Analyse multimodale de situations conflictuelles en contexte véhicule

Quentin Portes

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *30/06/2022* **par :**

QUENTIN PORTES

Analyse multimodale de situations conflictuelles en contexte véhicule

---

JURY

BRUNO ÉMILE	Maître de conférences	Rapporteur
GEORGES QUÉNOT	Maître de conférences	Examinateur
PASCALE SÉBILLO	Professeure des Universités	Rapporteure
FRÉDÉRIC LERASLE	Professeur des Universités	Directeur de thèse
JULIEN PINQUIER	Maître de conférences	Directeur de thèse

École doctorale et spécialité :
EDSYS : Informatique 4200018

Unité de Recherche :
Laboratoire d'analyse et d'architecture des systèmes (RAP)
Institut de Recherche en Informatique de Toulouse (SAMOVA)

Directeur(s) de Thèse :
Frédéric LERASLE et *Julien PINQUIER*

Rapporteurs :
Pascale SÉBILLO et *Bruno ÉMILE*

Résumé — Dans cette thèse nous étudions les interactions humaines afin d'identifier des situations conflictuelles dans l'habitacle d'un véhicule. Les humains utilisent le plus communément la vue et l'ouïe pour analyser les interactions. Cette tâche paraît anodine, mais reste complexe pour un modèle d'intelligence artificielle. Celui-ci doit capturer les informations vidéo et audio et les analyser pour prédire une situation conflictuelle. Notre approche est nouvelle en regard des travaux réalisés jusque-là sur ce sujet puisque les passagers sont contraints dans leurs mouvements dans l'habitacle et que la puissance de calcul embarqué pour cette tâche est limitée. Aucun travail, à notre connaissance, ne s'est intéressé à l'analyse des interactions humaines pour la détection de situations conflictuelles dans ce contexte et avec ces contraintes. Nos investigations s'appuient tout d'abord sur un corpus public (intitulé MOSI) d'analyse de sentiment pour se comparer à la littérature. Nous implémentons un modèle capable d'ingérer des données vidéo, audio et texte (transcription de l'audio) pour les fusionner et prendre une décision. Dans notre contexte applicatif, nous enregistrons par la suite un jeu de données multimodal d'interactions humaines simulant des situations plus ou moins conflictuelles dans un habitacle de véhicule. Cette base de données est exploitée afin d'implémenter des modèles de classification de bout-en-bout et paramétrique. Les résultats obtenus sont cohérents avec la littérature sur l'impact de chaque modalité sur les performances du système. Ainsi, le texte est respectivement plus informatif que l'audio et que la vidéo. Les différentes approches de fusion implémentées montrent des bénéfices notables sur les performances de classification mono-modalité. Le développement de nos systèmes est mené avec l'objectif de les intégrer sur une plateforme embarquée pour véhicule. Pour ce faire, les coûts en calculs de nos modèles sont considérés.

Mots clés : Analyse d'interaction, Multimodalité, Fusion

Abstract — In this thesis we study human interactions in order to identify conflictual situations in the vehicle cabin. Humans most commonly use sight and hearing to analyze interactions. This task seems trivial, but is complex for an artificial intelligence model. It must capture video and audio information and analyze it to make a prediction. Our approach is new compared to previous research on this topic since passengers are constrained in their movements in the cabin and the computing power on board for this task is limited. To our knowledge, no work has been done on the analysis of human interactions for conflictual situation detection in this context and with these constraints. Our investigations are first based on a public corpus of sentiment analysis to compare with the literature. We implement a model capable of ingesting video, audio and text data (audio transcription) to merge them and make a decision. In our application context, we then record a multimodal dataset of human interactions simulating more or less conflictual situations in a vehicle cockpit. This database is exploited to implement end-to-end and parametric classification models. The results obtained are consistent with the literature on the impact of each modality on the system performance. Thus, the text is respectively more informative than audio and video. The different fusion approaches implemented show significant benefits on the performance of single-modality classification. The developement step of all our systems are guided with the objective to integrate them on an on-board vehicul platform. For those purposes, the on-board capabilities of our models are measured and compared.

Keywords : Interactions analysis, Multimodality, Fusion

A ma famille et mes amis,

Remerciements

Tout d'abord, je souhaite adresser mes remerciements à Pascale Sébillo et Bruno Émile pour avoir accepté le rôle de rapporteur de thèse ainsi que pour leurs remarques particulièrement constructives concernant le manuscrit.

Je tiens à remercier chaleureusement et sincèrement mes deux directeurs de thèse, Frédéric Lerasle et Julien Pinquier. Un duo complémentaire qui par leurs discussions, conseils, relectures a participé à la réussite de cette thèse. Ils m'ont permis de me dépasser et de toujours viser le meilleur dans mes travaux.

Je remercie également José Mendes Carvalho pour m'avoir fait confiance à la suite de mon stage d'ingénieur en me proposant cette thèse industrielle. Nos échanges sur la compréhension des enjeux techniques et humains du monde de l'entreprise m'ont fait grandir. Je le remercie également pour son soutien tout au long de ce doctorat, pendant les moments difficiles, de solitude, induits par la Covid-19.

Je souhaite également remercier tous les participants ayant joué un rôle dans l'enregistrement du corpus, élément essentiel de cette thèse. Ils représentent la base, sans laquelle cette thèse n'aurait pu voir le jour. Je remercie plus particulièrement Walid Zaghdoud qui m'a aidé à mettre en place la plateforme matérielle permettant d'enregistrer le corpus.

Cette épreuve intellectuelle n'aurait jamais été possible sans les amis qui m'entourent et qui ont pu comprendre la difficulté de ce challenge. Je les remercie aussi de m'avoir permis de garder une vie sociale. Merci à Léo, Anoulak, Clara, Ludo (et Vale), Vivien et Vicente.

Je remercie finalement toute ma famille qui continue aujourd'hui de s'agrandir (croquette ?). Petit dernier d'une fratrie de trois, j'ai su m'inspirer, prendre exemple et marcher sur les traces de Audrey, ma grande sœur et Fabien, mon grand frère. La réussite dans mon parcours scolaire est sans nul doute inhérente à leur aide inestimable. Je te remercie également Léna pour ton soutien. Papa et Maman, même si trouver ma voie n'a pas été évident au début, vous avez toujours su me motiver, et me faire persévérer. Je vous témoigne par ces remerciements ma plus grande gratitude.

Pour terminer, je remercie ma compagne Blandine qui est entrée dans ma vie au début de la thèse. Tu m'as accompagné, soutenu et supporté tout au long de ce périple et cela malgré les difficultés. Tes encouragements, ta confiance et ton soutien moral ont permis la réussite de cette thèse. Merci pour tout, je vais maintenant t'accompagner pour la tienne et nous pourrons ensuite profiter de la vie.

Je remercie également toute ta famille pour son soutien ainsi que le petit havre de paix que Montchenon nous offrait.

Liste des publications

Conférences internationales

1. Q. Portes, J. Pinquier, F. Lerasle and J. M. Carvalho, Multimodal human interaction analysis in vehicle cockpit, IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis (US), September 2021.

2. Q. Portes, J. M. Carvalho, J. Pinquier and F. Lerasle, Multimodal Neural Network for Sentiment Analysis in Embedded Systems. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), Valetta (Malta), February 2020.

3. Q. Portes, J. M. Carvalho, J. Pinquier and F. Lerasle, Comparison of two approaches for human tense situation analysis in car cabin, The Thirteenth International Conference on Advances in Multimedia, (MMEDIA), Porto (Portugal) April 2021.

Conférences nationales

1. Q. Portes, J. Pinquier, F. Lerasle and J. M. Carvalho, Analyse multimodale d'interaction humaine dans le cockpit d'un véhicule. 18ème journées francophones des jeunes chercheurs en vision par ordinateur (ORASIS), Saint Ferréol, Septembre 2021.

2. (Accepté) Q. Portes, J. Pinquier, F. Lerasle and J. M. Carvalho, Analyse de situations conflictuelles dans l'habitacle d'un véhicule par apprentissage profond. Conférence francophone Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), Vannes, Juillet 2022.

Table des matières

Table des sigles et acronymes

TAL *Traitement Automatique du Langage*
RAM *Random Access Memory*
ROM *Read-Only Memory*
C3D *Convolutional 3D*
CNN *Convolutional Neural Network*
RNN *Recurrent Neural Network*
GRU *Gated Recurent Unit*
LSTM *Long Short-Terme Memory*
FC *Fully connected*
DST *Dempster Shafer Theory*
LLD *Low Level Descriptor*
MLP *Multi Layers Perceptron*
IHM *Interface Homme Machine*
ECU *Unité de Commande Electronique*
IVI *In-Vehicle Infotainment*
CPU *Central Processing Unit*
GPU *Graphics Processing Unit*
NPU *Network Processing Unit*
DSP *Digital Signal Processsor*
HAL *Hardware Abstraction Layer*
DST *Dempster Shafer Theory*
HOG *Histogram of Oriented Gradients*
HOF *Histogram of Optical Flow*
DFT *Discrete Fourier transform*

Table des figures

Liste des tableaux

Introduction

Contexte industriel et scientifique

Le sujet de cette thèse industrielle (CIFRE) est proposé par l'équipe DEA-LIM de Renault Software Labs de Toulouse. Cette entité du groupe Renault est spécialisée dans le développement logiciel lié au véhicule. Plus précisément, la DEA-LIM travaille sur tous les éléments associés au système d'infotainment du véhicule. Le sujet de la thèse nous amenant à analyser différents types de données, elle est de ce fait académiquement encadrée par deux laboratoires. Le premier est le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)-CNRS de Toulouse avec l'équipe « Robotique, Action et Perception » (RAP) qui nous apporte une expertise sur l'analyse image des vidéos. Le second laboratoire est l'Institut de Recherche en Informatique de Toulouse (IRIT) à Toulouse avec l'équipe « Structuration, Analyse et Modélisation de documents Vidéo et Audio » (SAMoVA) qui nous apporte une expertise sur l'analyse des données audio et texte.

Motivations

Situation conflictuelle et agression dans un véhicule

L'industrie automobile est aujourd'hui confrontée à des bouleversements majeurs. L'émergence de véhicules autonomes, partagés et connectés ouvre la porte à de nouveaux usages où l'expérience de l'utilisateur est modifiée, passant de conducteur à voyageur. Par ailleurs, de nouvelles contraintes voient le jour : contraintes environnementales, pénurie de composants, etc. Elles remettent en cause certains formats de voyage ou de transport comme le véhicule individuel par exemple. L'objectif est clairement de favoriser l'émergence de nouveaux modes de transport moins polluants, mais aussi moins coûteux pour les usagers. Cette transformation de l'industrie automobile et des usages entraîne de nouveaux besoins dont celui de la sécurité de l'habitacle. Si auparavant la probabilité de se retrouver avec un inconnu dans un véhicule (cas d'un auto-stoppeur) était extrêmement faible, les nouveaux types de mobilité, liés aux véhicules partagés notamment, entraînent une forte hausse dans l'apparition de ces situations (robot-taxi, partage de voyages, etc.). Or, un espace partagé aussi restreint que l'habitacle d'un véhicule peut entraîner des dérives comportementales (sentiment d'insécurité) posant par la suite des problèmes d'adoption et d'utilisation de ces nouveaux services de mobilité. Nous pouvons citer l'exemple récent de la société Uber qui a commencé à recenser les agressions survenues lors de l'utilisation de leur service de transport. De nombreux témoignages sont aujourd'hui disponibles pour dénoncer ces phénomènes. Le rapport interne [1] de

1. `https://www.uber.com/us/en/about/reports/us-safety-report/`

sécurité de la société a ainsi comptabilisé 5981 agressions sexuelles rapportées par des utilisateurs ou des conducteurs sur le territoire américain entre 2017 et 2018. Ce rapport dénombre également 19 homicides sur la même période. Ces problèmes de sécurité sont constatés dans de nombreux pays. Ils sont pris en considération par Uber en installant des systèmes de reconnaissance faciale dans les véhicules pour l'identification du conducteur. Cette problématique est aussi transposable dans une moindre mesure aux transports en commun. En 2019, 40,9% [2] des usagers d'Île-de-France ont ressenti au moins une fois de la peur en empruntant les transports en commun. Plus précisément, selon le rapport [3] de 2018, 136 540 personnes ont été victimes de vols et violences (physiques ou sexuelles) dans les transports en commun. Mettre en place des solutions de prévention/détection est bien plus complexe dans un tel environnement. Les caméras présentes actuellement n'ont pas vocation à être des lanceurs d'alerte de situation dangereuse. Elles servent uniquement à retrouver et identifier les agresseurs, mais ne permettent pas, hélas, d'anticiper pour éviter l'agression.

Les constructeurs automobiles comme Renault ont l'ambition d'accompagner la socialisation du véhicule en proposant des solutions d'autopartage de véhicule tel que le service Mobilize [4]. Afin d'offrir un environnement sécurisé à l'intérieur de ses véhicules, Renault souhaite explorer des techniques permettant de prévenir les situations de conflits. Des systèmes proactifs et robustes capables de détecter les situations conflictuelles dans l'habitacle du véhicule pour éviter les agressions deviennent alors un enjeu majeur pour rendre ces services attractifs.

Multimodalité et contexte véhicule

La mise en place d'un système proactif pour la prévention de situations conflictuelles passe par l'analyse des interactions entre les passagers. Les applications se rapprochant de cette problématique dans la littérature ou dans le milieu industriel sont l'analyse de sentiments ou d'émotions. Elles sont réalisées à l'aide de données issues de capteurs tels que des caméras et microphones. Les données vidéo sont utilisées pour l'analyse du visage afin de détecter les émotions et les déformations de la bouche. Les données audio quant à elles permettent d'analyser les intonations, la hauteur de la voix, etc. Finalement, le texte est généré par la transcription de l'audio afin de mesurer les sentiments issus des dialogues. Afin de réaliser ce type d'analyses dans un contexte véhicule qui est un environnement contraint par la puissance de calcul limitée et perturbé par les vibrations de la route, les bruits extérieurs, les changements de luminosité, etc., une solution intéressante est de fusionner les différentes sources de signaux (ou modalités) audio, vidéo et texte. Cette fusion de données permettrait d'augmenter les performances globales et la robustesse du système en profitant des forces

2. https://www.institutparisregion.fr/nos-travaux/publications/sentiment-dinsecurite-dans-les-transports-collectifs-franciliens/
3. https://mobile.interieur.gouv.fr/Media/SSMSI/Files/Les-vols-et-violences-dans-les-reseaux-de-transports-en-commun-en-2018-Interstats-Analyse-N-23
4. https://www.renaultgroup.com/groupe/nos-marques/mobilize/

de chacune des modalités. Il serait alors possible de concevoir des systèmes capables d'ingérer simultanément au minimum deux types de données différents. Les signaux vidéo, audio et texte sont déjà utilisés de manière indépendante pour l'analyse de sentiment et d'émotion. Des modèles les fusionnant sont apparus ces dernières années [NKK+11] sur des corpus publics, mais *a priori* aucun dans notre contexte applicatif *i.e.* véhicule partagé. Ce contexte particulier met en exergue de nombreux défis scientifiques et industriels auxquels nous souhaitons contribuer, ceci afin de pouvoir proposer des solutions d'analyse de situations conflictuelles entre passagers.

Challenges sous-jacents

Nous présentons ici les défis induits par l'analyse de discussions conflictuelles dans un habitacle de véhicule.

Du besoin d'un corpus spécifique - Les jeux de données disponibles dans le domaine du public semblent pertinents pour des objectifs de recherche [ZZPM16, BBL+08]. Toutefois, lorsque nous abordons des applications plus spécifiques et/ou industrielles afin de réaliser des preuves de concept, ces dernières ne sont que trop limitées. Le contexte étudié ici est particulier, car il nécessite des interactions entre deux individus, un habitacle de voiture et des flux de données audio-vidéo et texte synchronisé. Il n'existe *a priori* aucun corpus de ce type dans le domaine public.

Caractère temporel de l'analyse - L'évolution d'une interaction humaine et son analyse sont fortement corrélées au caractère temporel. Les humains ne changent pas de comportement à des fréquences élevées, ce qui implique l'obligation d'un suivi des interactions sur de longues périodes. Nous sommes ici en présence d'une analyse de l'ordre de la dizaine ou trentaine de secondes et non pas de la seconde. Cet aspect est important à prendre en considération pour la conception des réseaux de neurones. Des temps d'analyse trop courts ne permettent pas de capturer assez de contexte de la situation en cours. À l'inverse, des temps d'analyse trop longs ne sont pas acceptables du fait de la latence trop importante pour prévenir en cas de situation conflictuelle avérée.

Fusion de données hétérogènes et contrainte « embarquée » - Les modalités audio, vidéo et texte (signaux hétérogènes) combinées permettent d'améliorer les performances de prédiction [PCH+17, AYV19]. Cependant, la fusion n'est pas triviale, il est primordial d'obtenir avant tout des modalités parfaitement synchronisées puis d'extraire des caractéristiques pour chacune d'entre elles. Du fait des trois modalités auxquelles nous nous intéressons dans ce travail de recherche, les extractions peuvent être coûteuses en ressources de calcul. La multimodalité a alors tendance à générer des systèmes assez lourds, ce qui est paradoxal avec notre contexte embarqué où les ressources de calcul sont limitées. Nous devrons prendre en considération cette contrainte en optimisant

notre chaîne de traitement des données.

Forts de ces constats, nous détaillons ci-après les contraintes industrielles et le cahier des charges associé à ces travaux.

Cahier des charges industriel de cette thèse

Nous définissons ci-après le cahier des charges de la thèse, qui représente nos défis industriels. La finalité étant de réaliser une preuve de concept permettant de montrer que l'analyse de situations conflictuelles dans le contexte du véhicule est possible en prenant en considération, autant que possible, les performances embarquées. Les deux sections suivantes détaillent les trois contraintes définies par Renault.

Contrainte #1 : analyse proactive

Nous devons développer un système capable de détecter des comportements menant à une agression. L'agression physique en elle-même, lorsqu'elle advient, n'est pas au centre de nos recherches. Nous souhaitons, en effet, travailler sur les situations en amont, permettant de la prévenir. En définitive, le but est d'identifier des tensions et non des signaux de violence. L'enjeu est alors de détecter grâce à des données audio, texte et vidéo les situations menant potentiellement à l'agression. La contrainte est ici le fait que les passagers sont dans une configuration spatiale spécifique imposée par l'habitacle. Les passagers peuvent alors ne pas être visibles à la caméra et/ou le microphone peut ne pas capturer correctement le flux audio. Le domaine public ne disposant pas de jeux de données permettant l'étude de telles interactions, nous devons enregistrer notre propre corpus *in situ*. A minima, deux passagers doivent être présents dans l'habitacle et interagir en français.

Le protocole devra permettre d'enregistrer trois scénarios :

(i) curieux, avec des passagers qui discutent cordialement,

(ii) refus argumenté, avec le passager qui refuse cordialement la proposition du conducteur,

(iii) refus catégorique, quand le passager refuse la proposition du conducteur.

Contrainte #2 : performances vs ressources embarquées

Deux contraintes sont ensuite identifiées et liées à la notion de coût et d'embarquabilité. Premièrement, l'analyse étant multimodale (vidéo, audio et texte), nous devons déterminer la valeur ajoutée d'une modalité par rapport à une autre. En effet, du point de vue du coût en ressources, il s'agit de déterminer si une des modalités est plus consommatrice en ressources de calcul que les autres. Ainsi, nous étudions la valeur ajoutée de chaque capteur en analysant son ratio coûts/bénéfices au sein d'une architecture multimodale. Dans un second temps, nous devons implémenter deux types d'architectures afin de réduire

les coûts liés aux envois continus de données vidéo et audio vers une plateforme dite « cloud ». La première architecture est destinée à être embarquée dans le véhicule et la seconde est donc orientée « cloud computing ». En définitive, le modèle embarqué permet de faire une première détection qui permettra de générer un envoi de données vidéo et audio. Celui-ci déclenchera alors l'inférence d'un modèle dit « cloud ». L'architecture orientée embarquée doit ainsi avoir une compacité élevée, fonctionner avec des ressources en calcul limitées (latence et usage mémoire faible), cela en parallèle d'un environnement déjà existant (système de navigation, divertissement, système de recul, etc.).

En prenant en considération l'ensemble des défis scientifiques et industriels identifiés attenant à notre contexte particulier, nous allons dans cette thèse investiguer la problématique de la classification multimodale d'interactions humaines conflictuelles dans un contexte véhicule. Nous prenons également en considération les performances embarquées des modèles développés.

Plan du mémoire

Le chapitre 1 présente un état de l'art sur les modalités audio, vidéo et texte, ainsi que leur fusion, dans une problématique d'analyse des émotions et sentiments.

Le chapitre 2 vise à implémenter un premier système multimodal et comparer à la littérature les résultats de deux techniques de fusion sur un corpus public. Nous prenons en considération les coûts en ressources de calcul. Ces travaux préliminaires permettent de nous familiariser avec les trois modalités et la notion de compacité des modèles.

Le chapitre 3 développe la méthodologie associée à l'enregistrement de notre corpus multimodal dans un contexte véhicule.

Le chapitre 4 propose et compare les performances de deux modèles et de trois stratégies de fusions afin de catégoriser nos trois scénarios pré-cités : curieux, refus argumenté, refus catégorique.

Le chapitre 5 présente l'architecture logicielle Android à disposition dans les véhicules. Il résume les processus que nous avons mis en place dans nos travaux pour augmenter la compacité de notre chaîne de traitement des données. Un état de l'art des théories permettant de réduire la complexité et les coûts en ressources des modèles est alors effectué. Une étude quantitative compare les performances embarquées de nos deux systèmes.

Enfin, nous concluons sur l'ensemble des travaux qui ont été réalisés et les perspectives associées à l'issue de cette thèse.

Chapitre 1

État de l'art

Sommaire

1.1 Introduction

1.1.1 Enjeux et applications associées

Les dialogues, les interactions, les émotions et l'analyse des sentiments sont les principaux éléments pour comprendre les interactions humaines. Ces éléments sont considérablement étudiés dans les domaines de la sociologie et de la psychologie [CKT+16, TC08, LeD00, Lit16]. D'un point de vue industriel, la capture de ces informations, quel que soit le contexte d'application final, peut résoudre de nombreux problèmes tels que le filtrage des contenus sensibles sur les réseaux sociaux ou l'amélioration de la compréhension des interfaces homme-machine, etc. Les communautés de traitement du texte, de la vidéo et de l'audio ont étudié les interactions entre individus, les émotions et les sentiments dans leurs domaines respectifs (analyse unimodale). Récemment ces domaines se sont rejoints pour permettre des systèmes multimodaux avec pour objectif de tirer parti des forces de chacune des modalités dans le but d'améliorer à la fois les performances de prédiction et la robustesse. Dans un environnement maîtrisé, comme dans la majorité des corpus publics de la littérature, les performances

des analyses de sentiments/émotions sont convaincantes. *A contrario*, les environnements proches des applications du monde réel, qui sont par définition plus contraints, sont peu étudiés. Cela est principalement lié au manque de corpus public et aux topologies des modèles trop complexes qui ne peuvent être déployées sur une plateforme aux ressources limitées. Nous nous focalisons ici sur une analyse multimodale vidéo, audio et texte en prenant en considération les coûts en ressources de calculs de toutes les composantes de la chaîne de traitement.

1.1.2 Définitions

Nous définissons ci-après les notions « émotions » et « sentiments », utilisées par la suite au travers des différents chapitres pour adresser la problématique de l'analyse multimodale d'interaction humaine.

En sociologie, l'interaction sociale est définie comme une séquence dynamique d'actions sociales entre des individus (ou des groupes) qui modifient leurs actions et réactions en fonction des actions des partenaires d'interaction [Lit16]. Les interactions peuvent être verbales ou non verbales [BDNM12] (gestes, regards, sourires, attitudes, postures, etc.). Le langage humain inclut aussi des formes paraverbales [DTSD15] (intonation ou inflexion de la voix, débit et volume du discours, etc.).

Les interactions se catégorisent [LS16, BH96, Lin00] ainsi :

1. positives : coopération, participation, adaptation, intégration, émulation et compétition, etc.
2. négatives : conflit, lutte, rivalité, ségrégation, discrimination, insulte, etc.
3. ambivalentes : compétition, concurrence.

Lorsque nous interagissons, nous sommes amenés à exprimer des émotions et des sentiments [CS10]. Définir ces deux notions est peu intuitif, il en découle une multitude de définitions possibles. Les termes « sentiment » et « émotion » sont dans la majorité des cas utilisés de manière interchangeable, mais sont en réalité très différents.

L'émotion est liée à un état psychologique complexe tel que la peur, la colère ou le bonheur. Le sentiment est une attitude mentale produite par les émotions. Les émotions sont brutes, tandis que les sentiments sont organisés [Cab02].

Dans un contexte informatique, les applications d'analyse d'émotions et de sentiments sont définies comme suit :

— **Émotions** [HMKM17] - Le but est de définir dans quelle catégorie de la roue des émotions (voir roue de Plutchnick [1] en figure 1.1) se situe une donnée. Le champ de résolution des émotions étant très détaillé, la tâche est souvent ramenée aux émotions fondamentales : la peur, la colère, la joie, la tristesse.

1. `https://fr.wikipedia.org/wiki/Robert_Plutchik`

— **Sentiments** [Hov15] - Le but est de déterminer si les données, souvent textuelles, ont tendance à être positives, neutres ou négatives.

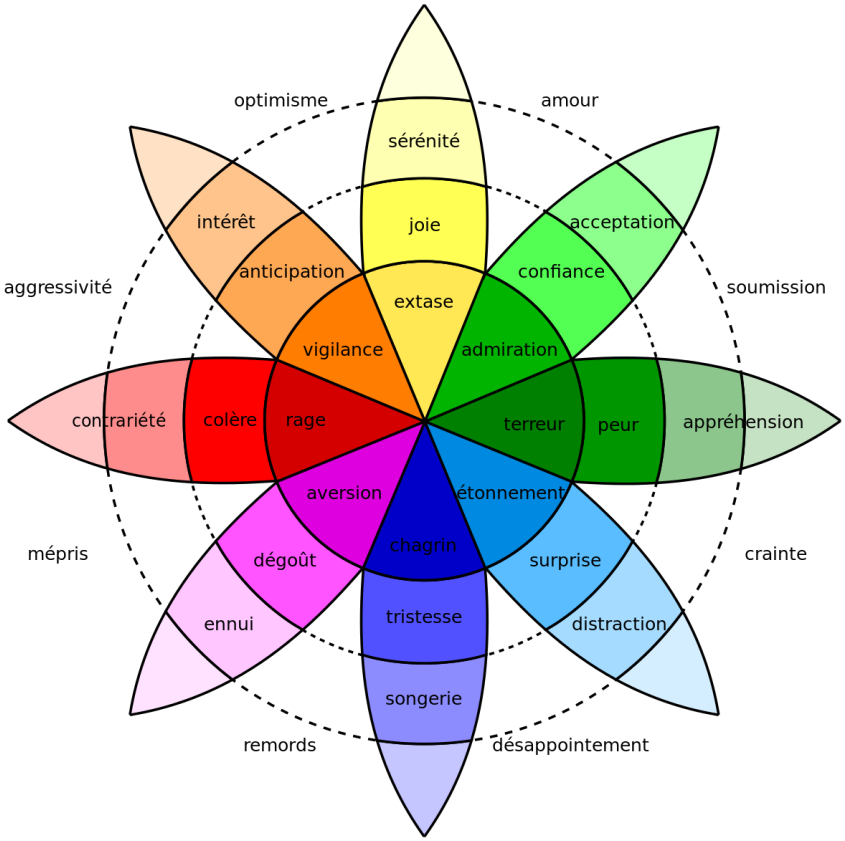

FIGURE 1.1 – Roue de Plutchnick représentant les différents états émotionnels.

Nos investigations se focalisent sur les relations multimodales entre les locuteurs. Nous étudions les interactions dites positives et négatives par des processus similaires à l'analyse de sentiment et d'émotion. L'objectif est de qualifier une interaction et déterminer si elle devient conflictuelle. Nous détaillons ci-après les architectures de l'état de l'art utilisées dans le domaine de l'apprentissage machine/profond pour analyser les émotions/sentiments et les interactions.

1.2 Analyse unimodale et temporelle

Préalablement, nous allons discuter de la temporalité dans les réseaux de neurones et présenter la théorie associée aux deux architectures de l'état de l'art les plus couramment utilisées pour cette tâche. Dans un second temps, nous détaillerons les outils d'analyses disponibles pour chacune des modalités (audio, vidéo et texte).

Notre application est dépendante du contexte donc de la cohérence temporelle des signaux. Capturer chronologiquement le contexte global et local est nécessaire pour comprendre l'évolution de scènes dans des vidéos. Les définitions des architectures mentionnées dans ce mémoire sont explicitées dans cette section. Il est difficile d'intégrer la notion de temporalité dans les architectures de réseaux de neurones. Le plus souvent, les *Recurrent Neural Network* (RNN) sont utilisés en amont ou aval d'un autre modèle plus performant pour capturer des caractéristiques locales (CNN). Les réseaux à convolution 3D représentent une alternative.

1.2.1 Réseaux de neurones à convolution 3D

Les réseaux de neurones convolutifs 2D sont majoritairement utilisés pour la classification d'images [DLLT21], la reconnaissance d'objets [RF17], l'extraction de caractéristiques dans des images [KSH12], etc. La couche de convolution utilise des filtres qui effectuent des opérations de convolution pendant qu'elle analyse l'entrée. Cette opération peut être appliquée sur des entrées à 1, 2 et 3 dimensions, seuls les axes sur lesquels se déplace le filtre et les dimensions du filtre de convolution changent. La convolution à 1 et 2 dimensions est aujourd'hui très populaire dans le milieu industriel et scientifique. Le besoin récent d'analyser des données à 3 dimensions tel que des vidéos, des images issues d'IRM, etc. ont fait émerger la convolution tridimensionnelle [TBF+15]. Le filtre de convolution 3D permet d'apprendre des descripteurs spatio-temporels sur un horizon de temps défini et ainsi prendre en compte le voisinage temporel d'un pixel en plus du voisinage spatial. Cette convolution appliquée à des données 3D génère un volume et non une image, préservant l'information temporelle, voir figure 1.2.

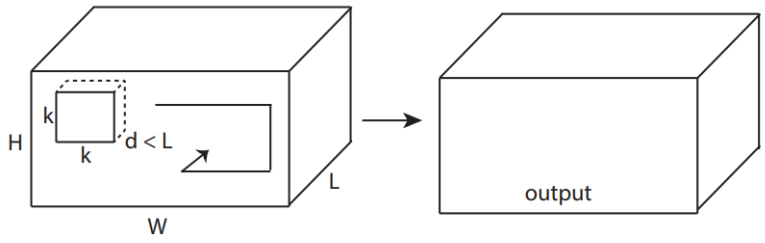

FIGURE 1.2 – Illustration de la convolution 3D. H (resp. W) représente la hauteur (resp. largeur) de l'image. L représente la profondeur temporelle (appelé clip vidéo). Le filtre est de hauteur et largeur k et de profondeur d.

La première architecture utilisant la *Convolutional 3D* (C3D) a été proposée par [TBF+15] : elle est définie dans le tableau 1.1. Elle est utilisée pour la reconnaissance d'action dans des vidéos sur un horizon temporel de 16 trames. Elle est composée de quatre couches élémentaires, la couche de « Convolution 3D » avec des filtres de tailles $[3 \times 3 \times 3]$ et un pas de un, la couche de « Pooling » de tailles $[2 \times 2 \times 2]$ excepté pour la première (pool1) $[1 \times 2 \times 2]$. La couche « Dense » entièrement connectée, puis une fonction « Softmax » qui permettent d'obtenir le score de confiance pour chacune des classes.

TABLE 1.1 – Modèle C3D. nf dénote le nombre de filtres, p la profondeur, h la hauteur, w la largeur du volume de sortie.

Couches	(nf, p, h, w)
Entrée initiale	(3, 16, 112, 112)
Convolution3D (conv1)	(64, 16, 112, 112)
MaxPooling3D (pool1)	(64, 16, 56, 56)
Convolution3D (conv2)	(128, 16, 56, 56)
MaxPooling3D (pool2)	(128, 8, 28, 28)
Convolution3D (conv3a)	(256, 8, 28, 28)
Convolution3D (conv3b)	(256, 8, 28, 28)
MaxPooling3D (pool3)	(256, 4, 14, 14)
Convolution3D (conv4a)	(512, 4, 14, 14)
Convolution3D (conv4b)	(512, 4, 14, 14)
MaxPooling3D (pool4)	(512, 2, 7, 7)
Convolution3D (conv5a)	(512, 2, 7, 7)
Convolution3D (conv5b)	(512, 2, 7, 7)
ZeroPadding3D (zeropadding3d)	(512, 2, 9, 9)
MaxPooling3D (pool5)	(512, 1, 4, 4)
Flatten (flatten)	(8192)
Dense (fc6)	(4096, 4096)
Dropout (dropout)	(4096, 4096)
Dense (fc7)	(4096, 4096)
Dropout (dropout)	(4096)
Dense (fc8)	(4096, nombre de classes)
Softmax	

Couche de pooling - Elle sous-échantillonne la donnée d'entrée par l'utilisation d'une fonction statistique pour garder l'information la plus importante. En réduisant la taille spatiale de la donnée d'entrée, le nombre de paramètres est réduit, diminuant le risque de sur-apprentissage. Le processus consiste à déplacer une fenêtre sur la matrice d'entrée et à appliquer la fonction maximum

ou moyenne. Les dimensions de la matrice de sortie sont données par :

$$\begin{aligned} W_{out} &= (W_{in} - K)/S + 1 \\ H_{out} &= (H_{in} - K)/S + 1 \\ \text{Depth}_{out} &= \text{Depth}_{in} \end{aligned} \tag{1.1}$$

Les hyperparamètres de cette couche sont : la taille de la fenêtre d'analyse K et le pas de déplacement de cette dernière S.

Couche dense ou entièrement connectée (FC) - Elle est utilisée pour connecter tous les neurones d'une couche précédente à ceux de la suivante.

Fonction softmax - Cette fonction transforme les K valeurs réelles d'entrée en K valeurs comprises entre 0 et 1, afin qu'elles puissent être interprétées comme des probabilités. La fonction softmax est utilisée lorsque les classes sont mutuellement exclusives :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1.2}$$

avec :

- $\vec{z}$, le vecteur d'entrée de la fonction softmax, composée de (z_0,..., z_K),
- e^{z_i}, la fonction exponentielle standard appliquée à chaque élément du vecteur d'entrée. Cela donne une valeur positive au-dessus de 0, qui est très petite si l'entrée est négative, et très grande si l'entrée est grande.
- le dénominateur $\sum_{j=1}^{K} e^{z_j}$, est le terme de normalisation. Il garantit que la somme de toutes les valeurs de sortie de la fonction sera égale à 1 et que chacune d'entre elles se situera dans l'intervalle [0, 1], constituant ainsi une distribution de probabilité valide.
- K est le nombre de classes.

