



République Tunisienne  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Tunis El Manar  
École Nationale d'Ingénieurs de Tunis



## Projet de Fin d'Année 2

Développement d'un système de  
reconnaissance des états émotionnels  
par analyse des données physiologiques  
et apprentissage machine.

*Elaboré par :*

Wajih Ounis

Mohamed Aziz Lachtar

*Encadré par :*

M. Zied LACHIRI

2<sup>ème</sup> Année informatique

2<sup>ème</sup> Année informatique

Année universitaire : 2022/2023



# Table des matières

<b>Table des figures</b>	<b>4</b>
<b>Abstrait</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>1 Classification des émotions à partir des signaux physiologiques :</b>	
<b>Etat de l'art</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Notions sur les émotions . . . . .	4
1.2.1 Définition de l'émotion . . . . .	4
1.2.2 Représentation des émotions . . . . .	4
1.3 L'activité physiologique et l'activation émotionnel . . . . .	7
1.3.1 Les modifications physiologiques concomitant des émotions .	7
1.3.2 Recherche antérieure sur la reconnaissance des émotions à partir des signaux physiologiques . . . . .	8
1.4 Défis et orientations futures . . . . .	10
1.5 Conclusion . . . . .	12
<b>2 Analyse des données physiologiques et conception du modèle de classification</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Les signaux physiologiques . . . . .	13
2.3 Méthodologie de collecte des données physiologiques . . . . .	16
2.3.1 Plan de l'expérience . . . . .	16
2.3.2 Interface d'annotation . . . . .	16
2.3.3 Les Vidéos . . . . .	17
2.3.4 Extraction de caractéristiques . . . . .	18
2.4 Méthodologie d'apprentissage machine utilisée . . . . .	21
2.4.1 Les bibliothèques et les frameworks . . . . .	21
2.4.2 Préparation des données . . . . .	22

2.4.3	Prétraitement des données . . . . .	22
2.4.4	Construction de modèle . . . . .	23
2.5	Explication de la méthode de l'évaluation et la validation des résultats	24
2.6	Conclusion . . . . .	26
<b>3</b>	<b>Implémentation et interprétation des résultats</b>	<b>28</b>
3.1	Introduction . . . . .	28
3.2	Prédiction des étiquettes des vidéos à partir des signaux physiologiques	28
3.2.1	Implémentation . . . . .	29
3.2.2	Évaluation et interprétation des résultats . . . . .	34
3.3	Prédiction des couples moyennes (arousal, valence) à partir des signaux physiologiques : modèle KNN . . . . .	40
3.3.1	Implémentation . . . . .	40
3.3.2	Interprétation des résultats . . . . .	44
3.4	Prédiction des couples moyennes (arousal, valence) à partir des signaux physiologiques : modèle CNN . . . . .	46
3.4.1	Implémentation . . . . .	47
3.4.2	Interprétation des résultats . . . . .	52
3.4.3	Résumé du modèle . . . . .	54
3.5	Conclusion . . . . .	56
	<b>Conclusion</b>	<b>57</b>
	<b>Références Bibliographiques</b>	<b>58</b>

# Table des figures

1.1	Liste des émotions basiques selon différents auteurs. . . . .	5
1.2	Quelques axes choisis par différents auteurs. . . . .	6
1.3	La représentation de quelques émotions sur deux axes. . . . .	7
1.4	Ensembles des projets antérieure dans la classification émotionnel à partir des signaux physiologiques. . . . .	10
2.1	Amusement. . . . .	15
2.2	Peur. . . . .	15
2.3	La configuration typique de l'expérience montre un participant regardant une vidéo et annotant à l'aide de JERI. . . . .	17
2.4	Le graphique de gauche montre les annotations d'un participant pour les différentes vidéos (voir Tableau 1) de l'expérience. Les annotations pour la vidéo "effrayant-2" par les cinq premiers participants (étiquetés comme p1-p5) peuvent être observées dans le graphique de droite. . . . .	17
2.5	La source, l'étiquette, l'ID utilisé, les attributs de valence-arousal prévus et la durée des vidéos utilisées dans la base de données. . . .	18
2.6	Les capteurs et les caractéristiques sélectionnées de chaque capteur.	19
2.7	"Violin" plots montrant la distribution des caractéristiques sélectionnées et des valeurs moyennes d'annotation (valence et arousal) pour différents types de vidéos. Les boîtes intégrées dans chaque "violin" plot montrent la plage interquartile (IQR) pour chaque variable considérée, tandis qu'un diamant jaune marque la moyenne de la distribution. . . . .	20
3.1	Code de chargement données des participants. . . . .	29
3.2	Fonction de synchronisation et d'échantillonnage des signaux physiologiques et des annotations émotionnelles. . . . .	30
3.3	Dataframe résultant. . . . .	30
3.4	Code de prétraitement des données. . . . .	31
3.5	Code de division des données en ensembles d'entraînement et de test.	32

3.6	Entraînement du modèle de prédiction des étiquettes de vidéo à partir des signaux physiologiques. . . . .	33
3.7	Les résultats obtenus. . . . .	34
3.8	Les résultats obtenus. . . . .	35
3.9	La fonction matrice de confusion. . . . .	35
3.10	La matrice de confusion obtenue. . . . .	36
3.11	Le graphique de la perte. . . . .	37
3.12	Le graphique de l'accuracy. . . . .	37
3.13	Le graphe du rappel. . . . .	38
3.14	Le graphe de la précision. . . . .	39
3.15	Le graphe de l'exactitude . . . . .	39
3.16	Code de création des nouveaux étiquettes. . . . .	40
3.17	Dataframe résultant. . . . .	41
3.18	Fonction de transformation des caractéristiques. . . . .	41
3.19	Fonction de transformation. . . . .	41
3.20	Fonction d'entraînement pour le KNN. . . . .	42
3.21	Boucle d'entraînement pour le KNN. . . . .	43
3.22	Graphique d'exactitude pour le KNN. . . . .	45
3.23	Graphique de précision pour le KNN. . . . .	45
3.24	Graphique de recall pour le KNN. . . . .	46
3.25	Code de création des nouveaux étiquettes. . . . .	47
3.26	Dataframe résultant. . . . .	48
3.27	Fonction de transformation des caractéristiques. . . . .	48
3.28	Fonction de transformation. . . . .	48
3.29	Ensembles des projets antérieure dans la classification émotionnel à partir des signaux physiologiques. . . . .	49
3.30	Fonction d'entraînement du modèle. . . . .	50
3.31	Boucle d'entraînement. . . . .	51
3.32	Graphique de précision 1. . . . .	52
3.33	Graphique de recall 1. . . . .	53
3.34	Graphique de précision 2. . . . .	54
3.35	Graphique de recall 2. . . . .	54
3.36	Résumé du modèle CNN. . . . .	55



# Abstrait

La reconnaissance des émotions humaines est une tâche complexe dans le domaine de l'informatique. Dans ce projet, nous avons proposé une approche de reconnaissance automatique des émotions basée sur des signaux physiologiques. Notre méthodologie consistait à extraire des informations caractéristiques des signaux physiologiques, suivie d'une étape de classification à l'aide d'un modèle CNN. Nous avons utilisé l'ensemble de données CASE, qui fournit des annotations continues des émotions ressenties par les participants lors du visionnage de vidéos. Nos résultats ont montré que notre méthode proposée a atteint un taux de précision élevé de 85% pour la reconnaissance des émotions, démontrant ainsi son potentiel d'application dans des scénarios réels. Notre étude contribue au développement de méthodes efficaces et non invasives pour la reconnaissance des émotions, qui pourraient être utiles dans divers domaines tels que la santé mentale, le divertissement et l'interaction homme-machine.



# Introduction

Les émotions jouent un rôle crucial dans le comportement humain et ont un impact significatif sur divers aspects de notre vie, notamment la communication, la prise de décision et le bien-être. La compréhension et l'analyse des émotions ont suscité un grand intérêt dans de nombreux domaines, notamment la psychologie, les neurosciences et l'interaction homme-machine. Les progrès récents des technologies d'acquisition des signaux physiologiques ont ouvert de nouvelles possibilités d'analyse émotionnelle en capturant en temps réel les réponses physiologiques associées aux émotions.

Un ensemble de données notable dans ce domaine est l'ensemble de données CASE (Continuously Annotated Signals of Emotion), qui a été développé pour pallier le manque d'ensembles de données physiologiques annotées de manière continue et simultanée pour l'informatique affective. CASE (Continuously Annotated Signals of Emotion), qui fournit des données continues et en temps réel de signaux physiologiques et d'annotations d'émotions ressenties par les participants lorsqu'ils regardent des vidéos.

L'objectif de ce projet est de fournir une approche efficace et précise pour la reconnaissance des émotions à partir des signaux physiologiques, avec des applications potentielles dans différents domaines.

Dans ce rapport, nous présentons une vue d'ensemble de l'état de l'art de l'analyse émotionnelle à l'aide de signaux physiologiques, en mettant l'accent sur l'ensemble de données CASE. Nous allons décrire en détail les différentes étapes de notre projet, de la préparation des données à l'évaluation des résultats, en passant par l'implémentation et l'interprétation des résultats. Nous soulignons les avancées actuelles, les défis et les orientations futures dans ce domaine, sur la base de la recherche et de la littérature existantes.

# Chapitre 1

## Classification des émotions à partir des signaux physiologiques : Etat de l'art

### 1.1 Introduction

Les émotions sont un aspect fondamental de l'expérience humaine, influençant tout, de la prise de décision aux interactions sociales. Cependant, la prévision précise des états émotionnels a toujours été une tâche difficile, car les émotions sont très subjectives et peuvent être influencées par divers facteurs. Les progrès technologiques récents, y compris l'apprentissage automatique et l'intelligence artificielle, ont ouvert de nouvelles possibilités pour prédire les états émotionnels avec plus de précision.

Dans ce chapitre, nous explorerons l'état actuel de la prédiction émotionnelle, en examinant les dernières recherches et développements dans le domaine. J'examinerai les diverses méthodes utilisées pour recueillir et analyser les signaux émotionnels, y compris les signaux physiologiques comme la fréquence cardiaque et la conductance cutanée. Nous explorerons également les dernières techniques d'apprentissage automatique et d'intelligence artificielle qui sont utilisées pour analyser ces signaux et faire des prédictions sur les états émotionnels.

## 1.2 Notions sur les émotions

### 1.2.1 Définition de l'émotion

L'émotion humaine peut être définie comme une expérience psychologique et physiologique complexe qui découle de l'évaluation d'une personne de son environnement et de son état interne. Les émotions impliquent un éventail de sentiments subjectifs, comme la joie, la colère, la peur, la tristesse et le dégoût, qui sont souvent accompagnés de sensations physiques, comme des changements dans la fréquence cardiaque, la respiration et la tension musculaire.

Les émotions jouent un rôle crucial pour façonner notre comportement et nous aider à naviguer dans notre environnement social et physique. Ils peuvent nous fournir de précieux renseignements sur nos besoins et nos désirs, nous aider à prendre des décisions et nous motiver à agir. En même temps, les émotions peuvent aussi être perturbatrices et accablantes, ce qui entraîne des conséquences négatives comme l'anxiété, la dépression et les conflits interpersonnels.

Comprendre les émotions humaines exige une approche interdisciplinaire qui s'appuie sur la psychologie, les neurosciences et la physiologie, entre autres domaines.

### 1.2.2 Représentation des émotions

Il existe de nombreux problèmes dans la manipulation informatique des émotions. Premièrement au niveau de leurs représentations, il s'agit de trouver un formalisme en accord avec les résultats psychologiques, tout en permettant une manipulation. Ensuite, à un événement, il faut pouvoir déterminer le potentiel émotionnel associé à celui-ci. En fonction du travail psychologique, certaines mesures tiennent compte des états émotionnels en tant que catégories, les autres en tant que construction multidimensionnelle.

#### L'approche catégorique

L'approche catégorique de la représentation des émotions est essentiellement une façon d'organiser les émotions en catégories ou types distincts. Cette approche suppose que les émotions peuvent être définies par un ensemble de caractéristiques uniques qui sont caractéristiques de ce type d'émotion. Par exemple, l'émotion de la colère est souvent caractérisée par des sentiments d'hostilité, de frustration et d'irritabilité, tandis que l'émotion de la tristesse est souvent caractérisée par des sentiments de perte, de déception et de désespoir. Chacune de ces émotions a un modèle distinct d'arousal physiologique, d'expérience subjective et d'expression comportementale, qui sont censés être spécifiques à cette émotion.

Afin d'étudier les émotions à l'aide de l'approche catégorielle, les chercheurs utilisent souvent des questionnaires d'auto-déclaration ou des entrevues structurées pour évaluer l'expérience d'une personne de différentes émotions. Ces mesures demandent généralement aux personnes d'évaluer l'intensité de leur expérience émotionnelle pour une variété d'émotions, comme la colère, la tristesse, la joie et la peur, entre autres. À partir de ces cotes, les chercheurs peuvent déterminer la prévalence de différentes émotions au sein d'une population, ainsi que la façon dont ces émotions sont liées à d'autres facteurs, comme la personnalité, le contexte social ou la santé mentale.

Bien que l'approche catégorique puisse être utile pour comprendre la structure de base des émotions et la façon dont elles sont vécues et exprimées, elle a des limites. Par exemple, cette approche peut négliger les différences individuelles dans l'expérience émotionnelle ou l'interaction complexe entre différentes émotions. De plus, certains chercheurs soutiennent que l'approche catégorielle simplifie à outrance la complexité des émotions en les réduisant à des catégories distinctes. Dans l'ensemble, l'approche catégorique de la représentation des émotions est une façon de comprendre et d'étudier les émotions en les organisant en catégories distinctes. Cependant, cette approche devrait être envisagée conjointement avec d'autres approches, comme l'approche dimensionnelle, qui peut fournir une compréhension plus nuancée des émotions en tant que processus dynamiques et continus .

