure.

C. Module d'Attention :

Un mécanisme d'attention temporelle qui pondère l'importance de chaque instant de la séquence.

D. Couches de Classification :

Un réseau feed-forward qui transforme les représentations temporelles en prédictions de classes d'actions.

4.2.2 Résultats et perspectives d'amélioration

L'entraînement du modèle s'est révélé contraignant sur le plan computationnel : chaque exécution nécessitait plusieurs heures à cause de la profondeur du réseau et du coût élevé du module xLSTM. À cela se sont ajoutés des problèmes récurrents d'explosion du gradient, qui ont rendu l'optimisation instable et ont empêché l'obtention de modèles véritablement satisfaisants dans le temps imparti. Ces limitations ne reflètent pas un blocage méthodologique, mais plutôt une contrainte temporelle forte, ne permettant pas d'explorer pleinement les stratégies de régularisation, de normalisation ou de tuning nécessaires pour stabiliser et finaliser le modèle.

5 Bilan

Ce stage a été une expérience agréable et enrichissante, autant sur le plan technique que personnel. J'ai découvert concrètement ce qu'implique un vrai projet de recherche, avec ses exigences, ses imprévus et surtout son besoin d'autonomie.

Durant ce stage, l'un des principaux défis que j'ai rencontrés était l'attente extrêmement longue pour obtenir des ressources et exécuter mes programmes. Les serveurs de Compute Canada étant partagés par des universités, des laboratoires et des entreprises de tout le pays, il fallait souvent patienter longtemps avant de pouvoir exécuter un job. En moyenne, près de 15 000 personnes étaient en file d'attente, ce qui faisait varier le délai d'exécution entre 45 minutes, plusieurs heures, voire parfois des journées complètes. Cela ralentissait fortement la phase d'expérimentation et ajoutait beaucoup de stress : il suffisait d'oublier un petit détail dans le code pour devoir patienter à nouveau 3 ou 4 heures avant de découvrir une simple erreur, puis recommencer. À cela s'ajoutaient les temps d'exécution très longs des modèles, les problèmes d'allocation des ressources souhaitées par Compute Canada et la gestion du transfert des données, car le dataset était volumineux et la communication avec les serveurs canadiens n'était pas toujours optimale.

Un autre défi important, surtout au début, était d'apprendre à structurer mes 12 semaines et à les planifier correctement. En recherche, si l'on n'a pas de sous-objectifs clairs et une méthodologie solide, on peut très vite perdre du temps, d'autant plus que ce type de stage demande énormément d'autonomie : savoir identifier les articles fiables, comprendre comment lire un article scientifique et en extraire l'essentiel, organiser son travail... Tout cela demande une vraie discipline. À côté de ces aspects techniques, il y a aussi eu les échanges culturels enrichissants, la réflexion sur les idées, les résultats et le regard critique à porter dessus.

En somme, ce stage m'a permis de gagner en compétences, en autonomie et en confiance. Malgré les défis rencontrés, j'en ressors avec une meilleure compréhension du travail de recherche et une réelle motivation pour continuer à progresser dans le domaine de l'IA et de la vision par ordinateur.

Références

[1] H. Duan, Y. Zhao, Y. Xiong, W. Liu et D. Lin, "Omni-sourced webly-supervised learning for video recognition," in *Computer Vision—ECCV 2020 : 16th European Conference*, H. Duan, éd., Glasgow : Springer, 2020, p. 670-688.

[2] S. Gowda, M. Rohrbach et L. Sevilla-Lara, "Smart frame selection for action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020. doi : 10.1609/aaai.v35i2.16235. adresse : https://doi.org/10.1609/aaai.v35i2.16235.

[3] W. Wu, X. Wang, H. Luo, J. Wang, Y. Yang et W. Ouyang, "Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, p. 6620-6630.

[4] L. Wang et al., "Videomae v2 : scaling video masked autoencoders with dual masking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, p. 14 549-14 560.

[5] S. Srivastava et G. Sharma, "Omnivec2—a novel transformer based network for large scale multimodal and multitask learning," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, p. 27 402-27 414.

[6] H. Lu, H. Jian, R. Poppe et al., "Enhancing video transformers for action understanding with VLM-aided training," *arXiv*, 2024. adresse : https://arxiv.org/abs/2403.16128.

[7] A. Rahnama, A. Esfahani et A. Mansouri, "Adaptive Frame Selection in Two Dimensional Convolutional Neural Network Action Recognition," 2022. doi : 10.1109/ICSPIS56952.2022.10044032. adresse : https://www.researchgate.net/publication/368726751_Adaptive_Frame_Selection_In_Two_Dimensional_Convolutional_Neural_Network_Action_Recognition.