De nombreux réseaux neuronaux multicouches se terminent par une avant-dernière couche qui produit des scores à valeur réelle avec lesquels il peut être difficile de travailler. Dans ce cas, la fonction softmax permet de les convertir en une distribution de probabilités normalisée, qui peut être affichée à un utilisateur ou utilisée comme entrée pour d'autres systèmes. Il est alors courant d'ajouter une fonction softmax comme couche finale d'un réseau neuronal.

1.2.2 Réseaux de neurones récurrents

Les RNN sont un type de réseau neuronal artificiel qui ingèrent des données séquentielles ou des données de séries temporelles. Ces algorithmes d'apprentissage profond sont couramment utilisés pour des problèmes temporels, tels que la traduction de langues [SS18], le traitement du langage naturel [XZ20], la reconnaissance vocale [SNL19], le sous-titrage d'images [KZS18], l'analyse audio [KHAS21], etc. Ils se distinguent par leur *mémoire*, car ils utilisent les informations des entrées précédentes pour influencer l'entrée et la sortie actuelles. La

sortie des RNN dépend des éléments antérieurs de la séquence. Les RNN sont définis en plusieurs types en fonction de la tâche à accomplir, voir figure 1.3.

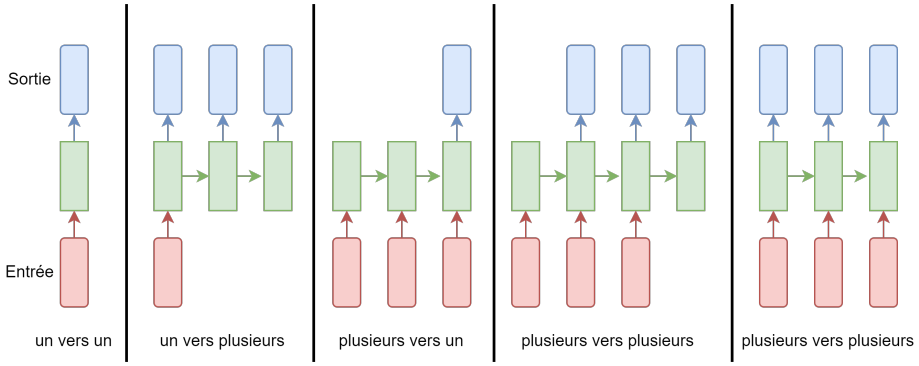

FIGURE 1.3 – Les différents types de RNN.

Chaque rectangle est une cellule élémentaire d'un RNN et les flèches représentent des fonctions. Les vecteurs d'entrée sont en rouge, les vecteurs de sortie sont en bleu et les vecteurs verts contiennent l'état du RNN. De gauche à droite :

1. réseau de neurones classique, l'entrée et la sortie sont de taille fixe, par exemple pour de la classification d'images [GDLG17].
2. sortie d'une séquence, par exemple pour la génération d'une légende (phrase) à partir d'une image [KZS18],
3. entrée de séquence, par exemple pour l'analyse des sentiments, où une phrase donnée est classée comme exprimant un sentiment positif ou négatif [Hov15],
4. séquence en entrée et séquence en sortie, par exemple pour la traduction automatique où un RNN ingère une phrase en anglais et produit une phrase en français [SS18],
5. entrée et sortie de séquences synchronisées, par exemple pour l'étiquetage morpho-syntaxique de texte qui consiste à déterminer pour chaque mot les informations grammaticales qui lui sont associées [BKY18].

Les RNN classiques ne sont pas utilisés à cause des phénomènes de disparition et d'explosion du gradient. Ce phénomène apparaît, car il est difficile de capturer les dépendances à long terme à cause du gradient multiplicatif qui peut être exponentiellement décroissant/croissant.

Deux types de cellules récurrentes sont alors utilisés pour compenser ce problème et former des réseaux récurrents : les cellules *Gated Recurent Unit* (GRU) et *Long Short-Terme Memory* (LSTM). La particularité de ces cellules est de pouvoir mettre à jour, supprimer l'information qui circule dans la cellule grâce à des portes (*gates*).

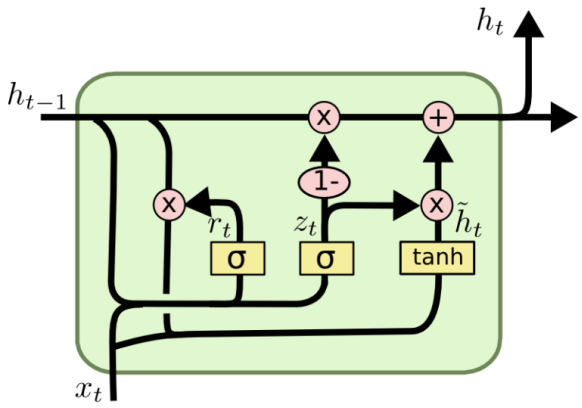

FIGURE 1.4 – Représentation visuelle d'une cellule GRU.

Cellule GRU - Elle est représentée sur la figure 1.4.

La cellule GRU se compose de deux portes *i.e.* la porte de mise à jour (z_t) et celle de remise à zéro (*reset*, r_t). Elle est définie par les équations suivantes :

$$\begin{aligned} z_t &= \sigma\left(W_z \cdot [h_{t-1}, x_t] + b_z\right) \\ r_t &= \sigma\left(W_r \cdot [h_{t-1}, x_t] + b_r\right) \\ \tilde{h}_t &= \tanh\left(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h\right) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned} \tag{1.3}$$

avec les notations :

— h_t, le vecteur de couche cachée,
— x_t, le vecteur d'entrée,
— b_z, b_r, b_h, les vecteurs de biais,
— W_z, W_r, W_h, les matrices de paramètres,
— $\sigma, \tanh$: les fonctions d'activation.

Cellule LSTM - Elle est une amélioration de la cellule GRU et *mémorise* l'information sur des séquences plus longues. Elle se compose de trois portes : la porte d'entrée (i_t), la porte d'oubli (f_t), et la porte de sortie (o_t). Elle est représentée sur la figure 1.5.

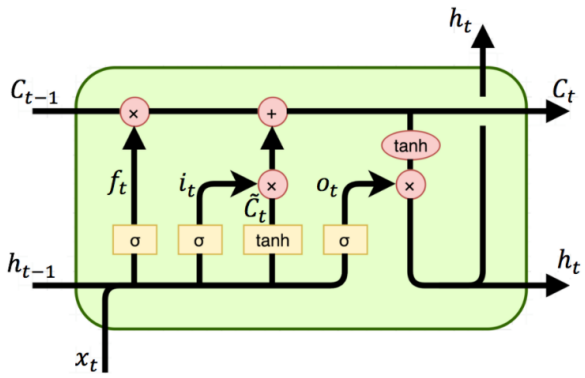

FIGURE 1.5 – Représentation visuelle d'une cellule LSTM.

Elle est définie par les équations (1.4).

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ \tilde{C}_t &= \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ h_t &= o_t \odot \tanh\left(C_t\right) \end{aligned} \tag{1.4}$$

avec les notations :

— h_t, C_t, les vecteurs de couche cachée,
— x_t, le vecteur d'entrée,
— b_f, b_i, b_c, b_o, les vecteurs de biais,
— W_f, W_i, W_c, W_o, les matrices de paramètres,
— $\sigma, \tanh$, les fonctions d'activation.

Les cellules élémentaires GRU ou LSTM sont ensuite mises bout à bout (horizontalement) pour former des couches plus complexes. Elles peuvent aussi être déclinées en deux variantes :

1. en empilant les couches de cellules sur la profondeur afin d'obtenir des réseaux profonds,
2. les réseaux neuronaux récurrents bidirectionnels [SP97] (BRNN) permettent de tenir compte des événements futurs dans leurs prédictions.

Les RNN sont capables de fonctionner dans deux modes différents : *stateless* et *stateful*. Par défaut, les RNN retiennent l'information à l'intérieur d'une séquence uniquement (*stateless*), correspondant au fonctionnement le plus commun résolvant une grande partie des problèmes. Dans ce mode les états cachés sont initialisés à zéro entre chaque nouvelle séquence. Dans certains cas, les

séquences peuvent être très longues et nécessitent d'être découpées en plusieurs sous-séquences pour être envoyées au modèle. Afin de garder le contexte entre elles, le mode *stateful* est nécessaire. Concrètement, il s'agit d'initialiser les états cachés (h_t et/ou C_t) des couches de RNN à l'itération N avec les états cachés de l'itération $N-1$.

1.2.3 Modalité vidéo

L'analyse d'image est le premier domaine à avoir bénéficié de modèles de bout-en-bout à apprentissage profond. Les réseaux CNN sont introduits pour la première fois avec les travaux de [LB98] avec une première architecture intitulée LeNet5 [LBD+89] utilisant la rétropropagation du gradient pour l'apprentissage des filtres de convolution. Les performances de ce modèle ont rapidement dépassé celles plus conventionnelles [PYLP11, WS13, ZWY+13], basées sur l'extraction de caractéristiques [SGLZ11] (*Histogram of Oriented Gradients* (HOG), *Histogram of Optical Flow* (HOF), *Discrete Fourier transform* (DFT), etc.) et suivies d'un classifieur [AW14] (par exemple SVM [EP01]).

De nouvelles architectures sont alors apparues permettant de résoudre de nouvelles problématiques et d'augmenter les performances de différentes tâches relatives au domaine du traitement d'images. Les modèles de l'état de l'art dans ce domaine sont aujourd'hui principalement basés sur des CNN. Nous pouvons recenser dans le tableau 1.2 les architectures de la littérature pour les tâches usuelles de traitement d'images.

TABLE 1.2 – Résumé des modèles disponibles dans la littérature pour les tâches usuelles de traitement image/vidéo.

Application	Citation
Segmentation d'image	[ZZZ+21]
Classification d'image	[DLLT21]
Détection d'objet	[RF17]
Génération d'image	[KLA+20]
Estimation de posture humaine	[CSWS17]
Suivi de multiples objets dans une vidéo	[WZL+20]
Classification d'action	[HKS17]
Reconnaissance de visage	[DGXZ19]
Flow optique	[JCL+21]
Amers du visage	[YNC+18]

Dans un contexte d'analyse d'émotions, de sentiments ou d'interactions humaines avec des données visuelles, il fait sens de se concentrer sur la zone du visage. En effet, il s'agit de la zone la plus déformable dans un tel contexte. Comme pour la reconnaissance d'action [GXX+18], où certaines actions ne peuvent pas être discriminées avec une seule image, nous pouvons analyser dans le temps le visage des passagers pour obtenir une représentation spatio-temporelle de ces derniers. Cela permet de capturer de nouveaux percepts tels les mouvements

de la tête/visage, des lèvres, des yeux et de la bouche. Cette technique est utilisée par [PCH+17] en appliquant un réseau C3D [TBF+15]. L'inconvénient des modèles C3D est la difficulté à les entraîner et leur besoin important en données annotées pour converger. De plus, il s'agit d'un modèle lourd, avec une compacité faible, rendant quasiment impossible son utilisation dans un environnement embarqué. Il est alors préférable d'utiliser un modèle R3D [HKS17] qui est plus compact (voir figure 1.6, issue de [HKS17]). Deux versions sont proposées (R3D : 18 et 34 couches). Eu égard à notre contexte embarqué, nous privilégions la version à 18 couches dans nos expérimentations.

Layer Name	Architecture	
	18-layer	34-layer
conv1	$7 \times 7 \times 7, 64$, stride 1 (T), 2 (XY)	
conv2_x	$3 \times 3 \times 3$ max pool, stride 2 $\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 3$
	average pool, 400-d fc, softmax	

FIGURE 1.6 – Modèle resnet3D avec deux variantes : 18 (resp. 34) couches sur la gauche (resp. droite). Chaque convolution est suivie par une fonction ReLu.

La fonction ReLU, définie par l'équation 1.5, est utilisée pour ajouter des non-linéarités dans le modèle afin d'apprendre des notions complexes. Elle est privilégiée aux autres fonctions d'activation, car les entraînements des modèles sont plus rapides avec cette dernière *i.e.* sa dérivée est plus simple à calculer.

$$f(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ x & \text{si} \quad x \geq 0 \end{cases} \tag{1.5}$$

L'idée générale du R3D est de remplacer toutes les convolutions bidimension-

nelles (2D) de l'architecture Resnet [HZRS15] par des convolutions tridimensionnelles (3D). Les connexions résiduelles permettent une meilleure rétropropagation du gradient vers les couches d'entrée du modèle favorisant la convergence de ce dernier.

Une alternative pour capturer précisément des mouvements des zones du visage est l'utilisation du flux optique. Le flux optique consiste à estimer le mouvement par pixel entre deux images consécutives. L'objectif est de trouver le déplacement d'un ensemble de caractéristiques ou de tous les pixels de l'image pour calculer leurs vecteurs de mouvement. Ce flux optique est ensuite souvent converti en image pour être empilé sur la profondeur (dimension du temps) puis un réseau R3D est utilisé pour faire la prédiction [IMS+17]. Cette approche n'a pas été implémentée ici, car la chaîne de traitement de l'image devient trop lourde pour un usage embarqué. Il est nécessaire d'inférer le modèle qui calcule le flux optique, convertir les résultats obtenus en image et répéter l'opération x fois pour avoir un minimum de 16 images en entrée du modèle R3D qui fait la prédiction.

Discussion - La quantité d'information apportée par chacune des modalités pour l'analyse d'émotion/sentiment n'est pas équirépartie. En effet, nous constatons que la modalité vidéo est souvent la moins informative, suivie de l'audio puis du texte [PCH+17, CHP+17, HSR18, AYV19]. Nous nous attendons donc à retrouver un tel constat dans nos futures évaluations.

La communauté Vision propose aujourd'hui une multitude d'outils pour l'analyse d'images et de vidéos. Pour notre contexte applicatif, nous orientons nos choix d'architectures vers des modèles de classification composés de couches CNN 2D et 3D. Dans le cas où ces derniers ne seraient pas en capacité d'extraire des caractéristiques pertinentes du visage du fait du statisme des passagers, imposés par l'habitacle de la voiture, nous envisageons l'exploitation de points clefs (anatomiques) du visage.

1.2.4 Modalité audio

Les applications audio sont aujourd'hui très variées avec la classification audio, la reconnaissance vocale, l'étiquetage automatique de la musique, la segmentation audio et la séparation des sources, le débruitage audio, la recherche d'informations musicales, les assistants virtuels tels qu'Alexa, Siri et Google Home, etc. Toutes ces tâches sont basées sur des modèles de bout-en-bout ou des modèles statistiques. Ces derniers sont encore très utilisés [PMP20a, CWP18] contrairement au domaine vidéo et texte. Une revue complète sur le sujet est disponible [PMP20b]. De récents travaux appliquent l'apprentissage autosupervisé pour apprendre des caractéristiques audio de haut niveau [RZP+20]. L'extraction de caractéristiques est d'une extrême importance puisque la performance du système dépend de la qualité de ces dernières. Elles doivent capturer suffisamment de propriétés audio invariantes au sein d'une même classe et de propriétés variantes entre différentes classes.

Plusieurs attributs élémentaires ont été introduits pour décrire différents types de signaux audio d'un point de vue psychoacoustique [MZB10] :

- **Durée -** Elle représente le temps pendant lequel l'énergie du son est perceptible.
- **Intensité -** Elle représente la perception humaine de la force ou de la faiblesse de sons de différentes intensités. L'intensité sonore d'un son est subjective, elle varie d'une personne à l'autre et se mesure par unités de sone et de phone [Ste55].
- **Hauteur du son -** C'est une propriété perceptive définie comme l'attribut intensif de la sensation auditive en fonction duquel un son peut être classé sur une échelle allant de doux à fort [Hou97]. La hauteur est mesurée *via* l'unité Mel.
- **Timbre -** Il est défini comme l'attribut de la sensation auditive qui permet à l'auditeur de juger que deux sons non identiques présentés de manière similaire et ayant la même intensité sonore (ainsi que la même hauteur) sont différents [Hou97].

L'extraction de caractéristiques audio tente de capturer les quatre attributs susmentionnés les plus adaptés en fonction du domaine d'application. Les caractéristiques audio possèdent cinq propriétés principales [MZB10] :

- **Format du signal -** Les caractéristiques sont calculées soit sur un codage linéaire soit sur une compression avec perte. La majorité des caractéristiques audio sont basées sur un codage linéaire.
- **Domaine -** Les caractéristiques peuvent appartenir aux domaines : temporel, fréquentiel, cepstral, fréquence de modulation et l'espace de phase reconstruit.
- **Échelle temporelle -** Les caractéristiques peuvent appartenir à trois catégories différentes : intraframe, interframe et globale. Dans les caractéristiques intraframe, le signal est considéré comme localement stationnaire. Chaque trame est prise en compte séparément, ce qui donne un vecteur de caractéristiques par trame (exemple : les MFCC que nous développons en pages *22*). *A contrario*, l'interframe capture le changement temporel d'un signal audio donné. Les caractéristiques rythmiques sont un exemple de caractéristiques interframes. Les caractéristiques globales sont calculées à partir de l'ensemble du signal.
- **Signification sémantique -** Elle comprend des caractéristiques perceptives basées sur les aspects de la perception humaine, tels que la hauteur, le rythme et les caractéristiques physiques décrivant les caractéristiques du signal sur la base de propriétés physiques et statistiques (transformée de Fourier).
- **Modèle sous-jacent -** Les caractéristiques sont calculées grâce à des modèles psychoacoustiques (banque de filtres).

Ayant caractérisé les attributs de haut niveau, nous pouvons définir les principales caractéristiques [MZB10] audio présentes dans la littérature. Nous les résumons dans la figure 1.7.

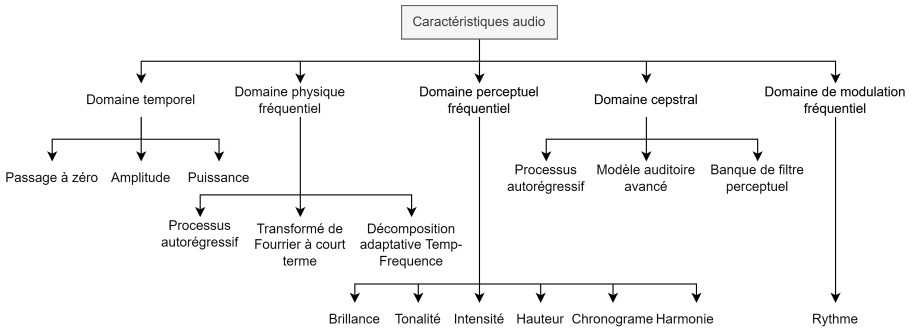

FIGURE 1.7 – Taxonomie des caractéristiques audio.

OpenSMILE [2], Librosa [3], Essentia [4] sont parmi les librairies les plus utilisées pour le calcul de caractéristiques audio. Nous privilégions OpenSMILE du fait que des fichiers de configuration avec des ensembles de paramètres audio prédéfinis sont fournis pour différentes tâches. Dans notre contexte de détection de situation conflictuelle, il est pertinent d'utiliser les fichiers de configuration pour l'analyse des émotions. Nous pouvons en recenser trois :

1. INTERSPEECH 2009 Emotion challenge [SSB09] comprenant 384 caractéristiques,
2. Emobase2010, basé sur les paramètres de [EWS09] avec 1582 caractéristiques,
3. ensemble de caractéristiques large pour les émotions, avec 6552 caractéristiques calculées.

Après extraction des caractéristiques audio, celles-ci sont envoyées à des modèles de classification, généralement des RNN [HI20, AS15, SCT21].

À l'opposé, nous avons les modèles dits de bout-en-bout. Ils sont apparus dans les années 2010 avec l'augmentation de la puissance de calcul et les bases de données grandissantes. Les performances globales se sont alors améliorées sur les applications suivantes :

— la classification audio [HCE+16] et musicale [CFSC16],
— la séparation de sources [LM19] et la segmentation audio [DOZ+20],
— la génération de musique [YCY17],
— la reconnaissance vocale [CJLV16],
— la transformation du texte vers l'audio [WSS+17].

Une alternative aux calculs de caractéristiques est l'analyse de représentations sous forme d'« image » de l'audio ou le traitement direct du fichier

2. `https://www.audeering.com/research/opensmile/`
3. `https://librosa.org/doc/latest/index.html`
4. `https://essentia.upf.edu/`

audio brut [CZB+19, KLN19]. Les trois transformations principales sont les suivantes (figure 1.8 [5]) :

(a) le spectrogramme,

(b) le Mel-spectrogramme,

(c) les coefficients cepstraux suivant l'échelle Mel (MFCC).

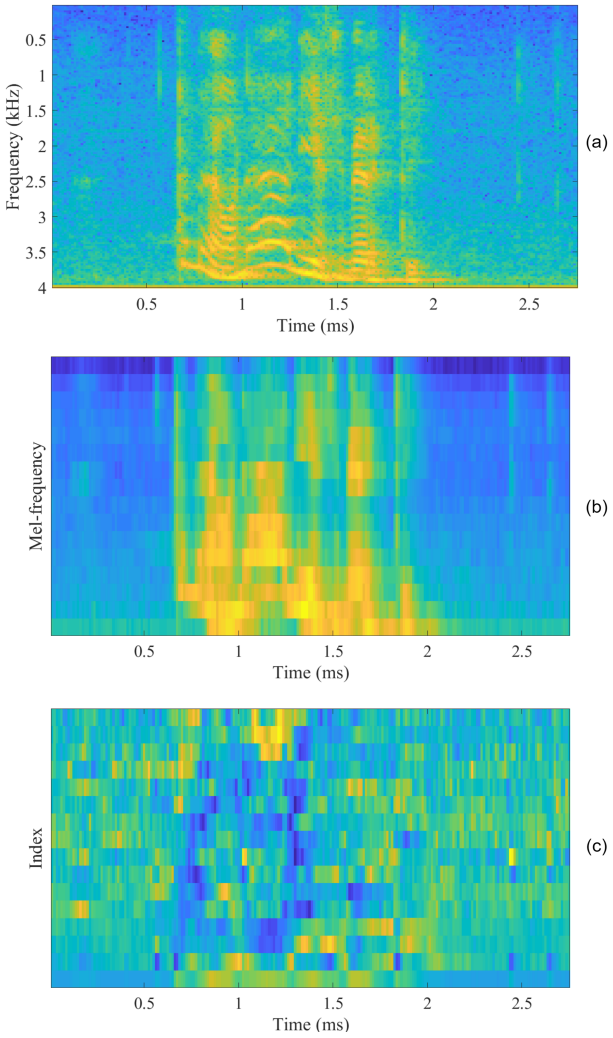

FIGURE 1.8 – Exemple de représentations visuelles d'un échantillon sonore, avec (a) spectrogramme, (b) Mel-spectrograme et (c) MFCC.

5. https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Spectrogramme - C'est une représentation visuelle du spectre des fréquences d'un signal audio en fonction de ses variations dans le temps. Il comprend donc les aspects temporels et fréquentiels du signal. Il est obtenu en appliquant la transformée de Fourier à court terme (STFT) sur le signal.

Mel-spectrogramme - Il convertit l'axe des fréquences en échelle Mel (équation (1.6)). En effet, l'oreille humaine se comporte de manière logarithmique, nous entendons plus les variations dans les basses fréquences.

$$m = 2595 \cdot \log\left(1 + \frac{f}{500}\right) \tag{1.6}$$

MFCC - Il s'agit de coefficients cepstraux calculés par une transformée en cosinus discret appliquée au spectre de puissance du signal. Les bandes de fréquence du spectre sont espacées de manière logarithmique selon l'échelle Mel.

Ces différentes transformations sont ensuite directement envoyées dans des architectures capables d'extraire automatiquement les caractéristiques pertinentes pour la tâche à accomplir. Les tâches de classification audio utilisent majoritairement la représentation image du signal. Les modèles utilisant des couches de CNN 2D sont aujourd'hui l'état de l'art pour l'analyse d'images. Il fait alors sens de les utiliser dans le domaine audio [KLN19].

La représentation image de l'audio permet d'utiliser les techniques d'augmentation de données appliquées dans le domaine vidéo et permet de tirer pleinement partie des propriétés de recherche locale de caractéristiques des CNN. Les propriétés d'invariance des CNN permettent une robustesse des performances. Toutefois un inconvénient majeur est le caractère temporel limité des CNN. Analyser des fichiers audio de plusieurs secondes devient rapidement consommateur de ressources de calcul. La temporalité est inhérente au signal audio, les RNN et le mécanisme d'attention sont au cœur des architectures de l'état de l'art pour toute tâche audio confondue.

Discussion - Les modèles de bout-en-bout nécessitent des corpus d'entraînement conséquents pour converger, comparés à d'extraction de caractéristiques manuelles. Les CNN ne sont pas adaptées aux données d'entrée de longueurs variables. Ils sont à privilégier pour la classification de fichiers audio homogènes en longueur et de durée inférieure à 10s. Concernant l'extraction de descripteurs, un total de 6552 caractéristiques n'est pas envisageable pour une solution embarquée. Nous favorisons la configuration qui en extrait 1582, car cette dernière est couramment utilisée dans la littérature [VKP+18, SGS+17, LZZ+19] et nous semble le meilleur compromis entre performances et quantité de données à traiter.

Dans le cadre de l'analyse de situations conflictuelles, il est nécessaire d'analyser des dialogues de durées variables : l'utilisation de RNN semble alors plus pertinente. De plus, pour l'analyse de sentiments, d'émotions et de dialogues,

l'extraction de caractéristiques statistiques, suivie d'un modèle d'apprentissage profond, est souvent la technique privilégiée [PCH+17, YBJ18].

1.2.5 Modalité texte

L'analyse du texte a connu une ascension fulgurante ces dernières années. Elle est aujourd'hui utilisée pour la classification et la synthèse de documents, l'analyse de commentaires produits, l'analyse des réseaux sociaux, l'analyse de sentiments, filtrage de messages spam, etc. [VBP17, VSP+17]. Le texte peut être exploré en étudiant les fréquences d'apparitions des mots [XC10], en recherchant du vocabulaire spécifique sur la base de dictionnaires de mots [XYY+19]. L'analyse statistique par tf-idf (Term Frequency-Inverse Document Frequency) [SJ88] est aussi couramment utilisée pour trouver l'importance des mots par rapport à l'ensemble d'un document/corpus. Elle est calculée en multipliant la fréquence du mot (tf) et la fréquence inverse du document (idf) :

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{1.7}$$

avec $|D|$ le nombre total de documents dans le corpus et $|\{d_j : t_i \in d_j\}|$ le nombre de documents où le terme t_i apparaît.

La théorie du tf-idf ne prenant pas en compte les relations entre les mots, l'utilisation de matrice co-occurrente pallie le problème [PSM14]. Ce processus a été ensuite devancé par des techniques d'apprentissage profond (par exemple le *word2vec* [MCCD13]).

À tout préalable pour traiter du texte, il est nécessaire de le rendre compréhensible par un ordinateur, pour ce faire il doit être converti en valeur numérique. Deux options sont couramment utilisées : l'encodage 1 parmi n (*one-hot encoding*) et l'encodage en entier unique. La seconde méthode est à privilégier, car les vecteurs générés sont moins *sparse* (moins de valeurs égales à zéro). À ce stade le vecteur peut être envoyé à un réseau de neurones, mais, lorsque nous travaillons sur de larges corpus, les vecteurs d'entrées deviennent tellement grands qu'ils ne sont pas exploitables. À cela s'ajoute la redondance de l'information. La couche *embedding* rentre alors en jeu en compressant les données entrantes. Elle peut être comparée à une table de correspondance qui relie les indices uniques à un vecteur dense (compressé).

La figure 1.9 illustre un exemple classique de représentation des relations entre les mots « roi », « reine », « homme », « femme ». Notons que le mot « roi » et le mot « homme » sont sémantiquement liés dans le sens où ils représentent tous deux un humain de sexe masculin. Cependant, le mot « roi » possède une caractéristique supplémentaire, à savoir la royauté. De même, que le mot « reine » possède le critère de royauté et « femme » représente le sexe féminin. Puisque la relation entre le « roi » et la « reine » (royauté masculine - royauté féminine) est similaire à la relation entre l'« homme » et la « femme » (humain masculin - humain féminin). En les soustrayant l'un de l'autre, nous

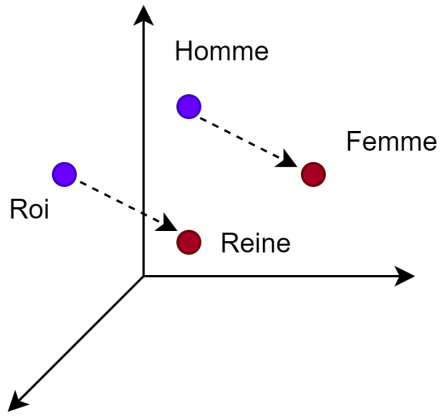

FIGURE 1.9 – Exemple de représentation des coefficients d'une couche d'*embedding*.

obtenons l'équation :

$$roi - reine = homme - femme \tag{1.8}$$

La soustraction de deux mots revient à soustraire leurs vecteurs. Si, par exemple, nous ne connaissons pas le féminin du mot « roi », nous pouvons le déterminer par l'équation précédente : $reine = roi - homme + femme$.

Plusieurs théories pour le calcul de l'*embedding* sont développées :
— GloVe [PSM14],
— Skip-gram (SG) [MCCD13],
— Continuous bag-of-words (CBOW) [MCCD13],
— C&W [CW08].

Ces couches sont capables d'apprendre des concepts tels que le masculin/féminin, capitale/ville, nom/verbe, etc. Les poids des couches de ces réseaux de neurones sont sauvegardés et réutilisables lorsque nous souhaitons développer une nouvelle application. Il arrive que ces couches ne soient pas directement utilisées, car le vocabulaire utilisé pour les entraîner est trop éloigné de l'application finale ou encore lorsque la langue est différente. Le critère de la compacité des modèles peut aussi être un frein à leur utilisation. Une alternative est de définir notre propre *embedding*. Il s'agit d'une couche entièrement connectée :

$$\mathrm{h}_i = \sum_{j=1}^{N_{dim}} w_{ij} \cdot x_j \tag{1.9}$$

où Xj est le mot et w la matrice de paramètres appris durant l'entraînement.

Embedding - Il est généralement en entrée de la taille du vocabulaire et en sortie un vecteur compressé de taille entre 100 et 300. Ensuite, il est possible de l'envoyer à des réseaux neurones profonds. Les couches récurrentes de type GRU ou LSTM sont privilégiées pour des tâches où les caractéristiques importantes sont contenues dans la longueur de la séquence (pour plus de détails sur ces réseaux récurrents, voir la sous-section 1.2.2). À l'inverse, les systèmes utilisant des CNN à une ou deux dimensions sont utilisés pour la recherche de caractéristiques locales ou à positionnement invariant. Les travaux suivants [YKYS17] comparent les trois méthodes (CNN, GRU, LSTM) pour différentes tâches de traitement de texte. Une disruption avec les réseaux *transformer* [VSP+17] et le concept « mécanisme d'attention » dans les réseaux de neurones profonds a permis des avancées majeures dans le domaine du *Traitement Automatique du Langage* (TAL). Citons ici les travaux de [DCLT19] qui ont développé le modèle BERT, il s'agit de la référence de l'état de l'art sur 11 tâches de traitement de texte pour la langue anglaise. Les travaux de [MMOS+20] ont permis d'entraîner le modèle BERT pour la langue française. Les modèles de question-réponse, classification, segmentation, résumé, analyse de sentiment, etc. pour le texte ont vu leurs performances s'améliorer significativement [6].

Discussion - Une solution pour l'analyse d'interactions humaines (afin de détecter des situations conflictuelles) est d'étudier les dialogues des interlocuteurs. Les méthodes citées précédemment sont toutes adaptées à cette tâche avec des performances plus ou moins variables et des contraintes associées du fait de notre contexte de langage oral et véhicule. En effet, les techniques statistiques peuvent avoir des lacunes pour capturer assez d'information pour permettre la classification. À l'inverse, les modèles récents comme les réseaux *transformer*, ont besoin d'être entraînés sur des jeux de données massifs pour obtenir de bons résultats. Il est préférable de réutiliser des modèles déjà entraînés comme *CamemBert* (entraîné sur 140 Gigaoctet de textes issus de Wikipédia). Les réseaux de neurones étant représentatifs des jeux de données sur lesquels ils sont entraînés, il est peu probable que ce type de modèle fonctionne dans notre contexte de langage oral où les phrases peuvent être mal construites et les répétitions très fréquentes. L'autre problématique liée au réseau *transformer* est la puissance de calcul nécessaire pour faire une inférence d'un tel modèle, non envisageable dans notre contexte. Finalement des techniques intermédiaires apparaissent plus pertinentes vis-à-vis de nos contraintes. Nous privilégierons d'entraîner notre propre couche d'*embedding*, utiliser des modèles RNN (LSTM ou GRU), en les combinant à une couche « d'attention ». L'utilisation de l'architecture HAN [YYD+16] analysant le texte au niveau des mots et de la phrase est aussi une solution envisageable. Dans la prochaine section, nous présenterons les différentes manières de combiner des signaux qu'ils soient hétérogènes, comme ceux que nous venons de présenter, ou non.

6. `https://paperswithcode.com/area/natural-language-processing`

1.3 Fusion multimodale

Notre vie quotidienne est multimodale, nous utilisons tous nos sens pour analyser des situations et prendre des décisions. Les signaux provenant de différentes modalités transportent des informations complémentaires sur des objets ou des scènes. Le concept de multimodalité est de combiner l'information provenant de plusieurs sources afin d'améliorer la robustesse et les performances de prédiction globale. Plus précisément, il s'agit d'exploiter les corrélations entre les sources. La fusion au sens large permet de combiner toutes sortes de données : capteurs, signaux analogiques ou numériques, images, sons, textes, données temporelles, etc. Les améliorations notables de la multimodalité sont démontrées dans différents domaines d'application comme la description d'images [KZS18], l'analyse du visage et des émotions [LSXC17, KBL+16], la reconnaissance vocale [FGL+17], etc. Cependant, il n'y a pas de consensus dans la littérature sur la définition exacte de ces différents types de fusion. Néanmoins, les termes « fusion tardive » et « fusion précoce » sont généralement présents. Le concept de « fusion intermédiaire » est parfois décrit comme une fusion précoce plus tardive. Nous détaillons ces trois fusions dans cette section.

1.3.1 Fusion précoce

La fusion au niveau des données est une méthode de fusion de plusieurs signaux avant de procéder à l'analyse (cf. figure 1.10). Cela consiste à combiner les données en supprimant la corrélation entre deux capteurs. Il existe de nombreuses solutions statistiques qui peuvent être utilisées avec notamment l'analyse en composantes principales, l'analyse de corrélation canonique et l'analyse en composantes indépendantes.