Auteur	Émotions basiques
Ekman et al. [77]	colère, dégoût, peur, joie, tristesse, surprise.
Izard [105]	colère, mépris, dégoût, détresse, peur, culpabilité, intérêt, joie, honte, surprise.
Plutchik [170]	acceptation, colère, anticipation, dégoût, peur, joie, tristesse, surprise.
Tomkins [205]	colère, intérêt, mépris, dégoût, détresse, peur, joie, honte, surprise.

FIGURE 1.1 – Liste des émotions basiques selon différents auteurs.

## L'approche dimensionnelle

L'approche dimensionnelle de la représentation des émotions suggère que les émotions sont mieux comprises comme existant sur un continuum, avec des émotions différentes variant en intensité et en valence. Valence désigne la qualité positive ou négative d'une émotion, tandis que l'intensité désigne le degré d'arousal ou d'activation associé à une émotion. Cette approche suggère que les émotions peuvent être décrites en fonction de leur position sur différentes dimensions, comme le modèle valence-éveil ou le modèle circumplexe. Dans ces modèles, les émotions

Auteur	Axe choisi
Russel [183]	Arousal/Valence
Cowie et al.[54]	Activation/ Évaluation

FIGURE 1.2 – Quelques axes choisis par différents auteurs.

sont représentées sous forme de points sur un espace bidimensionnel, la valence (positive-négative) sur un axe et l’arousal (basse-haute) sur l’autre. Par exemple, la joie pourrait être représentée comme haute valence et haute arousal, tandis que la tristesse pourrait être représentée comme basse valence et basse arousal. Cette approche est souvent utilisée dans l’étude des émotions à l’aide de mesures physiologiques. L’approche dimensionnelle des émotions est souvent utilisée dans la recherche qui implique des mesures physiologiques. Les chercheurs peuvent utiliser ces mesures pour évaluer comment différentes émotions sont associées à des changements dans l’activité cérébrale ou d’autres réponses physiologiques. En utilisant une approche dimensionnelle, les chercheurs peuvent mesurer plus précisément l’intensité et la valence des expériences émotionnelles, et mieux comprendre comment les émotions sont liées aux différentes réponses physiologiques.

En résumé, l’approche dimensionnelle de la représentation des émotions suggère que les émotions sont mieux comprises comme existant sur un continuum, avec différentes émotions variant dans leur intensité et leur valence. Cette approche utilise un modèle bidimensionnel pour représenter les émotions, la valence (positive ou négative) sur un axe et l’arousal (basse à haute) sur l’autre axe. Cette approche peut être utile pour étudier les émotions dans la recherche qui implique des mesures physiologiques.

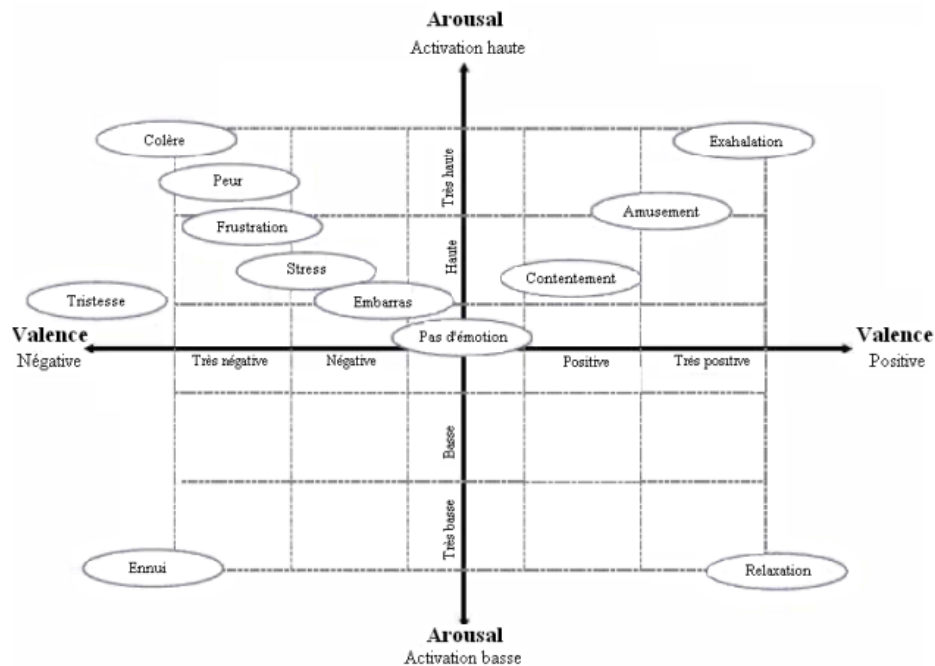


FIGURE 1.3 – La représentation de quelques émotions sur deux axes.

## 1.3 L'activité physiologique et l'activation émotionnel

Les émotions sont associées à des modèles spécifiques de changements physiologiques dans le corps, tels que des changements dans la fréquence cardiaque, la pression artérielle, la respiration, la tension musculaire et la conductance cutanée. Ces changements font partie de la réponse du système nerveux autonome du corps et sont activés lorsque nous éprouvons des émotions différentes. Bien que ces changements physiologiques puissent fournir de l'information sur les expériences émotionnelles, ils ne mesurent pas directement les émotions eux-mêmes. En mesurant ces changements, les chercheurs peuvent mieux comprendre comment les émotions sont vécues et exprimées.

### 1.3.1 Les modifications physiologiques concomitant des émotions

L'activation émotionnelle désigne l'expérience subjective de ressentir une émotion, comme se sentir heureux, triste ou anxieux. Cette expérience s'accompagne généralement de changements dans l'activité physiologique du corps, comme des

changements dans la fréquence cardiaque, la respiration et la tension musculaire. Ces changements physiologiques sont souvent appelés la réponse du corps "combat ou fuite", qui est une réponse automatique aux menaces ou défis perçus.

On pense que les changements physiologiques associés à l'activation émotionnelle sont en grande partie contrôlés par le système nerveux autonome (SNA), qui est responsable de la régulation de nombreuses fonctions automatiques du corps, comme la fréquence cardiaque et la respiration. Le SNA a deux branches principales, le système nerveux sympathique et le système nerveux parasympathique. Le système nerveux sympathique est responsable de l'activation de la réponse du corps "combat ou fuite", tandis que le système nerveux parasympathique aide à calmer le corps après la menace ou le défi a passé.

La recherche a montré que différentes émotions peuvent être associées à différents modèles d'activité physiologique dans le corps. Par exemple, la peur est souvent associée à une fréquence cardiaque accrue et à une respiration rapide, tandis que la tristesse est souvent associée à une diminution de la fréquence cardiaque et à un ralentissement de la respiration.

La relation entre l'activité physiologique et l'activation émotionnelle est complexe et bidirectionnelle. Autrement dit, l'activité physiologique peut influencer l'activation émotionnelle, et l'activation émotionnelle peut également influencer l'activité physiologique. Par exemple, si une personne est exposée à une situation stressante, comme un discours public, ses réponses physiologiques (comme une fréquence cardiaque accrue et une respiration rapide) peuvent contribuer à des sentiments d'anxiété ou de peur. Dans le même temps, l'expérience de l'anxiété ou de la peur peut également conduire à d'autres réponses physiologiques, telles que la transpiration ou le tremblement.

### **1.3.2 Recherche antérieure sur la reconnaissance des émotions à partir des signaux physiologiques**

Selon notre recension des écrits, presque tous les auteurs ont utilisé des classificateurs d'apprentissage automatique pour reconnaître les émotions. Les plus utilisées ont été les machines vectorielles de soutien (SVM)[12], suivies par K-voisins les plus proches (KNN)[5]. Les autres classificateurs utilisés sont Naive Bayes (NB), Multi-Yer Perceptron (MLP)[8], Logistic Regression (LR)[6] et Random Forest (RF)[11]. En plus de ces travaux qui utilisent des étiquettes artisanales, il existe d'autres solutions qui utilisent des approches d'apprentissage profond, comme les réseaux neuronaux artificiels (ANN)[13], les réseaux d'apprentissage profond (RAD)[**RAD**], les réseaux de croyances profondes (DBN)[4], les réseaux de mémoire à court terme (LSTM)[7] ou une combinaison de ces derniers avec le réseau neural convolutionnel (CNN)[10].

Le problème de classification des émotions s'est posé de l'une des trois façons suivantes :

- (i) identification d'émotions discrètes comme le bonheur, la peur ou le dégoût
- (ii) distinction entre la haute/basse arousal et la haute/basse valence
- (iii) recherche du quadrant, dans l'espace valence/éveil .

Dans les deux derniers cas, les chercheurs créent deux classificateurs, l'un pour discerner entre haute/basse valence et l'autre pour haute/basse arousal. Bien que la classification binaire soit la plus courante, il existe des travaux dans lesquels des chercheurs ont effectué la classification multi-classes . Il y a aussi des œuvres qui ont inclus toutes les émotions positives dans une classe et toutes les émotions négatives dans une autre, et parfois avec l'ajout de la classe neutre .

La figure résume les principales caractéristiques d'un sous-ensemble d'articles examinés, qui comprennent la base de données utilisée, les ondes cérébrales utilisées, les caractéristiques extraites, les classificateurs utilisés et l'ensemble des émotions reconnues. Ces œuvres ont été choisies en fonction de leur pertinence et de leur nouveauté.



MANHOB-HCI	PSD, DPSA ( $\theta, \alpha, \beta, \gamma$ )	SVM	arousal (3); valence (3)
DEAP	PSD, APSD ( $\theta, \alpha, \beta, \gamma$ )	NB	arousal (2); valence (2)
Video (Own)	SampEn, Spectral Centroid ( $\alpha, \beta, \gamma$ )	KNN, PNN	disgust, happy, surprise, fear and neutral (5)
DEAP	PSD ( $\theta, \alpha, \beta, \gamma$ )	DLN, SVM, NB	arousal (3); valence (3)
Video (Own)	DPSA, WT, WE, AE, FD, HE ( $\delta, \theta, \alpha, \beta, \gamma$ )	SVM	positive and negative (2)
DEAP	Pearson correlation, Phase coherence, MI ( $\theta, \alpha, \beta, \gamma$ )	SVM	arousal (2); valence (2)
SEED	PSD, DE, Differential/Rational asymmetry ( $\delta, \theta, \alpha, \beta, \gamma$ )	KNN, LR, SVM, DBN	positive, negative and neutral (3)
DEAP	PSD, STFT, HHS, HOC ( $\theta, \alpha, \beta, \gamma$ )	RF, SVM	anger, surprise, other (3)
DEAP	Statistical, PSD, HP, FD ( $\theta, \alpha, \beta, \gamma$ )	SVM	arousal (2); valence (2)
DEAP	WP, WE ( $\theta, \alpha, \beta, \gamma$ )	SVM, KNN	arousal (2); valence (2)
DEAP, Music	PSD, FD, differential asymmetry ( $\delta, \theta, \alpha, \beta, \gamma$ )	SVM, MLP, C4.5	arousal (2); valence (2)

FIGURE 1.4 – Ensembles des projets antérieure dans la classification émotionnel à partir des signaux physiologiques.

## 1.4 Défis et orientations futures

Variabilité des données : L'un des défis de l'analyse émotionnelle à l'aide de signaux physiologiques est la variabilité des réponses physiologiques d'un individu à l'autre, et même au sein d'un même individu au fil du temps. Les signaux physiologiques peuvent être affectés par divers facteurs tels que l'âge, le sexe, l'état de santé et les facteurs environnementaux, ce qui peut introduire de la variabilité dans les données. Le traitement de cette variabilité nécessite des modèles robustes et adaptables, capables de gérer les différences individuelles et de se généraliser à différentes populations et conditions.

Qualité des données et annotation : La qualité des données physiologiques et

la précision des annotations des émotions peuvent également constituer des défis pour l'analyse émotionnelle. Les signaux physiologiques sont sensibles au bruit, aux artefacts et aux erreurs de mesure, ce qui peut affecter la fiabilité de l'analyse. En outre, les annotations des émotions peuvent également être subjectives et varier en fonction de la méthode d'annotation ou des critères utilisés. Il est essentiel de garantir des données de haute qualité et des annotations précises des émotions pour construire des modèles de reconnaissance des émotions fiables et valides.

**Considérations éthiques :** L'analyse des émotions à l'aide de signaux physiologiques soulève des considérations éthiques liées à la protection de la vie privée, au consentement et aux biais potentiels. La collecte et l'analyse de données physiologiques, qui sont considérées comme sensibles et privées, nécessitent un examen attentif des lignes directrices éthiques et des réglementations en matière de protection des données. Le consentement et la transparence dans la collecte, l'utilisation et le partage des données doivent être garantis, et des mesures doivent être prises pour atténuer les biais potentiels dans les données et les modèles afin de garantir une analyse émotionnelle juste et éthique.

**Interprétabilité des modèles :** Les modèles d'apprentissage profond utilisés dans l'analyse émotionnelle, tels que les réseaux neuronaux, peuvent être complexes et difficiles à interpréter, ce qui complique la compréhension et l'interprétation des représentations apprises et des processus de prise de décision. L'interprétabilité des modèles est importante pour renforcer la confiance dans les systèmes de reconnaissance des émotions et pour mieux comprendre les mécanismes physiologiques sous-jacents des émotions. Le développement de modèles interprétables ou de méthodes permettant d'expliquer les décisions prises par les modèles constitue une orientation importante pour les recherches futures.

**Déploiement dans le monde réel :** Bien que la recherche sur l'analyse émotionnelle à l'aide de signaux physiologiques ait considérablement progressé, le déploiement dans le monde réel et les applications pratiques de ces modèles sont encore limités. Pour traduire les résultats de la recherche en applications pratiques dans le monde réel, telles que les soins de santé, l'interaction homme-machine ou les applications informatiques affectives, il faut relever les défis liés à l'extensibilité, à la robustesse et à la facilité d'utilisation des modèles. Il est nécessaire de poursuivre les recherches sur l'élaboration de solutions pratiques et évolutives pour le déploiement dans le monde réel.

**Contextes multiculturels et multilingues :** Les émotions sont influencées par la culture et la langue, et les réponses physiologiques aux émotions peuvent varier d'une culture et d'une langue à l'autre. La prise en compte des contextes multiculturels et multilingues dans l'analyse émotionnelle à l'aide de signaux physiologiques est une orientation future importante. Le développement de modèles sensibles à la culture et adaptables à la langue, capables de prendre en compte les différences

interculturelles et multilingues dans les réponses physiologiques aux émotions, peut améliorer l'applicabilité et la généralisation des modèles de reconnaissance des émotions.

## 1.5 Conclusion

En conclusion, la classification émotionnelle fondée sur les signaux physiologiques est un domaine de recherche actif depuis plusieurs décennies. La définition et la compréhension des émotions ont évolué au fil du temps, et les chercheurs ont identifié de nombreux signaux physiologiques qui sont associés à des états émotionnels spécifiques. Les travaux précédents sur ce sujet ont utilisé une variété de techniques pour extraire les caractéristiques et classer les émotions, y compris les classificateurs d'apprentissage automatique tels que les machines vectorielles de soutien, K-voisins les plus proches, Bayes naïfs, Perceptron multi-Layer, régression logistique, forêt aléatoire, Réseaux de neurones artificiels, réseaux d'apprentissage profond, réseaux de croyances profondes, réseaux de mémoire à court terme à long terme, et une combinaison de ceux-ci avec les réseaux de neurones convolutionnels. Le problème de classification des émotions a été abordé de diverses façons, y compris l'identification des émotions discrètes, la distinction entre haute/basse arousal et haute/basse valence, et trouver le quadrant dans l'espace valence/arousal.

Dans l'ensemble, bien que des progrès importants aient été réalisés dans le domaine de la classification émotionnelle fondée sur des signaux physiologiques, il reste des défis à relever, le développement de techniques d'extraction de caractéristiques plus robustes et l'amélioration de la précision des classificateurs. Des recherches plus poussées dans ce domaine pourraient fournir des renseignements précieux sur la relation entre la physiologie et les émotions.

# Chapitre 2

## Analyse des données physiologiques et conception du modèle de classification

### 2.1 Introduction

L'évaluation des émotions en informatique reste difficile en raison des contraintes de l'inspection directe et en temps réel dans des environnements réalistes. L'ensemble de données CASE (Continuously Annotated Signals of Emotion) offre une solution en se concentrant sur l'annotation continue et en temps réel des émotions ressenties par les participants tout en regardant des vidéos.

Dans ce chapitre, nous nous intéressons tout d'abord à décrire les différents signaux physiologiques (conductance de la peau, volume sanguin périphérique, volume respiratoire, signal électromyographie et température cutanée) utilisés pour la prédiction émotionnelle, ainsi que leurs relations avec les processus émotionnels. Ensuite, nous allons décrire la méthodologie utilisée pour l'analyse des données émotionnelles et la méthode d'apprentissage machine que nous avons utilisée pour ce projet. Enfin nous allons parler des méthodes de validation des résultats.

### 2.2 Les signaux physiologiques

- **Le signal ECG :** Le signal ECG permet de vérifier le bon fonctionnement du cœur chez un patient. Grâce à l'électrocardiographe, le personnel soignant peut visualiser la représentation graphique de l'activité électrique du cœur. L'analyse du signal ECG est utile dans le but de diagnostiquer des anomalies cardiaques et dans certaines situations, l'ECG peut être utilisé pour évaluer les réponses physiologiques liées aux émotions.

Bien qu'il n'y ait pas de lien direct entre les émotions et le signal ECG, il est vrai que certaines émotions peuvent affecter le rythme cardiaque, qui est enregistré dans le signal ECG. Les émotions telles que la peur, l'anxiété, le stress ou la colère peuvent provoquer une augmentation du rythme cardiaque, tandis que des émotions plus apaisantes telles que la relaxation peuvent entraîner une diminution du rythme cardiaque.

- **Le signal BVP :** Le signal BVP (Blood Volume Pulse) est un signal physiologique qui mesure les variations de volume sanguin dans les vaisseaux sanguins du corps, en particulier ceux situés dans les doigts.  
En utilisant des techniques de traitement de signal et de l'apprentissage automatique, les variations du signal BVP peuvent être analysées pour extraire des caractéristiques liées aux émotions, telles que l'amplitude, la fréquence ou la variabilité du signal. Ces caractéristiques peuvent ensuite être utilisées pour détecter ou classifier les émotions ressenties par une personne, telles que la colère, la peur, la tristesse ou la joie.
- **Le signal GSR :** (Galvanic Skin Response), également connu sous le nom de réponse électrodermale (EDR), est un signal électrodermique qui mesure la conductance de la peau en réponse à des stimuli émotionnels. La conductance de la peau est mesurée en utilisant des électrodes placées sur la peau, généralement sur les doigts ou la paume de la main.  
La réponse électrodermale (EDR) est une réponse physiologique qui est influencée par l'activité du système nerveux sympathique, qui est activé en réponse à des stimuli émotionnels tels que la peur, l'anxiété, le stress, l'excitation ou la surprise. Cette activation provoque une augmentation de la conductance de la peau, qui peut être mesurée à l'aide d'un GSR.
- **Le signal de respiration :** est un signal physiologique qui enregistre les mouvements respiratoires d'une personne. Il peut être mesuré à l'aide de différentes techniques, telles que la mesure de la pression respiratoire.  
En ce qui concerne la relation entre le signal de respiration et la détection d'émotions, il est intéressant de noter que la respiration peut être affectée par l'état émotionnel d'une personne. Les émotions telles que l'anxiété, le stress ou la colère peuvent entraîner une augmentation de la fréquence respiratoire et une diminution de l'amplitude respiratoire, tandis que les émotions apaisantes peuvent entraîner une diminution de la fréquence respiratoire et une augmentation de l'amplitude respiratoire.
- **La température cutanée :** également appelée température de la peau, fait référence à la température de la surface de la peau. Cette température peut

varier en fonction de nombreux facteurs, tels que l'environnement, l'activité physique, le stress, la maladie.

En ce qui concerne la détection des émotions, la température cutanée peut fournir des informations sur l'état émotionnel d'une personne. Par exemple, il a été démontré que les émotions négatives telles que la colère, la peur ou l'anxiété peuvent provoquer une augmentation de la température cutanée, tandis que les émotions positives telles que la joie ou la détente peuvent entraîner une diminution de la température cutanée.

Les figures 2.1 et 2.2 présentent respectivement les tracés des différents signaux physiologiques du sujet 1 pour les émotions 'amusement' et 'peur'.

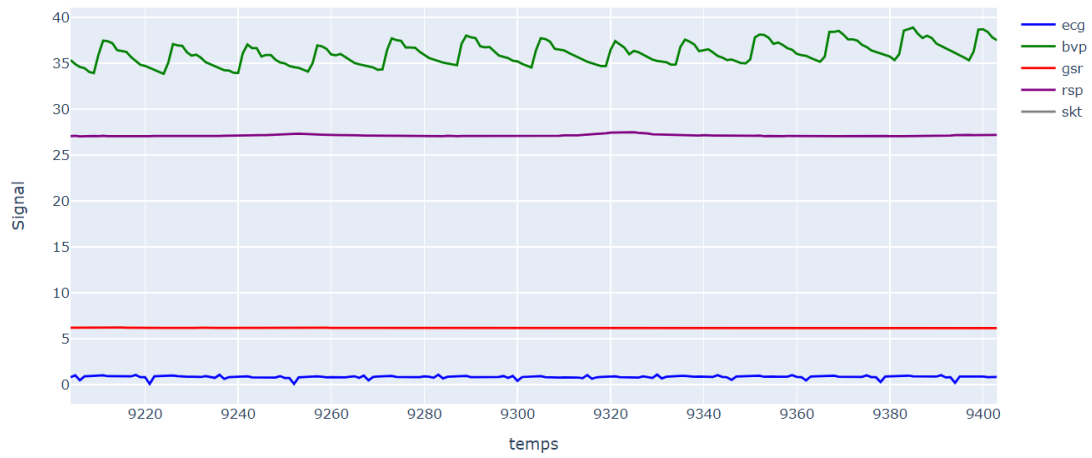


FIGURE 2.1 – Amusement.

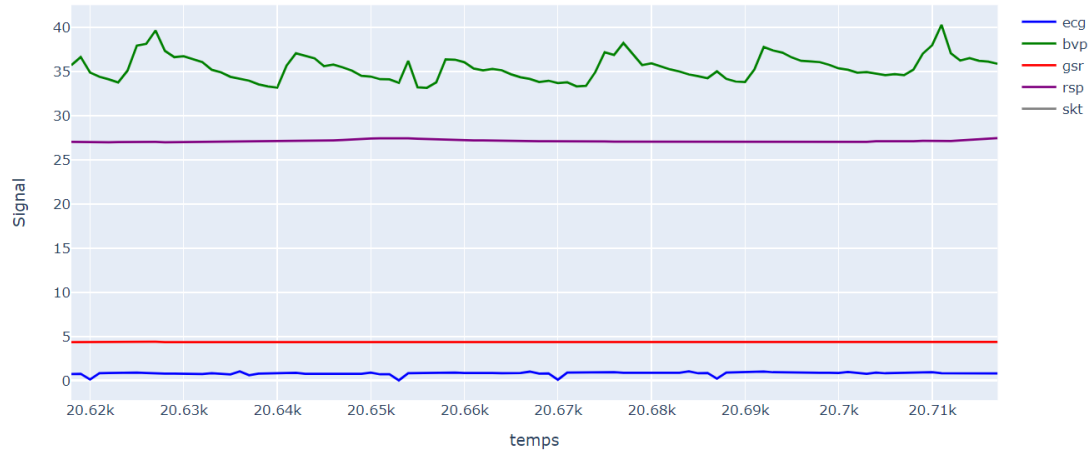


FIGURE 2.2 – Peur.

## 2.3 Méthodologie de collecte des données physiologiques

### 2.3.1 Plan de l'expérience

L'expérience a été mise en place selon un modèle intra-sujet. En conséquence, des mesures répétées ont été effectuées et tous les participants ont regardé et annoté les différents stimuli vidéo utilisés pour l'expérience. Afin d'éviter les effets de report, l'ordre des vidéos dans une session de visionnage a été modifié de manière pseudo-aléatoire, de sorte que la séquence vidéo résultante soit différente pour chaque participant. Afin d'isoler la réponse émotionnelle suscitée par les différentes vidéos, celles-ci ont été entrelacées différentes, elles étaient entrecoupées d'un écran bleu d'une durée de deux minutes. Cette période de deux minutes a également permis aux participants de se reposer entre deux annotations des vidéos [3].

### 2.3.2 Interface d'annotation

La Figure 3.36 (à droite) montre la conception de l'interface d'annotation. Elle est basée sur le modèle circulaire en 2D qui a été complété avec le Manikin d'auto-évaluation (SAM)[2] sur ses axes de coordonnées. Ces manikins représentent différents niveaux de valence (sur l'axe X) et d'excitation (sur l'axe Y), servant ainsi de guide non verbal aux participants pendant l'annotation. Le pointeur rouge dans la figure montre la position de repos/neutre. Les participants ont été instruits d'annoter leur expérience émotionnelle en déplaçant/maintenant le pointeur rouge dans la région appropriée de l'interface. La position de l'interface d'annotation à l'intérieur de la fenêtre de lecture de la vidéo est montrée dans la Figure 3.36 (au centre). Étant donné que l'annotation a été réalisée sur toute la durée de la vidéo, elle résulte en une trace 2D continue de l'expérience émotionnelle du participant (voir Figure 2.4).



FIGURE 2.3 – La configuration typique de l'expérience montre un participant regardant une vidéo et annotant à l'aide de JERI.

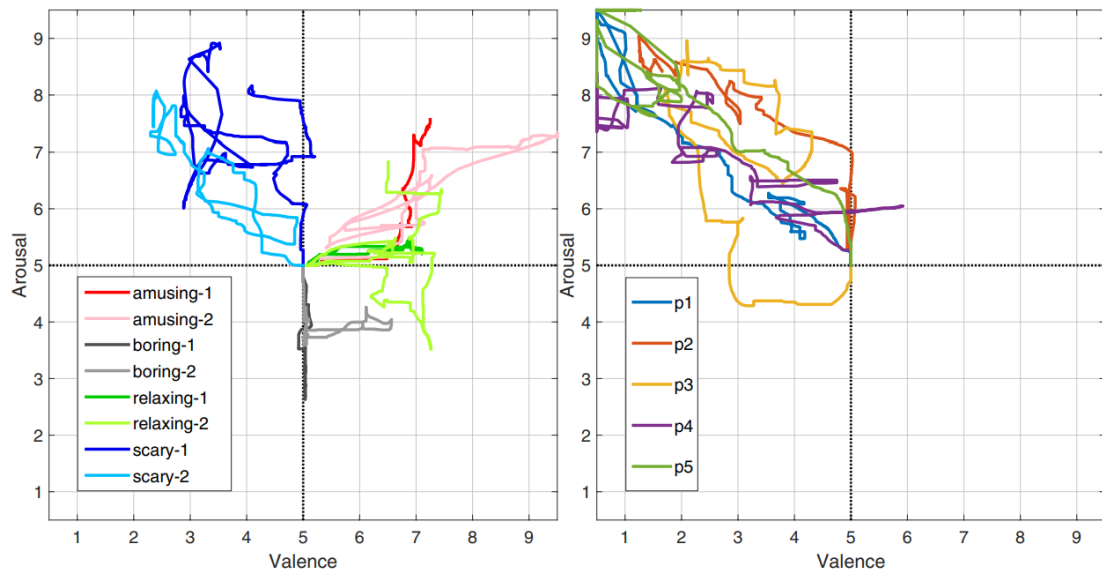


FIGURE 2.4 – Le graphique de gauche montre les annotations d'un participant pour les différentes vidéos (voir Tableau 1) de l'expérience. Les annotations pour la vidéo "effrayant-2" par les cinq premiers participants (étiquetés comme p1-p5) peuvent être observées dans le graphique de droite.