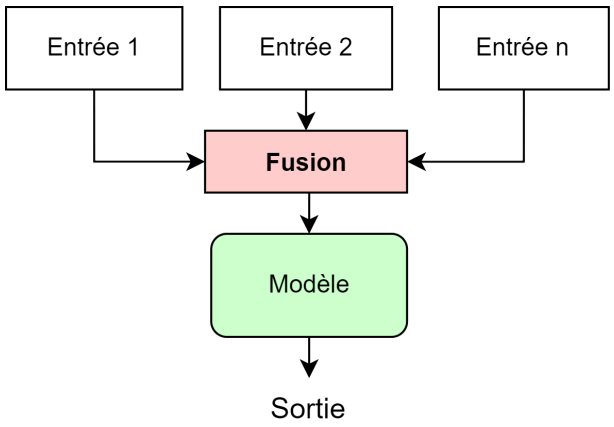

FIGURE 1.10 – Exemple de représentation des coefficients d'une couche d'*embedding* lors d'une fusion précoce.

La fusion précoce est applicable aux données brutes, ou aux données prétrai-

tées obtenues à partir de capteurs. Les caractéristiques des données doivent être extraites avant la fusion, sinon le processus est difficile à cause des taux d'échantillonnage pouvant être différents entre les modalités. La synchronisation des sources de données est également difficile lorsqu'une source de données est discrète et que les autres sont continues.

Si les données sont correctement alignées, les corrélations croisées entre les éléments de données peuvent être exploitées, offrant ainsi la possibilité d'augmenter les performances du système. Toutes ces contraintes rendent souvent l'utilisation de la fusion précoce difficile à mettre en œuvre.

1.3.2 Fusion tardive

La fusion tardive [SWS05], illustrée sur la figure 1.11, est la méthode de fusion la plus simple et la plus utilisée. Elle groupe les sources de données après traitement complet de chacune d'elles de manière indépendante, puis la fusion est effectuée lors d'une étape de prise de décision.

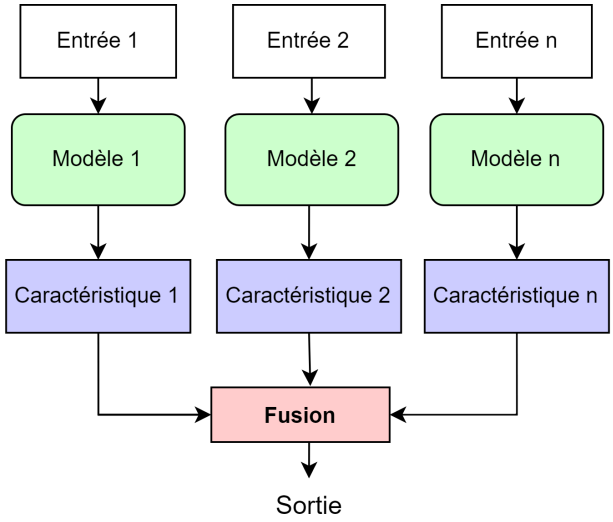

FIGURE 1.11 – Exemple de représentation des coefficients d'une couche d'*embedding* lors d'une fusion tardive.

Les règles de décision sont majoritairement effectuées par un vote pondéré (*soft voting*) ou majoritaire (*hard voting*), moyenne, règles de Bayes, méta-modèle, etc. Cette technique est plus simple que la méthode de fusion précoce, en particulier lorsque les sources de données sont très différentes les unes des autres en termes de taux d'échantillonnage, de dimensionnalité des données, etc. Finalement, aucune preuve ne montre que la fusion tardive est plus performante que la fusion précoce et inversement. En fonction du contexte applicatif, des types de sources fusionnées, des architectures, etc. les performances entre

les deux fusions peuvent varier considérablement.

Lorsque les flux de données d'entrée varient considérablement en termes de dimensionnalité et de taux d'échantillonnage, l'utilisation de la fusion tardive est plus simple et plus flexible. L'inconvénient est la nécessité d'entraîner séparément chacun des modèles unimodaux.

1.3.3 Fusion intermédiaire

Les deux fusions présentées précédemment sont très contrastées. Une alternative est l'utilisation de la fusion intermédiaire qui apporte plus de flexibilité (voir figure 1.12). Elle consiste à fusionner les données dans un espace commun de dimension inférieure, plus simple à analyser. En d'autres termes, il s'agit de compresser la donnée d'entrée en générant un espace latent. Un modèle apprend ensuite à faire la prédiction à partir de cet espace latent.

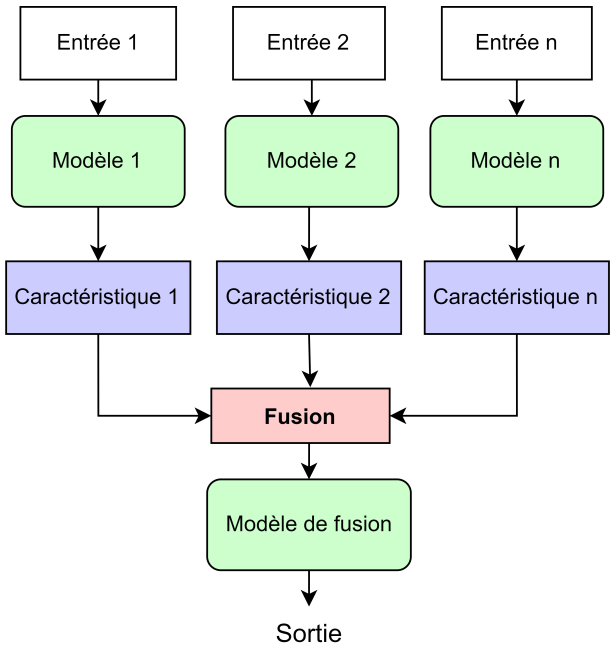

FIGURE 1.12 – Exemple de représentation des coefficients d'une couche d'*embedding* lors d'une fusion intermédiaire.

Cette méthode nécessite d'obtenir une base commune pour toutes les modalités avant la fusion. Après avoir obtenu les données dans une matrice commune (concaténation moyenne, *max-pooling*, etc.), elles sont ensuite analysées, par exemple à l'aide d'un réseau de neurones. La synchronisation des différentes modalités est aussi primordiale.

1.3.4 Discussion

Dans un contexte de fusion audio, vidéo et texte qui sont trois modalités hétérogènes, les fusions intermédiaires et tardives semblent les plus appropriées. Nous devons alors extraire en premier lieu des caractéristiques pour chacune des modalités de manière indépendante. De plus, dans la littérature, il existe *a priori* aucune architecture de fusion précoce pour les modalités audio, vidéo et texte combinées. Nous remarquons aussi que la synchronisation des signaux d'entrée est primordiale. Un moyen de surmonter cet inconvénient consiste à collecter les données ou les signaux à un taux d'échantillonnage commun.

1.4 Conclusion

Nos investigations portent sur l'analyse multimodale d'interactions humaines, dans le contexte du véhicule, pour discriminer des situations à tendance conflictuelle. Les contraintes d'un système embarqué font aussi partie de notre cahier des charges. Nous exploitons les modalités audio, vidéo et texte avec des modèles extrayant des caractéristiques statistiquement et/ou automatiquement. Eu égard à cet état de l'art, nos investigations ont pour objectif :

1. d'exposer plusieurs voies d'investigation de système d'analyse paramétrique ou bout-en-bout. Démontrer les avantages de la multimodalité avec des gains de performance de prédiction notables,
2. d'explorer différents types de fusion (intermédiaire *vs* tardive) pour déterminer la plus pertinente pour des modalités hétérogènes pour notre application,
3. d'évaluer différentes techniques pour modéliser les aspects local et global de la temporalité dans les réseaux de neurones,
4. d'identifier des modèles de références dans le domaine afin de se comparer,
5. de prendre en considération les contraintes embarquées dans le but d'augmenter la compacité des modèles implémentés.

Le chapitre suivant propose une étude des corpus publics se rapprochant de notre application. Un premier modèle ainsi que deux stratégies de fusion sont implémentés et évalués sur un corpus public proche de notre contexte applicatif afin de se comparer à la littérature.

Chapitre 2

Analyse multimodale pour la classification de sentiments

Sommaire

2.1 Introduction

Nous présentons dans ce chapitre l'implémentation d'un système multimodal capable d'ingérer des données vidéo, audio et texte. Celle-ci est entraînée sur un corpus public pour la classification de sentiments. Ces premières expérimentations

ont pour vocation de nous familiariser avec différents éléments : l'étude des modalités hétérogènes audio, vidéo et texte ; la fusion de celles-ci ; ainsi que la notion de compacité des réseaux de neurones. Deux leviers seront particulièrement considérés pour l'étude de la réduction des coûts en calcul : la compacité des modèles développés et le pré-traitement des données. Nous souhaitons également comparer notre premier modèle de fusion avec des modèles *benchmark* de la littérature. Comprendre et exploiter un corpus public nous donnera également des informations pertinentes lorsque nous construirons le nôtre (voir chapitre 3).

2.2 Corpus public pour l'analyse de sentiments ou émotions

Cette section détaille les jeux de données publiques pour catégoriser les sentiments de personnes émettant des avis sur des objets, films ou livres. Une analyse qualitative en fonction de notre cahier des charges est ensuite menée afin de choisir le corpus correspondant à nos besoins.

2.2.1 Analyse des différents corpus multimodaux pour la classification de sentiments

Les jeux de données publics comportant trois modalités pour l'analyse de sentiments ou d'émotions sont rares, et à notre connaissance inexistants, si nous désirons des dialogues en français. Nous avons pu en recenser six dans la littérature. Plusieurs critères sont importants dans le choix de notre premier jeu de données. En effet, nous souhaitons nous rapprocher de notre contexte applicatif final qui est l'habitacle du véhicule. Nos critères sont au minimum les suivants : un jeu d'acteur naturel sans suivi de script, un enregistrement vidéo où le plan de vue sur le visage est de face, car cohérent avec l'instrumentation d'un cockpit de voiture. Voici les six corpus recensés :

— MOUD [1] [PRMM13],
— CMU-MOSI [2] [ZZPM16],
— CMU-MOSEI [3],
— ICT-MMMO [4] [WWK+13],
— Youtube [5] [MMD11],
— IEMOCAP [6] [BBL+08].

Afin de choisir le jeu de données final parmi cette sélection, nous détaillons leurs caractéristiques globales dans le tableau 2.1, puis une analyse plus fine est menée dans la section 2.2.2.

1. http://multicomp.cs.cmu.edu/resources/moud-dataset/
2. http://multicomp.cs.cmu.edu/resources/cmu-mosi-dataset/
3. http://multicomp.cs.cmu.edu/resources/cmu-mosei-dataset/
4. http://multicomp.cs.cmu.edu/resources/ict-mmmo-dataset/
5. http://multicomp.cs.cmu.edu/resources/youtube-dataset-2/
6. https://sail.usc.edu/iemocap/

TABLE 2.1 – Comparaison de six corpus. « Emotion » et « Sentiment » indiquent la manière dont le corpus est annoté.

Corpus	Tours de parole	Locuteurs	Sentiment	Emotion	Durée
MOUD	400	101	✓	x	59min
MOSI	2199	89	✓	x	2h36
MOSEI	23453	1000	✓	✓	65h53
ICT-MMMO	340	200	✓	x	13h58
YouTube	300	50	✓	x	29min
IEMOCAP	10000	10	x	✓	11h28

2.2.2 Choix du corpus MOSI

Parmi les jeux de données susmentionnés, nous choisissons MOSI. Ce choix est réalisé pour les raisons énoncées ci-après. Tout d'abord, les locuteurs agissent naturellement en comparaison à IEMOCAP, où l'on demandait aux sujets d'agir ou de suivre un scénario. Le deuxième point est le langage, les corpus avec des dialogues français n'existant pas nous avons opté pour MOSI, où les critiques sont en anglais. En effet, il est plus aisé de faire une analyse quantitative avec des critiques en anglais par rapport à l'ensemble de données MOUD où les locuteurs sont espagnols. Concernant le flux audio et vidéo, ils intègrent intrinsèquement des variations de luminosité, d'arrière plan, de qualité, etc. Cet aspect est pertinent pour notre application finale, car l'habitacle d'un véhicule est soumis à de nombreuses fluctuations de lumière, de bruits parasites, etc. Par ailleurs, les participants dans le corpus de données YouTube sont jeunes (14 ans). Dans notre application finale, les sujets n'auront pas moins de 18 ans. De plus, il ne possède pas assez d'orateurs différents avec peu de tours de paroles. Le corpus MOUD n'est pas assez important en volume de donnée. Les ensembles de données (MOSEI, ICT-MMMO et IEMOCAP) sont trop volumineux pour être utilisés avec nos ressources informatiques disponibles. Enfin, le jeu de données MOSI répond aussi à plusieurs autres attentes, notamment :

— le nombre de locuteurs différents permet d'avoir une bonne diversité,
— la durée totale nous permet des temps de calcul convenables,
— les visages sont en vue de face, ce qui correspond bien à nos besoins.

MOSI est également un corpus très populaire dans la littérature pour l'analyse multimodale de sentiments [PCH+17, CHP+17, HSR18]. Il nous permet de nous comparer à d'autres travaux, mais aussi d'avoir plus d'informations issues de la communauté scientifique, notamment sur les réglages des hyperparamètres des systèmes concurrents.

Le jeu de données MOSI contient 93 vidéos enregistrées grâce à 89 locuteurs différents pour approximativement 2h36 de vidéo. Il est divisé et annoté en 2199 sous-séquences (tours de parole) représentant 26 457 mots. Les locuteurs sont répartis en 48 femmes et 41 hommes. Les sujets de discussion sont des critiques en anglais de films ou de livres.

L'angle de vue de la caméra du corpus MOSI est donné sur la figure 2.1.

FIGURE 2.1 – Exemples de vues frontales du corpus MOSI.

2.3 Modèle de bout en bout pour la classification de sentiments

L'information « sentiment » étant véhiculée par différents canaux (ou modalités), nous étudions *a priori* ceux qui véhiculent au mieux cette information, c'est-à-dire la vidéo, l'audio et le texte. Ce sont aussi les plus étudiés dans la littérature [PCH+17, CHP+17, HSR18, AYV19]. Ces canaux sont hétérogènes. Nous concevons intuitivement trois réseaux de neurones, un pour chaque modalité. Ensuite, les caractéristiques extraites sont fusionnées pour obtenir la prédiction finale. La fusion tardive s'apprête ici le mieux, car nous avons besoin dans un premier temps d'extraire les informations par modalité. Chacune d'elle requiert un modèle spécifique à cause des dimensions différentes des données d'entrées.

La figure 2.2 illustre notre système (*pipeline*) unimodal et multimodal. Les trois cases violettes représentent le prétraitement des données. Les trois cases vertes illustrent les modèles d'apprentissage profond. Les prédictions sont faites sur deux classes de sentiments : positif ou négatif, afin de pouvoir nous comparer avec la littérature.

Les sections suivantes détaillent l'implémentation de chacune des trois modalités.

2.4 Analyse vidéo

Nous privilégions la convolution 3D avec un modèle R3D modifié [HKS17], dans le but de réduire les besoins en ressources de calcul, favorisant la compa-

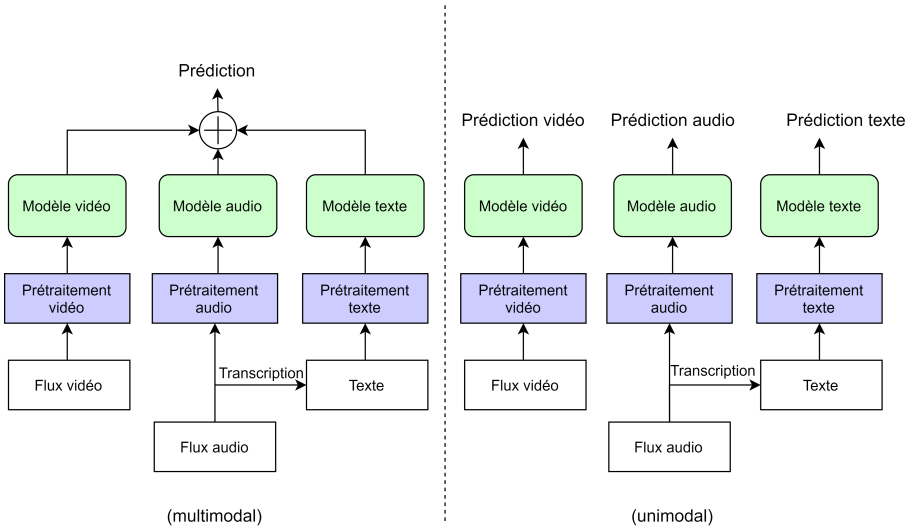

FIGURE 2.2 – Schéma global des systèmes multimodal et unimodal.

cité. Une seule image contenant trop peu de caractéristiques visuelles pour la classification de sentiments, nous décidons d'utiliser des réseaux neuronaux capables de capter des caractéristiques spatio-temporelles (évolution de l'information entre un nombre donné d'images consécutives). En effet, nous pensons que les mouvements faciaux tels que les clignements de paupières, la déformation de la bouche ou encore les balancements de la tête sont informatifs. Les réseaux 3D-CNN (C3D) [TBF+15] et le 3D-CNN résiduel (R3D) [HKS17] ont été appliqués avec succès dans le passé pour la reconnaissance d'actions [SUD19, GXX+18]. En raison des contraintes de calcul plus faibles et de ses capacités en matière de reconnaissance ou de classification d'actions, nous privilégions le modèle R3D. Ces deux types de modèles sont alimentés par une entrée à quatre dimensions définies comme suit :

$$R^{f*c*h*w} \tag{2.1}$$

où f est le nombre d'images, c est le nombre de canaux (trois par exemple pour les images RGB), h est la hauteur des images et w est la largeur des images.

Avant d'alimenter le modèle, nous extrayons et recadrons le visage en utilisant le détecteur de points clés de la librairie Dlib [Kin09] au lieu de détecteurs de visages. Grâce à cette technique, nous obtenons un alignement précis de la tête entre chaque image consécutive, point important pour l'opération de convolution 3D. Nous utilisons le menton, les oreilles et les sourcils gauche et droit pour déterminer une boîte englobante et recadrer le visage à cette taille. Ensuite, les images sont ré-échantillonnées à 50px * 50px (voir exemple sur la figure 2.3). Cette résolution permet d'obtenir les meilleures performances de prédiction avec un coût en calcul le plus faible. Pour rappel, la convolution consiste à parcourir

une image avec un filtre et appliquer le produit matriciel de Hadamard entre ces deux éléments avec un pas spécifique. La taille de l'image d'entrée impacte alors directement le nombre de convolutions nécessaire pour la parcourir. Pour un pas et une taille de filtre fixe, plus l'image est grande plus il y a de convolution, augmentant les besoins en ressources de calcul. Nous avons alors cherché les meilleures performances du modèle puis réduit la résolution des images d'entrée jusqu'à ce que les performances commencent à chuter. Le point d'inflexion nous donne alors ta taille d'image d'entrée du modèle.

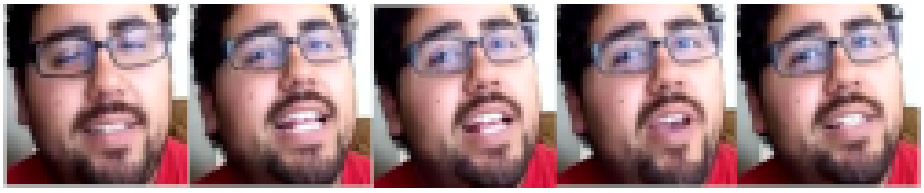

FIGURE 2.3 – Exemple de visage échantillonné en 50×50 pixels.

Dans la lignée des choix précédents, et afin de réduire la charge de calcul, nous modifions également les dernières couches de convolution 3D du modèle R3D originel. La modalité vidéo étant peu informative, nous réduisons le nombre de filtres de 512 à 350. Cette amélioration réduit le nombre de paramètres de 13 millions, ce qui est cohérent avec notre choix de vouloir réduire les coûts en ressources de calcul (voir tableau 2.2). Ce tableau montre que le modèle C3D n'est pas une solution adéquate pour les systèmes embarqués. À performances équivalentes, environ 57% de F1 score (voir la section 2.10 pour le détail de cette métrique), le modèle R3D réduit drastiquement le nombre de paramètres et la taille mémoire du modèle d'un facteur deux comparé au C3D. Finalement, notre variante R3D réduit par un facteur 3 le nombre de paramètres et par un facteur 2 la taille mémoire, en opposition au modèle C3D. Cela représente une amélioration significative pour des résultats équivalents.

TABLE 2.2 – Comparaison de trois modèles CNN basés sur la vidéo.

Modèle	#Paramètres	Usage Mémoire vive *Random Access Memory* (RAM)
C3D	63,32 M	300 MB
R3D	33,18 M	265 MB
Notre R3D	**20,78 M**	**166 MB**

Le tableau 2.3 décrit le modèle R3D [HKS17] modifié, consacré à la vidéo. Ce dernier ingère 16 images. Il est composé de cinq blocs résiduels convolutionnels 3D avec un nombre croissant de filtres sur les couches basses. Les dernières couches des réseaux convolutionnels contiennent les caractéristiques de haut niveau qui sont faiblement nombreuses dans notre cas, car la modalité vidéo est peu informative. Nous réduisons jusqu'à 350 le nombre de filtres dans le

bloc conv5, en dessous de cette valeur les performances commencent à chuter. Ensuite, les 350 caractéristiques extraites passent par une couche dense pour inférer la prédiction finale.

TABLE 2.3 – Modèle R3D modifié pour l'analyse de la vidéo. Avec H la hauteur, L la largeur et P la profondeur des filtres de convolutions. Nf dénote le nombre de filtre et Nb le nombre de blocs.

Couches	[(H, L, P), Nf] x Nb
Conv1	[(7 x 7 x 3), 64] x 1
Conv2	[(3 x 3 x 3), 64] x 2
Conv3	[(3 x 3 x 3), 128] x 2
Conv4	[(3 x 3 x 3), 256] x 2
Conv5	[(3 x 3 x 3), 350] x 2
FC	(350,2)

2.5 Analyse Audio

Pour l'analyse audio, nous expérimentons deux techniques.

Tout d'abord, la classification à l'aide d'un réseau neuronal convolutif CNN. L'objectif est de transformer le signal en une image spectrale (représentation temps-fréquence), puis de l'envoyer au CNN. Cette technique est largement utilisée pour la classification de la parole et de la musique [HCE+17] ou pour la classification des émotions et du genre [AVTP17].

Le deuxième modèle est une classification utilisant un LSTM [HS97]. Pour rappel, les LSTM sont aujourd'hui utilisés pour analyser des données temporelles. Leur particularité est liée à leur capacité à mémoriser des informations sur une longue période. Les caractéristiques sonores sont extraites toutes les 10 ms avec une fenêtre glissante de 60 ms grâce à la librairie OpenSMILE [EWS10]. Nous utilisons Emobase2010 [SSB+10]. Se référer à la section 1.2.4 pour plus de précision sur les caractéristiques audio.

L'ensemble contient 1582 caractéristiques qui résultent d'une base de 34 descripteurs bas niveau (*Low Level Descriptor* (LLD)) auxquels sont ajoutés 34 coefficients delta correspondants. 21 fonctions sont appliquées (moyenne arithmétique, maximum, minimum, quartile, percentile, etc.) à chacun de ces 68 paramètres (soit 1428 caractéristiques). À cela s'ajoute, 19 fonctions appliquées aux 4 paramètres basés sur le *pitch* (fréquence fondamentale) et à leurs quatre dérivées (soit 152 caractéristiques). Enfin, le nombre de *pitch onsets* (pseudo syllabes) et la durée totale de l'entrée sont ajoutés (soit 2 caractéristiques).

Dans le but de réduire les coûts en ressource de calcul, nous réduisons le nombre de caractéristiques sans impacter les performances. Parmi les 21 fonctions, nous en déterminons empiriquement 14 qui donne les meilleures performances réduisant à 1054 le nombre de caractéristiques. Les fonctions retirées sont principalement liées au calcul des percentiles et quartiles.

Le tableau 2.4 illustre notre modèle audio. Il est alimenté par un tenseur de taille fixe : la largeur est le nombre de caractéristiques, et la hauteur est le nombre de pas de temps. Le modèle est composé de deux couches LSTM de 800 cellules chacune, suivies d'une couche dense pour prédire les sentiments. Le nombre de cellules pour le modèle LSTM est fixé au départ à 600, vis-à-vis des travaux de [PCH+17], puis affiné empiriquement à 800.

TABLE 2.4 – Modèle audio.

Couches	(Entrée, Sortie) x taille
LSTM	(1054,800) x 2
FC	(800,2) x 1

2.6 Analyse du texte

Concernant la modalité « texte », nous proposons dans un premier temps d'extraire manuellement les caractéristiques. Après avoir créé une liste de tous les mots du jeu de données, nous la filtrons pour privilégier les adjectifs et les verbes [BCP+07, PR17] représentant un vocabulaire de 860 mots uniques.

Les mots sont ensuite encodés en nombre entier unique. Enfin, les phrases sont définies avec une longueur fixe et passent par la couche de vectorisation de mot (*embedding*) pour compresser l'espace des caractéristiques d'entrée en un espace plus petit.

Cette couche d'*embedding* est généralement le modèle de Google entraîné sur 100 milliards de mots provenant de Google News [MSC+13]. Le poids de ce modèle coûte plus de 3,5 Go à charger en mémoire vive. Ce n'est pas une solution applicable dans un environnement embarqué. Par conséquent, nous décidons d'entraîner notre propre couche *d'embedding*. Elle reçoit en entrée un vecteur de taille 860 correspondant aux adjectifs et verbes les plus importants/représentés dans le corpus et génère ensuite un vecteur de caractéristique compressé de taille 100. Le LSTM (voir tableau 2.5) est alimenté pour finalement prédire les sentiments. Le modèle est structuré en deux couches de RNN de type LSTM de 32 cellules chacune, suivies d'une couche dense pour la prédiction finale. Le choix de 32 cellules est ici déterminé empiriquement.

TABLE 2.5 – Modèle texte.

Couches	(Entrée, Sortie) x taille
Embedding	(860,100) x 1
LSTM	(100,32) x 2
FC	(32,2) x 1

2.7 Fusion tardive

Deux stratégies de fusion sont implémentées, la première est basée sur un modèle mathématique avec la théorie de l'évidence [Sha76]. La seconde entraîne une couche de neurones dense. La théorie de l'évidence est adaptée pour modéliser la confiance de différents canaux. Nous choisissons de l'implémenter car elle permet de combiner facilement différentes sources d'information. La fusion pilotée par l'apprentissage machine, utilise une couche entièrement connectée qui nécessite peu de ressources de calcul.

2.7.1 Fusion par la théorie de l'évidence

Dempster Shafer Theory (DST), ou théorie de l'évidence, combine les preuves d'informations provenant de plusieurs événements pour calculer la croyance de l'occurrence d'un autre événement. Soit $\Theta = \{X_0, X_1, ..., X_n\}$ un ensemble fini appelé cadre de discernement. 2^Θ fait référence à chaque sous-ensemble mutuellement exclusif possible des éléments de Θ.

Chaque sous-ensemble reçoit une valeur de croyance dans $[0, 1]$. Dans cette modélisation, l'incertitude est estimée sur la base de la métrique du rappel.

La probabilité de masse, désignée par $m(X)$, est utilisée pour attribuer une preuve à une modalité donnée X :

$$0 \leq m(X_i) \leq 1, \quad \sum_{X \subseteq \Theta} m(X) = 1, \quad m(\emptyset) = 0 \tag{2.2}$$

Dans notre cas, nous avons trois probabilités de masse $m_V(X)$, $m_A(X)$, $m_T(X)$, une pour chaque modalité, respectivement pour la modalité vidéo, audio et texte. Chaque modèle produit un nombre de probabilités égal au nombre de classe. Nous calculons également le score de rappel de chaque modèle modélisant ici le terme d'incertitude de la DST. Le rappel mesure le pourcentage d'échantillons positifs correctement classés.

Avec tous ces éléments, nous pouvons calculer la fusion DST. La masse conjointe vidéo/texte est :

$$k_{V,T} = \sum_{X_i \cap X_j = \emptyset} m_V(X_i) \times m_T(X_j) \tag{2.3}$$

$$m_{VT}(Z) = \frac{1}{1 - k_{V,T}} \sum_{Xi \cap Xj = Z} m_V(Xi) \times m_T(X_j) \tag{2.4}$$

La masse commune vidéo/texte/audio est :

$$k_{VT,A} = \sum_{X_i \cap X_j = \emptyset} m_{VT}(X_i) \times m_A(X_j) \tag{2.5}$$

$$m_{VT,A}(Z) = \frac{1}{1 - k_{VT,A}} \sum_{X_i \cap X_j = Z} m_{VT}(X_i) \times m_A(Xj) \tag{2.6}$$

avec m_x des matrices de taille 3×1 et $x = V, T, A, VT$. Les 2 premières colonnes représentent les probabilités de la classe négative et de la classe positive. La dernière colonne est l'incertitude. Xi représente les éléments de m_x.

Pour obtenir la performance finale, le F1 score est calculé, nous utilisons l'indice de la valeur maximale des 2 premières colonnes. L'indice renvoie la prédiction de l'étiquette (i.e. 0 ou 1). Ensuite, le score final (F1 score) est calculé en utilisant la prédiction et la vérité du terrain.

Cette stratégie de fusion ne nécessite pas d'entraînement supplémentaire et son intégration est peu coûteuse en termes de calcul. Dans un contexte plus général, l'inconvénient de la DST est le coût en ressources de calcul qui augmente avec le nombre de signaux possibles à fusionner.

2.7.2 Fusion par couche dense avec apprentissage

Les modalités étant indépendantes et n'apportant chacune pas le même niveau d'information, nous cherchons empiriquement le ratio de caractéristiques entre les modalités nous donnant les meilleurs résultats. Les sorties de chaque modèle unimodal sont alors modifiées pour avoir respectivement 32 caractéristiques pour l'audio et la vidéo et 16 caractéristiques pour le texte (voir figure 2.4 pour un schéma détaillé). Ensuite, ces trois vecteurs sont concaténés afin d'obtenir un vecteur final de taille 80. Une couche de *max pooling* à une dimension est appliquée pour obtenir 39 caractéristiques. Le choix de cette opération a pour but d'éviter d'utiliser une couche entièrement connectée FC supplémentaire pour compresser l'entrée. L'opération de maxpooling consiste à sous-échantillonner l'entrée en prenant la valeur maximale sur une fenêtre de taille fixe qui est déplacée sur l'entrée. Pour finir, une (FC) de 78 paramètres (2 classes $\times$ 39) est appliquée pour obtenir la prédiction de sentiment. Nous avons remarqué que plus le modèle est contraint (réduction du nombre de paramètres entre les couches), meilleures sont les performances. Inversement, avoir trop peu de caractéristiques est défavorable. Le concept « d'entonnoir » est très largement utilisé dans le domaine de l'apprentissage profond pour compresser (réduire les dimensions) des données d'entrées [TZ15] et garder uniquement les informations pertinentes.

Cette fusion de par sa compacité est en adéquation avec nos considérations d'embarquabilités.

2.8 Implémentation

Contrairement à [PCH+17], nous ne considérons pas le niveau inter-tours de parole. Un tour de parole est une unité de discours continue commençant et se terminant par une pause explicite. Nous considérons les tours de parole indépendamment les uns des autres. Cette modélisation permet d'utiliser que deux modèles LSTM (1 pour le modèle audio et 1 pour le modèle texte).

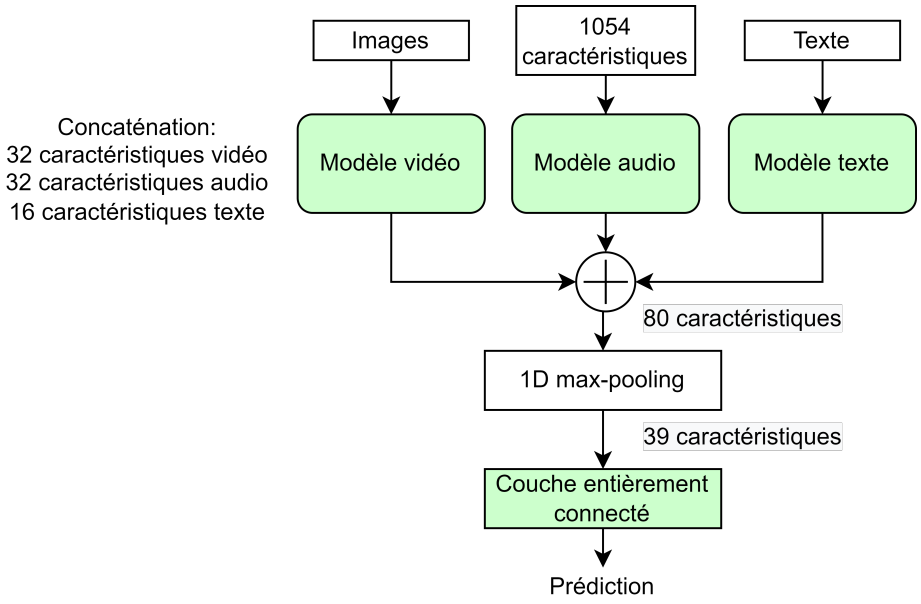

FIGURE 2.4 – Fusion des caractéristiques par une couche entièrement connectée.

2.8.1 Apprentissage par transfert

Pour entraîner le modèle multimodal, nous utilisons l'apprentissage par transfert (voir [PY10] pour plus de détails). À défaut de commencer le processus d'apprentissage à partir de zéro, nous partons d'un modèle qui a appris à résoudre un problème similaire. Cette technique réduit considérablement le temps d'entraînement et les données nécessaires à l'apprentissage. L'apprentissage par transfert comprend deux techniques différentes : le développement de modèles et le pré-entraînement. Ces techniques sont couramment utilisées pour l'apprentissage profond [HZRS15, KSH12, RW17].

Nous mettons en place la méthode dite de pré-entraînement. Elle consiste dans un premier temps à choisir un modèle source pré-entraîné parmi des modèles disponibles (ici les modèles unimodaux). Ensuite, le modèle sélectionné est réutilisé comme point de départ pour un modèle (ici le modèle multimodal) correspondant à la deuxième tâche d'intérêt. Cela peut impliquer l'utilisation de tout ou partie du modèle, en fonction de la technique de modélisation utilisée. Pour finir, le modèle peut être adapté ou affiné en fonction des nouvelles données d'entrées-sorties disponibles pour la tâche concernée. Dans notre cas la tâche est identique entre les modèles unimodaux et le multimodal. Nous utilisons les modèles unimodaux de chacune des modalités à leurs meilleurs points de précision pour entraîner le modèle multimodal final. L'utilisation du pré-entraînement est nécessaire. En effet, la fusion par couche entièrement connectée ne converge pas si nous commençons l'entraînement à partir de zéro. Elle est

composée de 78 paramètres (initialisés aléatoirement) qui contraignent le réseau.