### 2.3.3 Les Vidéos

L'objectif de la séance de pratique était de familiariser les utilisateurs avec l'interface d'annotation et non de les former explicitement aux différents types de vidéos émotionnelles auxquelles ils seraient confrontés lors de l'expérience principale. Pour cette séance, cinq courtes vidéos (durée : environ 1 minute chacune), entrecoupées d'un écran bleu de transition de 10 secondes, ont été utilisées. Ces



Source	Video-Label	Video-ID	Intended Attributes		Dur. [s]
			Valence	Arousal	
Hangover	amusing-1	1	med/high	med/high	185
When Harry Met Sally	amusing-2	2	med/high	med/high	173
European Travel Skills	boring-1	3	low	low	119
Matcha: The way of Tea	boring-2	4	low	low	160
Relaxing Music with Beach	relaxing-1	5	med/high	low	145
Natural World: Zambezi	relaxing-2	6	med/high	low	147
Shutter	scary-1	7	low	high	197
Mama	scary-2	8	low	high	144
Great Barrier Reef	startVid	10	—	—	101
Blue screen with end credits	endVid	12	—	—	120
Blue screen	bluVid	11	—	—	120

FIGURE 2.5 – La source, l’étiquette, l’ID utilisé, les attributs de valence-arousal prévus et la durée des vidéos utilisées dans la base de données.

vidéos visaient à susciter des états émotionnels excitants, tristes, relaxants, effrayants et heureux [1]. Plus d’informations sur toutes ces vidéos sont disponibles dans la figure 2.5

### 2.3.4 Extraction de caractéristiques

L’extraction des caractéristiques a été réalisée de manière itérative sur les fichiers de données physiologiques pour chaque participant. Tout d’abord, les données d’un participant donné ont été segmentées en morceaux pour les différents stimuli vidéo. Ensuite, à partir des données de capteurs relatives à chaque morceau vidéo, plusieurs caractéristiques ont été extraites.

Les caractéristiques physiologiques sélectionnées sont présentées dans la figure 2.6. En raison des différences interpersonnelles, les participants ont une valeur de référence différente pour chacune de ces caractéristiques extraites. Ces différences peuvent être préjudiciables à la comparaison des caractéristiques entre tous les participants et ont donc été éliminées en utilisant la standardisation en Z-score pour chaque participant. La même approche a été utilisée pour les données d’annotation.

Les diagrammes en violon de la Figure 2.7 montrent les distributions des caractéristiques sélectionnées et des données d’annotation, pour les 4 labels de vidéo différents (voir figure 2.5). À partir de la figure, il est évident que certaines des caractéristiques physiologiques (considérer les huit premiers panneaux) caractérisent des types spécifiques de vidéos. Par exemple, les vidéos effrayantes entraînent des valeurs élevées de SCR (réponse cutanée galvanique) et de fréquence cardiaque élevée, tandis que les vidéos amusantes provoquent des taux de respiration accélérée et une activité des muscles zygomatiques [9]. Les vidéos ennuyeuses et relaxantes, comme prévu, entraînent des valeurs similaires pour toutes les caractéristiques. Cet effet peut également être observé dans nos données, où les niveaux d’excitation rapportés (voir le panneau en bas à gauche) pour les vidéos effrayantes sont plus élevés que pour les autres vidéos. De même, l’activité des muscles zygomatiques, qui a été rapportée comme étant corrélée positivement à la valence (voir le panneau en bas à gauche), présente également des schémas similaires dans nos données.

Sensor	Feature Selected
ECG	mean HR
BVP	Standard Deviation (SD) of NN-intervals (SDNN)
GSR	mean SCR
Respiration	mean RR
Skin Temperature	SD of Temperature (SDT)
EMG-zygomaticus	mean amplitude (mean Zygo)
EMG-corrugator	mean amplitude (mean Corr)
EMG-trapezius	mean amplitude (mean Trap)

FIGURE 2.6 – Les capteurs et les caractéristiques sélectionnées de chaque capteur.

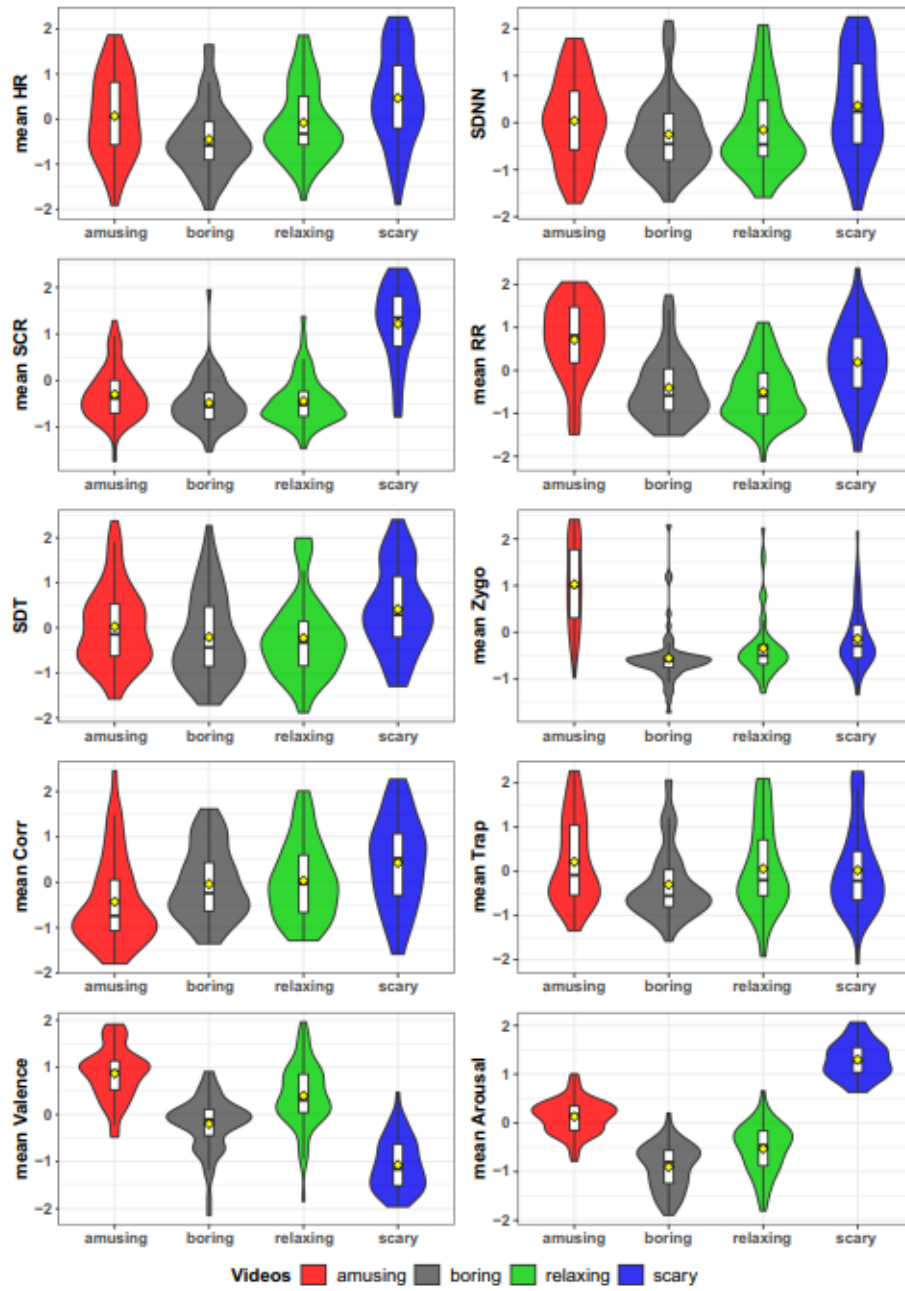


FIGURE 2.7 – "Violin" plots montrant la distribution des caractéristiques sélectionnées et des valeurs moyennes d'annotation (valence et arousal) pour différents types de vidéos. Les boîtes intégrées dans chaque "violin" plot montrent la plage interquartile (IQR) pour chaque variable considérée, tandis qu'un diamant jaune marque la moyenne de la distribution.

## 2.4 Méthodologie d'apprentissage machine utilisée

### 2.4.1 Les bibliothèques et les frameworks

Dans le projet de reconnaissance des états émotionnels à partir des signaux physiologiques et d'apprentissage machine, nous avons utilisé plusieurs bibliothèques et frameworks pour différents aspects du projet. Voici une explication de chaque bibliothèque et framework que nous avons utilisé :

- **TensorFlow** : est un framework open-source d'apprentissage automatique et de traitement des données, développé par Google. nous avons utilisé TensorFlow pour construire et entraîner votre modèle de réseau de neurones convolutionnels (CNN) pour la reconnaissance des états émotionnels.
- **Scikit-learn** : est une bibliothèque open-source d'apprentissage automatique pour Python. nous avons utilisé Scikit-learn pour la normalisation des données en utilisant les fonction "MinMaxScaler" et "StandardScaler", la division en ensembles d'entraînement et de test en utilisant la fonction "train\_test\_split", ainsi que pour la sélection des lignes en fonction du temps.
- **Numpy** : est une bibliothèque open-source pour Python qui prend en charge les calculs mathématiques avec des tableaux et des matrices multidimensionnels. nous avons utilisé Numpy pour effectuer des calculs mathématiques sur les données, tels que la réduction de la dimensionnalité et la préparation des données pour l'entraînement.
- **Pandas** : est une bibliothèque open-source pour Python qui fournit des structures de données faciles à utiliser pour l'analyse de données. nous avons utilisé Pandas pour le chargement des données des signaux physiologiques et des annotations émotionnelles à partir de fichiers CSV, la fusion des données, et la cartographie des émotions.
- **Plotly** : est une bibliothèque open-source pour Python qui fournit des outils de visualisation de données interactifs. nous avons utilisé Plotly pour la visualisation des signaux physiologiques pour une vidéo particulière.
- **Matplotlib** : Matplotlib est une bibliothèque open-source pour Python qui fournit des outils de visualisation de données statiques.
- **Keras** : est une bibliothèque open-source d'apprentissage automatique pour Python qui fournit une interface de haut niveau pour la construction de modèles de réseaux de neurones. nous avons utilisé Keras pour la construction de votre modèle de réseau de neurones convolutionnels pour la reconnaissance des états émotionnels.

En somme, nous avons utilisé une combinaison de bibliothèques et de frameworks pour les différentes étapes de votre projet, allant de la normalisation des données et la visualisation des signaux physiologiques à la construction et l'entraînement de notre modèle de réseau de neurones convolutionnels.

### 2.4.2 Préparation des données

Dans le cadre de ce projet, nous avons utilisé une approche méthodologique rigoureuse pour préparer les données avant de les utiliser pour entraîner notre modèle de prédiction des émotions. Nous avons effectué plusieurs étapes de prétraitement des données pour améliorer leur qualité et garantir qu'elles soient synchronisées et cohérentes.

Pour cela, nous avons commencé par mapper les émotions à chaque vidéo en utilisant une fonction de mappage. Ensuite, nous avons sélectionné les lignes de données à intervalles réguliers et fusionné les données de signaux physiologiques avec les annotations en utilisant la fonction de concaténation de Pandas. Nous avons également appliqué la moyenne et la médiane mobiles aux signaux physiologiques pour réduire le bruit en utilisant des fonctions de traitement de données.

Enfin, nous avons filtré les données pour ne garder que les signaux synchronisés en utilisant les fonctions de conversion de dates et les fonctions de manipulation de données de Pandas. L'ensemble de ces étapes de préparation de données ont permis d'obtenir des données de haute qualité pour l'entraînement de notre modèle de prédiction des émotions.

Dans l'ensemble, les étapes de préparation de données jouent un rôle essentiel dans la construction de tout modèle de machine learning. Nous avons donc pris soin d'appliquer des techniques de prétraitement de données rigoureuses pour garantir des données de haute qualité pour notre modèle de prédiction des émotions. En fin de compte, cette approche méthodologique a permis d'obtenir des résultats précis et fiables pour la prédiction des émotions dans notre ensemble de données.

### 2.4.3 Prétraitement des données

Dans tout projet de Machine Learning, le Prétraitement des données est une étape cruciale. Cette étape comprend plusieurs sous-étapes :

1. **Normalisation des données** : permet de mettre toutes les caractéristiques à la même échelle, de sorte qu'elles aient la même influence sur le modèle de Machine Learning. nous avons normalisé les données des signaux physiologiques en utilisant la fonction "MinMaxScaler()" et "StandardScaler()" de la

bibliothèque Scikit-learn pour transformer les données en une distribution centrée sur 0 avec une variance de 1. nous avez appliqué la normalisation sur les caractéristiques des signaux physiologiques, qui étaient stockées dans un DataFrame.

2. **La réduction de la dimensionnalité** : est une étape qui consiste à réduire le nombre de caractéristiques ou de variables dans un ensemble de données. Cette étape peut être importante lorsque l'ensemble de données contient un grand nombre de caractéristiques, ou lorsque certaines des caractéristiques sont redondantes ou inutiles pour le modèle de Machine Learning. Nous avons réduit la dimensionnalité des données en extrayant uniquement les caractéristiques pertinentes pour la reconnaissance des états émotionnels, en sélectionnant les caractéristiques des signaux physiologiques qui étaient les plus utiles pour la classification des états émotionnels.
3. **Division en ensembles d'entraînement et de test** : permet de vérifier les performances du modèle de Machine Learning sur un ensemble de données distinct de celui sur lequel il a été entraîné. Nous avons divisé les données normalisées et les étiquettes (labels) des émotions en ensembles d'entraînement et de test en utilisant la fonction "train\_test\_split()" de la bibliothèque Scikit-learn, et les stockant dans des tableaux NumPy distincts.
4. **Reformattage des données** : consiste à reformater les données pour qu'elles soient compatibles avec le modèle de Machine Learning utilisé. Nous avons reformatté les données d'entraînement et de test pour les adapter à l'entrée de notre modèle de réseau de neurones convolutionnels, en utilisant la fonction "reshape()" de NumPy.

En résumé, on a préparé les données en normalisant les données des signaux physiologiques et en reformatant les données pour les adapter à l'entrée de votre modèle de réseau de neurones convolutionnels.

#### 2.4.4 Construction de modèle

Dans le cadre de notre projet, nous avons choisi de construire un modèle séquentiel de réseaux de neurones convolutionnels (CNN) pour prédire les émotions en utilisant des signaux physiologiques enregistrés pendant que les participants regardaient des vidéos.

Pour construire le modèle, nous avons utilisé la bibliothèque Tensorflow et créé une architecture CNN avec des couches de convolution, des couches de pooling et des couches dense. La fonction d'activation utilisée était "relu" pour les couches de convolution et la dernière couche a utilisé une activation "softmax" pour produire une distribution de probabilité sur les classes.

Ensuite, nous avons formé le modèle en utilisant les données d'entraînement et validé ses performances sur les données de test. Nous avons utilisé la fonction de perte "sparse\_categorical\_crossentropy" et l'optimiseur "adam" pour minimiser la perte pendant l'entraînement.

Nous avons également utilisé des callbacks pour arrêter l'entraînement lorsque la précision atteint une certaine valeur pour éviter l'overfitting. Enfin, nous avons évalué les performances du modèle en utilisant des métriques telles que l'exactitude (accuracy), la précision et le rappel pour chaque classe.

En conclusion, la construction et la formation du modèle CNN ont été réalisées avec succès en utilisant Tensorflow et les performances ont été évaluées en utilisant des métriques appropriées. Ce modèle a été utilisé pour prédire les émotions à partir des signaux physiologiques et peut être amélioré en utilisant des techniques de régularisation pour éviter l'overfitting et en ajustant les hyperparamètres du modèle pour optimiser les performances.

## 2.5 Explication de la méthode de l'évaluation et la validation des résultats

La partie d'évaluation du modèle est une étape cruciale dans tout projet d'apprentissage automatique. Cette étape consiste à évaluer les performances du modèle sur un ensemble de données de test distinct de celui utilisé pour l'entraînement, et donc à analyser les performances du modèle de classification de la valence et de l'arousal. Le but est d'estimer la capacité du modèle à généraliser et à prédire de manière précise les résultats sur de nouvelles données.

Dans le cadre de ce projet, l'évaluation du modèle se fait en utilisant la méthode `evaluate()` de Keras. Cette méthode prend en entrée les données de test (caractéristiques et cibles) et calcule la perte et la précision du modèle sur ces données. La perte est une mesure de l'erreur du modèle, tandis que la précision mesure la performance globale du modèle.

En plus de la méthode "`evaluate()`", d'autres mesures de performance peuvent également être utilisées pour évaluer le modèle, telles que la précision, le rappel et la mesure F1. Ces mesures peuvent être obtenues en utilisant des fonctions de métrique telles que `accuracy_score()`, `recall_score()`, `precision_score()` et `f1_score()` de la bibliothèque Scikit-learn. Nous avons également affiché une matrice de confusion pour mieux comprendre les performances du modèle. Aussi, un objet de rappel (callback) a été utilisé pour arrêter l'entraînement du modèle lorsqu'une précision de 90% est atteinte. Cette technique est souvent utilisée pour économiser des ressources de calcul lorsqu'il est clair qu'un certain niveau de précision a été atteint, plutôt que de poursuivre l'entraînement du modèle sans fin.

Les métriques d'évaluation sont des outils essentiels pour mesurer la performance des modèles d'apprentissage automatique. Dans le contexte de notre projet, et pour évaluer la performance de notre modèle de prédiction de l'état émotionnel, nous avons utilisé :

- **La précision** : mesure le pourcentage de prédictions correctes parmi les prédictions positives. Cela signifie le pourcentage de fois où le modèle a correctement prédit la valence ou l'activation émotionnelle.  
$$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$$
où TP (True Positive) représente le nombre de prédictions correctes positives, TN (True Negative) représente le nombre de prédictions correctes négatives, FP (False Positive) représente le nombre de prédictions incorrectes positives et FN (False Negative) représente le nombre de prédictions incorrectes négatives.
- **L'exactitude** : (accuracy) mesure le pourcentage de prédictions correctes parmi toutes les prédictions. Cela signifie le pourcentage de fois où le modèle a correctement prédit la valence ou l'activation émotionnelle parmi toutes les prédictions qu'il a effectuées.  
$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$
- **F1-Score** une moyenne harmonique de la précision et du rappel. Il est utilisé pour trouver un équilibre entre la précision et le rappel. C'est une mesure utile lorsque les classes sont déséquilibrées, c'est-à-dire lorsqu'il y a un grand écart entre le nombre de données dans chaque classe.  
$$\text{F1-score} = 2 * \text{TP} / (2 * \text{TP} + \text{FN} + \text{FP})$$
- **Recall** mesure le pourcentage de prédictions correctes parmi toutes les instances de la classe réelle. cela signifie le pourcentage de fois où le modèle a correctement prédit une valence ou une activation émotionnelle parmi toutes les instances de cette valence ou activation émotionnelle.  
$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative})$$
- **La perte** (loss) est une mesure qui indique la performance du modèle pendant l'entraînement. Il mesure à quel point les prédictions du modèle sont éloignées des valeurs réelles. L'objectif est de minimiser la perte afin que le modèle puisse prédire avec précision les valeurs de la valence et de l'activation émotionnelle pour de nouvelles données.  
$$\text{Loss} = -(y * \log(y\_pred) + (1-y) * \log(1-y\_pred))$$
- **La matrice de confusion** est un tableau qui montre le nombre de prédictions correctes et incorrectes pour chaque classe. Elle est très utile pour visualiser la performance d'un modèle de classification. Elle nous permet de voir combien de fois le modèle a correctement prédit la valence ou l'activation émotionnelle et combien de fois il s'est trompé.



Il est important de noter que l'évaluation du modèle ne doit pas être effectuée uniquement sur les données de test, mais également sur les données de validation. En effet, la validation croisée est une méthode qui consiste à diviser le jeu de données en plusieurs parties et à entraîner le modèle sur chacune de ces parties, en utilisant une partie différente pour la validation à chaque fois. Cette méthode permet d'évaluer les performances du modèle de manière plus robuste en réduisant les risques de surapprentissage (overfitting) et en maximisant la généralisation.

En outre, nous pouvons visualiser les performances du modèle à chaque epoch en utilisant la bibliothèque Matplotlib, En traçant la perte (loss) et l'exactitude (accuracy) du modèle au cours de l'entraînement et de la validation.

Le premier graphe affiche l'évolution de la perte (loss) du modèle sur les ensembles de données d'entraînement et de validation en fonction du nombre d'epochs. On peut observer si la perte diminue de manière stable au fil des epochs ou s'il y a des surapprentissages (overfitting) du modèle.

Le deuxième graphe montre l'évolution de l'exactitude (accuracy) du modèle sur les ensembles de données d'entraînement et de validation en fonction du nombre d'epochs. On peut observer si l'exactitude du modèle augmente et se stabilise à un certain point ou s'il y a un surapprentissage (overfitting) du modèle.

En utilisant ces graphes, on peut déterminer si le modèle est en train de s'améliorer au fil des epochs ou s'il y a un surapprentissage (overfitting) du modèle. Si la perte (loss) diminue et que l'exactitude (accuracy) augmente de manière stable au fil des epochs, cela signifie que le modèle fonctionne bien. Si la perte (loss) diminue de manière stable mais que l'exactitude (accuracy) du modèle sur les données de validation commence à diminuer ou stagner, cela peut indiquer un surapprentissage (overfitting) du modèle. Dans ce cas, il est nécessaire de réduire la complexité du modèle ou d'ajouter des régularisations pour améliorer les performances du modèle sur les ensembles de données de validation.

En conclusion, l'évaluation du modèle est une étape cruciale dans tout projet d'apprentissage automatique. Elle permet de mesurer la capacité du modèle à généraliser et à prédire de manière précise les résultats sur de nouvelles données.

## 2.6 Conclusion

Dans ce chapitre, nous avons présenté une approche de reconnaissance automatique des émotions, fondée sur le traitement des signaux physiologiques. La démarche consiste à extraire des informations caractéristiques des signaux, préparer les données en sélectionnant des sujets et en pré-traitant les signaux, suivie par une étape de classification basée sur un réseau de neurones convolutif et finalement

par une évaluation du modèle en utilisant des métriques de performance. En fin de compte, cette méthode pourrait être utilisée pour développer des applications dans divers domaines tels que la psychologie, la santé mentale et l'interaction homme-machine.

# Chapitre 3

## Implémentation et interprétation des résultats

### 3.1 Introduction

Dans ce chapitre, nous allons présenter l'implémentation de notre modèle de reconnaissance automatique des émotions basé sur le traitement des signaux physiologiques. Nous allons également interpréter les résultats obtenus à travers une analyse détaillée des performances de notre modèle et une comparaison avec les résultats précédemment publiés sur le même ensemble de données.

### 3.2 Prédiction des étiquettes des vidéos à partir des signaux physiologiques

#### Le but

Le but de ce modèle est de prédire les étiquettes émotionnelles des vidéos à partir des signaux physiologiques est de construire un système de reconnaissance des émotions en temps réel qui peut aider à comprendre les réactions émotionnelles d'une personne à divers stimuli, tels que des vidéos ou des images. La prédiction des étiquettes des vidéos vise à prédire l'étiquette émotionnelle générale associée à une vidéo, telle que la colère, la tristesse ou la joie. Cela implique de classer une vidéo en une catégorie émotionnelle discrète et prédéfinie.

### 3.2.1 Implémentation

#### Préparation des données

L'ensemble des données comprend les données physiologiques et d'annotation de 30 participants qui ont regardé plusieurs stimuli vidéo validés. Pour cela, la fonction `choose_subjects(Number)` (voir figure 3.3) est utilisée pour charger les données des participants à partir des fichiers CSV correspondants pour construire les DataFrames des annotations et des signaux physiologiques. Le paramètre `Number` est utilisé pour spécifier le nombre de participants que nous voulons charger.

```
def choose_subjects(Number):  
    annotations_df = pd.read_csv("drive/MyDrive/case_dataset-master/data/interpolated/annotations/sub_1.csv")  
    physiological_df = pd.read_csv("drive/MyDrive/case_dataset-master/data/interpolated/physiological/sub_1.csv")  
    for i in range(2, Number+1):  
        chemin1 = 'drive/MyDrive/case_dataset-master/data/interpolated/annotations/sub_{}.csv'.format(i)  
        chemin2 = 'drive/MyDrive/case_dataset-master/data/interpolated/physiological/sub_{}.csv'.format(i)  
        annotations_df = pd.concat([annotations_df, pd.read_csv(chemin1)], axis=0)  
        physiological_df = pd.concat([physiological_df, pd.read_csv(chemin2)], axis=0)  
    return annotations_df, physiological_df  
  
anno_df, physio_df = choose_subjects(10)
```

FIGURE 3.1 – Code de chargement données des participants.