2.8.2 Réglage des hyperparamètres

Nous effectuons une tâche de classification binaire, car dans la littérature les travaux sur le corpus MOSI prédisent majoritairement deux classes. Chaque apprentissage de chaque modèle est alors effectué en utilisant la perte d'entropie croisée comme fonction de coût. Pour la classification binaire, la formule de la perte d'entropie croisée devient :

$$l = -y \log(p) + (1 - y) \log(1 - p) \tag{2.7}$$

avec p la prédiction du réseau et y la vérité terrain associée.

Nous considérons deux optimiseurs différents : la descente de gradient stochastique pour entraîner le modèle vidéo et l'optimiseur Adam [KB17] pour le modèle audio et texte. Empiriquement, nous avons constaté qu'Adam est plus performant pour optimiser des réseaux avec des données d'entrée éparses, ce qui est aussi constaté par [KB17].

Concernant la régularisation, nous utilisons un *dropout* [WWL13] de 0,4 pour le modèle LSTM audio et texte afin de réduire la variance.

Le taux d'apprentissage est choisi avec précision pour chaque modalité et pour le modèle de fusion entièrement connecté. Pour entraîner les modèles unimodaux, le taux d'apprentissage est graduel démarrant de 10^{-3} à 10^{-5}. Comme nous utilisons l'apprentissage par transfert, nous réduisons ce taux pour le modèle multimodal. Par défaut, il démarre à 10^{-4} jusqu'à 10^{-6}, tandis que le taux d'apprentissage de la couche FC est multiplié par dix par rapport au taux d'apprentissage par défaut.

2.8.3 Découpage du corpus en apprentissage et test

Un point clé lorsque nous travaillons sur l'analyse des sentiments, d'émotions ou encore sur l'analyse d'interactions, est la dépendance au locuteur. L'idée est d'évaluer les capacités du modèle à généraliser lorsqu'il découvre un nouveau locuteur. Ainsi, nous décidons qu'en phase de test, le modèle n'a jamais vu le locuteur. Ce fonctionnement est le plus contraignant pour un réseau de neurones. Afin de comparer nos performances avec la littérature, nous avons divisé l'ensemble de données comme [CHP$^+$17] et [PCH$^+$17] : les 62 premières vidéos ($\approx$ 70%) du jeu de données sont utilisées pour l'entraînement et les autres ($\approx$ 30%) sont utilisées pour les tests.

2.9 Potentiel d'embarquabilité

En référence au cahier des charges, nous considérerons les performances potentielles d'embarquer nos travaux. Nous avons ici deux leviers d'action : un

TABLE 2.6 – Détails du corpus MOSI.

	Apprentissage	Tests
Nb de vidéos	62	31
Nb de tours de parole	1447	752
Nb de locuteurs	58	31
Homme	33	15
Femme	25	16
Durée (min)	≈ 85	≈ 50
Nb de phrases	1447	752
Nb de mots	17296	9161

orienté sur la réduction de la complexité des modèles et un sur la phase de prétraitement des données avant ingestion par le modèle. Sur le flux vidéo, en particulier dans le jeu de données MOSI, les images vidéo consécutives représentent des informations redondantes. Pour pallier ce problème, nous réduisons la fréquence d'images de la vidéo. Nous le réduisons d'un facteur 4, 8, 16, 32. Les performances les plus notables sont pour le facteur 8. Nous pouvons voir qualitativement la différence entre un facteur 1 et 8 sur un exemple en figure 2.5.

FIGURE 2.5 – Exemples de réduction de la fréquence d'images. La première ligne représente une vidéo avec des images successives. La deuxième ligne montre la même vidéo réduite avec un facteur 8.

Pour l'analyse audio, nous testons deux techniques afin de réduire les coûts en calculs. Pour la première, nous évitons la dépendance à un extracteur de caractéristiques spécifique. Le signal audio est transformé en images (voir l'état de l'art en section 1.2.4) puis un extracteur de caractéristiques de type CNN 2D suivi d'une couche dense effectuent la classification. Les résultats ne sont pas probants. La deuxième technique utilise OpenSMILE comme extracteur de caractéristiques puis nous utilisons un modèle LSTM suivi d'une couche dense

pour la classification. En réduisant la matrice d'entrée du LSTM, le coût en calcul est réduit. Après essais et erreurs, nous réduisons la largeur de la matrice d'entrée à seulement 1054 entités (voir la section 2.5 pour la sélection des paramètres).

Les données textuelles représentent la transcription du flux audio. Toutes les phrases représentent un total de 26457 mots (dont 3003 mots uniques). Pour améliorer les performances embarquées, nous filtrons tous les mots. Après quelques expérimentations, nous avons retenu seulement les adjectifs et les adverbes. Cette configuration fournit de meilleures performances que de prendre tous les mots [BCP+07, PR17]. Le filtrage réduit le nombre de mots unique à 860. Un autre paramètre est la longueur des phrases fixée à 30 mots (la valeur est trouvée empiriquement). Le lecteur peut se référer au tableau 2.7 où les dix mots les plus et moins fréquents sont recensés.

Table 2.7 – Fréquence d'apparition des mots dans le corpus.

	Les plus présents	Les moins présents
Mots	really	catastrophic
	good	mid
	whole	oldest
	I	rough
	little	meanwhile
	not	papa
	pretty	guys
	sad	overly
	awesome	upbeat
	funny	relatable

2.10 Évaluations et analyses associées

Dans un premier temps, cette section présente les évaluations et les compare à la littérature. Dans un deuxième temps, nous proposons une analyse qualitative illustrée par quelques résultats sur quelques inférences. Les performances embarquées sont aussi discutées en fin de section.

Toutes les évaluations sont menées avec l'utilisation de la métrique du score F1 défini comme suit :

$$F1_{score} = 2 * \frac{Rappel * Précision}{Rappel + Précision} \tag{2.8}$$

Avec :

$$Précision = \frac{T_p}{T_p + F_p} \tag{2.9}$$

Et :

$$Rappel = \frac{T_p}{T_p + F_n} \tag{2.10}$$

2.10.1 Évaluations quantitatives

Comme nous pouvons le voir dans le tableau 2.8, chaque modalité ne porte pas la même quantité d'informations. La vidéo est légèrement au-dessus de l'aléatoire avec un score F1 de 57%. La modalité audio arrive en deuxième position avec un score F1 de 65,5%. Au final, le texte a le meilleur score F1 avec 77,1%. Nos résultats de fusion sont de 78% pour la fusion DST et 80% pour la fusion FC, montrant respectivement une amélioration de 1% et 3% par rapport aux modèles unimodaux. La fusion FC obtient la meilleure amélioration et les 78 paramètres sont insignifiants au regard des performances embarquées (i.e. augmentation de la charge mémoire et du calcul). Cette solution est donc à privilégier pour un contexte embarqué.

En comparaison à [PCH$^+$17], nous améliorons les performances de classification audio et vidéo (gains de 1,4% et 5,2% respectivement). Par contre, notre modèle texte réalise 1% moins bien. Cette réduction des performances est due au fait que nous entraînons notre propre couche *d'embeddings* sur MOSI à la place d'utiliser celle de Google. Les performances de notre modèle (fusion FC) sont de 0,3% inférieur à [PCH$^+$17] et sont donc au niveau de l'état de l'art.

Nos résultats sont en cohérence avec la littérature sur l'ordre d'importance des modalités (la vidéo est peu informative, l'audio et le texte sont respectivement moyennement et très informatifs) (voir [PCH$^+$17, CHP$^+$17, HSR18]).

Table 2.8 – Comparaison des variantes proposées, en terme de score F1.

Modalité	Source	F1 score (%)
Unimodal	Vidéo	57,2
	Audio	65,5
	Texte	77,1
Unimodal [PCH$^+$17]	Vidéo	55,8
	Audio	60,3
	Texte	78,1
Fusion DST	Vidéo + Audio + Texte	78,0
Fusion FC	**Vidéo + Audio + Texte**	**80,0**
bc-LSTM [PCH$^+$17]	Vidéo + Audio + Texte	80,3

Comparer les performances vis-à-vis de la compacité avec des modèles *benchmarks* est intéressant pour notre contexte applicatif. Nous pouvons de cette manière comparer notre travail avec les performances de [PCH$^+$17] :

— Vidéo : nous réduisons par 3 le nombre de paramètres et par 2 l'utilisation de la mémoire.
— Audio : nous réduisons par 6 le nombre de caractéristiques audio extraites qui alimente le modèle LSTM.
— Texte : nous réduisons de 3,5 Go l'utilisation mémoire du modèle.

— Fusion : notre modèle de fusion ne comprend que 78 paramètres au lieu d'un LSTM bidirectionnel composé de 600 cellules.
— Global : trois LSTM bidirectionnels sont utilisés après chaque modalité pour capturer les relations entre les tours de parole, rajoutant 1800 cellules. Notre système ne les possède pas.

Comparé à [PCH+17] qui est notre référence pour le corpus MOSI (voir table 2.9), nous réduisons de 2,2 le nombre de paramètres et l'utilisation de la mémoire par 13,8.

TABLE 2.9 – Besoin en ressources de calcul pour chaque modèle.

Modèle	Paramètres	Occupation mémoire
R3D	20,78 M	166 MB
LSTM Audio	11,25 M	133,5 MB
LSTM Texte	112 k	1,3 MB
DST fusion	32,14 M	300,8 MB
FC fusion	32,14 M + 78	300,8 MB
bc-LSTM [PCH+17]	≈ 70 M	≈ 4,15 Go

Nous remarquons que la modalité texte est cruciale sur le jeu de données MOSI. La transcription apporte 77% des informations avec seulement 112 k paramètres et 1,3 Mo de mémoire. Notre système conduit à des performances au niveau de l'état de l'art tout en réduisant considérablement les coûts en calcul et le besoin d'espace mémoire.

2.10.2 Évaluations qualitatives

Pour cette analyse, nous récupérons tous les fichiers mal classés pour chaque modalité afin de mieux comprendre les mauvaises classifications. Plusieurs limites sont identifiables dans ce corpus.

Une limite mineure semble venir du fait que le locuteur peut exprimer des sentiments en totale contradiction avec le sentiment du film. Il est difficile pour le système de différencier les sentiments de l'orateur de ceux du film. Par exemple, le sujet raconte calmement : « I love the war scene ». Il est classé comme négatif par le modèle audio et texte. Mais la vérité de terrain est positive. À ce moment, le sentiment du locuteur est positif avec le sentiment « love », mais le sentiment exprimé par le contexte du film peut être interprété comme négatif avec le mot « war ».

Les mauvaises classifications peuvent provenir d'autres éléments, comme, par exemple, les durées du son ou la longueur du texte qui sont parfois extrêmement courtes. L'audio peut également contenir une pause très prolongée ou le texte ne pas contenir assez de mots. Ces deux facteurs sont responsables d'un manque de contexte, en particulier pour les couches du LSTM, induisant une

mauvaise classification. Nous retenons quelques exemples de phrases courtes et peu informatives : « And it would make sense » ou « I wish I weren't ».

Une autre limitation est liée à la neutralité de certains enregistrements ou des mots contenus dans les phrases qui ne donnent pas assez de sens pour être classés correctement. Un exemple de phrase dénuée de sens est : « I would like to quickly talk about Machete ».

Enfin, la dernière limite identifiée dans ce jeu de données est la qualité de l'enregistrement qui peut impacter la classification. Plusieurs enregistrements sont d'une qualité vidéo extrêmement mauvaise et d'une qualité audio critique avec des bruits parasites générés par des webcams anciennes.

2.11 Conclusion

Les performances embarquées des modèles sont souvent omises dans la littérature. D'autant plus pour les systèmes multimodaux qui ont tendance à être très coûteux en calcul. Notre système composé d'un modèle par modalité et d'une fusion tardive conduit, sur le corpus MOSI, à des performances similaires à la littérature, mais avec une compacité élevé. En effet, il réduit de 2,2 le nombre de paramètres et de 13,8 la charge mémoire. Les performances obtenues sur le corpus MOSI démontrent également que la vidéo est moins informative que l'audio et surtout le texte dans notre contexte.

D'une part, ces travaux nous ont familiarisés avec les modalités vidéo, audio et texte. D'autre part, ils nous ont permis d'investiguer la fusion tardive. Cette étude a donné lieu à une publication dans la conférence internationale VISAPP 2020 (cf. page v).

Le chapitre suivant présente le corpus multimodal enregistré sur véhicule Renault pour répondre à notre problématique : étude d'interactions conflictuelles en contexte véhicule.

Chapitre 3

Corpus pour l'analyse d'interactions in situ

Sommaire

3.1 Introduction

De manière générale, les jeux de données publiques permettent de développer de nouveaux modèles d'état de l'art, mais sont limités et souvent inadaptés lorsque les cas d'usages deviennent plus spécifiques. Il existe aujourd'hui une disparité entre les corpus et les modèles de réseaux de neurones présentés dans la littérature scientifique et le monde industriel. Pour pallier ce constat, nous enregistrons un corpus dans notre contexte applicatif. L'objectif de ce chapitre est de détailler le protocole ainsi que les réflexions initiées préalablement afin d'obtenir un corpus exploitable pour l'analyse d'interactions humaines multimodales en contexte véhicule. Il met en lumière les étapes de la définition des scénarios, du matériel d'enregistrement, de la phase d'enregistrement et de la mise en forme des données. La section 2.2.1 recense tous les corpus publics qui se rapprochent de notre étude. Cet exercice montre qu'il n'existe hélas aucun corpus en accès libre correspondant à notre cahier des charges.

3.2 Réflexions préliminaires

Une motivation centrale, pour la création de ce jeu de données, est le réalisme des scénarios joués. Nous souhaitons des données où les acteurs ne surjouent pas leurs émotions ou n'utilisent pas un vocabulaire spécifique. Nous retrouvons majoritairement dans les corpus publics d'analyse d'émotions et de sentiments des scènes non réalistes avec des accentuations sur les expressions faciales par exemple. L'enregistrement d'un corpus est très coûteux en ressources (temps, matériel, disponibilité des équipements industriels, etc.) et reste très chronophage. Plusieurs de nos choix sont effectués afin de respecter au mieux ces contraintes ; ils seront détaillés dans les sections ci-après. Le cahier des charges initial est l'étude des agressions dans les véhicules avec pour objectif de détecter les prémisses et l'évolution des comportements avant d'arriver à la situation extrême, étant ici l'agression. La proactivité est la notion clef pour un tel contexte applicatif. En raison de contraintes protocolaires, nous avons dû écarter cette idée. En effet, lorsque des participants sont placés dans des situations d'agression, un suivi physiologique est obligatoire après leur participation. Certains participants pourraient avoir des souvenirs refoulés qui ressurgissent, nécessitant un entretien psychologique. Nous avons donc opté pour des scénarios de vente forcée qui se rapprochent de notre idée initiale. Ils permettent de générer une gradation dans l'exécution de ces scénarios, gradation que nous cherchons à détecter.

3.2.1 Scénarios

Dans le but de mettre les participants en situation et d'optimiser leur temps de passage, un enchaînement de phases est défini. Plusieurs phases d'action sont jouées séquentiellement générant une vidéo de huit minutes (voir figure 3.1). À la fin du temps imparti, il est demandé au participant d'arrêter.

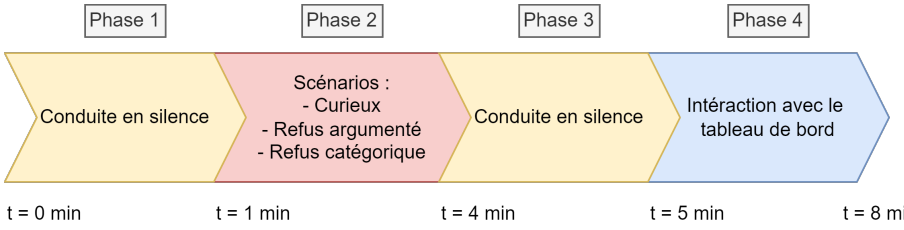

FIGURE 3.1 – Les quatre phases jouées lors de l'enregistrement d'un scénario.

Les phases 1 et 3 de conduite en silence sont ajoutées pour plusieurs raisons. Premièrement, il s'agit de laisser du temps aux participants pour se mettre en situation ainsi que de s'habituer à la diffusion de la vidéo sur l'écran. L'intérêt second est de permettre au conducteur de réfléchir à une idée pour le scénario de vente qu'il va devoir jouer. Finalement, les passages de conduite en silence

permettent de faciliter le découpage des vidéos lors du post-traitement des enregistrements.

La phase 2 est toujours entrecoupée de deux phases de silence. Cette phase où les participants jouent les différents scénarios est celle qui est étudiée tout au long de cette thèse.

L'évolution du cahier des charges nous a conduits aux trois scénarios de vente nommée comme ci-après (i.e les trois classes) :

— **Curieux** : l'objectif est ici de capturer une interaction cordiale,
— **Refus argumenté** : le passager refuse poliment la proposition du conducteur,
— **Refus catégorique** : le passager refuse catégoriquement la proposition du conducteur.

La problématique initiale étant bipartie (détecter des situations conflictuelles ou non), nous pouvons nous questionner sur le besoin de trois classes. Les scénarios et classes sont définis afin que les comportements *i.e.* les percepts des passagers soient implicitement distincts lors du jeu d'acteur des passagers. Des variations de percepts entre classes sont nécessaires afin de permettre la classification. Nous avons choisi trois classes, car nous souhaitons une gradation dans les comportements, définie comme suit :

curieux (comportement nominal)

$\longrightarrow$ **refus argumenté** (légèrement conflictuel)

$\longrightarrow$ **refus catégorique** (conflictuel)

De plus, pour une classification binaire, la crainte est d'avoir des distributions de données ambiguës selon les scénarios à interpréter. Par exemple, nous avons supposé que si la quantité de mots négatifs est équirépartie entre les deux classes les plus opposées, la classification sera compliquée, voire impossible. L'idée est alors de rajouter une troisième classe dont la distribution se situe entre les deux précédentes. Cela permet de forcer implicitement les participants à jouer des variations entre les différents scénarios et d'avoir *a priori* des données séparables.

Le rôle du conducteur est donc primordial pour l'instauration de la tension dans l'habitacle. La gradation apparaîtra naturellement lorsque le passager refusera plus ou moins la proposition et que le conducteur continuera d'insister.

L'équilibre des données entre les différentes classes dans un corpus est un problème majeur pour les réseaux de neurones [CJK04] (voir [RMM18] pour une revue complète sur cette problématique). Celle-ci est inhérente au monde réel, par exemple dans la vie quotidienne nous partons de l'hypothèse que les conflits seront moins représentés que les comportements nominaux. Nous avons essayé de prendre en compte cet aspect en ajoutant une dissymétrie dans la quantité de données de chaque classe avec 50% pour la classe **curieux** et 25% pour chacune des deux autres classes. Ces valeurs ont vocation à montrer l'apparition des effets de déséquilibre dans nos résultats sans augmenter considérablement la difficulté de classification.

La phase 4, appelée « interaction avec le tableau de bord » est destinée à l'équipe Interface Homme Machine (HMI) de Renault. Elle consiste à avoir une interaction (allumer/éteindre la climatisation, changer la température, etc.) avec l'écran du tableau de bord tout en conservant la tâche cognitive. Cette dernière consiste à simuler une tâche de conduite en regardant une vidéo de roulage en vue première personne jouée sur l'écran posé sur le capot de la voiture. Des données véhicule de type Controller Area Network (CAN) [CT09] sont enregistrées. Tous les véhicules sont aujourd'hui équipés d'un bus CAN, il permet de faire transiter toutes les données entre les capteurs et les calculateurs. Son atout principal réside dans sa capacité à gérer les erreurs de transmission (erreurs bits, perte de données, etc.) qui peuvent survenir dans des environs hostiles comme la voiture. Pour Renault, l'intérêt est par exemple de déterminer au bout de combien d'appuis sur l'écran et la durée mise par le conducteur pour parvenir à ce réglage. Cette phase 4 est destinée aux autres équipes R&D Renault et n'est pas étudié dans cette thèse.

Au final, nous avons deux minutes de conduite en silence, trois minutes de scénario ainsi que trois minutes d'interaction avec le tableau de bord.

3.2.2 Protocole d'enregistrement

Tout d'abord, nous fixons à deux le nombre de personnes au sein du véhicule : un conducteur et un passager arrière assis sur le siège de droite (configuration Uber). Il s'agit de notre cas nominal que nous pourrons étendre dans de futurs travaux à davantage de personnes. Concernant les phases de scénarios, afin d'instaurer le conflit, une unique consigne a été donnée au conducteur : « jouer en permanence un scénario de vente insistant ». Le conducteur doit être force de proposition même si le passager refuse afin d'instaurer une tension dans la situation. Ces indications nous amènent à un vocabulaire, des intonations et des variations vocales plus naturelles. Avant chaque enregistrement, nous sollicitons le conducteur pour savoir s'il possède un objet ou un service à vendre. Si ce n'est pas le cas, nous lui en suggérons un parmi les trois variantes suivantes :

- une proposition de vente d'une télévision, d'un smartphone ou autre,
- une proposition de payer son trajet de covoiturage, sans passer par l'application (Blablacar, Uber, etc.),
- une proposition de changer d'itinéraire.

Le passager arrière suit la consigne qui lui est donnée. Il joue une des trois classes (**curieux**, **refus argumenté** et **refus catégorique**). Nous avons ainsi le passager qui impose la situation et le conducteur qui subit le scénario. Afin de limiter les biais, nous mettons en place un protocole en double aveugle. Le conducteur ne connaît pas le scénario joué par le passager arrière et inversement le passager arrière ne connaît pas le rôle du conducteur. Pour ce faire, les rôles sont indiqués à chacun sur des papiers avec les instructions associées.

3.2.3 Annotation des données

L'annotation est une étape très chronophage. Pour la réduire au maximum, nous annotons la vidéo de manière globale, comparée à d'autres corpus où les annotations sont faites au niveau du *tour de parole* [ZZPM16] (voir schéma 3.2). Le scénario est donné aux participants avant le début de l'enregistrement, le label de toute la vidéo est défini à cet instant avec un nom de fichier unique. Ce choix a pour conséquence de générer des étiquettes erronées si les passagers jouent mal leur rôle. Nous reviendrons sur ces limites par la suite.

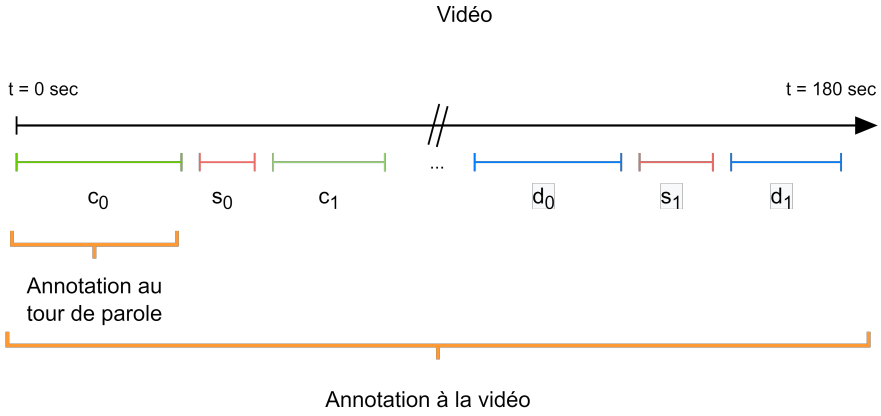

FIGURE 3.2 – Différence entre annotation globale de la vidéo et annotation au niveau du tour de parole. c_x dénote les tours de parole du conducteur, d_x ceux du passager arrière et s_x les silences.

3.2.4 Taille du jeu de données

Ne connaissant pas la quantité de données *a priori* nécessaire, nous avons fixé comme objectif d'enregistrer une quantité de données similaire aux corpus publics MOSI que nous avons étudié par ailleurs (voir chapitre 2.2.1). En effet, lorsque nous travaillons sur des réseaux de neurones profonds il est peu intuitif de connaître *a priori* la quantité de données nécessaire pour faire converger un modèle. De manière générale, plus la tâche à effectuer est complexe, plus il faudra collecter un grand nombre de données. Nous avons estimé à 20 le nombre de sujets potentiels et fixé leur temps de passage dans le véhicule à 32 minutes par paires de participants. Cela permet de ne pas monopoliser trop longtemps les volontaires, mais aussi de ne pas les épuiser. En effet, suite à plusieurs tests préliminaires, des durées supérieures à huit minutes ont été jugées trop fatigantes pour les participants. Chaque binôme joue une fois le scénario **curieux** et une fois le scénario **refus argumenté** ou **refus catégorique** générant respectivement 16 minutes de scénario **curieux**, huit minutes de scénario **refus argumenté** et huit minutes de **refus catégorique**.

À cette étape, le protocole d'enregistrement est entièrement défini, nous allons décrire dans les prochaines sections l'aspect technique de l'enregistrement. Plus précisément, nous présenterons le matériel utilisé ainsi que le prétraitement des données.

3.3 Plateforme sensorielle

Ne connaissant pas à l'avance l'angle de vue et les emplacements microphones les plus pertinents nous avons opté pour un dispositif d'enregistrement multi-capteurs (voir figure 3.3). Il est composé de six caméras, quatre microphones et d'un écran posé sur le capot d'un Renault Dacia Duster. Pour des raisons de sécurité, d'organisation et de simplification protocolaire, la voiture dans laquelle les scénarios sont joués est à l'arrêt. La question du réalisme de la situation est alors soulevée. À l'arrêt, le conducteur n'a plus la tâche cognitive de conduite, nous essayons d'obvier au problème en ajoutant un grand écran d'ordinateur sur le capot de la voiture. Il diffuse une vidéo de roulage en vue première personne et permet également de donner les indications de changements de phase (voir figure 3.1) aux passagers. L'écran est placé en face du conducteur et est également visible par le passager. Toutes les interactions avec la voiture sont autorisées (volant, commodos, levier de vitesse, etc.).

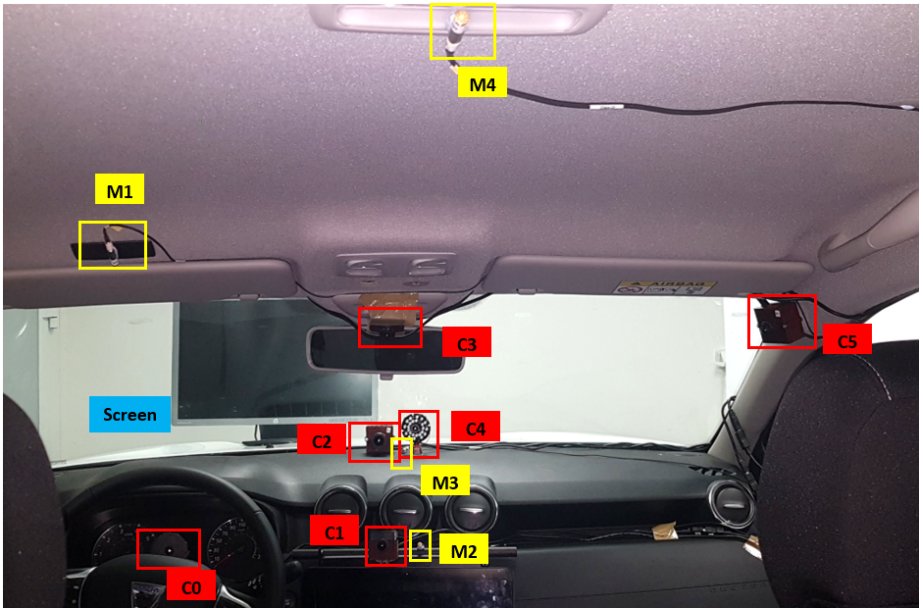

FIGURE 3.3 – Vue intérieure de l'habitacle de la voiture avec le matériel d'enregistrement et les capteurs déployés.

3.3.1 Flux vidéo

Trois types de caméras sont déployés (voir le tableau 3.1 pour les détails exhaustifs de ces dernières). Elles diffèrent par leur résolution, leur angle de vue et leur focale. Toutes les caméras enregistrent à 25 images par seconde.

TABLE 3.1 – Spécifications des 6 caméras (voir figure 3.3).

ID	Type	Résolution	Angles de vues
C0	Infrarouge	640×480	Orientation sur le visage du conducteur
C1	Fisheye	960x720	Vision frontale de l'habitacle
C2	Focus manuel	1920×1080	Vision sur le conducteur et le passager
C3	Fisheye	960x720	Vision plafonnier de l'habitacle
C4	Infrarouge	640×480	Vision sur le conducteur et le passager
C5	Fisheye	960x720	Vision latérale de l'habitacle

Nos travaux privilégient la caméra C2 car elle présente la meilleure qualité d'image et d'éclairage (voir figure 3.4). Il s'agit d'une caméra avec mise au point manuelle et de résolution 1920 × 1080 pixels. Elle est positionnée de manière à avoir un angle de vue frontal (voir figure 3.3). Les données issues des autres caméras ne sont pas exploitées dans cette thèse.

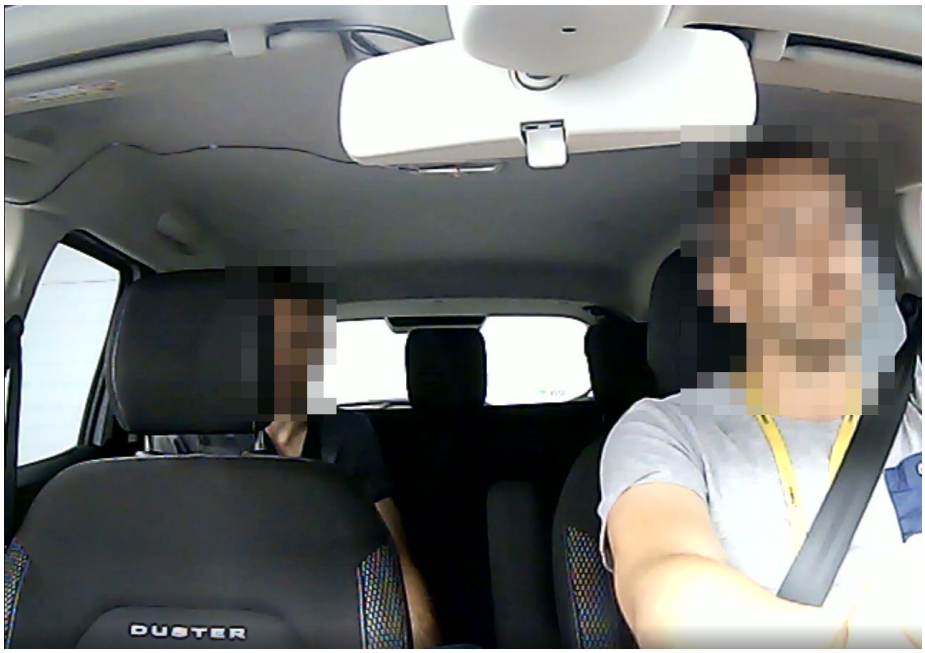

FIGURE 3.4 – Champ de vue de la caméra C2.

3.3.2 Flux audio

Quatre microphones identiques sont placés dans différentes zones de l'habitacle. Le tableau 3.2 récapitule leurs emplacements. Le flux audio est enregistré à une fréquence d'échantillonnage de 44 kHz en mono, 32 bits.

TABLE 3.2 – Spécifications des quatre microphones (voir Figure 3.3).

ID	Modèle	Emplacement
M1	Brüel&Kjaer prepolarized 1/4 type 4958	Plafonnier du conducteur
M2		Face au conducteur
M3		Sous le pare-brise
M4		Plafonnier du passager

Les travaux réalisés dans le cadre de cette thèse utilisent uniquement le microphone plafonnier du conducteur (M1), car il s'agit de l'emplacement des microphones dans les voitures Renault. Le rectangle noir visible sous le microphone M1 dans l'image 3.3 est le microphone appartenant à la voiture. Cette zone avec celle du rétroviseur intérieur sont privilégiées par les constructeurs automobiles de nos jours.

Chaque flux est sauvegardé au format brut (pas de compression en direct) pour ne pas perdre en qualité. Ce choix a aussi pour objectif de réduire le coût *Central Processing Unit* (CPU) afin de pouvoir enregistrer simultanément les 6 flux vidéo et les 4 flux audio.

Le schéma fonctionnel de la plateforme d'enregistrement est résumé dans la figure 3.5. Il représente les bus de connexions entre les différents composants matériels permettant l'enregistrement. Les quatre microphones se connectent sur un amplificateur pour ensuite être numérisés par la carte son reliée en USB à l'ordinateur. Pour les caméras, chacune d'elle est reliée à un port USB de l'ordinateur.

3.4 Préparation au stockage du jeu de données

La non-compression en direct des données génère un décalage temporel variable entre les flux audio et vidéo. Au début de chaque enregistrement, nous demandons au conducteur de faire un « clap » de synchronisation avec les mains. La première étape de post-traitement est donc de synchroniser ces flux grâce à Adobe Première Pro [1]. Ensuite, les premières secondes de début et fin de vidéo sont supprimées pour enlever les éléments indésirables générés au démarrage et à la fin du processus d'enregistrement. Finalement, les vidéos sont exportées au format MPEG-4.

Pour obtenir le texte, le flux audio est transcrit. La première approche basée sur l'utilisation d'une transcription automatique de la parole (Automatic Speech

1. `https://www.adobe.com/fr/`

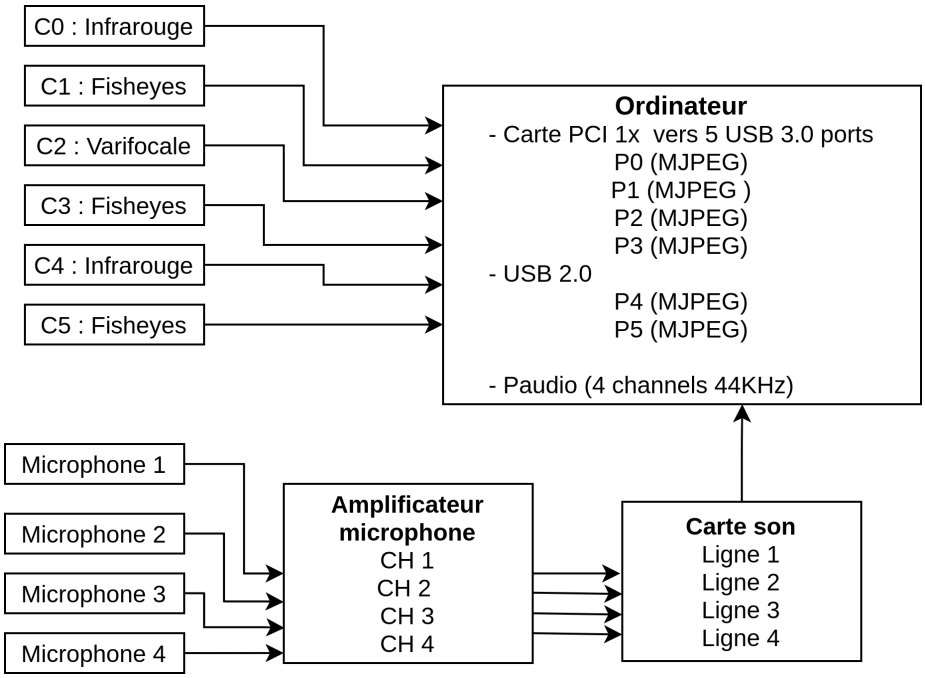

FIGURE 3.5 – Schéma fonctionnel de la plateforme d'enregistrement.