Pour préparer les données en vue de l'entraînement du modèle de prédiction des étiquettes de vidéo à partir des signaux physiologiques, nous avons utilisé la fonction "synchronize\_and\_resample(physio\_df, anno\_df)" a été développée pour synchroniser les données physiologiques et les annotations émotionnelles, et pour échantillonner les signaux physiologiques à des intervalles de temps réguliers (50 ms, car elles ont été enregistrées à des moments différents et sur des fréquences d'échantillonnage différentes. En synchronisant les deux, nous nous assurons que les données correspondantes sont disponibles pour chaque instant de temps.

```
def synchronize_and_resample(physio_df, anno_df):

    # Synchronize the dataframes
    physio_df['datetime'] = pd.to_datetime(physio_df['daqtime'], unit='ms')
    anno_df['datetime'] = pd.to_datetime(anno_df['jstime'], unit='ms')
    # Find the common timestamps between the two dataframes
    common_times = np.intersect1d(physio_df['daqtime'], anno_df['jstime'])

    # Filter the dataframes based on the common timestamps
    df_physio_synced = physio_df[physio_df['daqtime'].isin(common_times)]
    df_annotations_synced = anno_df[anno_df['jstime'].isin(common_times)]
    # Resample physiological signals to 50ms intervals
    df_physio_synced = df_physio_synced.set_index(['video', 'datetime'])
    df_physio_synced = df_physio_synced.reset_index()
    # Merge dataframes on video ID and synchronized timestamps
    merged_df = pd.merge(df_physio_synced, df_annotations_synced, on=['video', 'datetime'], how='inner')
    # Check for missing or invalid data
    merged_df = merged_df.dropna()
    return merged_df
```

FIGURE 3.2 – Fonction de synchronisation et d'échantillonnage des signaux physiologiques et des annotations émotionnelles.

	video	datetime	daqtime	ecg	bvp	gsr	rsp	skt	emg_zygo	emg_coru	emg_trap	jstime	valence	arousal	emotion
0	10	1970-01-01 00:00:00.000	0	0.846	38.172	11.749	27.610	29.495	7.288	8.684	6.550	0	5.000	5.000	3
1	10	1970-01-01 00:00:00.000	0	0.846	38.172	11.749	27.610	29.495	7.288	8.684	6.550	0	5.000	5.000	3
2	10	1970-01-01 00:00:00.000	0	0.843	36.418	24.945	27.629	31.661	8.272	7.698	-52.631	0	5.000	5.000	3
3	10	1970-01-01 00:00:00.000	0	0.843	36.418	24.945	27.629	31.661	8.272	7.698	-52.631	0	5.000	5.000	3
4	10	1970-01-01 00:00:00.050	50	0.813	38.239	11.729	27.572	29.477	7.371	7.987	6.427	50	5.000	5.000	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
136077	7	1970-01-01 00:38:51.350	2331350	0.843	36.040	29.104	27.058	28.345	9.671	9.589	-75.216	2331350	2.393	8.247	4
136078	7	1970-01-01 00:38:51.400	2331400	0.944	35.304	29.104	27.087	28.348	10.490	8.972	-74.807	2331400	2.393	8.247	4
136079	7	1970-01-01 00:38:51.450	2331450	1.076	35.778	29.077	27.087	28.345	8.233	10.369	-75.136	2331450	2.393	8.247	4
136080	7	1970-01-01 00:38:51.500	2331500	1.119	38.520	29.061	27.078	28.341	9.343	11.068	-74.847	2331500	2.393	8.247	4
136081	7	1970-01-01 00:38:51.550	2331550	0.803	39.459	29.046	27.077	28.341	8.808	9.752	-75.258	2331550	2.393	8.247	4

136082 rows × 15 columns

FIGURE 3.3 – Dataframe résultant.

Pour le prétraitement des données, nous avons développé La fonction "preprocess\_data(merged\_df)",(voir figure 3.4, qui est utilisée pour extraire les caractéristiques physiologiques et la variable cible à partir du dataframe, qui contient les données synchronisées et échantillonnées.

La fonction commence par extraire les caractéristiques physiologiques du dataframe en sélectionnant les colonnes correspondantes et en stockant les valeurs dans un tableau, on a choisi de conserver les huit caractéristiques physiologiques. Ensuite, elle applique une normalisation MinMax aux caractéristiques en utilisant la classe MinMaxScaler de la bibliothèque sklearn.preprocessing. Cette normalisation met toutes les caractéristiques à l'échelle dans la plage [0, 1], ce qui est important pour faciliter l'apprentissage du modèle et obtenir de meilleurs résultats de prédiction.

Ensuite, la fonction extrait la variable cible (étiquette de vidéo) du dataframe et la stocke dans un tableau séparé. Elle convertit ensuite les tableaux contenant les caractéristiques et les étiquettes en tableaux Numpy pour être utilisés dans l'entraînement du modèle.

```
def preprocess_data(merged_df):  
    features_df = merged_df.iloc[:,3:11]  
    features_scaled=features_df.values  
    #features_scaled = features_scaled.reshape(-1,1) #uncomment if using 1 feature  
    min_max_scaler = sk.preprocessing.MinMaxScaler()  
    features_scaled=min_max_scaler.fit_transform(features_scaled)  
  
    # Extract the target variable  
    target_df=merged_df.iloc[:,14]  
    target=target_df.values  
    return features_scaled, target
```

FIGURE 3.4 – Code de prétraitement des données.

Afin de diviser les données en ensembles d'entraînement et de test, pour l'entraînement d'un modèle de réseau de neurones convolutionnel (CNN),la fonction "prepare\_data(features, target)", (voir figure 3.6. Elle prend en entrée deux tableaux Numpy : features qui contient les caractéristiques prétraitées des signaux physiologiques, et target qui contient les étiquettes de vidéo correspondantes.

La fonction divise les données en ensembles d'entraînement et de test en utilisant la fonction train\_test\_split de la bibliothèque sklearn.model\_selection. Elle utilise un argument test\_size pour spécifier la taille de l'ensemble de test par rapport à l'ensemble d'entraînement, et un argument random\_state pour spécifier la graine aléatoire pour la reproductibilité des résultats.

Ensuite, la fonction reformate les caractéristiques en un tableau 3D qui peut être utilisé par le modèle de prédiction. Cela est nécessaire car le modèle de prédiction prend en entrée des données 3D sous la forme de (nombre d'échantillons, longueur

de la séquence, nombre de caractéristiques). Enfin, la fonction renvoie les ensembles d'entraînement et de test reformés pour les caractéristiques et les étiquettes de vidéo.

La division en ensembles d'entraînement et de test permet également de mesurer l'exactitude du modèle sur des données inconnues, ce qui peut aider à identifier les problèmes de surapprentissage et de sous-apprentissage.

```
def prepare_data(features, target, test_size=0.2, random_state=6):  
  
    from sklearn.model_selection import train_test_split  
  
    # Diviser les données en ensembles d'entraînement et de test  
    x_Train, x_Test, y_Train, y_Test = train_test_split(features, target, test_size=test_size, random_state=random_state)  
  
    # Reformater les features en un tableau 3D  
    x_Train = x_Train.reshape((x_Train.shape[0], x_Train.shape[1], 1))  
    x_Test = x_Test.reshape((x_Test.shape[0], x_Test.shape[1], 1))  
  
    return x_Train, x_Test, y_Train, y_Test
```

FIGURE 3.5 – Code de division des données en ensembles d'entraînement et de test.

### Définition et entraînement du modèle

La fonction `train_model(x_train, y_train, epochs, batch_size, validation_split)` est utilisée pour entraîner un modèle de prédiction des étiquettes de vidéo à partir des signaux physiologiques. Elle prend en entrée les tableaux de données d'entraînement `x_train` et `y_train`, ainsi que des paramètres tels que le nombre d'époques `epochs`, la taille de lot `batch_size` et le pourcentage de données à utiliser pour la validation `validation_split`. Elle renvoie le modèle entraîné et l'historique de l'apprentissage.

Le modèle décrit est un modèle de réseau de neurones à convolution (CNN) avec plusieurs couches de convolutions et des couches de mise en commun (max pooling). Les convolutions permettent d'extraire des caractéristiques importantes des signaux physiologiques en les analysant en petits segments, tandis que les couches de mise en commun réduisent la taille de la représentation en éliminant les informations redondantes. La dernière couche de convolution est suivie d'une couche Flatten, qui permet de transformer la sortie en un vecteur 1D pour l'entrée dans une couche dense.

La couche Dense est utilisée pour intégrer les informations de toutes les convolutions et pour classifier les signaux physiologiques.

Le modèle a été choisi en raison de sa capacité à extraire des caractéristiques importantes des signaux physiologiques en les analysant en petits segments et à

```
def train_model(x_train, y_train, epochs, batch_size, validation_split):

    from tensorflow.keras import metrics
    from sklearn.metrics import precision_score, recall_score, f1_score
    from tensorflow.keras.metrics import Precision, Recall
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from keras.models import Sequential
    from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Dropout

    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(x_train.shape[1], 1)))
    model.add(Conv1D(filters=128, kernel_size=1, activation='relu'))
    model.add(MaxPooling1D(pool_size=1))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(8, activation='softmax'))

    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics='accuracy')
    model.fit(x_train, y_train, epochs=100, batch_size=batch_size)

    precision, recall = model.evaluate(x_test, y_test)

    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=["accuracy"])
    history = model.fit(x_train, y_train, validation_split=validation_split, epochs=epochs, batch_size=batch_size)

    precision, recall = model.evaluate(x_test, y_test)

    return model, history
```

FIGURE 3.6 – Entraînement du modèle de prédiction des étiquettes de vidéo à partir des signaux physiologiques.

modéliser des relations non linéaires entre les caractéristiques et les étiquettes de la vidéo.

La fonction d'activation 'relu' a été utilisée dans les couches de convolution pour introduire une non-linéarité dans le modèle et pour empêcher la saturation des neurones. La fonction 'softmax' a été utilisée pour la couche de sortie car elle permet de normaliser les sorties pour qu'elles représentent des probabilités qui indiquent la classe prédite pour chaque entrée.

La fonction de perte utilisée est 'sparse\_categorical\_crossentropy', qui est appropriée pour les tâches de classification multi-classes avec des étiquettes entières. Cette fonction de perte mesure la différence entre la distribution de probabilité prédite par le modèle et la distribution de probabilité des étiquettes réelles, et minimise cette différence pendant l'entraînement.

Le modèle est compilé en utilisant la fonction d'optimisation utilisée pour entraîner le modèle est 'adam', qui est une méthode d'optimisation adaptative couramment utilisée pour les réseaux de neurones. Elle ajuste le taux d'apprentissage pour chaque paramètre en fonction de l'historique des mises à jour de chaque paramètre, ce qui permet de trouver plus rapidement le minimum global de la fonction de perte.



Ensuite, la fonction `fit()` est utilisée pour entraîner le modèle en utilisant les données d'entraînement et de validation, et l'historique de l'apprentissage est retourné. Enfin, la fonction évalue le modèle sur les données de test en utilisant la méthode `evaluate()`. Cette méthode calcule la perte et l'exactitude du modèle sur les données de test, et retourne les scores. La fonction renvoie le modèle entraîné et l'historique de l'apprentissage.

La fonction `train_model()` est donc une partie essentielle du projet, car elle permet de construire et entraîner le modèle de prédiction des étiquettes de vidéo à partir des signaux physiologiques. Les paramètres tels que le nombre d'époques et la taille de lot peuvent être ajustés pour améliorer les performances du modèle, et les résultats de la fonction peuvent être utilisés pour évaluer la précision du modèle et son aptitude à prédire les étiquettes de vidéo à partir des signaux physiologiques.

### 3.2.2 Évaluation et interprétation des résultats

La partie évaluation du modèle est une étape cruciale dans tout projet d'apprentissage automatique, car elle permet de mesurer la performance du modèle sur un ensemble de données de test et de valider sa capacité à généraliser à d'autres ensembles de données.

Dans ce projet, nous avons évalué la performance de notre modèle de reconnaissance d'émotions à partir de signaux physiologiques en utilisant plusieurs métriques de performance, telles que l'exactitude, la précision, le rappel et le score F1. Nous avons créé une fonction "`evaluate_model(model, x_Test, y_Test)`" (voir figure 3.7 pour calculer ces métriques à partir des prédictions du modèle et des vérités terrain sur l'ensemble de test. Les résultats obtenus pour ces métriques pour le modèle en question sont présentes dans la figure 3.8.

```
def evaluate_model(model, x_test, y_test):
    from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

    prediction = model.predict(x_test)
    y_pred = np.argmax(prediction, axis=1)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    print('Accuracy:', accuracy)
    print('Precision:', precision)
    print('Recall:', recall)
    print('F1-score:', f1)
    return accuracy, precision, recall, f1
```

FIGURE 3.7 – Les résultats obtenus.

Ces résultats indiquent que le modèle a une très bonne performance de prédiction des étiquettes des vidéos à partir des signaux physiologiques. Le modèle a une

précision élevée (0.9819) qui indique qu'il prédit correctement la plupart des étiquettes positives et n'a que peu de faux positifs. Le recall également élevé (0.9818) indique que le modèle prédit correctement la plupart des étiquettes positives, et n'a que peu de faux négatifs. Le F1-score, qui est une moyenne pondérée de la précision et du recall, est également élevé (0.9819).

```
evaluate_model(model, x_Test, y_Test)

307/307 [=====] - 2s 5ms/step
Accuracy: 0.9818478482561697
Precision: 0.981935851019809
Recall: 0.9818478482561697
F1-score: 0.9818641034739268
(0.9818478482561697, 0.981935851019809, 0.9818478482561697, 0.9818641034739268)
```

FIGURE 3.8 – Les résultats obtenus.

Nous avons également utilisé une matrice de confusion, en utilisant la fonction "print\_confusion\_matrix(model, x\_Test, y\_Test)", voir figure 3.9, pour visualiser la performance du modèle en montrant le nombre de vrais positifs, de faux positifs, de vrais négatifs et de faux négatifs pour chaque classe de la variable cible. Cette matrice nous permet de mieux comprendre la capacité du modèle à classifier correctement les signaux physiologiques en fonction de leurs émotions associées.

```
def print_confusion_matrix(model, x_test, y_test):

    import numpy as np
    from sklearn.metrics import confusion_matrix
    # Get predictions for the test set
    y_pred = model.predict(x_test)

    # Convert predicted probabilities to predicted class labels
    y_pred = np.argmax(y_pred, axis=1)

    # Create confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Print confusion matrix
    print('Confusion matrix:')
    print(cm)
```

FIGURE 3.9 – La fonction matrice de confusion.

Dans ce cas, la matrice de confusion montre les résultats de la prédiction des étiquettes de vidéos à partir des signaux physiologiques, avec six classes possibles dans la figure 3.10. Les résultats montrent que le modèle a prédit avec une très haute précision la majorité des classes, à l'exception de quelques erreurs mineures.

Plus précisément, on peut observer que la première classe a été prédite correctement dans 1356 cas, sans aucune confusion avec les autres classes. La deuxième classe a également été prédite avec une très haute précision, avec 1086 prédictions correctes. La troisième classe a été prédite avec un score parfait de 1550.

La quatrième classe a également été prédite avec une haute précision, avec 1324 prédictions correctes.

Cependant, on peut observer quelques erreurs dans la prédiction de la cinquième classe, où 10 prédictions ont été faites pour la première classe, 15 pour la deuxième classe, 13 pour la troisième classe, et 30 pour la quatrième classe. Néanmoins, la majorité des prédictions pour la cinquième classe étaient correctes, avec 3813 prédictions correctes. Pour la sixième classe, le modèle a fait seulement 3 erreurs avec une précision de 99.4

Dans l'ensemble, la matrice de confusion montre que le modèle a prédit avec succès les étiquettes des vidéos à partir des signaux physiologiques, avec une très haute précision pour la majorité des classes.

Les résultats obtenus sont présentés dans la figure ??.

```
print_confusion_matrix(model, x_Test, y_Test)

307/307 [=====] - 1s 3ms/step
Confusion matrix:
[[1356   0   0   0  17   0]
 [   0 1086   0   0  22   0]
 [   0   0 1550   0  43   0]
 [   0   0   0 1324  25   0]
 [  10  15  13  30 3813   0]
 [   0   0   0   3   0  499]]
```

FIGURE 3.10 – La matrice de confusion obtenue.

En plus de l'évaluation du modèle, nous avons également visualisé les résultats de l'entraînement en utilisant la fonction "plot\_training" pour tracer l'historique de la perte et de l'exactitude pendant l'entraînement. Cette visualisation nous permet de détecter les problèmes de sur-apprentissage ou de sous-apprentissage du modèle et de surveiller sa progression au fil des époques.

La fonction plot\_training(history) permet de tracer deux graphes :

1. Graphique de la perte d'entraînement et de la perte de validation par rapport aux epochs. Ce graphique montre comment la perte d'entraînement et de validation évoluent à mesure que le modèle s'entraîne. Si la perte d'entraînement continue de diminuer, mais que la perte de validation commence à augmenter, cela peut indiquer que le modèle est en train de sur-apprendre aux données d'entraînement et qu'il ne généralise pas bien aux données de validation. Dans ce cas, il peut être nécessaire d'arrêter l'entraînement plus tôt ou d'utiliser des techniques de régularisation pour éviter le sur-apprentissage.
2. Graphique de l'accuracy d'entraînement et de l'accuracy de validation par rapport aux epochs. Ce graphique montre comment l'accuracy d'entraînement et de validation évoluent à mesure que le modèle s'entraîne. Si l'accuracy d'entraînement continue d'augmenter mais que l'accuracy de validation stagne ou diminue, cela peut indiquer un sur-apprentissage du modèle.

Le graphique de la perte 3.11 montre une diminution constante de la perte d'entraînement et de validation, ce qui suggère que le modèle apprend correctement et ne sur-apprend pas. Le graphique de l'accuracy 3.15 montre également une augmentation constante de l'accuracy d'entraînement et de validation, ce qui indique que le modèle est capable de bien généraliser aux données de validation et n'a pas de problème de sur ou sous-apprentissage.

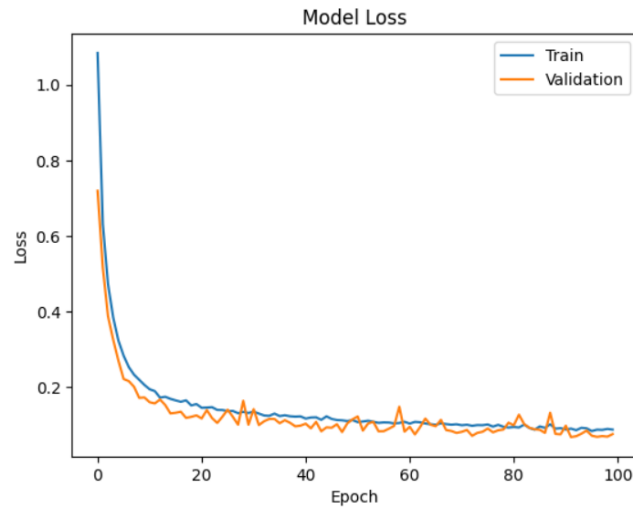


FIGURE 3.11 – Le graphique de la perte.

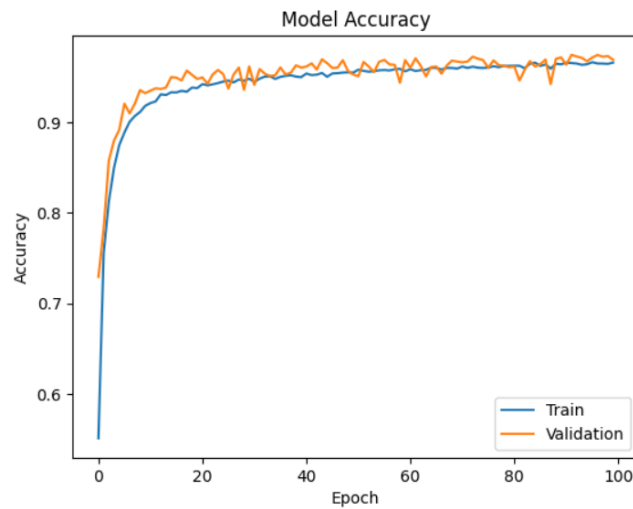


FIGURE 3.12 – Le graphique de l'accuracy.

## Entraînement sur plusieurs sujets

D'après les résultats de l'entraînement, nous pouvons voir que les résultats de rappel sont généralement élevés dans la figure 3.13, allant de 0,93 à 0,97. Ceci indique que le modèle est bon pour identifier les vrais positifs, c'est-à-dire pour prédire correctement les étiquettes des vidéos et l'activation émotionnelle basées sur des signaux physiologiques. La note de rappel la plus élevée est obtenue pour le deuxième sujet, tandis que la note de rappel la plus faible est obtenue pour le quatrième sujet, où la note de rappel est de 0,93.

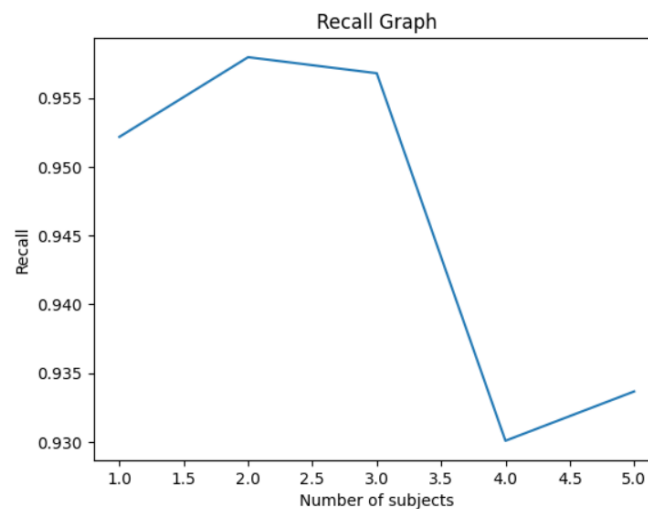


FIGURE 3.13 – Le graphe du rappel.

En revanche, la figure 3.14 montre que les scores de précision sont relativement élevés, allant de 0,93 à 96. Cela indique que le modèle a un pourcentage de prédictions correctes parmi les prédictions positives très élevés, Cela signifie le pourcentage de fois où le modèle a correctement prédit les étiquettes vidéos et l'activation émotionnelle. La note de précision la plus élevée est obtenue pour le deuxième sujet, tandis que la note de précision la plus faible est obtenue pour le quatrième sujet.

La figure 3.15 montre que les scores de l'exactitude (accuracy) sont relativement élevés, allant de 0,85 à 94 pour les 5 sujets. Cela indique que le pourcentage de fois où le modèle a correctement prédit les étiquettes vidéos ou l'activation émotionnelle parmi toutes les prédictions qu'il a effectuées est élevé. La note de l'exactitude la plus élevée est obtenue pour le deuxième sujet, tandis que la note de l'exactitude la plus faible est obtenue pour le quatrième sujet.

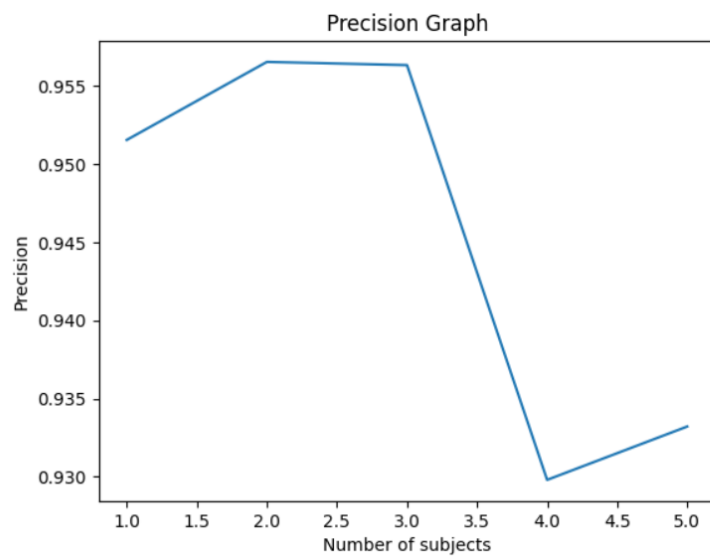


FIGURE 3.14 – Le graphe de la précision.

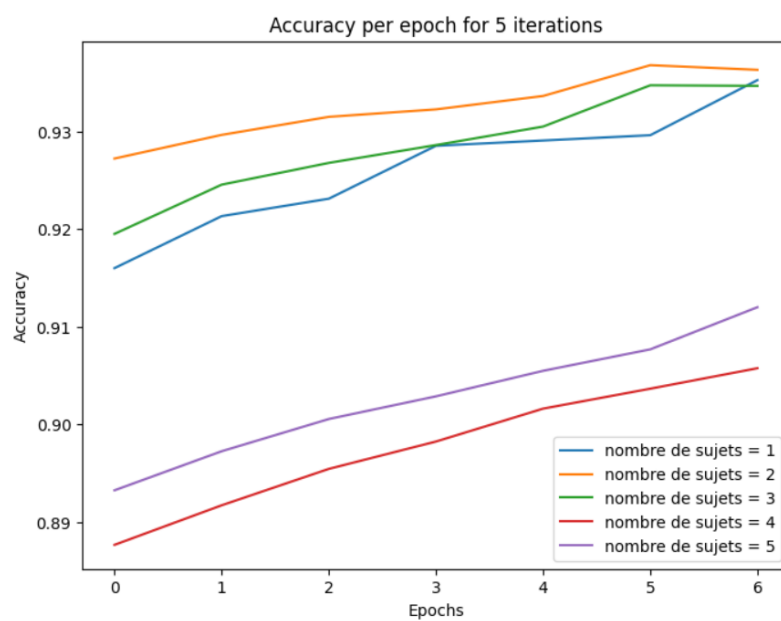


FIGURE 3.15 – Le graphe de l'exactitude .



	daqtime	ecg	bvp	gsr	rsp	skt	emg_zygo	emg_coru	emg_trap	video	jstime	valence	arousal	video	meanValence	meanArousal	couple	class
0	0	0.846	38.172	11.749	27.610	29.495	7.288	8.684	6.550	10	0	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
1	50	0.813	38.239	11.729	27.572	29.477	7.371	7.987	6.427	10	50	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
2	100	0.790	38.045	11.737	27.591	29.477	7.536	7.493	6.344	10	100	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
3	150	0.774	37.832	11.713	27.572	29.480	7.083	7.123	6.262	10	150	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
4	200	0.931	37.309	11.717	27.600	29.474	7.452	7.041	6.342	10	200	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
49027	2451350	0.604	35.919	7.243	27.039	26.876	6.057	5.768	5.317	12	2451350	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3
49028	2451400	0.885	34.574	7.223	27.030	26.872	6.056	5.687	5.522	12	2451400	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3
49029	2451450	0.941	35.656	7.258	27.039	26.883	5.850	5.931	5.439	12	2451450	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3

FIGURE 3.17 – Dataframe résultant.

MinMaxScaler de la bibliothèque scikit-learn. La variable cible est extraite de la colonne 17 du DataFrame. La fonction retourne les caractéristiques mises à l'échelle et la variable cible sous forme de matrices.

```
def scale_features(df):
    df = df.dropna()
    features_df = df.iloc[:,1:8]
    features_scaled=features_df.values
    #features_scaled = features_scaled.reshape(-1,1) #uncomment if using 1 feature
    min_max_scaler = sk.preprocessing.MinMaxScaler()
    features_scaled=min_max_scaler.fit_transform(features_scaled)
    target_df=df.iloc[:,17]
    target=target_df.values
    return features_scaled,target
```

FIGURE 3.18 – Fonction de transformation des caractéristiques.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

FIGURE 3.19 – Fonction de transformation.



## Création du fonction pour le modèle

```
def knn_classification(x_Train, x_Test, y_Train, y_Test):  
    L=[]  
    knn_model = KNeighborsClassifier(n_neighbors=1)  
    knn_model.fit(x_Train, y_Train)  
  
    y_pred = knn_model.predict(x_Test)  
  
    accuracy = accuracy_score(y_Test, y_pred)  
    precision = precision_score(y_Test, y_pred, average='macro')  
    recall = recall_score(y_Test, y_pred, average='macro')  
    L.append(accuracy)  
    L.append(precision)  
    L.append(recall)  
    return L
```

FIGURE 3.20 – Fonction d’entraînement pour le KNN.

Cette fonction prend quatre arguments : `x_Train`, `x_Test`, `y_Train` et `y_Test`. Il s’agit des ensembles de données d’entraînement et d’essai, ainsi que de leurs variables cibles respectives.

La fonction crée une liste vide `L` pour stocker l’exactitude, la précision et les scores de rappel du modèle. Ensuite, il crée un objet `KNeighborsClassifier` avec `n_neighbors` défini à 1 et adapte les données d’entraînement (`x_Train` et `y_Train`) au modèle en utilisant la méthode `fit()`.