Transcription, ASR) est abandonnée. En effet, la spécificité du contexte oral (répétitions, interjections et mots isolés) ainsi que la mauvaise construction des phrases (sujet-verbe-complément) limitent les performances des ASR actuels. Le texte est finalement transcrit manuellement. Nous utilisons le logiciel ELAN [2]. Il s'agit d'un logiciel libre d'annotation manuelle conçu pour créer, éditer et visualiser des annotations textuelles pour des données vidéo et audio. Nous transcrivons le flux audio de chaque tour de parole en texte, ce qui donne un total de 2026 *tours de parole*.

Afin de simplifier les démarches de conformités avec les normes RGPD en vigueur, le corpus est sauvegardé uniquement sur un ordinateur local et l'accès est restreint aux personnes impliquées dans la thèse.

3.5 Informations intrinsèques du corpus

Le jeu de données se compose de 22 participants (4 femmes et 18 hommes). Chacun d'eux a joué une fois en tant que conducteur et une fois en tant que passager, générant un total de 44 vidéos. L'ensemble des interactions donne 2026 tours de parole, représentant environ 22k mots. Le nombre de mots uniques est de 2082. Après traitement de ces données brutes, ce sont 1h48 de données vidéo

2. `https://archive.mpi.nl/tla/elan`

qui sont enregistrées, soit 54 min pour la classe **curieux**, 27 min pour la classe **refus argumenté** et 27 min pour la classe **refus catégorique**.

Nous comparons dans le tableau 3.3 ces caractéristiques avec celles du corpus MOSI sur lequel nous avons déjà travaillé (voir chapitre 2). Cette comparaison permet de nous donner une intuition générale sur les possibilités du jeu de données Renault.

TABLE 3.3 – Comparaison des corpus MOSI et Renault.

	MOSI	Renault
Durée	2h15	1h48
Nb de vidéos	93	44
Nb de locuteurs	89	22
Nb de tours de parole	2199	2026
Nb de mots	26 457	22 000
Nb de mots uniques	3003	2022
Interactions	Non	Oui
Contexte véhicule	Non	Oui

3.6 Analyse du corpus Renault

Nous cherchons ici à tracer des courbes de descripteurs (caractéristiques) locaux basées sur des moyennes, fréquences, etc. pour déduire des comportements spécifiques dans chacune des classes. Comme les humains ne changent pas leurs émotions ou leurs comportements toutes les secondes, nous avons effectué des analyses sur des fenêtres de 10, 15 et 20 secondes. Ces valeurs nous donnent les courbes les plus exploitables, avec des valeurs supérieures à 20s nous obtenons des courbes plates et lissées, en dessous 10s il y a trop de variation et de bruit, ne permettant pas une analyse. Notre première intuition pour identifier ces descripteurs est inspirée de [BPFAO10].

3.6.1 Analyse du flux audio

L'analyse des données sonores (voir figures 3.6 et 3.7) permet de déduire les caractéristiques suivantes :

- **la durée moyenne d'une interaction -** Dans une conversation normale, les durées de prise de parole tendent à être équiréparties entre les participants.
- **la durée moyenne du temps de parole -** Il s'agit de la longueur moyenne d'un *tour de parole*. Dans un contexte d'interaction, la durée d'un discours est un bon indicateur de qui mène la conversation et de qui veut la clore.
- **le silence moyen -** C'est un indicateur de l'intensité d'un dialogue. Plus il y a de silence, plus la discussion est pauvre et tend vers une situation

de refus.

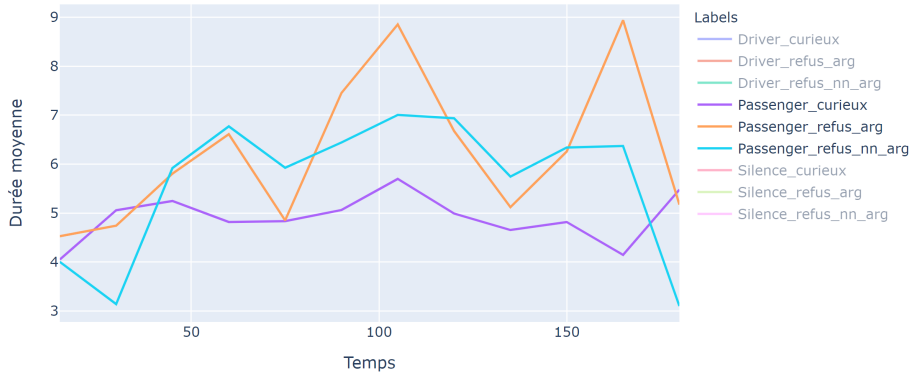

FIGURE 3.6 – Durée moyenne du temps de parole (en seconde) pour le passager arrière.

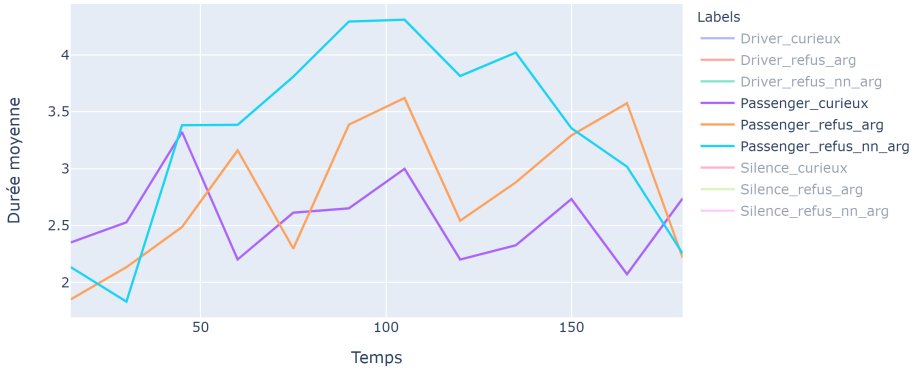

FIGURE 3.7 – Durée moyenne d'une interaction (en seconde).

Les deux premiers paramètres sont communs au conducteur et au passager arrière tandis que le silence est commun aux deux.

Pour rappel, c'est le passager qui impose le scénario. Les courbes complémentaires pour chaque classe et le passager sont accessibles en annexe. Nous détaillons maintenant le calcul de ces caractéristiques en se basant sur la figure 3.8.

La durée moyenne d'une interaction - Nous calculons la somme des j_i (par classe) dans l'intervalle de 15s pour chaque vidéo puis cette valeur est divisée par le nombre de vidéos :

$$\text{Durée d'une interaction moyenne}_{t=15} = \frac{\sum_{V=0}^{43} \sum_{i=0}^{n} V j_i}{44} \tag{3.1}$$

où j_i est la durée du tour de parole ou de silence, avec $j \in (c, s, d)$ et i un tour de parole/silence. n est le nombre de tours de paroles dans la fenêtre de 15s.

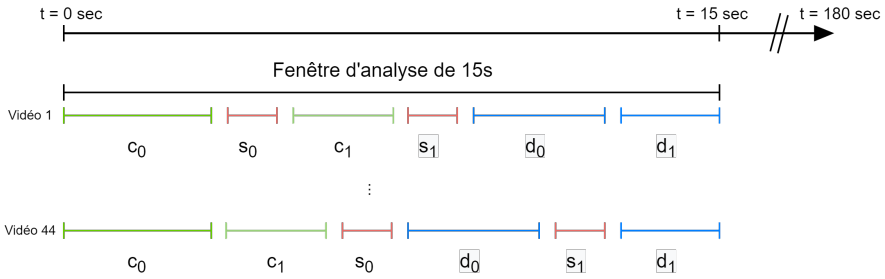

FIGURE 3.8 – Explication du calcul des caractéristiques. La durée d'un tour de parole/silence i est définie par : j_i avec $j \in (c, s, d)$.

Durée moyenne du temps de parole ou de silence - les j_i sont additionnées (par classes) puis divisées par le nombre de fenêtres sur cet intervalle pour chaque vidéo. Puis, les valeurs obtenues sont additionnées et finalement divisées par le nombre de vidéos, soit :

$$\text{Durée moyenne du temps de parole}_{t=15} = \frac{\sum_{V=0}^{43} \frac{\sum_{i=0}^{n} V j_i}{n}}{44} \tag{3.2}$$

Avec V qui dénote une vidéo.

Les participants n'étant pas de véritables acteurs et leurs dialogues n'étant pas scriptés, nous avons observé sur les vidéos et les courbes deux phases de transitions. La première est la mise en place du scénario : les participants démarrent naturellement la conversation avec un dialogue de courtoisie et de salutation sur les 30 premières secondes. La seconde est celle de la fin : les sujets ont parfois manqué d'inspiration, provoquant un essoufflement des échanges sur les 20 dernières secondes. Cela est confirmé sur les courbes : les temps de prise de parole et les durées des interactions sont anarchiques avec des variations décroissantes puis croissantes pour ensuite croître et se stabiliser après les 30 premières secondes. À partir de 150s environ **la durée moyenne du temps de parole** ainsi que la majorité des courbes tracées tendent à chuter drastiquement.

3.6.2 Analyse du flux vidéo

La modalité vidéo est *a priori* peu informative au regard de notre problématique. Généralement l'étude des émotions ou sentiments nous amène à analyser le visage, car il s'agit de la zone la plus déformable (bouche et yeux). Dans le contexte véhicule tous les percepts sont encore plus atténués d'une part à cause de la tâche cognitive et d'autre part à cause du du mouvement corporel

limité des passagers au sein de l'habitacle, de la ceinture de sécurité, etc. Après l'analyse des vidéos, nous avons remarqué un comportement caractéristique des conducteurs, qui regardent dans le rétroviseur intérieur pour chercher un lien visuel avec leur interlocuteur. Nous appellerons ce percept **contact visuel**. Il est défini comme la fréquence à laquelle le conducteur regarde dans le rétroviseur intérieur. Comme il est concentré sur sa tâche de conduite, il n'a pas d'autre choix que de regarder dans le rétroviseur pour interagir visuellement avec son interlocuteur. Concrètement, pour extraire ce percept, nous utilisons Dlib [Kin09] comme détecteur de visages, suivi de l'extracteur des angles d'orientation du visage via hyperface [RPC19a]. Enfin, l'algorithme de k-means [Llo82] appliqué sur les axes *Yaw* et *Pitch* permet de déduire la paire d'angles d'Euler lorsque le conducteur regarde dans le rétroviseur intérieur (couleur verte sur la figure 3.9). L'axe des abscisses représente les mouvements horizontaux et les ordonnées représentent les mouvements verticaux de la tête. En fixant empiriquement le nombre de clusters à trois, nous obtenons le meilleur découpage permettant d'extraire les regards dans le rétro intérieur. Si le nombre de clusters augmente, les ensembles bleu et rouge sont subdivisés. S'il est à deux, le cluster bleu et vert s'étend chacun jusqu'à la moitié du cluster rouge, le cluster vert n'est alors plus représentatif du regard rétroviseur intérieur. L'axe *Tilt* n'est pas informatif dans le contexte véhicule.

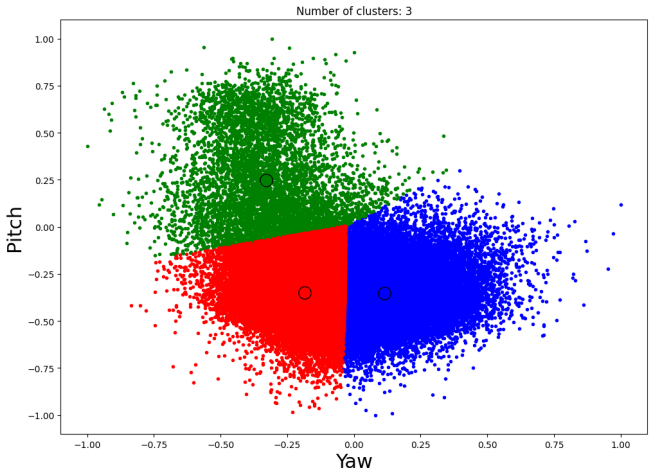

FIGURE 3.9 – Clustering des angles d'orientation de la tête du conducteur.

Nous avons aussi observé un second comportement caractéristique propre au passager arrière. Ce dernier a l'habitude de se rapprocher entre les deux sièges avant, lorsqu'il est engagé dans la conversation, en situation inverse il se retranche dans son siège. C'est un bon indicateur pour savoir si le passager est intéressé par la conversation. Nous réduisons naturellement la distance avec

notre interlocuteur lorsque nous sommes engagés dans une discussion. Dans le cockpit, le passager arrière se balance (ou non) entre les deux banquettes avant-arrière. Ces balancements corporels sont alors observés par la caméra. Pour capturer cette information, nous utilisons à nouveau Dlib pour détecter le visage du passager arrière, sur chaque image. Il s'agit d'une caractéristique binaire que nous appellerons **visibilité passager**, soit nous voyons le passager, soit non. Nous ne cherchons pas à déterminer la taille du visage. Connaissant la cadence vidéo, nous convertissons ensuite cette information en temps.

La visualisation de ces deux percepts en fonction du temps sont disponibles en annexes avec les tracés 5.15 et 5.14.

Ensuite, nous synchronisons au niveau du *tour de parole* les paramètres audio et vidéo, car le système est alimenté par des flux synchrones.

Pour conforter les choix faits précédemment nous calculons la matrice de corrélation de Pearson (équation (3.3)) pour toutes les caractéristiques susmentionnées avec pour objectif de mettre en évidence des corrélations linéaires entre les paires $(X,\ Y)$ de caractéristiques.

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{3.3}$$

où σ désigne l'écart-type, μ la moyenne et $\mathbb{E}$ l'espérance mathématique.

Les tableaux 3.4 et 3.5 exhibent des corrélations évidentes entre les caractéristiques sonores et visuelles. La première est l'augmentation du contact visuel entre le conducteur et le passager avec l'augmentation de la visibilité du passager. La seconde est la moyenne du silence qui diminue avec les contacts visuels. Ces corrélations confirment l'existence d'un lien entre la vidéo et l'audio dans l'interaction entre passager et conducteur pour les scénarios ciblés.

Les acronymes suivants sont définis pour les cinq caractéristiques :

— Msp désigne la durée moyenne des interactions pour le conducteur et le passager,
— Mdur la durée moyenne du temps de parole pour le conducteur et le passager,
— EyeC le contact visuel,
— Msil la moyenne de silence,
— Pvisi la visibilité du passager.

TABLE 3.4 – Corrélations de Pearson pour le conducteur.

Conducteur	Msp	Mdur	EyeC	Msil	Pvisi
Msp	1	-	-	-	-
Mdur	0.5	1	-	-	-
EyeC	0.37	-0.03	1	-	-
Msil	-0.56	-0.21	**-0.37**	1	-
Pvisi	0.4	0.07	**0.84**	-0.23	1

TABLE 3.5 – Corrélations de Pearson pour le passager.

Passager	Msp	Mdur	EyeC	Msil	Pvisi
Msp	1	-	-	-	-
Mdur	0.74	1	-	-	-
EyeC	-0.14	-0.22	1	-	-
Msil	0.22	0.19	**-0.37**	1	-
Pvisi	-0.21	-0.18	**0.84**	-0.23	1

3.6.3 Analyse du flux texte

Concernant la modalité texte, nous traçons la fréquence des mots et le tf-idf (Term Frequency-Inverse Document Frequency, [SJ88]) pour tenter d'exhiber des fréquences ou des distributions spécifiques de mots propres à chaque scénario. Cette méthodologie est courante dans l'exploration et l'analyse de textes. Nous calculons aussi le delta tf-idf (1-gram) absolu entre les deux classes opposées (i.e. **curieux** et **refus catégorique**) et traçons le graphique 3.10.

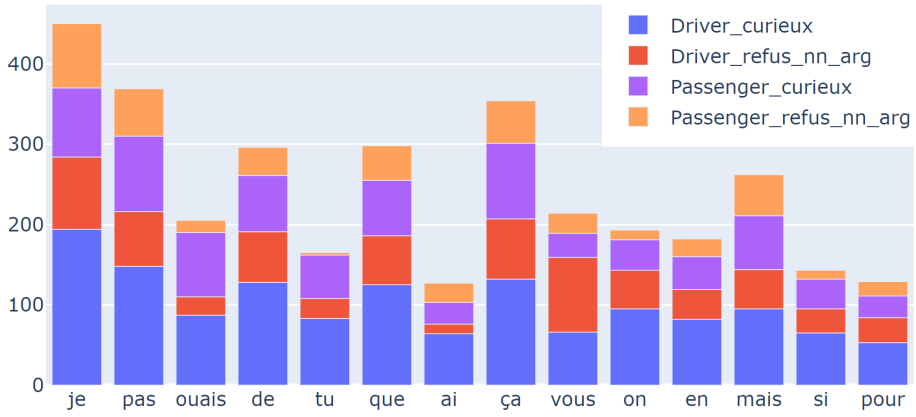

FIGURE 3.10 – 14 mots les plus représentés du corpus par ordre décroissant en partant de la gauche.

Ce tracé permet de mettre en évidence les mots les plus opposés entre les deux classes les plus distantes. Pour chaque barre du graphique, nous avons quatre couleurs représentant chaque passager dans leur classe respective. Le conducteur est représenté par les couleurs bleu et rouge respectivement pour les classes **curieux** et **refus catégorique**. Le passager correspond à la couleur violette pour la classe **curieux** et orange pour la classe **refus argumenté**. Si nous nous concentrons sur les apparitions du mot « je » dans la classe **curieux**, il est utilisé au total 194 fois contre 90 fois dans la classe opposée. Pour le passager nous avons 86 utilisations pour la classe **curieux** et 80 pour la classe antagonisme. En faisant de même pour le mot « vous », nous pouvons *a priori*

supposer que le conducteur ajoute de la distance dans son discours en passant du pronom personnel « je » dans la classe **curieux** à « vous » dans la classe **refus catégorique**. Le changement est moins flagrant pour le passager arrière. Ce type de supposition est difficile à trouver. La modalité texte n'est hélas pas riche, avec seulement 2082 mots différents. Ces constatations ont été identifiées intuitivement, une analyse exhaustive basée sur de l'apprentissage machine est alors plus pertinente. Un modèle de bout-en-bout sera étudié dans le chapitre suivant.

3.7 Conclusion

Dans ce chapitre, nous avons présenté les étapes permettant l'enregistrement d'un corpus en contexte véhicule pour la problématique de détection de situation conflictuelle. Les réflexions initiales ont permis de mener à bien cette campagne d'enregistrement en obtenant un corpus vidéo de 1h48, exercice certes incontournable eu égard au cahier des charges Renault mais hélas fastidieux et chronophage. Concernant l'utilisation des différents flux vidéo et audio, nous avons défini un fonctionnement nominal avec deux passagers dans l'habitable et l'utilisation d'une caméra et d'un microphone parmi ceux disponibles afin de montrer la faisabilité. Ce fonctionnement et processus d'annotation peuvent ensuite être étendus à plus de deux personnes dans l'habitacle. Le choix de la caméra ($C2$) et du microphone ($M1$) peut être modifié ou complété pour améliorer les performances, faire de l'augmentation de données ou faire de la fusion multi-capteurs, etc.

L'analyse statistique met en évidence l'existence de caractéristiques dans chacune des trois modalités pour les scénarios ciblés, et ainsi valide nos intuitions. Nous avons identifié un total de sept caractéristiques. Cinq caractéristiques (deux par passager et une commune aux deux) capturent les évolutions de comportements dans l'audio. Certes, la modalité vidéo est un flux vidéo peu informatif dans notre contexte applicatif. Néanmoins, nous avons mis en évidence une caractéristique par passager capturant des percepts visuels. La modalité texte montre aussi l'existence de caractéristiques entre les différentes classes permettant l'utilisation de techniques de classification.

Le corpus est limité en taille, car la plateforme sensorielle installée sur le DACIA Duster a hélas dû être démontée suite aux acquisitions. En effet, le véhicule est partagé entre les différentes équipes présentes sur le site de Renault Software Labs et son utilisation est administrée par des plages horaires/journalières à réserver.

Sur la base de ce corpus et des caractéristiques audio/vidéo/texte sous-jacentes identifiées, le chapitre suivant décrit nos diverses architectures neuronales pour la classification de nos trois scénarios.

Chapitre 4

Analyse d'interactions humaines dans l'habitacle

Sommaire

4.1 Introduction

Ce chapitre, central dans ce mémoire, porte sur l'analyse multimodale des interactions humaines pour la détection de situations conflictuelles sur notre corpus Renault. En nous appuyant sur les connaissances décrites dans le chapitre 2 et sur notre corpus multimodal (voir chapitre 3), nous souhaitons montrer la faisabilité de la prédiction de telles situations tout en montrant les gains d'une fusion multimodale sur les performances de cette prédiction. La compacité des modèles est aussi au centre de notre attention lors du développement de nos différents systèmes. Nous proposons ici de développer deux modèles. Un premier dit de bout-en-bout (*end-to-end*) dénoté (BB) ingérant les trois flux de données (audio, vidéo et texte) bruts. Puis, un second modèle est proposé, avec pour objectif d'améliorer la compacité. Ce modèle dénoté (BB+P) utilise des caractéristiques (descripteurs) statistiques extraites manuellement, car intuitives, pour les modalités audio et vidéo. La finalité étant de pouvoir prédire

les trois classes issues de notre corpus : **curieux**, **refus argumenté** et **refus catégorique**. Des évaluations, quantitatives et qualitatives, sont aussi menées. Nous concluons ce chapitre par une étude comparative des performances de compacité de nos deux chaînes de traitements.

4.2 Analyse de bout-en-bout

L'extraction d'informations d'un jeu de données peut être réalisée de deux manières. La plus commune consiste à calculer intuitivement des paramètres statistiques (moyennes, écarts-types, quartiles, fréquences, percentiles, etc.) sur lesquels des algorithmes de classification (ou de *clustering*) sont appliqués (SVM, forêt d'arbre aléatoire, régression, etc.). Ces caractéristiques ont un sens et sont humainement interprétables. *A contrario*, nous avons les modèles dit de bout-en-bout (*end-to-end*) qui, pour des données brutes (issues de capteurs), sont capables d'extraire l'information pertinente pour accomplir différentes tâches (classification, régression, génération, etc.). Ces techniques sont moins explicables, car boîte noire (voir [XUD+19] pour plus de détails). Cette section vise à présenter un modèle capable d'éviter la recherche manuelle de caractéristiques en donnant les données brutes en entrée du modèle afin d'obtenir en sortie une prédiction. Il doit donc être capable de trouver implicitement les caractéristiques importantes dans le texte, l'audio et la vidéo grâce au processus d'apprentissage.

4.2.1 Analyse du texte

Les analyses effectuées dans la section 3.6 nous orientent vers l'utilisation de modèles de bout-en-bout pour la modalité texte. Cependant, cette approche nous confronte à plusieurs challenges.

Le premier est relatif à l'usage de la langue française. En effet, tous les systèmes et modèles pré-entraînés tels que Spacy [1] [HJ15], NLTK [2] [LB02] et BERT [3] [DCLT19] sont adaptés à l'analyse de l'anglais, mais sont peu performants sur la langue française. Les alternatives existantes pour la langue française sont limitées à des modèles entraînés sur de l'ancien français ou du français écrit. Nous obtenons donc de très mauvais résultats sur le modèle *Transformer* nommé CamenBERT [MMOS+20] qui est entraîné sur 139 Go de texte Wikipédia.

Le second est la faible richesse du texte de notre corpus qui rend inefficaces deux topologies : tf-idf [SJ88] ou *embedding*, combiné avec un modèle récurrent seul [WPZ18]).

Ainsi, nous privilégions le réseau d'attention hiérarchique (HAN) représenté par le tableau 4.1. Celui-ci est initialement conçu pour classer des documents textes [YYD+16]. Dans des données texte, au niveau micro, les mots d'une phrase ne contribuent pas tous avec la même importance pour la compréhension de la phrase. Il en est de même au niveau macro, les phrases ne participent pas

1. Industrial-Strength Natural Language Processing
2. Natural Language Toolkit
3. Bidirectional Encoder Representations from Transformers

toutes avec le même poids à la compréhension du document. Nous avons donc choisi ce modèle, car il a la capacité d'encoder les mots (embedding + GRU bidirectionnel sur les mots) et de déterminer les plus importants (une attention sur les mots). Ensuite la représentation des mots est envoyée à la couche suivante qui va encoder la phrase (GRU bidirectionnel sur les phrases) et pour finir déterminer l'importance (une attention sur les phrases) de chacune d'elles dans le document. Le double mécanisme d'attention permet de se concentrer à la fois sur l'importance des mots et des phrases.

Table 4.1 – Définition des couches du modèle HAN.

Couches	(Entrée, Sortie) x taille
Embedding	(500,100) x 1
GRU bidirectionnel (mots)	(100,32) x 16
Attention (mots)	(32,32) x 1
GRU bidirectionnel (phrases)	(32,32) x 16
Attention (phrases)	(32,32) x 1
Dense	(32,3) x 1

Son implémentation originale est modifiée en remplaçant la couche GRU analysant les phrases par une *stateful* GRU. Cette modification permet au modèle de garder un historique temporel des phrases précédentes. Avant de pouvoir envoyer des données textuelles à un réseau de neurones, il est nécessaire d'appliquer des transformations. Comme détaillé dans l'état de l'art en section 1.2.5, nous appliquons l'encodage des mots en entier unique qui est ensuite envoyé au HAN intégrant la couche *d'embedding*. Les paramètres de cette dernière convergent durant la phase d'entraînement. Pour rappel, il s'agit d'une couche entièrement connectée. Elle est en entrée de la dimension du vocabulaire du corpus et en sortie un vecteur compressé de la dimension que nous souhaitons, généralement entre 100 et 300. Dans notre implémentation la taille du document correspond à la fenêtre d'analyse de 35 secondes. Voici la configuration des hyper-paramètres de notre HAN, fixés empiriquement :

- *l'embedding* est constitué d'un vocabulaire des 500 mots les plus importants du corpus (définis en utilisant la technique du tf-idf décrit en section 3.6.3). La sortie est un vecteur de taille 100, valeur que nous avons divisée par trois, comparativement à l'implémentation originelle, car notre corpus est moins riche.
- 16 cellules pour le GRU des mots et le GRU des phrases. Le modèle originel en possédant 64, nous l'avons réduit pour la même raison.

4.2.2 Analyse audio

Une alternative à l'analyse manuelle développée dans la section 3.6.1 est d'extraire automatiquement des caractéristiques audio. Cette technique a déjà été exploitée dans la section 2.5, relative aux travaux sur le corpus public MOSI,

et donnant des résultats probants. Les paramètres audio sont calculés grâce à la boîte à outils OpenSMILE [EWS10], en utilisant le même fichier de configuration emobase2010 [SSB+10] (1581 caractéristiques) utilisé dans le chapitre 2.5. La paramétrisation s'effectue uniquement sur les tours de parole du passager arrière de plus d'une seconde. Pour rappel, le passager arrière est le seul à connaître le scénario joué avec le conducteur qui, lui, subit les scénarios. Filtrer les tours de parole de moins d'une seconde permet de ne pas prendre en compte les interjections de type : « heu », « ouais », « d'accord » ainsi que les répétitions de mots très présents dans le langage oral. Les caractéristiques sont extraites toutes les 10ms avec une fenêtre glissante de 60ms. Comparé aux travaux du chapitre 2.5 nous ajoutons une étape qui les moyennent sur la longueur du tour de parole. En effet, les variations audio au niveau de la prononciation du mot sont très faibles dans notre corpus (ce qui est commun dans la langue française comparée à la langue anglaise). Les accentuations étant plus perceptibles au niveau du tour de parole [Vai02] dans la langue française, nous les moyennons sur la longueur de cette dernière. Pour pallier au peu d'intonation/accentuation de notre corpus nous gardons ici les 1581 caractéristiques du fichier de configuration d'« emobase2010 » au lieu des 1054 trouvées empiriquement pour l'analyse audio de la section 2.5. Les caractéristiques sont ensuite envoyées séquentiellement à un réseau récurrent de type *stateful* GRU. La matrice qui nourrit le modèle est de la taille du nombre de tours de parole dans les 35 secondes de la fenêtre d'analyse par les 1581 caractéristiques. Le tableau 4.2 ci-après synthétise notre modèle.

TABLE 4.2 – Définition des couches du modèle audio.

Couches	(Entrée, Sortie) x taille
GRU	(1581,1581) x 12
GRU	(1581,1581) x 12
Dense	(12,3) x 1

Le modèle 2.4 est composé d'un empilement de deux couches GRU composées chacune de 12 cellules. Finalement, la couche dense effectue la prédiction. La dimension de la couche dense et du nombre de cellules est déterminée empiriquement.

4.2.3 Analyse vidéo

La modalité vidéo est la moins informative dans notre contexte et l'est aussi dans la littérature pour l'analyse de sentiments et d'émotions [PCH+17, CHP+17, AYV19]. Le contexte véhicule restreint les mouvements du conducteur et ceux du passager arrière. Seuls les mouvements de la tête, des paupières et de la bouche peuvent nous donner des informations qui sont, elles aussi, limitées, car le conducteur ne doit pas dévier de sa tâche de simulation de conduite. Pour rappel, il s'agit de la vidéo de conduite en vue première personne diffusée sur l'écran posé sur le capot de la voiture. Eu égard à nos travaux sur

le corpus MOSI, en section 2.4, nous expérimentons en premier lieu le modèle R3D [HKS17]. Les résultats n'étant pas concluants, nous avons implémenté plusieurs autres modèles de la communauté vision qui prennent en compte la notion de temporalité :

- convLSTM [SCW+15],
- convolution + RNN [SVSS15] : combinaison d'un modèle convolutionnel 2D suivi d'un modèle récurrent,
- R3D + RNN [WASD21] : combinaison d'un modèle R3D suivi d'un modèle récurrent,
- flux optique [IMS+17] + R3D,
- flux optique + CNN + GRU/LSTM [HS97].

Les éléments ci-dessus n'ont pas donné de résultats convaincants, les modèles convergent pendant la phase d'entraînement, mais les performances s'effondrent durant la phase de validation. Ce résultat est caractéristique d'un problème de sur-apprentissage. L'utilisation de techniques de régularisation n'a pas permis de pallier le problème. Deux hypothèses sont envisagées : une quantité de données insuffisante ou bien les modèles ne sont pas en mesure de capturer les bonnes caractéristiques permettant la classification.

Forts de ce constat, nous implémentons deux nouvelles solutions.

La première utilise un vecteur de 128 caractéristiques pour encoder le visage du conducteur dans chaque image extraite grâce à la librairie Dlib [Kin09]. Le modèle fourni est entraîné sur plusieurs corpus différents avec des centaines de milliers de visages différents. Ensuite, les caractéristiques sont empilées par fenêtre de 35 secondes et envoyées à un réseau récurrent de type GRU ou LSTM. Les résultats ne sont pas concluants là non plus.

Une alternative est d'extraire les points anatomiques/amers du visage (*landmarks*). Le principe est de récupérer 68 repères (le contour des yeux, du visage ou encore du nez) définis par un couple (x, y) dans le repère image (voir figure 4.1). Ceux-ci sont calculés pour chaque image constituant les vidéos en utilisant les librairies Dlib [4]. Cette implémentation donne de premiers résultats satisfaisants. Nous ajoutons les angles d'orientation de la tête calculés comme précédemment en section 2.4 avec *Hyperface* [RPC19b] afin d'améliorer les performances. Finalement un total de 139 caractéristiques *i.e.* 68×2 (le couple (x, y)) plus les trois angles du visage encodent le visage du conducteur.

L'extraction spatio-temporelle de ces points capture explicitement les mouvements de toutes les zones du visage. Dans le but d'augmenter les performances et de réduire la consommation en ressources de calcul, nous avons sous-échantillonné le nombre d'images des vidéos d'un facteur 5. En effet, les images consécutives sont redondantes. Si deux images consécutives sont soustraites, comme illustré sur la figure 4.2, nous remarquons que leurs déplacements sont faibles. En espaçant le nombre d'images, les différences sont plus prononcées. La couleur rouge indique les zones qui ont changé.

Une fois les données extraites, nous les envoyons à un réseau récurrent de

4. `https://ibug.doc.ic.ac.uk/resources/300-W/`

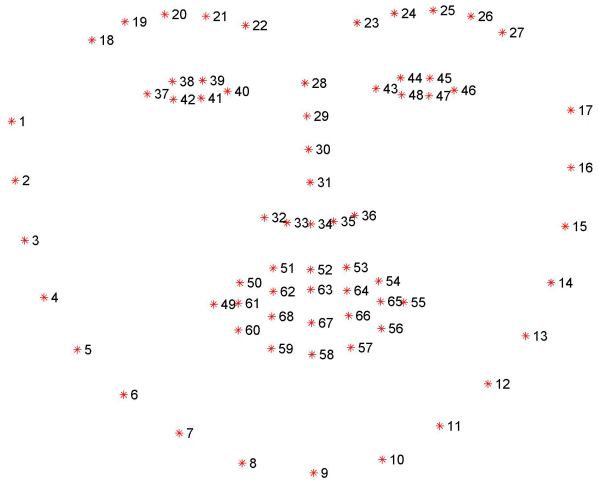

FIGURE 4.1 – Les 68 amers extraits par le modèle Dlib

FIGURE 4.2 – Différences observées entre deux images successives du flux vidéo.

type *stateful* GRU. La matrice qui nourrit le modèle est de la taille du nombre d'images dans chaque tour de parole pour une fenêtre de 35 secondes d'analyse, divisé par le facteur 5, par les 139 caractéristiques. Le nombre de cellules des couches GRU et de la couche dense est aussi trouvé empiriquement. Le tableau 4.3 synthétise notre modèle.

TABLE 4.3 – Définition des couches du modèle vidéo.

Couches	(Entrée, Sortie) x taille
GRU	(139,139) x 4
GRU	(139,139) x 4
Dense	(92,3) x 1

4.2.4 Fusion tardive

Les flux entrants (vidéo, audio et texte) sont par nature hétérogènes, cela nous impose quelques contraintes de modélisation pour notre stratégie de fusion. L'état de l'art effectué au chapitre 1 détaille les différentes façons de fusionner des données. Nous privilégions les deux stratégies de fusion illustrées sur la figure 4.3.

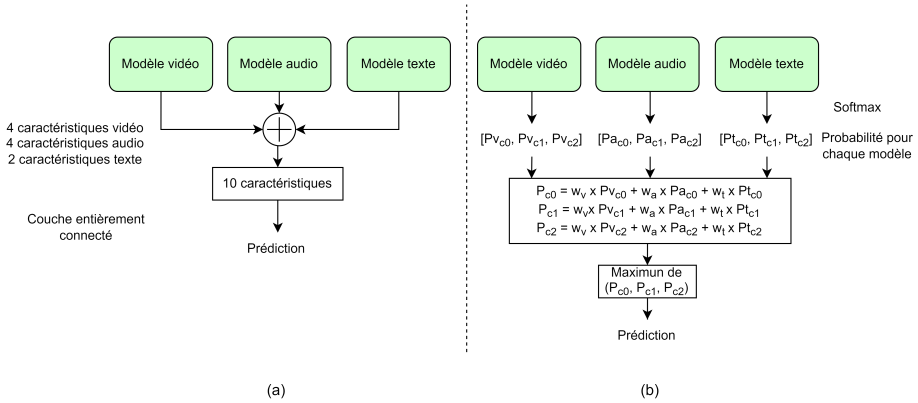

FIGURE 4.3 – Les deux fusion implémentées : (a) fusion tardive par couche dense, (b) fusion tardive par moyenne pondérée.

Fusion tardive par couche dense

Cette fusion nécessite d'entraîner à nouveau la dernière couche des modèles unimodaux ainsi que la couche dense utilisée pour la fusion. Deux étapes sont alors nécessaires.

La première consiste à concaténer les caractéristiques extraites par chacun des modèles. À l'instar de la méthode utilisée sur le corpus MOSI (voir section 2.7.2), nous déterminons empiriquement le nombre et le ratio de caractéristiques entre les modalités pour obtenir les meilleures performances. Pour rappel, les différentes modalités n'ont respectivement pas le même impact sur les performances de prédiction. Un total de 10 caractéristiques sont concaténées, quatre pour le texte et l'audio et deux pour la vidéo. Leur nombre est ici plus faible, car le corpus est moins riche et les caractéristiques sont extraites à des fréquences plus faibles.

La deuxième étape consiste à ajouter une couche entièrement connectée, composée de 30 paramètres (10×3) permettant la classification en trois classes.

Fusion tardive par moyenne pondérée

Les modèles génèrent également en sortie le score de confiance (probabilité) associé à chacune des classes grâce à la fonction *softmax*, notée σ :

$$\sigma(\mathbf{z})_j = \frac{\mathrm{e}^{z_j}}{\sum_{k=1}^{K} \mathrm{e}^{z_k}} \text{ pour tout } j \in \{1, \ldots, K\} \tag{4.1}$$

avec Z un vecteur d'entrée et la somme des K éléments de $\sigma(\mathbf{z})_j = 1$.

Nous utilisons ces scores de confiance en appliquant une moyenne pondérée. La moyenne mathématique est directement écartée de nos expérimentations, car elle ne permet pas de tirer parti des forces de chacune des modalités. Cette fusion par moyenne pondérée est définie comme suit :

$$\begin{aligned} P_{c_0} &= w_v.Pv_{c_0} + w_a.Pa_{c_0} + w_t.Pt_{c_0} \\ P_{c_1} &= w_v.Pv_{c_1} + w_a.Pa_{c_1} + w_t.Pt_{c_1} \\ P_{c_2} &= w_v.Pv_{c_2} + w_a.Pa_{c_2} + w_t.Pt_{c_2} \end{aligned} \tag{4.2}$$

avec P_{j_k} qui désigne la probabilité d'avoir la classe k avec le modèle j pour $k \in \{c_0, c_1, c_2\}$ et $j \in \{\text{texte}(t), \text{audio}(a), \text{vidéo}(v)\}$.

La dernière étape consiste à sélectionner le maximum des P_{j_k} pour obtenir la classe correspondante. Les poids de la moyenne pondérée sont fixés avec les connaissances acquises précédemment : $w_a = 0,2$, $w_v = 0,1$ et $w_t = 0,7$. Cela signifie que la modalité vidéo est peu informative, suivie de l'audio, puis du texte (la plus pertinente ici). Les performances obtenues avec ces deux stratégies de fusion sont discutées dans la section 4.5.

4.3 Analyse paramétrique combiné au modèle bout-en-bout

Le modèle d'analyse de texte (HAN) utilisé à la section précédente 4.2.1 possède de bonnes performances et est peu coûteux en ressources de calcul, ce dernier point est évoqué plus en détail au chapitre suivant 5. Il est donc réutilisé ici. Nous devons toutefois concevoir un nouveau modèle exploitant des caractéristiques extraites manuellement pour la vidéo et l'audio.

4.3.1 Extraction de caractéristiques audio et vidéo

Nous souhaitons extraire des caractéristiques de plus haut niveau pour la modalité audio et vidéo afin de les fusionner avec le modèle HAN analysant le texte. L'objectif est de réduire les coûts en ressources de calcul en réduisant la taille des modèles analysant l'audio et la vidéo. Pour rappel, nous avons un total de cinq caractéristiques différentes extraites manuellement (cf. section 3.6) :

1. la durée moyenne de prise de parole,

2. la durée moyenne du temps de parole,
3. la moyenne des silences,
4. le contact visuel du conducteur,
5. la visibilité du passager arrière.

Les valeurs concernant la parole (caractéristiques 1 et 2) sont calculées pour le conducteur et le passager donnant un total de sept caractéristiques. La caractéristique des silences (3) est commune à tous les passagers. Nous utilisons directement ces dernières comme données d'entrée pour notre modèle. Elles sont ici inférées sur des fenêtres de 35 secondes afin de se synchroniser sur la longueur des fenêtres ingérées par le modèle texte. Le but est d'avoir une synchronisation totale entre les différentes modalités.

Ce vecteur de sept caractéristiques est envoyé à un simple perceptron multicouche (*Multi Layers Perceptron* (MLP)) ayant logiquement une couche d'entrée de taille sept, une couche cachée de quatre neurones et une couche de sortie de taille trois correspondant aux classes **curieux**, **refus argumenté**, **refus catégorique** (voir tableau 4.4). Nous faisons le choix d'un MLP car il s'agit du modèle classique lorsque les données d'entrées sont des vecteurs à une dimension *i.e.* pas de dimension spatiale ni temporelle. De plus, les sept caractéristiques calculées véhiculent déjà la notion de temporalité locale, un réseau récurrent n'est alors pas utile pour ces deux modalités.

TABLE 4.4 – Modèle audio-vidéo.

Couches	(Entrée, Sortie) x taille
Entrée	(7,4) x 1
	ReLU
Couche cachée	(4,4) x 1
	ReLU
Sortie	(4,3) x 1

4.3.2 Fusion multimodale temporelle

Lors de l'analyse d'interactions, il est pertinent de prendre en compte la notion de temporalité locale et globale. Naturellement, l'historique et le contexte permettent de mieux interpréter une interaction humaine à l'instant présent. Il en va de même pour un réseau de neurones profond. Dans notre modèle de bout-en-bout, la temporalité globale est prise en compte dans les modèles d'extraction de caractéristiques avant fusion grâce aux trois différents *stateful* GRU. Ici, nous ne pouvons pas faire cela pour les modalités audio et vidéo, car un MLP est utilisé. Le problème est pallié en ajoutant un modèle temporel à la fusion, soit une couche GRU. Cela permet de garder le contexte global entre chaque fenêtre de 35 secondes. La figure 4.4 schématise notre système complet. Les deux premières composantes vertes réfèrent à l'extraction de caractéristiques

faite par chacun des modèles, la flèche bleue représente l'empilement des vecteurs de caractéristiques.

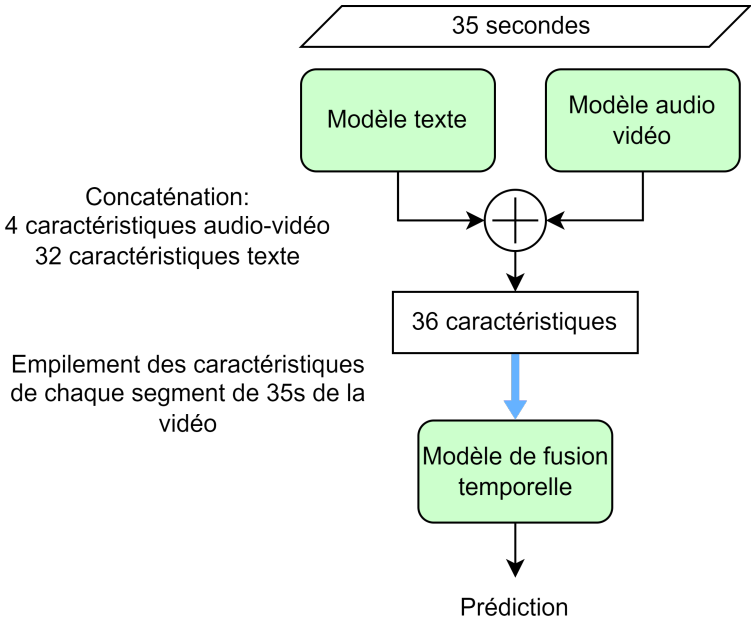

FIGURE 4.4 – Notre modèle de fusion multimodale temporelle.

La fusion se déroule de la manière suivante : un total de 36 caractéristiques sont extraites, les 32 premières sont extraites du texte à l'aide du modèle HAN (voir section 4.2.1) et les quatre restantes sont extraites des sept paramètres audio-vidéo. Cette opération est répétée autant de fois qu'il y a de fenêtres d'analyse de 35 secondes dans la vidéo (en moyenne nous en avons cinq). Une fois le tenseur de taille 5×36 généré, il est envoyé au modèle de fusion temporelle. Finalement, une FC donne la prédiction finale du scénario. Le nombre de caractéristiques est trouvé empiriquement pour chaque modèle. La différence du nombre de caractéristiques texte et audio-vidéo pour la fusion est liée au niveau d'information apporté par chacune d'elle pour la classification finale. Pour rappel, le texte est plus informatif, suivi par l'audio puis la vidéo.

Pour le modèle audio-vidéo nous avons sept valeurs pour lesquelles nous voulons extraire les quatre caractéristiques les plus pertinentes. La couche de sortie du modèle (voir tableau 4.4) est modifiée en $(4, 4) \times 1$. Les réseaux de neurones profonds utilisés pour l'extraction de caractéristiques appliquent l'effet « entonnoir » dans leur topologie : il s'agit d'utiliser une couche comportant moins de neurones que la couche précédente [TZ15]. Cela permet de réduire la dimension des couches au fur et à mesure que le modèle devient profond. La présence d'une telle couche encourage le réseau à compresser les représentations des caractéristiques afin de s'adapter au mieux à l'espace de dimension dispo-

nible. La couche cachée du MLP audio-vidéo (voir tableau 4.4) est alors fixée à quatre neurones pour réduire quasiment par deux l'entrée.

4.4 Implémentation

Le réglage des hyper-paramètres est vital dans le domaine de l'apprentissage profond. Sur les corpus publics, de nombreux travaux sont menés sur leur réglage, nous simplifiant ainsi cette étape. Dans notre contexte, la multimodalité accroît rapidement le nombre d'hyper-paramètres, engendrant un nombre de combinaisons trop important pour utiliser des techniques de *grid search* ou de *random search*. Par nos connaissances *a priori* et par gain de temps (limitation des ressources en calcul), nous utilisons une recherche empirique (ici par dichotomie) afin de trouver un ensemble de valeurs consistantes. Dans nos expérimentations, la fenêtre d'analyse temporelle glissante est fixée à $T = 35$ secondes. Celle-ci exhibe empiriquement les meilleures performances sur la modalité texte pour différents ensembles de validation. Afin de synchroniser les données audio, vidéo et texte en entrée du système, nous appliquons cette fenêtre d'analyse de 35 secondes aux trois modalités.

L'évolution du dialogue donne des informations primordiales qu'il est nécessaire de réussir à capturer. Pour cela, nous utilisons un *stateful* GRU. Les couches de RNN (GRU ou LSTM) mémorise uniquement les informations dans une séquence. Une séquence peut être une phrase, un ensemble de caractéristiques, etc. À chaque nouvelle séquence, les états cachés sont initialisés à zéro, ce qui signifie que les informations extraites précédemment ne sont pas utilisées. Dans notre implémentation, nous remplaçons l'initialisation à zéro par les états cachés de l'itération précédente. Appliquée aux couches RNN de notre modèle (BB), cette stratégie permet de garder la trace de l'évolution de toutes les caractéristiques du début à la fin de la vidéo.

Pour mémoriser le contexte et être en concordance avec les *stateful* RNN, il est impératif d'entraîner les modèles vidéo par vidéo. Chaque vidéo est découpée en environ cinq clips (180/35, *i.e.* la durée d'une vidéo divisée par la taille de la fenêtre d'analyse). Plus précisément, pour éviter que les phrases ne soient coupées, la fenêtre d'analyse est dynamique, variant autour de 35 secondes. Ensuite, les clips sont envoyés chronologiquement un par un au modèle. Cette méthode génère seulement 220 échantillons d'apprentissage (44×5, *i.e.* le nombre de vidéos multiplié par le nombre de clips). Afin d'augmenter cet ensemble, nous décalons le début de la fenêtre d'analyse pour générer 400 échantillons. Ce décalage consiste à passer plusieurs fois sur chaque vidéo et à chaque itération le point de départ de la fenêtre d'analyse est décalé de quelques secondes.

Comme évoqué précédemment, l'exécution des scénarios par les « acteurs » nous oblige à ne pas considérer les 20-30 premières secondes de nos échantillons d'entraînement. Nous les avons donc supprimées lors des phases d'entraînement et de test.

Pour entraîner le modèle multimodal, nous utilisons comme pour les travaux sur le corpus MOSI, des techniques de pré-entraînement. Les modèles unimo-

daux sont d'abord entraînés pendant environ 80 époques sur leurs données respectives. Ensuite, lorsqu'ils atteignent leur meilleure précision, leurs poids sont sauvegardés. Puis, au début de la phase d'entraînement multimodale, les poids sauvegardés précédemment sont utilisés pour initialiser les poids du modèle multimodal. Nous avons donc uniquement les poids du modèle de fusion qui sont initialisés aléatoirement. Ce processus permet d'améliorer les performances finales et de réduire le temps d'entraînement.

Pour notre problématique de classification multi-classes, nous privilégions la perte d'entropie croisée :

$$-\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right) \tag{4.3}$$

où y est un indicateur binaire, 0 si la classe c prédite ne correspond pas à la classe de l'observation o et respectivement 1 si c prédit est égal à c de o. p est la probabilité prédite de l'observation o pour la classe c. M est le nombre de classes.

Elle permet de mesurer l'erreur de la classe actuelle en ne tenant pas compte des erreurs des autres classes.

4.5 Évaluations

Cette section présente les évaluations obtenues pour les deux modèles implémentés. Nous détaillons les performances de chacune des modalités séparément et des différentes stratégies de fusion. Une discussion des analyses est ensuite proposée dans la section relative aux évaluations qualitatives.

4.5.1 Analyse quantitative

Un point clé, lors de travaux sur l'analyse du comportement ou des émotions, est la dépendance au locuteur. En effet, l'intérêt est d'évaluer la capacité de l'algorithme à généraliser lorsqu'il traite un nouvel individu. Nous générons aléatoirement cinq fichiers d'entraînement/test différents (*i.e.* validation croisée à cinq blocs, *5-fold Cross-Validation*). Pour chacun d'eux, nous divisons le jeu de données en 80% (18 participants) et 20% (4 participants).

Nous utilisons la micro-précision comme métrique pour évaluer nos modèles. La micro-précision est définie par l'équation (4.4). Celle-ci est pertinente lorsque nous n'avons pas une équipartition d'échantillons dans chaque classe, ici 50% des données dans la classe « curieux », et 25% pour les classes « refus argumenté » et « refus catégorique ».

$$\text{Micro-précision} = \frac{(TP_1 + TP_2 + \ldots + TP_n)}{(TP_1 + TP_2 + \ldots + TP_n + FP_1 + FP_2 + \ldots + FP_n)} \tag{4.4}$$

Il s'agit de la somme des vrais positifs (TP) divisée par la somme des vrais positifs et faux négatifs (FP) pour chacune des classes n.

Nous complétons notre métrique d'évaluation par son écart-type, *i.e.* la moyenne des cinq écarts types induits par la stratégie de validation croisée.

Modèle de bout-en-bout (BB)

Les performances obtenues démontrent les potentialités de notre modèle de fusion. Nous obtenons une micro-précision de 70% pour la modalité texte, 70,6% pour l'audio et 65,6% pour la vidéo. La modalité texte offre les meilleures performances si nous prenons en considération l'écart-type. Notre modèle de fusion par couche entièrement connectée (FC) donne de bons résultats en fusion trimodale (vidéo, audio et texte) : elle améliore la micro-précision de 11% pour atteindre 81,6%. La fusion naïve par moyenne pondérée est moins performante avec un gain de 5,4% seulement. Considérant la fusion bimodale, les performances sont améliorées de 8,8% pour le couple texte-audio et de 5,2% pour le couple texte-vidéo. Les modalités audio et vidéo combinées n'améliorent pas les performances. Si nous retirons les relations (*stateful* RNN) entre les fenêtres d'analyses, la micro-précision chute d'environ 5%.

TABLE 4.5 – Performances obtenues pour le modèle de bout-en-bout en validation croisée sur cinq blocs. FC dénote la stratégie par couche dense et (N) la stratégie par moyenne pondérée.

Mode	Modalité	Micro-précision ± écart-type
Unimodal	Vidéo (V)	65,6% ± 4
	Audio (A)	70,6% ± 4,9
	Texte (T)	70% ± 0,8
Bimodal	A + V (FC)	61% ± 3,9
	T + A (FC)	79,4% ± 5,9
	T + V (FC)	75,8% ± 7,4
Trimodal	**A + V + T (FC) (BB)**	**81,6% ± 5,9**
	A + V + T (N)	78% ± 5,3

L'écart-type peut paraître élevé au premier abord, mais il est à replacer dans le contexte d'indépendance au locuteur. En considérant la taille de notre corpus et la variabilité des participants, cela engendre de fortes variations. *A priori* une augmentation du corpus réduirait ces valeurs. À cela s'ajoute le fait que la modalité texte est la seule ingérant les données des deux passagers. En effet, les données audio sont celles du passager arrière uniquement et la modalité vidéo analyse seulement le visage du conducteur.

Le contexte d'indépendance au locuteur est très contraignant pour un réseau de neurones. Par exemple, les assistants vocaux présents dans les smartphones utilisent un modèle d'intelligence artificielle générique, dit du « Monde ». Il est entraîné sur des corpus colossaux (milliers d'heures) et il est ensuite spécialisé à un utilisateur par une phase de ré-estimation du modèle du monde. Il est par exemple demandé à l'utilisateur, dans le cas des téléphones Android, de répéter plusieurs fois « OK Google ».

Pour notre application, si nous nous plaçons dans le contexte de véhicule partagé, il est envisageable d'utiliser des techniques similaires : par exemple, en mentionnant aux utilisateurs que le service sera activé au bout d'un certain temps d'utilisation. Pour tenter de modéliser cet aspect, et ainsi passer à une connaissance du locuteur, nous entraînons notre modèle multimodal FC sur les 90 premières secondes de chacune des 44 vidéos et testons sur les 90 secondes restantes. La micro-précision passe à **88,2%**. À l'échelle de notre corpus, cette amélioration de 6,6% comparée au modèle (FC) montre qu'une phase de spécification serait bénéfique en termes de performances. Elle souligne aussi le manque de données pour entraîner notre modèle de bout-en-bout.

Modèle de bout-en-bout + paramétrique (BB+P)

Le tableau 4.6 synthétise les performances obtenues. Le modèle audio/vidéo obtient une micro-précision de 60% et le textuel 70%. Cette nouvelle implémentation donne de bonnes performances. En effet, elle améliore la micro-précision de 11% pour atteindre 81%. Les valeurs d'écart-type sont comprises entre 0,8 et 1,2 : ceci indique une bonne stabilité du modèle en fonctionnement « indépendant du locuteur ».

TABLE 4.6 – Performances moyennes sur cinq ensembles de validation croisée.

Modalité	Micro-précision
A + V	60% ± 1,12
T	70% ± 0,8
A + V + T (BB+P)	**81% ± 1,2**

La figure 4.5 nous montre les prédictions du modèle pour un ensemble de validation. La métrique tracée est la micro-précision en fonction du temps. Plus précisément, il s'agit de la micro-précision pour chaque fenêtre ($T = 35s$) à leur instant respectif t. Il est intéressant de noter que, lorsque le modèle prend en compte 90 secondes de la vidéo, il est capable de classifier avec une précision (99%). La vidéo utilisée pour tracer la courbe 4.5 est de la classe **curieux**. Elle est représentative des vidéos de bonne qualité et les participants sont en cohérence avec le scénario demandé.

4.5.2 Analyse qualitative

Une étude des clips, ainsi que l'analyse statistique effectuée à la section 3.6, nous montrent des phases transitoires en début et en fin de vidéo. Les sujets n'ont parfois pas pu jouer leur rôle en adéquation avec le scénario demandé. Si nous ne prenons pas en compte les 30 premières et les 20 dernières secondes, les 130 secondes restantes sont des scénarios comparables à une situation de discussion réelle.

Nous avons aussi mis en place une validation croisée *leave-one-out*. Il s'agit de mettre toutes les vidéos en phase d'entraînement excepté une. Le but est

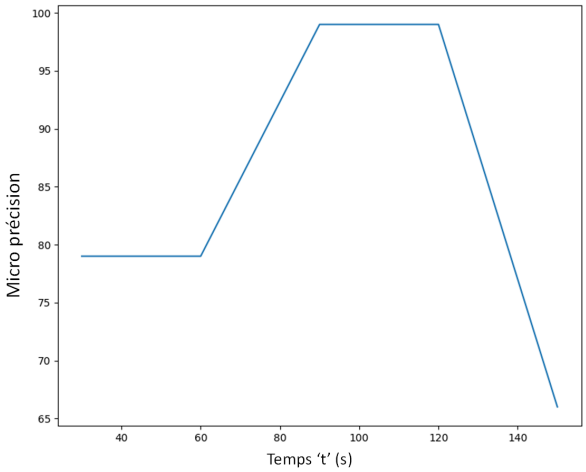

FIGURE 4.5 – Exemple d'évolution temporelle de la micro-précision sur une vidéo du scénario « curieux ».

d'identifier les vidéos délivrant de mauvaises performances pour les analyser plus en détail. Cette étude met en exergue trois vidéos permettant d'identifier trois problèmes :

— Dans la première vidéo, le conducteur a mis plus de 60 secondes à commencer le scénario comparé aux autres qui mettent environs 30 secondes.
— Sur la seconde, les participants ne parlaient pas assez fort, ils sont donc peu audibles. De plus le conducteur n'articule pas assez, rendant la modalité audio peu informative.
— Sur la dernière vidéo, il y a deux réserves. D'une part, le conducteur est mal installé dans son siège : son visage est alors partiellement perçu par la caméra. D'autre part, un problème du matériel audio est survenu : nous constatons la présence de bruits parasites sur tout l'enregistrement. Ces deux problèmes perturbent l'extraction du visage et le calcul des caractéristiques sonores.

D'autres erreurs sont dues aux mauvaises prédictions du modèle. Nous nous attendons à ce que la distribution des données de la catégorie « refus argumenté » soit au centre des deux autres. Lors de la création du corpus, les scénarios et classes sont définis afin que les comportements *i.e.* les percepts ou caractéristiques des passagers soient implicitement distincts lors du jeu d'acteur des passagers. La gradation souhaitée est la suivante : **curieux** (comportement nominal) $\longrightarrow$ **refus argumenté** (légèrement conflictuel) $\longrightarrow$ **refus catégorique** (conflictuel). Les distributions de données peuvent alors se superposer/mélanger du fait que les sujets n'ont parfois pas pu jouer leur rôle en adéquation avec le comportement demandé. Ce phénomène apparaît surtout sur les classes **refus argumenté** et **refus catégorique**. Sur la figure 4.6 ces deux

classes ont beaucoup de faux positifs et faux négatifs. Le modèle (BB+P) est plus performant sur la classe **refus catégorique** et **curieux**. Quant à (BB) il est plus efficace sur les scénarios **refus argumenté**.

		Classes prédites					
		Système (BB+P)			Système (BB)		
		cur	ref_arg	ref_cat	cur	ref_arg	ref_cat
Classes réelles	cur	13	0	0	10	3	0
	ref_arg	1	4	2	0	7	0
	ref_cat	2	2	15	1	9	9

FIGURE 4.6 – Matrice de confusion pour nos deux modèles. cur dénote la classe curieux, ref_arg = refus argumenté et ref_cat = refus catégorique

L'espace restreint de l'habitacle est également une limite pour la modalité vidéo, car les passagers sont la plupart du temps statiques dans cet environnement, limitant la pertinence du mouvement inter-images pour différencier les scénarios.

Certaines erreurs peuvent aussi être induites par le processus d'annotation. En effet, déterminer le début et la fin d'un tour de parole peut parfois être difficile, car les répétitions et interjections rendent le processus de délimitation compliqué. Les superpositions de parole peuvent aussi complexifier l'annotation. Pour rappel un tour de parole est une unité de discours continue commençant et se terminant par une pause explicite. Il en résulte un impact sur les systèmes utilisant des caractéristiques statistiques (fréquence, moyenne, etc.). La transcription manuelle peut aussi comporter des erreurs et, dans une moindre mesure, impacter les performances du modèle texte, même si les modèles de bout-en-bout sont robustes aux erreurs aléatoires. La figure 4.7 nous expose des exemples d'annotations avec le logiciel ELAN.

4.6 Étude comparative des deux modèles

Il a été constaté [ATY+19] l'existence d'un seuil de quantité de données pour lequel les techniques utilisant l'apprentissage profond surpassent les méthodes plus classiques (statistiques, apprentissage machine, etc.). En dessous de ce

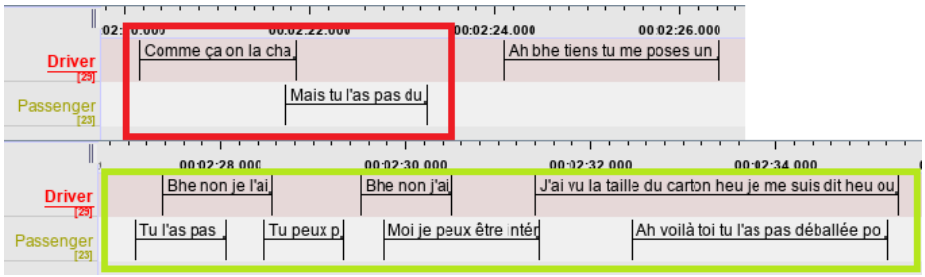

FIGURE 4.7 – Exemple d'annotation facile en rouge et plus difficile en vert.

seuil les méthodes plus anciennes obtiennent des performances équivalentes ou meilleures que les modèles de *deep learning*, cf. figure 4.8.

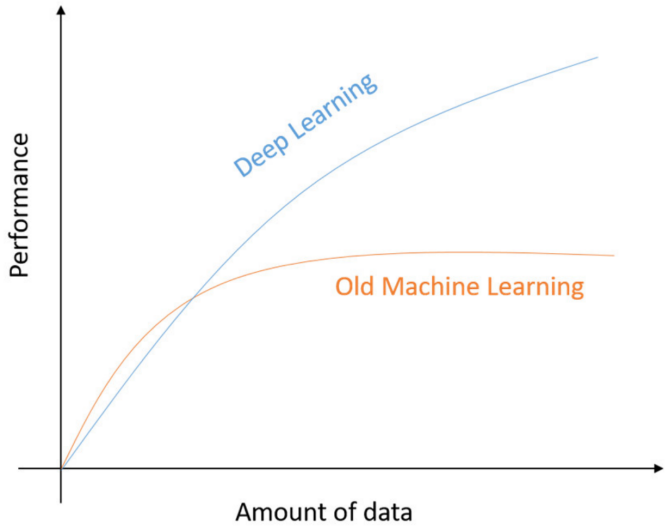

FIGURE 4.8 – Performances des modèles par apprentissage machine et profond *vs* quantité de données [ATY+19].

Eu égard à nos performances et de la taille de notre corpus, il semble que nous soyons dans cette situation. En mettant en parallèle nos deux modèles, nous obtenons le tableau 4.7. L'utilisation de caractéristiques extraites manuellement (BB+P) donne des résultats plus consistants avec des écarts-types nettement en dessous du modèle de bout-en-bout (BB). Ce dernier permet d'utiliser chaque modalité indépendamment et ne nécessite pas de trouver des caractéristiques manuelles qui peuvent s'avérer représenter une tâche fastidieuse. Le (BB+P) ne contient pas assez de percepts sur la vidéo ou l'audio seul pour permettre une classification avec ces deux modalités séparées. L'écart-type du (BB) est probablement lié au manque de données, nous faisons nos mesures sur des ensembles

de données avec « indépendant au locuteur». C'est-à-dire qu'en phase de test, le modèle n'a jamais vu le locuteur pour lequel il reçoit les données. De ce fait, il a besoin d'être entraîné sur de nombreuses personnes différentes afin de pouvoir généraliser correctement.

TABLE 4.7 – Mise en parallèle des modèles (et variantes proposées).

Modèle	Modalité	Micro-précision
Bout-en-bout (BB)	Vidéo (V)	65,6% ± 4
	Audio (A)	70,6% ± 4,9
	Texte (T)	70% ± 0,8
	A + V (FC)	61% ± 3,9
	V + A + T (FC)	**81,6% ± 5,9**
Bout-en-bout + paramétrique (BB+P)	T	70% ± 0,8
	A + V	60% ± 1,12
	V + A + T	**81% ± 1,2**

En analysant les mauvaises classifications de chacun des modèles pour un même ensemble de validation, le constat est que les erreurs ne sont pas sur les mêmes fenêtres d'analyse (voir figure 4.9).

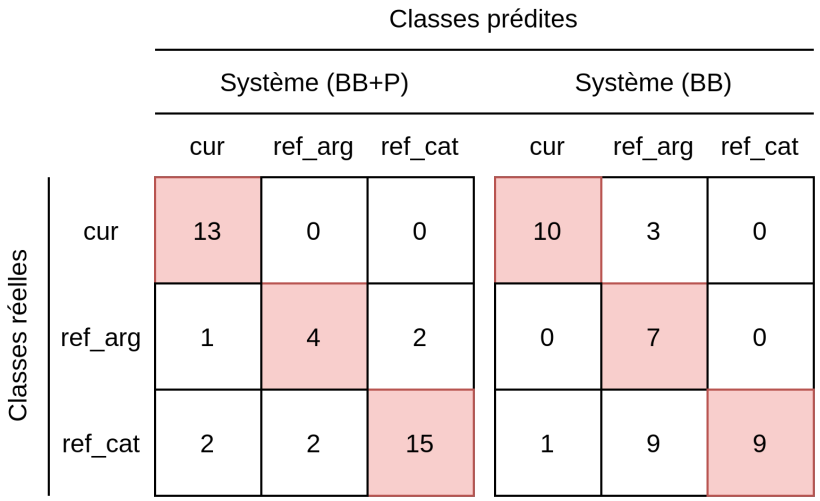

FIGURE 4.9 – Comparaison des deux matrices de confusion. Elles représentent l'inférence de chacun des modèles pour un même ensemble de test.

Le modèle (BB+P) possède de meilleures capacités à catégoriser les vidéos de la classe **curieux** et **refus catégorique** qui sont les classes les plus opposées. Le modèle (BB) est performant sur la classe **refus argumenté**.

Fort de ce constat, un point d'amélioration important serait l'utilisation de

méthodes d'apprentissage d'ensemble telles le *boosting* ou le *bootstrap aggregating (ou bagging)* pour améliorer les performances de prédiction : voir [GGD+20, SJB+18] pour une revue complète sur le sujet. Ces techniques sont pertinentes dans notre cas, car les modèles implémentés sont compacts ce qui nous permet de les inférer plusieurs fois sans avoir des temps de latence trop importants. D'autre part, la complémentarité des deux modèles est en adéquation avec les concepts du *bagging* et du *boosting* qui consistent à avoir des ensembles de modèles dits « faibles » pour former un modèle « fort ». Les principes des deux concepts sont définis ci-après.

Bagging - C'est un méta-algorithme conçus pour améliorer la stabilité et la précision des algorithmes d'apprentissage automatique. Il diminue la variance et permet d'éviter le surapprentissage. Il est généralement appliqué aux forêts d'arbre décisionnel.

Boosting - C'est une technique de modélisation d'ensemble qui tente de construire un classificateur fort à partir d'un certain nombre de classificateurs faibles. Le modèle fort est construit en utilisant des modèles faibles en série. Tout d'abord, un modèle est construit à partir des données d'apprentissage. Ensuite, un deuxième modèle est construit et essaye de corriger les erreurs présentes dans le premier modèle. Cette procédure est poursuivie et des modèles sont ajoutés jusqu'à ce que l'ensemble des données d'apprentissage soit prédit correctement ou que le nombre maximum de modèles soit ajouté.

Les principales causes d'erreur dans l'apprentissage de modèles sont dues aux bruits, aux biais et à la variance. Les méthodes d'ensembles permettent de les minimiser, car elles combinent plusieurs estimations provenant de différents modèles. Les performances peuvent donc être plus stables. Ces techniques ont l'inconvénient d'accroître d'autant la latence globale que chaque modèle soit inféré. Tous les modèles participant à la décision doivent être inférés. Nous pouvons l'envisager dans notre cas en contraignant le nombre d'inférences.

4.7 Conclusion

Dans ce chapitre, nous proposons un premier modèle exploitant le corpus enregistré. Celui-ci est capable d'analyser des ajouts de 35 secondes de données vidéo, audio et texte simultanément pour détecter des situations conflictuelles dans l'habitacle d'une voiture. Nous montrons que l'ajout de connexions entre les fenêtres d'analyse (*stateful* GRU) et le processus d'entraînement permet de garder le contexte tout au long de l'analyse et améliore les performances d'environ 5%. Les performances sont satisfaisantes malgré un écart-type qui peut paraître élevé. Ce dernier est *a priori* induit par le manque de données. La fusion démontre, comme attendu, des gains substantiels ($+11,6\%$ par rapport au texte) en termes de performances. De plus, la spécialisation aux locuteurs du véhicule montre des gains supplémentaires de 6,6% par rapport à

notre meilleure stratégie de fusion (FC). En effet, une phase d'adaptation d'un système générique « Monde » à un individu est envisageable.

Puis, une nouvelle version est proposée en combinant un réseau profond et un modèle paramétrique (BB+P). Les résultats sont en cohérence avec le modèle (BB), avec une performance de prédiction de 81% et un gain apporté par la fusion de 11% comparé au modèle texte. Cette version s'inscrit aussi dans une problématique d'embarquabilité dans un véhicule. En effet, le passage des sept caractéristiques audio-vidéo dans le modèle MLP est peu coûteux (puissance de calcul, mémoire, etc.). De plus, le modèle HAN offre de bonnes performances pour un modèle compact. Cet aspect embarquabilité/intégration sera développé dans le chapitre suivant.

En marge des performances de classification, l'intérêt du modèle (BB) est d'avoir une chaîne de traitement autosuffisante ingérant les données brutes en entrée, là où (BB+P) nécessite une intervention humaine afin de déterminer et calculer les caractéristiques pertinentes, tâche pouvant être fastidieuse.

Ce chapitre a permis deux publications en conférences internationales : IEEE ITSC pour les travaux sur le modèle (BB+P) et MMEDIA pour les travaux sur le modèle (BB). Un article est en cours de soumission dans la conférence francophone RFIAP. Les références sont détaillées page v.

Le chapitre suivant se focalise sur les aspects intégration, en vue d'une application industrielle embarquée de toute la chaîne de traitement. Plus précisément, nous expliquons le fonctionnement d'un point de vu macro de la plateforme embarqué dans les véhicules. Nous présentons aussi les améliorations théoriques et les performances de compacités de nos deux systèmes.

Chapitre 5

Intégration : vers une application industrielle embarquée

Sommaire

5.1 Introduction

La réduction drastique des prix des cartes graphiques et du matériel informatique permet aujourd'hui de moins se soucier de l'optimisation des programmes informatiques et de la taille des réseaux de neurones profonds. Appliqué au domaine des systèmes embarqués ce constat n'est plus vrai. Tout est optimisé pour réduire au maximum l'utilisation des ressources. La compacité des réseaux de neurones est alors un aspect qui devient primordial. Il s'agit d'une problématique investiguée dans le secteur industriel, mais aussi académique. Trois leviers majeurs sont explorés. Tout d'abord, la mise sur le marché de puces matérielles optimisées pour le calcul matriciel. Ensuite, côté logiciel, Tensorflow et Pytorch, les principales librairies pour programmer et entraîner des

réseaux de neurones proposent de nouvelles fonctionnalités pour l'optimisation des modèles. Notons que chaque fabricant de puces (Qualcom, Nvidia, Apple, etc.) développe sa propre librairie optimisée pour leur matériel. Finalement, le dernier levier d'action concerne la topologie des modèles. De récents modèles ont vu le jour à partir de 2015 avec YOLO V1 [RDGF16] puis Mobilenet V1 [HZC+17] en 2017. Ces modèles sont conçus pour être légers avec des temps d'exécution faibles pour la détection d'objets ou reconnaissance d'image. Les avancées sur la compacité des modèles sont aujourd'hui principalement centrées sur des applications de vision par ordinateur. Les autres contextes applicatifs pour le domaine des systèmes embarqués sont aujourd'hui peu considérés. Nous souhaitons ici aborder cet aspect en analysant les performances de notre chaîne de traitement.

Nous présentons dans ce chapitre les avantages d'un système embarqué et l'architecture des systèmes Android présents dans les véhicules. Ensuite, nous développons les optimisations disponibles dans la littérature pour augmenter la compacité des modèles puis les performances de notre chaîne de traitement sont présentées et discutées.

5.1.1 Pourquoi privilégier l'embarquabilité ?

Le *cloud computing* permet de déplacer les tâches de calcul lourdes vers des grappes de calculateur de haute performance. Les informations à calculer sont alors envoyées vers le serveur de calcul à travers une requête internet, le résultat du calcul est ensuite renvoyé au demandeur.

L'inférence embarquée présente de nombreux avantages comparée aux implémentations dans le *cloud i.e.* :

- **Temps de latence**. Il n'est pas nécessaire d'envoyer une requête sur une connexion réseau et d'attendre une réponse. Ceci peut être critique par exemple pour les applications vidéo qui traitent des images successives provenant d'une caméra.
- **Disponibilité**. L'application fonctionne même en dehors de la couverture réseau. Dans le cas d'une application de détection de situation conflictuelle, il n'est pas envisageable d'avoir un système fonctionnant par intermittence.
- **Vitesse.** Le nouveau matériel spécifique au traitement des réseaux neuronaux présents sur le matériel récent permet des calculs nettement plus rapides qu'une unité centrale polyvalente, seule.
- **Confidentialité**. Les données ne quittent pas l'appareil. Ceci permet de se conformer plus simplement aux normes RGPD [1] en vigueur.
- **Coût**. Aucune ferme de serveurs n'est nécessaire lorsque tous les calculs sont effectués sur l'appareil Android. De plus, les coûts d'un opérateur téléphonique ne seraient pas négligeables si des données vidéo devaient être envoyées sur un serveur.

1. `https://www.cnil.fr/fr/reglement-europeen-protection-donnees`

Tous ces constats justifient donc le choix d'une application embarquée comme première détection. Des fonctionnements hybrides (embarqué-cloud) sont également envisageables.

5.1.2 Architecture haut niveau d'un système Android automobile

Dans un véhicule, de nombreux sous-systèmes (*Unité de Commande Electronique* (ECU), capteurs, etc.) s'interconnectent entre eux, mais aussi avec le système d'infodivertissement embarqué (*In-Vehicle Infotainment* (IVI)) via diverses topologies de bus de données. Les constructeurs tendent aujourd'hui à unifier les calculateurs pour converger vers un unique système recevant toutes les données du véhicule. Actuellement, l'IVI s'apparente à ce fonctionnement, il reçoit les données de nombreux capteurs. Il est notamment en charge de l'écran, de toutes les interactions homme-machine associées (régler la climatisation, le volume, etc.), du GPS, de la caméra de recul, des appels téléphoniques, etc. L'IVI est la plateforme prédisposée à faire fonctionner des algorithmes d'apprentissage profond, car elle possède toutes les ressources matérielles : CPU, *Graphics Processing Unit* (GPU), *Network Processing Unit* (NPU), *Digital Signal Processsor* (DSP), etc.

Un diagramme d'architecture de haut niveau du système d'exploitation automobile Android est présenté en figure 5.1. Ce système se compose des éléments génériques suivants :

- **Applications**. Également appelée couche *Interface Homme Machine* (IHM), elle contient les applications utilisateur et système. L'idéal est de concevoir les applications de manière à ce que la plupart des exigences commerciales essentielles soient déplacées vers la couche de service. Une telle conception facilite l'évolutivité et les mises à jour.
- **Couche de service**. Tous les services système sont inclus dans cette couche. Il est intéressant de noter que les équipementiers peuvent utiliser la couche de service comme un bouclier de sécurité et éviter le contact direct entre les applications et la couche d'abstraction matérielle.
- **Couche d'abstraction matérielle (*Hardware Abstraction Layer* (HAL))**. La HAL expose les interfaces automobiles aux services du système de manière à obtenir une architecture agnostique pour les véhicules. Le cadre d'application, les services système et HAL sont les composants essentiels de la plateforme Android Automotive OS : ces couches facilitent l'échange de données entre les calculateurs des véhicules et les applications.
- **Noyau Linux (BSP)**. Il est le noyau sous-jacent de l'architecture Android Automotive. Il contient tous les logiciels bas niveau (drivers) permettant de communiquer avec le matériel (microphone, Bluetooth, USB, WIFI, Caméra, etc.).

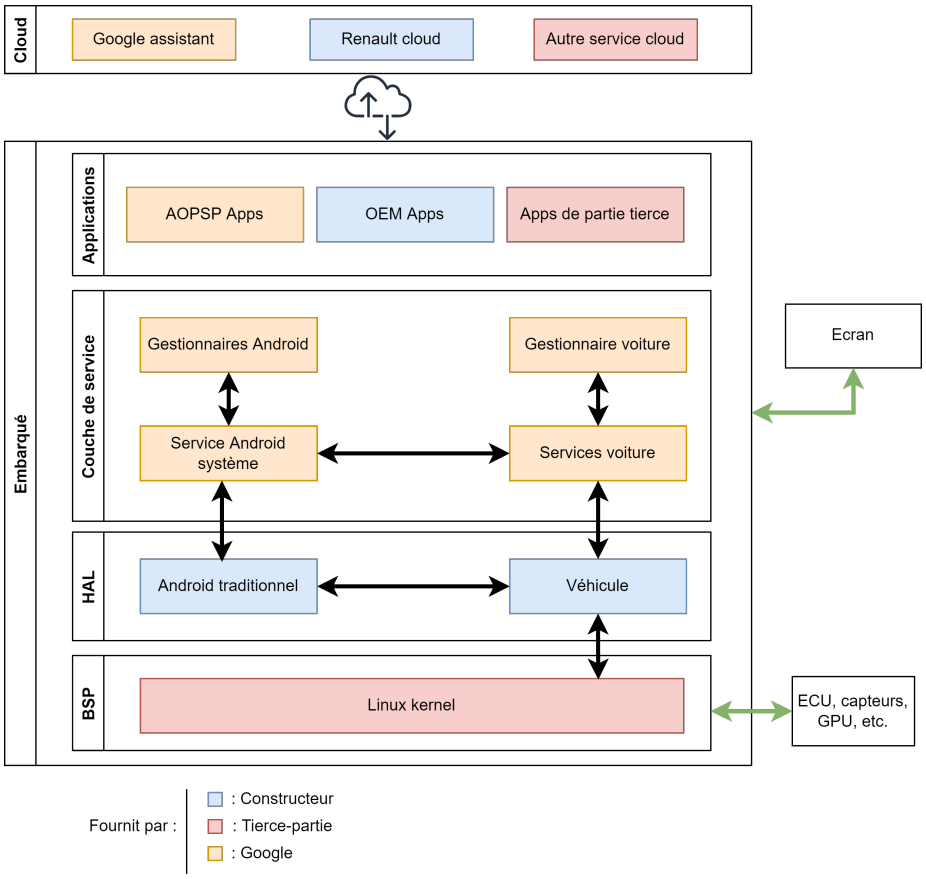

FIGURE 5.1 – Architecture d'un système infodivertissement automobile.

5.2 Optimisation pour améliorer la compacité

Cette section détaille les techniques couramment utilisées pour augmenter la compacité d'un modèle et réduire ses besoins en ressources de calcul. La compacité est liée au nombre d'opérations élémentaires (multiplication, addition, etc.) et au nombre de paramètres du modèle. Le premier critère impacte les coûts en calcul (CPU, GPU, etc.) et le second l'emplacement mémoire utilisé : RAM,*Read-Only Memory* (ROM), espace disque.

Nous n'avons pas exploré toutes les possibilités durant cette thèse. Rappelons que nos travaux se focalisent sur le choix des modèles, sur les couches élémentaires utilisées et sur le traitement des données avant ingestion par les modèles. Nous discutons donc ci-après uniquement de la partie inférences des modèles. L'entraînement est effectué sur une plateforme cloud dédiée.

Lorsque nous travaillons sur des ressources embarquées avec des réseaux de

neurones, il est important de prendre en compte toute la chaîne de traitement de la donnée. Comme illustré sur la figure 5.2, il y a trois sources consommatrices de ressources.

La première (en bleu) correspond à l'enregistrement de la donnée avec l'encodage.

Ensuite, l'extraction des « caractéristiques primaires » (en orange) permet de calculer les éléments qui seront envoyés au réseau de neurones. Pour la modalité vidéo, nous avons besoin d'un modèle d'apprentissage profond capable d'extraire un visage et ses amers anatomiques associés. Concernant la modalité audio, OpenSMILE calcule un jeu de paramètres. Finalement nous avons la transcription de l'audio en texte qui est aujourd'hui difficilement embarquable, les solutions proposées par les industriels du secteur sont principalement des modèles cloud car, actuellement, les modèles embarqués ne sont pas assez performants. Des travaux internes chez Renault se focalisent sur ce verrou technologique, avec notamment des solutions basées sur des vocabulaires restreints.

En fin de chaîne, nous avons l'extraction de « caractéristiques secondaire » en rouge (figure 5.2) qui ingère les caractéristiques primaires pour en générer des intermédiaires qui sont utilisées par la fusion pour effectuer la classification Tous ces extracteurs nécessitent d'être embarqués et sont consommateurs de ressources. Ces contraintes sont à prendre en considération.

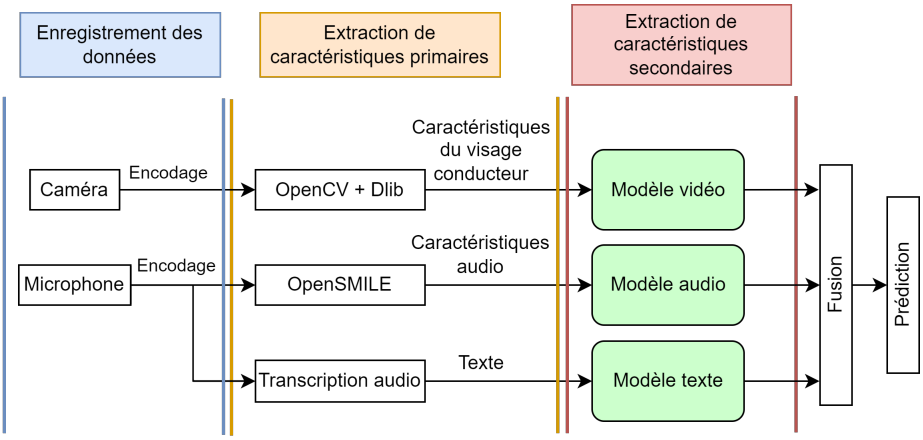

FIGURE 5.2 – Chaîne de traitement des données : extraction de caractéristiques primaires et secondaires.

Pour les caractéristiques primaires, nous avons limité les besoins en ressources de calcul eu égard aux considérations suivantes :

1. baisse de la fréquence de calcul des caractéristiques, en réduisant notamment le nombre d'images traitées par seconde,
2. sélection et réduction du nombre de caractéristiques audio à leur minimum, sans dégradation des performances.

Pour l'extraction de caractéristiques secondaires, nous nous sommes focalisés sur :

1. le choix de nos couches de réseaux de neurones, en privilégiant les couches élémentaires,
2. la réduction de la taille des modèles et de leurs hyperparamètres pour chacune des modalités jusqu'au point d'inflexion de la courbe performance *vs* taille du modèle,
3. la limitation du vocabulaire du texte aux mots les plus importants.

Après ces considérations, il est possible d'appliquer plusieurs techniques et processus pour améliorer les performances. Nous détaillons ci-après les plus usuelles.

5.3 Outils d'optimisation des modèles

Nous présentons ici les outils de la littérature qui permettent d'augmenter la compacité des modèles des réseaux de neurones. Le but est de réduire au maximum les critères suivants :

1. **la latence**, représentant la quantité de temps nécessaires pour réaliser une inférence pour un modèle donné. Une faible latence réduit aussi la consommation électrique.
2. **la taille du modèle**. En la diminuant, le besoin en espace de stockage, le temps de téléchargement du modèle et l'utilisation de la RAM sont moindres.

La latence est importante : elle est directement impactée par la taille des modèles. Sa réduction permet de ne pas avoir à stocker les données au fur et à mesure qu'elles arrivent (*overflow*). Elle impacte aussi la qualité de la détection, car si les données arrivent plus vite que leur vitesse de traitement, il faut vider (*flush*) la mémoire ; ceci engendre alors une perte de données.

L'augmentation de la compacité par rapport à un modèle de référence peut réduire les performances. Un équilibre acceptable est alors à définir lors du processus de développement entre performances *vs* compacité.

5.3.1 Noeuds de calcul superflus

Lors de la sauvegarde du modèle, les noeuds de calculs liés à la phase d'entraînement sont présents et doivent être retirés. Les librairies de Tensorflow et Pytorch permettent de supprimer les éléments liés à la rétropropagation du gradient (ou *backpropagation*) et à la régularisation (*dropout, batch normalisation*, etc.).

5.3.2 Quantification

Cette méthode (*quantization*) consiste à réduire le nombre de bits sur lesquels un nombre est numériquement encodé [NZC19]. Deux attributs permettent de caractériser un nombre au format numérique : la plage dynamique, qui réfère à la plage de représentation d'un nombre et la résolution représentant la distance entre deux nombres (voir le tableau 5.1).

Dans le domaine de l'apprentissage profond le 32-bit (FP32) virgule flottante est le format prédominant pour encoder les nombres, car il est plus versatile que l'entier (INT) avec la contrepartie d'augmenter l'emplacement mémoire.

TABLE 5.1 – Représentation des encodages les plus couramment utilisés. INT8/4 pour un entier signé encodé sur 8/4 bits et FP32 pour les nombres à virgules flottantes encodés sur 32 bits.

Encodage	Plage dynamique	Résolution
INT8	$[-128..127]$	255
INT4	$[-8..7]$	15
FP32	$\pm 3,4 \times 10^{38}$	$4,2 \times 10^{9}$

La quantification peut être appliquée de deux manières : *online*, avec l'application du procédé durant la phase d'entraînement du modèle ou *offline i.e.* après avoir sauvegardé le modèle. Nous pouvons observer les bénéfices de la quantification sur la figure 5.3 [2], sachant que les résultats sont obtenus sur un smartphone Google pixel 2.

Model	Top-1 Accuracy (Original)	Top-1 Accuracy (Post Training Quantized)	Top-1 Accuracy (Quantization Aware Training)	Latency (Original) (ms)	Latency (Post Training Quantized) (ms)	Latency (Quantization Aware Training) (ms)	Size (Original) (MB)	Size (Optimized) (MB)
Mobilenet-v1-1-224	0.709	0.657	0.70	124	112	64	16.9	4.3
Mobilenet-v2-1-224	0.719	0.637	0.709	89	98	54	14	3.6
Inception_v3	0.78	0.772	0.775	1130	845	543	95.7	23.9
Resnet_v2_101	0.770	0.768	N/A	3973	2868	N/A	178.3	44.9

FIGURE 5.3 – Exemple d'amélioration de la compacité de modèle CNN grâce à la quantification.

Si nous considérons le modèle InceptionV3, après quantification, les performances de prédiction sont réduites de 1% avec une amélioration drastique de la latence de pratiquement 50% « Latency (Post Training quantized » et une réduction de l'emplacement mémoire « size (optimized) » d'un facteur 4.

2. `https://www.tensorflow.org/lite/performance/model_optimization`

La quantification est une technique performante et nécessaire dans le cas de systèmes embarqués. Il convient de déterminer si la chute de performance est acceptable vis-à-vis de l'application.

5.3.3 Élagage

Cette technique (*pruning*, [HABN+21]) consiste à mettre à zéro les poids du modèle ayant peu d'amplitude (peu d'impact sur la sortie du modèle). La motivation sous-jacente est le sur-paramétrage des modèles. Ces redondances peuvent être supprimées en mettant leur poids à zéro. Un critère de type seuil est utilisé pour déterminer si un poids est conservé ou non. Il résulte une augmentation de matrice creuse (*sparcity*) des poids du modèle. Une matrice creuse ou *sparse* contient un grand nombre de zéros. Il devient alors ensuite plus facile à compresser, réduisant ainsi sa taille mémoire. Les bénéfices de l'élagage sont illustrés sur la figure 5.4 [3].

Model	Non-sparse Top-1 Accuracy	Random Sparse Accuracy	Random Sparsity	Structured Sparse Accuracy	Structured Sparsity
InceptionV3	78.1%	78.0%	50%	75.8%	2 by 4
		76.1%	75%		
		74.6%	87.5%		
MobilenetV1 224	71.04%	70.84%	50%	67.35%	2 by 4
MobilenetV2 224	71.77%	69.64%	50%	66.75%	2 by 4

FIGURE 5.4 – Exemple d'amélioration de la compacité de modèle CNN grâce à l'élagage.

Nous constatons pour le modèle InceptionV3, une perte de seulement 0,1% de précision pour une mise à zéro de la moitié des poids. Cette méthode réduit la taille mémoire du modèle, mais pas les besoins en ressources de calcul, car le modèle en lui-même n'est pas modifié. Tout comme la quantification, cette technique peut être utilisée pendant ou après l'entraînement. Notons que quantification et élagage peuvent être combinés.

5.3.4 Modèle professeur-étudiant

Aussi appelée *Knowledge distilation*, cette technique est introduite pour la première fois par [BCNM06] et généralisée ensuite par [HVD15]. Il s'agit d'une méthode de compression pour laquelle un petit modèle (*étudiant*) est entraîné à imiter un plus gros modèle (*professeur* pré-entraîné. La figure 5.5 illustre ce modèle.

3. `https://www.tensorflow.org/model_optimization/guide/pruning?hl=en`

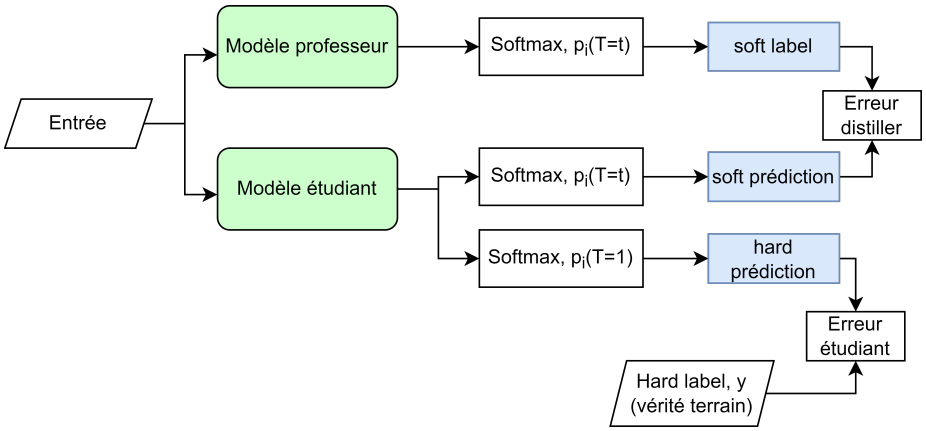

FIGURE 5.5 – Modèle générique professeur-étudiant.

Lors de la distillation, la connaissance est transférée du modèle de l'enseignant à l'étudiant en minimisant une fonction de perte, cf. équation (5.3), dans laquelle la cible est la distribution des probabilités de classes prédites par le modèle de l'enseignant *i.e.* la sortie d'une fonction softmax sur les prédictions du modèle de l'enseignant.

$$\text{Erreur\_distiller} = \alpha * \mathcal{H}\left(y, \sigma\left(z_s; T=1\right)\right) \tag{5.1}$$

$$\text{Erreur\_étudiant} = \beta * \mathcal{H}\left(\sigma\left(z_t; T=\tau\right), \sigma\left(z_s, T=\tau\right)\right) \tag{5.2}$$

$$\mathcal{L}(x; W) = \text{Erreur\_distiller} + \text{Erreur\_étudiant} \tag{5.3}$$

avec x l'entrée, W les paramètres du modèle étudiant, y la vérité terrain, $\mathcal{H}$ la perte d'entropie croisée, σ_i est la fonction softmax paramétrisée par la température T. α et β sont des coefficients et z_s, z_t sont les vecteurs de prédiction brute (ou logits).

Les travaux de [BCNM06] indiquent que $T \in [1, 20]$ et $\beta = 1 - \alpha$ avec $\alpha = \beta = 0,5$ donnent les meilleurs résultats.

Le modèle professeur étant déjà entraîné, les valeurs de la fonction softmax seront élevées. Afin de pallier ce problème, la « softmax temperature » est introduite et définie ainsi :

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} \tag{5.4}$$

Si $T = 1$, p_i est équivalent à la fonction softmax.

Le modèle étudiant peut être professeur avec moins de couches ou alors un modèle élagué [ARBK18] ou quantifié [PPA18], comme détaillé précédemment. Cette technique réduit à la fois l'emplacement mémoire et la latence.

5.3.5 Regroupement des poids du modèle

Ce principe réduit la taille du modèle en remplaçant les poids similaires dans une couche par la même valeur. Ces valeurs sont trouvées en exécutant un algorithme de *clustering* sur les poids entraînés du modèle. Le nombre de *clusters* est déterminé empiriquement. Ensuite chaque matrice de poids est remplacée par l'index de son centroïde. La finalité de cette méthode est très similaire à l'élagage, car les deux ont pour principe de réduire le nombre de poids dans le modèle.

La figure 5.6 [4] montre par exemple pour les modèles MobileNetV1/2 une réduction par deux de la taille mémoire « Size of compressed.tflite » en perdant au maximum 1% de précision « Top-1 accuracy ».

Model	Original		Clustered			
	Top-1 accuracy (%)	Size of compressed .tflite (MB)	Configuration	# of clusters	Top-1 accuracy (%)	Size of compressed .tflite (MB)
MobileNetV1	70.976	14.97	Selective (last 3 Conv2D layers)	16, 16, 16	70.294	7.69
			Selective (last 3 Conv2D layers)	32, 32, 32	70.69	8.22
Full (all Conv2D layers)	32	69.4	4.43			
MobileNetV2	71.778	12.38	Selective (last 3 Conv2D layers)	16, 16, 16	70.742	6.68
			Selective (last 3 Conv2D layers)	32, 32, 32	70.926	7.03
Full (all Conv2D layers)	32	69.744	4.05			

FIGURE 5.6 – Résultats de regroupement de poids pour 4 modèles CNN.

5.4 Performances de notre chaîne de traitement

Cette section vise à analyser quantitativement et qualitativement l'extraction de caractéristiques secondaires et primaires (resp. rouge et orange sur la figure 5.2). Par souci de confidentialité, nous ne pouvons pas divulguer de données (temps d'exécution, caractéristiques matérielles, performances, etc.) liées à la plateforme embarquée intégrée sur les véhicules Renault.

5.4.1 Extraction de caractéristiques secondaires

Le tableau 5.2 synthétise les caractéristiques intrinsèques de chacune de nos implémentations.

4. `https://www.tensorflow.org/model_optimization/guide/clustering`

TABLE 5.2 – Caractéristiques embarquées des modèles proposés dans la section 3.6.1. Np désigne le nombre de paramètres, Um l'usage mémoire et Ti le temps d'inférence ou latence.

Modèle	Modalité	Np	Um	Ti (ms)
Bout-en-bout (BB)	Vidéo	2 056	11,4 kB	5,4
	Audio	58 576	236,9 kB	0,462
	Texte	68 636	279 kB	12,43
	Fusion (FC)	54	1,8 kB	0,0348
	Total	**129 322**	**529,1 kB**	**18,33**
Bout-en-bout + paramétrique (BB+P)	Vidéo-audio	62	2,4 kB	0,0975
	Texte	68 636	279 kB	12,43
	Fusion (GRU)	767	5,3 kB	0,1872
	Total	**69 465**	**286,7 kB**	**12,72**

Nous obtenons un total de 129 322 paramètres pour le modèle (BB) et 69 465 pour le (BB+P). Concernant l'usage mémoire, le modèle (BB) est à 529,1 kB et 286,7 kB pour (BB+P).

Pour la latence, nous l'avons évaluée sur une carte graphique RTX 2080 Ti. Elle est calculée sur un ensemble de test et moyennée sur les inférences. Les mesures représentent la latence pour une fenêtre d'analyse de 35 secondes. L'objectif est d'étudier relativement les temps de latence par modalité au sein d'un même système et entre les deux systèmes (BB) et (BB+P). Pour le (BB), la latence totale est de 18,33 ms, répartie en 12,43 ms (67,82%) pour le modèle texte, 5,4 ms (29,46%) pour la vidéo, 0,462 ms (2,52%) pour le modèle audio et 0,0348 ms (0,18%) pour le modèle de fusion. Nous constatons que le modèle audio possède 28,5 fois plus de paramètres que celui de la vidéo avec une latence 12 fois plus faible. Cela est causé par la quantité de caractéristiques primaires ingérées, pour les modèles RNN, plus la profondeur temporelle (pas de temps) est élevée plus la latence est élevée.

Le modèle (BB+P) possède un temps d'inférence total de 12,72 ms dont 12,43 ms (97,76%) pour la modalité texte, 0,0975 ms (0,76%) pour le modèle audio-vidéo et (1,47%) pour la fusion. La modalité texte est dans les deux cas la plus consommatrice de ressources et possède la latence la plus importante.

Finalement, nous avons amélioré la compacité avec le modèle (BB+P) de **30,62%** pour le temps d'inférence, de **45,81%** pour l'usage mémoire et de **46,28%** pour le nombre de paramètres. Ces modèles sont très peu consommateurs de ressources. En effet, nous sommes à moins de 1 MB de mémoire et très largement inférieur au million de paramètres. Par exemple, si nous avions privilégié un modèle récent de type *transformer* pour analyser le texte (camemBERT [MMOS+20]), ce dernier aurait consommé 966 MB de mémoire pour 110 millions de paramètres. En comparaison, notre modèle HAN consomme 3462 fois moins de mémoire et 1602 fois moins de paramètres. Un modèle *transformer* n'est pas envisageable à l'heure actuelle dans notre système embarqué.

Nous ne pouvons pas implémenter les techniques d'élagage ou de quanti-

fication post-entraînement, car la librairie Pytorch avec laquelle nous avons programmé nos modèles ne les implémente pas encore pour les couches GRU. Les techniques d'optimisation de modèles récurrents sont peu étudiées dans la littérature comparée aux réseaux convolutionnels. Elles sont de ce fait peu implémentées dans les librairies de programmation de réseau de neurones (Tensorflow et Pytorch). Une solution serait de les appliquer durant la phase d'entraînement. Un autre problème est lié à l'exportation des modèles vers le format « mobile ». Tous les nœuds de calculs présents dans le modèle ne peuvent pas être convertis dans ce format. Le modèle HAN en est l'exemple avec une boucle « for » qui ne peut pas être exportée.

5.4.2 Extraction de caractéristiques primaires

Nous détaillons les performances des extracteurs de caractéristiques primaires que nous n'avons pas optimisés. Ainsi, le tableau 5.3 recense les temps d'inférence et les tailles des modèles utilisés pour chaque système (BB et BB+P). Les valeurs sont obtenues avec des inférences sur CPU. Pour rappel, le traitement primaire est le même entre les deux systèmes pour la modalité texte et vidéo avec l'extraction des amers du visage et de l'orientation du visage.

TABLE 5.3 – Latences et utilisation mémoire des extracteurs de caractéristiques primaires utilisés dans le chapitre 4. Np désigne le nombre de paramètres, Um l'usage mémoire et Ti le temps d'inférence ou latence.

Modèles	Extracteurs	Um	Ti (ms)
Bout-en-bout (BB)	Dlib	63,4 MB	20
	hyperface	120,2 MB	49
	OpenSMILE	0	66,7 pour 1s d'audio
	Transcription	0	0,2
Bout-en-bout + paramétrique (BB+P)	Dlib	63,4 MB	20
	hyperface	120,2 MB	49
	Transcription	0	0,2

Nous pouvons évaluer pour 35 secondes de données la latence de nos deux modèles :

— **modèle BB**. La latence audio s'élève à 2,3 secondes ($35 \times 66,7$), la vidéo est enregistrée à 25 images par seconde soit une latence de 60,37 secondes ($35 \times 25 \times (49 + 20)$). Pour rappel nous avons réduit par cinq le calcul des caractéristiques soit une latence totale de 12,07 secondes ($60,37/5$). Pour le texte, une requête/réponse à un service cloud de Google ou Amazon est nécessaire avec une latence d'environ 0,2 seconde. Finalement, pour 35 secondes d'analyse la latence est de 14,37 secondes.

— **modèle BB + P**. L'unique différence pour ce modèle est l'extraction audio qui n'est pas nécessaire. La latence est alors de 12,07 secondes. Au final, la latence est améliorée de 16% et l'utilisation de la mémoire est identique avec un total de 183,6 MB entre les deux modèles.

Les valeurs de latence paraissent conséquentes, notamment sur les modèles images. Elle représente 82,4% du temps pour (BB) et 98% pour le (BB+P). Cela est lié à plusieurs facteurs. Tout d'abord les modèles extrayant les caractéristiques ont été inférés sur un CPU, ensuite les modèles utilisés tels que hyperface ou openSMILE ne sont pas optimisés, même pour des machines de types serveur. L'utilisation d'hyperface comme extracteur d'angle d'orientation du visage peut être évitée en le calculant grâce aux amers fournis par le modèle Dlib. Cela réduirait de 71,5% la latence liée au calcul des caractéristiques du visage, soit 3,5 secondes ($35 \times 25 \times 20/5$). Dans nos travaux, pour l'extraction de caractéristiques primaires, les images sont envoyées au modèle de manière unitaire, un traitement par paquet (*batch*) réduirait considérablement la latence.

Comme nous pouvons le constater, l'extraction de caractéristiques primaires n'est pas le point sur lequel nous avons mis l'accent dans nos travaux. En effet, Qualcomm, qui est le fournisseur des plateformes véhicule, propose tout l'accompagnement logiciel pour l'intelligence artificielle optimisée pour leur matériel. Ils proposent une implémentation d'une multitude de modules élémentaires :

— détection des visages dans l'habitacle,
— inférence de leurs orientations,
— localisation de points anatomiques du visage,
— alignement du visage (correction perspective),
— orientation et état des yeux,
— inférence d'une émotion.

De ce fait, il est plus pertinent d'utiliser ces fonctionnalités totalement optimisées pour le traitement de données vidéo/image.

Un autre point est l'utilisation croisée des applications présentes sur la plateforme. Concrètement, il est possible de réutiliser les caractéristiques extraites par les inférences d'autres applications pour notre tâche. Cette optimisation permet de réduire grandement les besoins en ressources de calcul.

5.4.3 Chaîne de traitement global

Finalement, les latences des deux systèmes peuvent être résumées via la figure 5.7. Dans le cas d'un fonctionnement séquentiel (le moins avantageux) de la chaîne de traitement, la latence est de 12,519 secondes pour (BB) et de 10,512 secondes pour (BB+P). Si nous passons en fonctionnement parallèle, la latence devient égale à la latence la plus grande de la chaîne de traitement. Dans l'ensemble, l'extraction primaire représente 99,84% pour (BB) et 99,88% du temps pour (BB+P). Les optimisations doivent être orientées sur l'extraction primaire afin d'obtenir rapidement des gains en compacité.

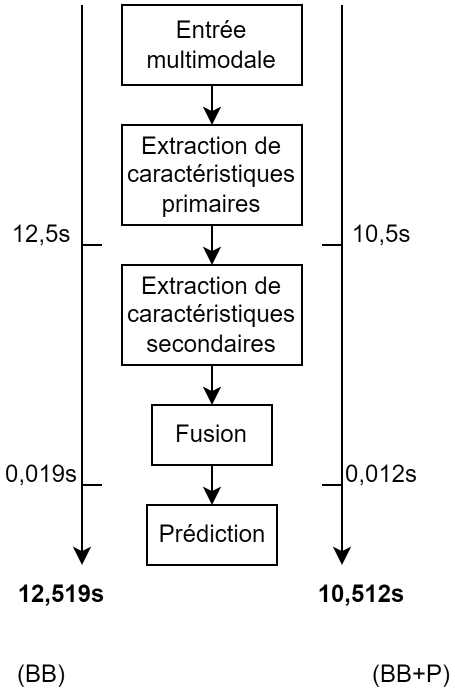

FIGURE 5.7 – Résumé des temps de traitements associés à notre chaîne de traitement.

5.5 Conclusion

Nous avons pris en considération la compacité de deux éléments de la chaîne de traitement des données. Sur la partie « extraction de caractéristiques primaires », nous avons réduit de 16% la latence. Pour « l'extraction de caractéristiques secondaires », nous avons réduit la taille de nos modèles. La conception du (BB+P) est la plus compacte avec une réduction de 34,7% de la latence, de 45,6% sur l'utilisation mémoire et de 46,2% sur le nombre de paramètres comparé au modèle (BB). Nous avons aussi évoqué les difficultés liées au passage de modèle développé sur serveur vers un système embarqué, avec notamment la limitation de certains nœuds de calcul ne pouvant pas être exporté vers un modèle dit « mobile ».

Finalement, les techniques classiques d'optimisation (élagage, quantification) évoquées dans ce chapitre n'ont pas pu être implémentées, car elles ne sont pas disponibles dans la librairie Pytorch pour les modèles RNN post-entraînement. Les principaux gains de compacité peuvent être obtenus en optimisant l'extraction de caractéristiques primaires du traitement image en utilisant les modèles fournis par le constructeur Qualcomm.

Conclusion et perspectives

Synthèse des travaux et plus-values associées

Rappelons que l'objectif de ces investigations est l'évaluation de la faisabilité de l'analyse d'interactions humaines pour la recherche de situations conflictuelles en contexte véhicule. Nous synthétisons ci-après nos principales contributions.

La première, détaillée au chapitre 2, est liée au fait que nous nous sommes basés sur un **corpus public (MOSI), avec implémentation d'un modèle multimodal (audio, vidéo, texte) pour l'analyse de sentiments en prenant en considération la compacité du modèle**. Grâce à cette pré-étude, nous avons confirmé deux de nos intuitions. D'une part, sur la manière de traiter des données hétérogènes avec un modèle par modalité extrayant des caractéristiques suivies d'une fusion plus ou moins tardive. D'autre part, évaluer les performances en augmentant la compacité du modèle : la prise en considération des ressources embarquées n'induit pas de chute drastique de performance.

Pour pallier le manque de corpus public dans notre contexte applicatif, nous avons **enregistré notre propre corpus** *in situ*, exercice par définition chronophage ! Il est composé de scènes réalistes jouées sans script ni jeu d'acteur. Les 22 participants ont été enregistrés dans un véhicule à l'arrêt avec trois scénarios différents (curieux, refus argumenté, refus catégorique) permettant de simuler une gradation de situations conflictuelles. Ce corpus est détaillé au chapitre 3.

Partant de ce corpus, un premier **modèle dit bout-en-bout (noté BB)**, cf. chapitre 4.2, repose sur un modèle capable d'extraire des caractéristiques automatiques pour chacune des modalités pour des fenêtres d'analyse de 35 secondes. Il s'agit de notre modèle de référence. Nous avons une extraction des amers du visage et de l'orientation de la tête pour la modalité vidéo. Concernant la modalité audio, 1581 caractéristiques sont calculées et permettent de capturer les informations d'intensité, hauteur de la voix, etc. Le texte est encodé grâce à un modèle HAN capable de capturer les informations pertinentes au niveau du mot et de la phrase. La temporalité est prise en compte au niveau local avec des fenêtres d'analyse de 35 secondes et à un niveau global grâce au *stateful* RNN.

Nous proposons ensuite un **modèle dit paramétrique (noté BB+P)**, voir chapitre 4.3, afin d'augmenter sa compacité. Le modèle texte est gardé et nous modifions les extracteurs de caractéristiques pour la vidéo et l'audio. La stratégie est alors de calculer sept valeurs extraites manuellement caractérisant le comportement du conducteur et le passager. Elles sont calculées sur des fenêtres d'analyses de 35 secondes, peu coûteuses en ressources de calcul. Pour cette version le contexte local est aussi pris en compte par la durée de la fenêtre d'analyse, le contexte global est pris en compte au niveau du modèle de fusion.

Parmi les différentes **stratégies de fusion** implémentées dans les deux modèles *i.e.* : (1) théorie de l'évidence, (2) couche entièrement connectée, (3) RNN de type GRU et (4) moyenne pondérée, nous constatons que la couche

entièrement connectée et la GRU sont les plus pertinents, car induisant des meilleurs gains de performance. Nous avons donc proposé une implémentation originale de la temporalité afin de capturer l'évolution d'interactions humaines aboutissant à des performances satisfaisantes.

Le tableau 5.4 présente une synthèse des performances de nos deux modèles les plus performants.

TABLE 5.4 – Récapitulatif de nos deux meilleurs systèmes. F désigne le type de fusion, Mp la micro-précision, Nb_p le nombre de paramètres, Tm la taille mémoire et Ti le temps d'inférence.

Modèle	F	Mp	Nb_p	Tm	Ti (ms)
BB	FC	81,6% ± 5,9	129 322	527,3 kB	19,49
BB+P	GRU	81% ± 1,2	69 465	286,7 kB	12,72

En absolu, le modèle naïf (BB) donne de meilleures performances. Pour un modèle plus compact, le (BB+P) est à privilégier. Leurs propriétés (usage mémoire, latence, etc.) les rendent tous deux embarquables. De plus, l'analyse des matrices de confusion associées à ces deux modèles originaux montre une complémentarité. Cela permet d'ouvrir la voie à de futurs travaux sur des modèles ensemblistes ou à une hybridation cloud/embarqué.

L'implémentation de ces deux modèles, leurs évaluations et une étude comparative sur notre corpus (chapitres 4) constituent à nos yeux les contributions principales de cette thèse. Nous avons aussi proposé **un processus d'apprentissage** original. Les modèles ingèrent des sous-séquences ou clips audio-vidéo par ordre chronologique. En combinant ce processus avec le mode *stateful* des RNN, les capacités de mémorisation du contexte global sont améliorées ainsi que les performances de prédiction. La **compacité** ainsi que **les possibilités d'intégration** d'un modèle d'apprentissage profond dans un environnement logiciel Renault déjà existant ont été évoquées. Des investigations supplémentaires sont en cours dans ce sens.

Perspectives

Ces travaux ouvrent de nombreuses perspectives à court et moyen terme. Nous les évoquons dans un potentiel ordre chronologique de réalisation.

Compatibilité embarquée - Comme évoqué au chapitre 5 afin de passer à une application entièrement embarquée, il est nécessaire de posséder uniquement des nœuds de calculs exportables au format « mobile ». Le point bloquant se situe au niveau du modèle HAN qui doit alors être transformé pour se passer de la boucle « for ».

Évaluation du taux de compacité des modèles - Une fois les modèles exportés au format mobile il est possible d'évaluer leurs performances en appliquant les théories axées sur la compacité des modèles présentés au chapitre 5. Implémenter un réseau « professeur-étudiant » pour augmenter la compacité du modèle texte et s'affranchir de la boucle « for » serait une solution tout à fait pertinente.

Évaluation sur plateforme réelle - Pour exploiter ces travaux dans les processus d'industrialisation de Renault, des évaluations précises et chiffrées sont nécessaires. Dans un premier temps, il sera important d'effectuer des mesures d'inférences embarquées sur le CPU. Puis, après optimisation, elles devront être réalisées sur une implémentation parallélisée utilisant les différentes unités de calcul présentes sur la plateforme (GPU, DSL, NPU). Les mesures de latence, utilisation mémoire, consommation énergétique, etc. de toute la chaîne de traitement des données devront être prises en considération.

Approche emsembliste - Les résultats obtenus nous montrent une complémentarité dans les prédictions de nos deux modèles. Tirer parti de ces forces en utilisant des techniques telles que le *boosting* ou *bagging* pourrait permettre d'accroître notablement les performances. Les deux modèles sont par nature compacts, envisager une fusion est alors possible.

Exploitation des autres capteurs du corpus - Pour rappel, notre corpus est composé de six flux vidéo et de quatre flux audio avec différents angles de vue et positionnements du microphone. Nous avons alors la possibilité de fusionner les différents capteurs vidéo et profiter des différents angles de vue pour enrichir la modalité vidéo (qui pour rappel est la moins informative). De même pour l'audio, il est possible de fusionner les différentes sources et/ou faire de l'augmentation de données en considérant chaque flux comme une variante des autres.

Modification de la fonction de coût - Lors de nos travaux, nous avons naturellement utilisé la fonction de coût d'entropie croisée couramment utilisée dans les problèmes de classifications. Cette dernière calcule l'erreur du modèle en pénalisant fortement les erreurs. L'erreur est ensuite propagée au modèle de fusion puis aux modèles unimodaux. La quantité d'information apportée par chacune des branches du modèle est toujours récompensée de la même manière. Une alternative serait de créer une fonction de coût spécifique à l'apprentissage multimodal. Il faudrait par exemple prendre en compte l'erreur apportée par chacune des branches (avant fusion) puis l'erreur du modèle après fusion et les combiner de manière à obtenir une rétropropagation de l'erreur pour chacune des branches en fonction de leurs erreurs respectives.

Mise à l'échelle du corpus - Le système de bout-en-bout présenté au chapitre 4 met en exergue une limitation concernant la quantité de données dans

le corpus. Augmenter le volume de données pourrait réduire considérablement la variance des résultats proposés et augmenter les performances globales. La réutilisation du protocole et du matériel d'enregistrement présenté dans le chapitre 3 permettrait de limiter le temps nécessaire à une nouvelle session de recueil de données.

Annexes

Nous présentons ici les tracés complémentaires de la section 3.6.1 liée aux caractéristiques « Durée moyenne d'une interaction » et « Durée moyenne du temps de parole et du silence ». Les tracés supplémentaires des caractéristiques « contact visuel » et « visibilité du passager » sont ajoutés.

FIGURE 5.8 – Durée moyenne d'une interaction (en sec.) pour le conducteur.

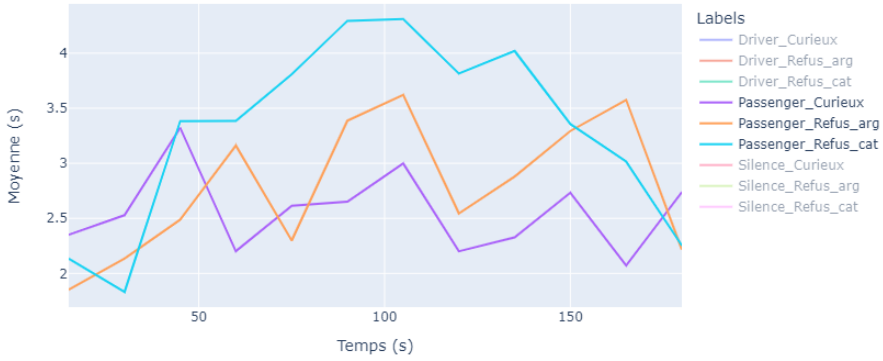

FIGURE 5.9 – Durée moyenne d'une interaction (en sec.) pour le passager arrière.

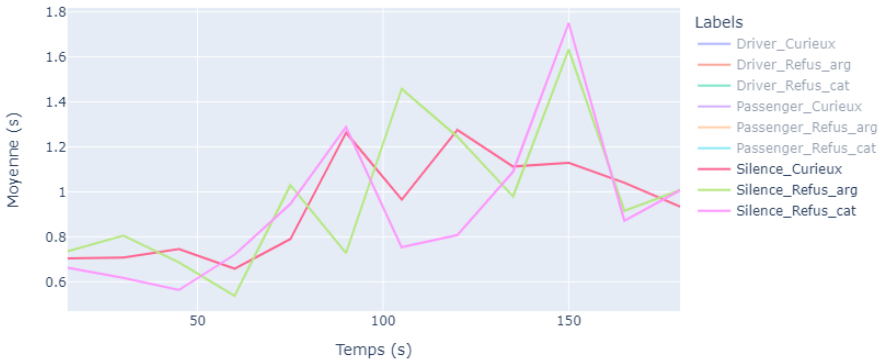

FIGURE 5.10 – Durée moyenne d'un silence (en sec.).

FIGURE 5.11 – Durée moyenne du temps de parole (en sec.) pour le conducteur.

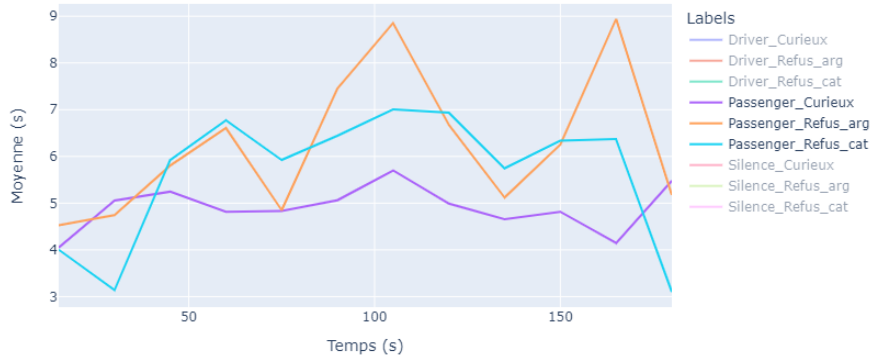

FIGURE 5.12 – Durée moyenne du temps de parole (en sec.) pour le passager arrière.

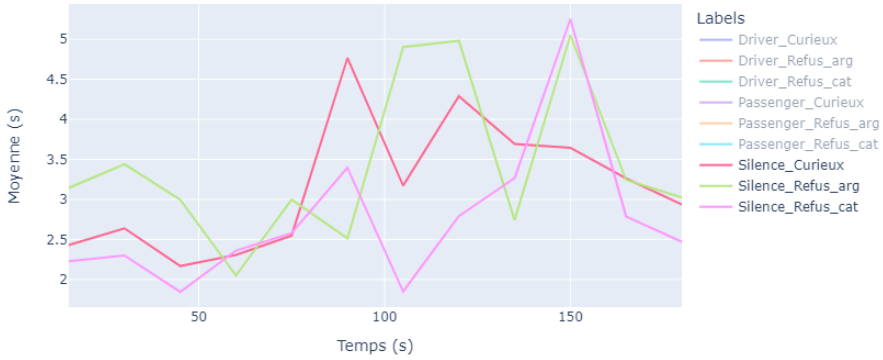

FIGURE 5.13 – Durée moyenne du silence (en sec.).

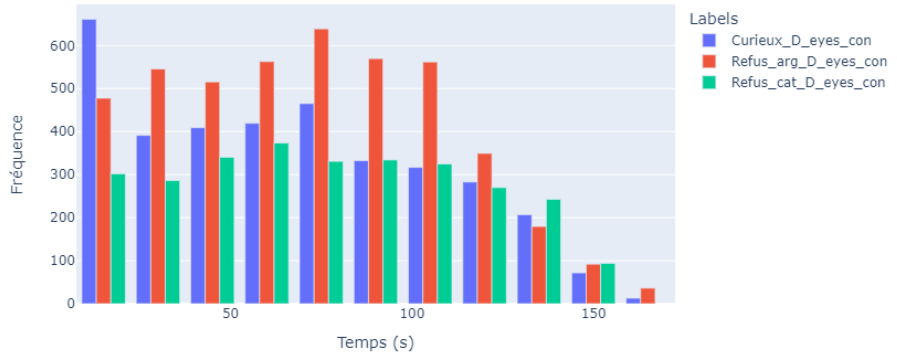

FIGURE 5.14 – Nombre de regards dans le rétroviseur intérieur du conducteur.

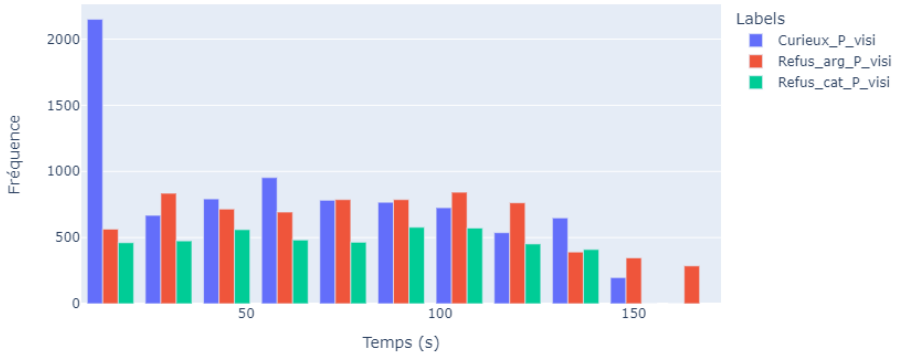

FIGURE 5.15 – Nombre de fois où le passager est visible à la caméra.

Bibliographie

[ARBK18] Anubhav Ashok, Nicholas Rhinehart, Fares Beainy, and Kris M. Kitani. N2n learning : Network to network compression via policy gradient reinforcement learning. In *International Conference on Learning Representations*, 2018.

[AS15] Swapna Agarwalla and Kandarpa Kumar Sarma. Composite feature set for mood recognition in dialectal assamese speech. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 691–695, 2015.

[ATY+19] Md. Zahangir Alom, Tarek Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Nasrin, Mahmudul Hasan, Brian Essen, Abdul Awwal, and Vijayan Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 03 2019.

[AVTP17] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. Realtime convolutional neural networks for emotion and gender classification. *arXiv :1710.07557 [cs]*, 2017.

[AW14] Khawlah Hussein Ali and Tianjiang Wang. Learning features for action recognition and identity with deep belief networks. In *2014 International Conference on Audio, Language and Image Processing*, pages 129–132, 2014.

[AYV19] A. Agarwal, A. Yadav, and D. K. Vishwakarma. Multimodal sentiment analysis via rnn variants. In *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science Engineering (BCD)*, 2019.

[BBL+08] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP : interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4) :335–359, 2008.

[BCNM06] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 535–541. Association for Computing Machinery, 2006.

[BCP+07] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V.S. Subrahmanian. Sentiment analysis : Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[BDNM12] Martin Bruder, Dina Dosmukhambetova, Josef Nerb, and Antony S. R. Manstead. Emotional signals in nonverbal interaction : dya-

dic facilitation and convergence in expressions, appraisals, and feelings. *Cognition & Emotion*, 26(3) :480–502, 2012.

[BH96] Diane S. Berry and Jane Sherman Hansen. Positive affect, negative affect, and social interaction. *Journal of Personality and Social Psychology*, 71(4) :796–809, 1996. Place : US Publisher : American Psychological Association.

[BKY18] Cenk Anil Bahcevan, Emirhan Kutlu, and Tugba Yildiz. Deep neural network architecture for part-of-speech tagging for turkish language. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 235–238, 2018.

[BPFAO10] Benjamin Bigot, Julien Pinquier, Isabelle Ferrané, and Régine André-Obrecht. Looking for relevant features for speaker role recognition (regular paper). In *INTERSPEECH, Makuhari, Japan, 26/09/10-30/09/10*, pages 1057–1060, http ://www.isca-speech.org/, 2010. International Speech Communication Association (ISCA).

[Cab02] Michel Cabanac. What is emotion ? *Behavioural Processes*, 60(2) :69–83, 2002.

[CFSC16] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *CoRR*, abs/1609.04243, 2016.

[CHP+17] Erik Cambria, Devamanyu Hazarika, Soujanya Poria, Amir Hussain, and R. B. V. Subramaanyam. Benchmarking multimodal sentiment analysis. *arXiv :1707.09538 [cs]*, 2017.

[CJK04] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial : Special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1) :1–6, jun 2004.

[CJLV16] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell : A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964, 2016.

[CKT+16] Daniel T. Cordaro, Dacher Keltner, Sumjay Tshering, Dorji Wangchuk, and Lisa M. Flynn. The voice conveys emotion in ten globalized cultures and one remote village in Bhutan. *Emotion*, 16 :117–128, 2016. Place : US Publisher : American Psychological Association.

[CS10] Géraldine Coppin and David Sander. *Théories et concepts contemporains en psychologie de l'émotion.* Systèmes d'interaction émotionnelle. Hermès Science publications-Lavoisier, Paris, 2010. ID : unige :34368.

[CSWS17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[CT09] Hanxing Chen and Jun Tian. Research on the controller area network. In *2009 International Conference on Networking and Digital Society*, volume 2, pages 251–254, 2009.

[CW08] Ronan Collobert and Jason Weston. A unified architecture for natural language processing : Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery.

[CWP18] Stuart Cunningham, Jonathan Weinel, and Richard Picking. High-level analysis of audio features for identifying emotional valence in human singing. In *AM'18 : Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, AM'18, New York, NY, USA, 2018. Association for Computing Machinery.

[CZB+19] Yangkang Chen, Guoyin Zhang, Min Bai, Shaohuan Zu, Zhe Guan, and Mi Zhang. Automatic waveform classification and arrival picking based on convolutional neural network. *Earth and Space Science*, 6(7) :1244–1261, 2019.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[DGXZ19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface : Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[DLLT21] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet : Marrying convolution and attention for all data sizes. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[DOZ+20] Marc Delcroix, Tsubasa Ochiai, Katerina Zmolikova, Keisuke Kinoshita, Naohiro Tawara, Tomohiro Nakatani, and Shoko Araki. Improving speaker discrimination of target speech extraction with time-domain speakerbeam. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 691–695, 2020.

[DTSD15] Laurence Devillers, Marie Tahon, Mohamed El Amine Sehili, and Agnes Delaborde. Détection des états affectifs lors d'interactions parlées : robustesse des indices non verbaux. *Revue TAL*, 2015.

[EP01] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines : Theory and applications. In *Machine Learning and Its Applications*, volume 2049, pages 249–257, 01 2001.

[EWS09] Florian Eyben, Martin Wöllmer, and Björn Schuller. Openear — introducing the munich open-source emotion and affect recognition toolkit. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–6, 2009.

[EWS10] Florian Eyben, Martin Wöllmer, and Björn Schuller. opensmile – the munich versatile and fast open-source audio feature extractor. In *ACM Multimedia*, pages 1459–1462, 01 2010.

[FGL+17] Weijiang Feng, Naiyang Guan, Yuan Li, Xiang Zhang, and Zhigang Luo. Audio visual speech recognition with multimodal recurrent neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 681–688, 2017. ISSN : 2161-4407.

[GDLG17] Tianmei Guo, Jiwen Dong, Henjian Li, and Yunxing Gao. Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 721–724, 2017.

[GGD+20] Sergio González, Salvador García, Javier Del Ser, Lior Rokach, and Francisco Herrera. A practical tutorial on bagging and boosting based ensembles for machine learning : Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, 64 :205–237, 2020.

[GXX+18] Yongbin Gao, Xuehao Xiang, Naixue Xiong, Bo Huang, Hyo Jong Lee, Rad Alrifai, Xiaoyan Jiang, and Zhijun Fang. Human action monitoring for healthcare based on deep learning. *IEEE Access*, 2018.

[HABN+21] Torsten Hoefler, Dan Alistarh, Tan Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in Deep Learning : Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241) :1–124, Sep. 2021.

[HCE+16] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. *CoRR*, abs/1609.09430, 2016.

[HCE+17] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.

[HI20] H. M Mahmudul Hasan and Md. Adnanul Islam. Emotion recognition from bengali speech using rnn modulation-based categorization. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 1131–1136, 2020.

[HJ15] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[HKS17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. *arXiv :1708.07632 [cs]*, 2017.

[HMKM17] Nida Manzoor Hakak, Mohsin Mohd, Mahira Kirmani, and Mudasir Mohd. Emotion analysis : A survey. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 397–402, 2017.

[Hou97] A.J.M. Houtsma. Pitch and timbre : Definition, meaning and use. *Journal of New Music Research*, 26(2) :104–115, 1997.

[Hov15] Eduard H. Hovy. What are Sentiment, Affect, and Emotion ? Applying the Methodology of Michael Zock to Sentiment Analysis. In Núria Gala, Reinhard Rapp, and Gemma Bel-Enguix, editors, *Language Production, Cognition, and the Lexicon*, pages 13–24. Springer International Publishing, Cham, 2015.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 :1735–80, 12 1997.

[HSR18] M. G. Huddar, S. S. Sannakki, and V. S. Rajpurohit. An ensemble approach to utterance level multimodal sentiment analysis. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, pages 145–150, 2018.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[HZC+17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications, 2017.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv :1512.03385 [cs]*, 2015.

[IMS+17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0 : Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[JCL+21] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9772–9781, October 2021.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv :1412.6980 [cs]*, 2017.

[KBL+16] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, Raul Chandias Ferrari, Mehdi Mirza, David Warde-Farley, Aaron Courville, Pascal Vincent, Roland Memisevic, Christopher Pal, and Yoshua Bengio. EmoNets : Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2) :99–111, 2016.

[KHAS21] Ahmed A. Khamees, Hani D. Hejazi, Muhammad Alshurideh, and Said A. Salloum. Classifying audio music genres using cnn and rnn. In Aboul-Ella Hassanien, Kuo-Chi Chang, and Tang Mincong, editors, *Advanced Machine Learning Technologies and Applications*, pages 315–323, Cham, 2021. Springer International Publishing.

[Kin09] Davis E. King. Dlib-ml : A machine learning toolkit. *J. Mach. Learn. Res.*, page 1755–1758, dec 2009.

[KLA+20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[KLN19] Taejun Kim, Jongpil Lee, and Juhan Nam. Comparison and analysis of samplecnn architectures for audio classification. *IEEE Journal of Selected Topics in Signal Processing*, 13(2) :285–297, 2019.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[KZS18] Philip Kinghorn, Li Zhang, and Ling Shao. A region-based image caption generator with refined descriptions. *Neurocomputing*, 272 :416–424, 2018.

[LB98] Yann LeCun and Yoshua Bengio. *Convolutional Networks for Images, Speech, and Time Series*, page 255–258. MIT Press, Cambridge, MA, USA, 1998.

[LB02] Edward Loper and Steven Bird. Nltk : The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective*

Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1, ETMTNLP '02, page 63–70. Association for Computational Linguistics, 2002.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4) :541–551, 12 1989.

[LeD00] Joseph E LeDoux. Emotion circuits in the brain. *Annual Review of Neuroscience*, 23 :155–184, 2000.

[Lin00] Karen D. Lincoln. Social Support, Negative Social Interactions, and Psychological Well-Being. *The Social service review*, 74(2) :231–252, June 2000.

[Lit16] William Little. *Introduction to Sociology - 2nd Canadian Edition*. BCcampus, October 2016.

[Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982.

[LM19] Yi Luo and Nima Mesgarani. Conv-tasnet : Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8) :1256–1266, 2019.

[LS16] Hyo Jung Lee and Maximiliane E. Szinovacz. Positive, negative, and ambivalent interactions with family and friends : Associations with well-being. *Journal of Marriage and Family*, 78(3) :660–679, 2016.

[LSXC17] Huibin Li, Jian Sun, Zongben Xu, and Liming Chen. Multimodal 2d+3d facial expression recognition with deep fusion convolutional neural network. *IEEE Transactions on Multimedia*, 19(12) :2816–2831, 2017. Conference Name : IEEE Transactions on Multimedia.

[LZZ+19] Sunan Li, Wenming Zheng, Yuan Zong, Cheng Lu, Chuangao Tang, Xingxun Jiang, Jiateng Liu, and Wanchuang Xia. Bimodality fusion for emotion recognition in the wild. In *ICMI '19 : 2019 International Conference on Multimodal Interaction*, ICMI '19, page 589–594, New York, NY, USA, 2019. Association for Computing Machinery.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[MMD11] Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. Towards multimodal sentiment analysis : Harvesting opinions from the web. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, page 169–176, New York, NY, USA, 2011. Association for Computing Machinery.

[MMOS+20] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah,

and Benoît Sagot. CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219. Association for Computational Linguistics, 2020.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural and Information Processing System (NIPS)*, 2013.

[MZB10] Dalibor Mitrović, Matthias Zeppelzauer, and Christian Breiteneder. Chapter 3 - features for content-based audio retrieval. In *Advances in Computers : Improving the Web*, volume 78 of *Advances in Computers*, pages 71–150. Elsevier, 2010.

[NKK+11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[NZC19] Prateeth Nayak, David Zhang, and Sek Chai. Bit efficient quantization for deep neural networks. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pages 52–56, 2019.

[PCH+17] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 873–883. Association for Computational Linguistics, 2017.

[PMP20a] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4) :614–626, 2020.

[PMP20b] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Audio features for music emotion recognition : a survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.

[PPA18] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018.

[PR17] Y. H. P. P Priyadarshana and L. Ranathunga. Verb sentiment scoring : A novel approach for sentiment analysis based on adjective-verb-adverb combinations. In *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 533–540, 2017.

[PRMM13] Verónica Pérez-Rosas, Rada Mihalcea, and Louis-Philippe Morency. Utterance-level multimodal sentiment analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 973–982, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014. Association for Computational Linguistics.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10) :1345–1359, 2010.

[PYLP11] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal Processing*, 91(4) :773–781, 2011.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection, 2016.

[RF17] Joseph Redmon and Ali Farhadi. Yolo9000 : Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[RMM18] Neelam Rout, Debahuti Mishra, and Manas Kumar Mallick. Handling imbalanced data : A survey. In M. Sreenivasa Reddy, K. Viswanath, and Shiva Prasad K.M., editors, *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, Singapore, 2018. Springer Singapore.

[RPC19a] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface : A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1) :121–135, 2019.

[RPC19b] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface : A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1) :121–135, 2019.

[RW17] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification : A comprehensive review. *Neural Computation*, 29 :1–98, 2017.

[RZP+20] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993, 2020.

[SCT21] Yanan Song, Yuanyang Cai, and Lizhe Tan. Video-audio emotion recognition based on feature fusion deep learning method. In *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 611–616, 2021.

[SCW+15] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Waikin Wong, and Wang-chun Woo. Convolutional lstm network : A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 802–810, Cambridge, MA, USA, 2015. MIT Press.

[SGLZ11] Ling Shao, Ruoyun Gao, Yan Liu, and Hui Zhang. Transform based spatio-temporal descriptors for human action recognition. *Neurocomputing*, 74(6) :962–973, 2011.

[SGS+17] Saurabh Sahu, Rahul Gupta, Ganesh Sivaraman, Wael AbdAlmageed, and Carol Espy-Wilson. Adversarial Auto-Encoders for Speech Based Emotion Recognition. In *Proc. Interspeech 2017*, pages 1243–1247, 2017.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. Google-Books-ID : wug9DwAAQBAJ.

[SJ88] Karen Sparck Jones. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.

[SJB+18] Yash Singhal, Ayushi Jain, Shrey Batra, Yash Varshney, and Megha Rathi. Review of bagging and boosting classification performance on unbalanced binary classification. In *2018 IEEE 8th International Advance Computing Conference (IACC)*, pages 338–343, 2018.

[SNL19] Apeksha Shewalkar, Deepika Nyavanandi, and Simone Ludwig. Performance evaluation of deep neural networks applied to speech recognition : Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, 9 :235–245, 10 2019.

[SP97] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11) :2673–2681, 1997.

[SS18] Sandeep Saini and Vineet Sahula. Neural machine translation for english to hindi. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–6, 2018.

[SSB09] Björn Schuller, Stefan Steidl, and Anton Batliner. The INTERSPEECH 2009 emotion challenge. In *Proc. Interspeech 2009*, pages 312–315, 2009.

[SSB+10] Bjorn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Muller, and Shrikanth S Narayanan. The INTERSPEECH 2010 paralinguistic challenge. *INTERSPEECH 2010*, page 4, 2010.

[Ste55] S S Stevens. The measurement of loudness. *The Journal of the Acoustical Society of America 27, 815*, page 15, 1955.

[SUD19] Anton Saveliev, Mikhail Uzdiaev, and Malov Dmitrii. Aggressive action recognition using 3d cnn architectures. In *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, 2019.

[SVSS15] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, 2015.

[SWS05] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, page 399–402, New York, NY, USA, 2005. Association for Computing Machinery.

[TBF+15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[TC08] John Tooby and Leda Cosmides. The evolutionary psychology of the emotions and their relationship to internal regulatory variables. In *Handbook of emotions*, page 114–137, 2008.

[TZ15] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[Vai02] Jacqueline Vaissière. Cross-linguistic prosodic transcription : French vs. English. In N.B. Volskaya, N.D. Svetozarova, , and P.A. Skrelin, editors, *Problems and methods of experimental phonetics. In honour of the 70th anniversary of Pr. L.V. Bondarko*, pages pp. 147–164. St Petersburg State University Press, 2002.

[VBP17] Vikas K Vijayan, K. R. Bindu, and Latha Parameswaran. A comprehensive study of text classification algorithms. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1109–1113, 2017.

[VKP+18] Valentin Vielzeuf, Corentin Kervadec, Stéphane Pateux, Alexis Lechervy, and Frédéric Jurie. An occam's razor view on learning audiovisual emotion recognition with small training sets. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, page 589–593, New York, NY, USA, 2018. Association for Computing Machinery.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors,

Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[WASD21] Jin Wu, Yi Yuan An, Qian Wen Shi, and Wei Dai. Behavior recognition algorithm based on the fusion of se-r3d and lstm network. *IEEE Access*, 9 :141002–141012, 2021.

[WPZ18] Jin Wang, Bo Peng, and Xuejie Zhang. Using a stacked residual LSTM model for sentiment intensity prediction. *Neurocomputing*, 322 :93–101, 2018.

[WS13] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[WSS+17] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron : A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017.

[WWK+13] M. Wöllmer, F. Weninger, T. Knaup, B. Schuller, C. Sun, K. Sagae, and L. Morency. Youtube movie reviews : Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3) :46–53, 2013.

[WWL13] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[WZL+20] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 107–122, Cham, 2020. Springer International Publishing.

[XC10] Yan Xu and Lin Chen. Term-frequency based feature selection methods for text categorization. In *2010 Fourth International Conference on Genetic and Evolutionary Computing*, pages 280–283, 2010.

[XUD+19] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai : A brief survey on history, research areas, approaches and challenges. In Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan, editors, *Natural Language Processing and Chinese Computing*, pages 563–574, Cham, 2019. Springer International Publishing.

[XYY+19] Guixian Xu, Ziheng Yu, Haishen Yao, Fan Li, Yueting Meng, and Xu Wu. Chinese text sentiment analysis based on extended sentiment dictionary. *IEEE Access*, 7 :43749–43762, 2019.

[XZ20] Jianqiong Xiao and Zhiyong Zhou. Research progress of rnn language model. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 1285–1288, 2020.

[YBJ18] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118, 2018.

[YCY17] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet : A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions. *CoRR*, abs/1703.10847, 2017.

[YKYS17] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *ArXiv*, abs/1702.01923, 2017.

[YNC+18] Yongzhe Yan, Xavier Naturel, Thierry Chateau, Stefan Duffner, Christophe Garcia, and Christophe Blanc. A survey of deep facial landmark detection. In *RFIAP*, Paris, France, June 2018.

[YYD+16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[ZWY+13] Jun Zhu, Baoyuan Wang, Xiaokang Yang, Wenjun Zhang, and Zhuowen Tu. Action recognition with actons. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[ZZPM16] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi : Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos, 2016.

[ZZZ+21] Fangrui Zhu, Yi Zhu, Li Zhang, Chongruo Wu, Yanwei Fu, and Mu Li. A unified efficient pyramid transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2667–2677, October 2021.