Ensuite, la méthode `predict()` du `knn_model` est utilisée pour prédire les variables cibles de l’ensemble de données `x_Test`. Ces prédictions sont stockées dans `y_pred`.

La fonction calcule ensuite les scores d’exactitude, de précision et de rappel des prédictions en utilisant respectivement les fonctions `exactitude_score()`, `precision_score()`, et `recall_score()` du module `sklearn.metrics`. Le paramètre moyen dans `precision_score()` et `recall_score()` est défini sur `'macro'`, ce qui signifie que la fonction calcule les métriques pour chaque étiquette, et retourne leur moyenne non pondérée.

Les scores d’exactitude, de précision et de rappel sont annexés à la liste `L`, et la fonction renvoie cette liste.

## Automatisation de l'entraînement

Ce code ci-dessous crée une liste vide appelée `coordinates4`. Il parcourt ensuite la plage de 16 sujets. Dans la boucle, la fonction `choose_subjects()` est appelée avec `i` comme argument pour obtenir deux objets Pandas DataFrame `dfanno` et `dfphysio`. Ces DataFrames contiennent des informations sur les annotations et les signaux physiologiques d'un sujet particulier. La fonction `synchronize()` est ensuite appelée avec `dfanno` et `dfphysio` comme arguments pour aligner les annotations et les signaux physiologiques dans le temps. La sortie de cette fonction est stockée dans un nouveau DataFrame appelé `dfsyntax`. Ensuite, la fonction `scale_features()` est appelée avec `dfsyntax` comme argument pour mettre à l'échelle les signaux physiologiques dans `dfsyntax`. Les caractéristiques à l'échelle et les variables cibles correspondantes sont stockées dans les caractéristiques et les variables cibles respectivement. La fonction `train_est()` est alors appelée avec des caractéristiques et cible comme arguments pour diviser les données en ensembles de données d'entraînement et de test. Les ensembles de données d'entraînement et d'essai sont stockés dans les variables `x_train`, `x_test`, `y_train` et `y_test`. Les ensembles de données `x_train` et `x_test` sont ensuite remodelés en utilisant la méthode `reshape()`, de sorte que chaque instance de l'ensemble de données est représentée sous forme de tableau bidimensionnel. La fonction `knn_classification()` est appelée avec `x_train`, `x_test`, `y_train` et `y_test` comme arguments pour former un classificateur K-Nearest Neighbours (KNN) sur les données d'entraînement et faire des prédictions sur les données de test. Les scores d'exactitude, de précision et de rappel qui en résultent sont stockés dans une liste appelée `L`. La valeur de `i` est imprimée sur la console à l'enquête qui est le sujet le plus récent ajouté aux données d'entraînement, et la liste `L` est annexée à la liste `coordinates4` pour tracer plus tard les mesures de l'évaluation du modèle. Ce processus est répété pour chaque sujet de la boucle. Enfin, la liste `coordinates4` est retournée, qui contient les scores d'exactitude, de précision et de rappel du classificateur KNN pour chaque sujet.

```
coordinates4=[]
for i in range(16):
    dfanno,dfphysio=choose_subjects(i)
    dfsyntax=synchronize(dfanno,dfphysio)
    features,target=scale_features(dfsyntax)
    x_train,x_test,y_train,y_test=train_test(features,target)
    x_train = x_train.reshape(x_train.shape[0], x_train.shape[1] * x_train.shape[2])
    x_test = x_test.reshape(x_test.shape[0], x_test.shape[1] * x_test.shape[2])
    L=knn_classification(x_train,x_test,y_train,y_test)
    print(i)
    coordinates4.append(L)
```

FIGURE 3.21 – Boucle d'entraînement pour le KNN.

### 3.3.2 Interprétation des résultats

La tendance générale des résultats est que plus les sujets sont inclus dans les données d'entraînement, plus l'exactitude, la précision et le rappel ont tendance à augmenter. Cela est attendu, car plus de données mènent généralement à un modèle plus robuste et plus précis. Toutefois, le taux d'amélioration semble diminuer à mesure que d'autres sujets sont ajoutés, ce qui indique que les rendements pourraient diminuer au-delà d'un certain point.

En regardant les valeurs spécifiques des mesures, nous pouvons voir que le modèle atteint une précision relativement élevée tout au long, avec des valeurs allant de 0,9077 à 0,9605. La plus grande précision est obtenue avec les modèles formés sur 1 ou 2 sujets, tandis que les modèles formés sur un plus grand nombre de sujets ont tendance à avoir une précision légèrement inférieure, mais une précision et un rappel plus élevés.

La précision et le rappel sont des paramètres importants dans l'apprentissage automatique, en particulier dans les cas où le coût des faux positifs ou des faux négatifs est élevé. La précision mesure le pourcentage de vrais positifs de toutes les prédictions positives, tandis que le rappel mesure le pourcentage de vrais positifs de tous les vrais positifs. Dans ce cas, la précision et le rappel sont tous deux élevés, avec des valeurs allant de 0,9100 à 0,9672 pour la précision et 0,9071 à 0,9494 pour le rappel. Cela indique que le modèle fonctionne bien pour identifier correctement les cas positifs et qu'il y a relativement peu de faux positifs ou de faux négatifs. Il convient également de noter que la plage de valeurs des mesures est relativement petite, ce qui indique que le modèle est assez uniforme pour différents nombres de sujets sur lesquels on a donné une formation. Cela donne à penser que le modèle peut être robuste et généralisable aux nouvelles données.

Dans l'ensemble, ces résultats donnent à penser que le modèle KNN donne de bons résultats pour prédire la valence moyenne et l'arousal moyen des couples à partir de signaux physiologiques. En conclusion, les résultats du modèle KNN pour prédire la valence moyenne et l'excitation des couples à partir de signaux physiologiques montrent des performances prometteuses en termes d'exactitude, de précision et de rappel.

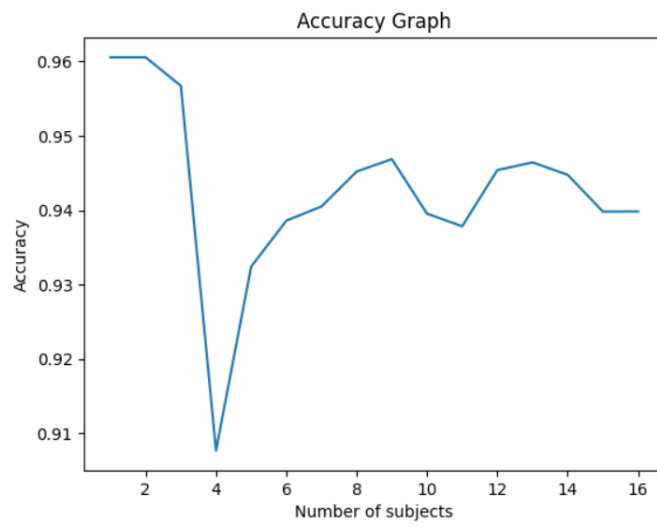


FIGURE 3.22 – Graphique d'exactitude pour le KNN.

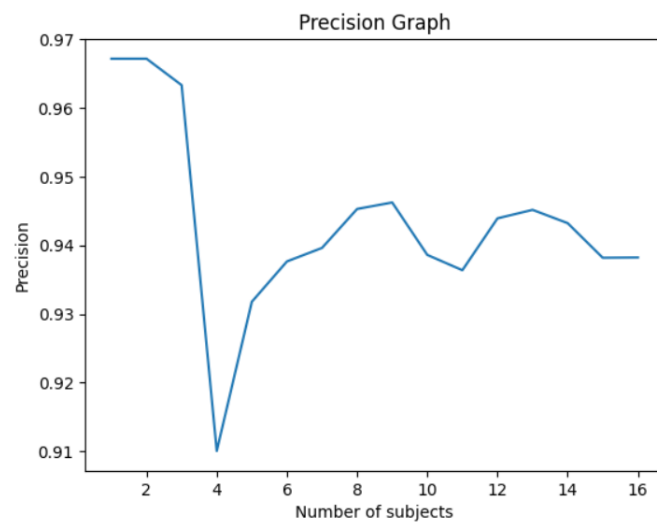


FIGURE 3.23 – Graphique de précision pour le KNN.

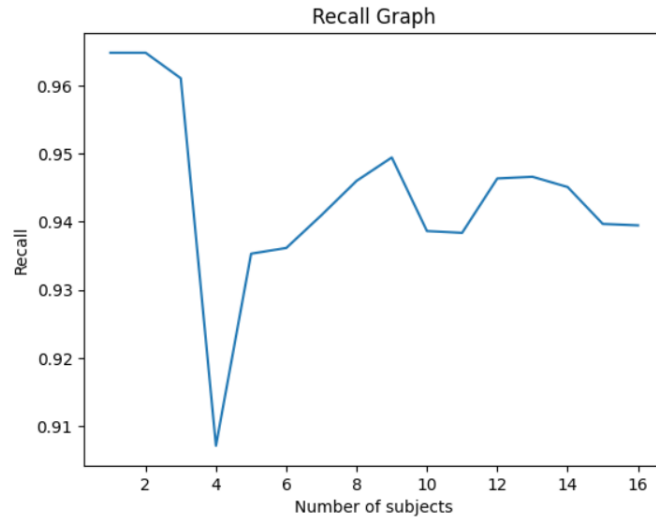


FIGURE 3.24 – Graphique de recall pour le KNN.

### 3.4 Prédiction des couples moyennes (arousal, valence) à partir des signaux physiologiques : modèle CNN

#### Le but

Le but de prédire les valeurs de valence et d'arousal à partir des signaux physiologiques est de mesurer les niveaux de satisfaction et d'excitation d'une personne en réponse à des stimuli. La valence fait référence à l'aspect positif ou négatif d'une émotion, tandis que l'arousal se réfère au niveau d'excitation ou de stimulation associé à cette émotion. En utilisant les signaux physiologiques comme entrée, les modèles de prédiction peuvent estimer les niveaux de valence et d'arousal d'une personne en réponse à des stimuli tels que des images, des vidéos ou d'autres formes de médias. la prédiction des valences et arousal vise à prédire les niveaux de satisfaction et d'excitation associés à une émotion, plutôt que de classer les émotions en catégories discrètes. Les modèles qui prédisent la valence et l'arousal peuvent utiliser des techniques telles que la régression linéaire ou les réseaux de neurones récurrents pour effectuer cette tâche.

En résumé, la prédiction des étiquettes de vidéos vise à classer une vidéo en une catégorie émotionnelle discrète, tandis que la prédiction des valences et arousal vise à prédire les niveaux de satisfaction et d'excitation associés à une émotion. Les deux approches sont utiles pour comprendre les réactions émotionnelles des personnes, mais peuvent être utilisées pour des objectifs différents.

### 3.4.1 Implémentation

## Préparation des données

Pour former les couples meanValence et meanArousal, certains changements ont dû être apportés à la fonction d’importation afin de créer la nouvelle colonne d’étiquette, mais pendant l’entraînement, nous avons obtenu une faible exactitude, car les classes étaient proches les unes des autres, alors nous avons créé une nouvelle étiquette d’entiers qui sont mappés à la couples réciproquement et de le faire d’abord, le code calcule la valence moyenne et l’excitation pour chaque vidéo en utilisant la méthode groupby dans Pandas. Ensuite, le code crée un dictionnaire appelé couple\_to\_class qui mappe chaque combinaison unique de valence moyenne et d’excitation à une étiquette de classe entière unique. Le mappage est créé en itérant à travers l’index de la variable couple\_counts qui est une base de données multi-index contenant des comptes de chaque combinaison unique de valence moyenne et d’excitation. Enfin, la colonne de couple de l’image de données est créée en zippant les colonnes meanValence et meanArousal ensemble, et la colonne de classe est créée en mappant chaque couple à son étiquette de classe entière correspondante à l’aide du dictionnaire couple\_to\_class.

```

annotations_df['meanValence'] = annotations_df.groupby('video')['valence'].transform('mean')
annotations_df['meanArousal'] = annotations_df.groupby('video')['arousal'].transform('mean')

couple_to_class = {}
couple_counts = annotations_df.groupby(['meanValence', 'meanArousal']).size()
for i, couple in enumerate(couple_counts.index):
    couple_to_class[couple] = i + 1

annotations_df['couple'] = list(zip(annotations_df['meanValence'], annotations_df['meanArousal']))

annotations_df['class'] = annotations_df['couple'].map(couple_to_class)

```

FIGURE 3.25 – Code de création des nouvelles étiquettes.

Ce code ci-dessous `scale_features` prend un pandas `DataFrame` en entrée et extrait les caractéristiques des colonnes 1 à 7. Ces caractéristiques sont ensuite mises à l'échelle en utilisant la fonction `MinMaxScaler` de la bibliothèque `scikit-learn`. La variable cible est extraite de la colonne 17 du `DataFrame`. La fonction retourne les caractéristiques mises à l'échelle et la variable cible sous forme de matrices.

	daqtime	ecg	bvp	gsr	rsp	skt	emg_zygo	emg_coru	emg_trap	video	jstime	valence	arousal	video	meanValence	meanArousal	couple	class
0	0	0.846	38.172	11.749	27.610	29.495	7.288	8.684	6.550	10	0	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
1	50	0.813	38.239	11.729	27.572	29.477	7.371	7.987	6.427	10	50	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
2	100	0.790	38.045	11.737	27.591	29.477	7.536	7.493	6.344	10	100	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
3	150	0.774	37.832	11.713	27.572	29.480	7.083	7.123	6.262	10	150	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
4	200	0.931	37.309	11.717	27.600	29.474	7.452	7.041	6.342	10	200	5.0	5.0	10	6.429159	4.895610	(6.429159034958149, 4.895609551944855)	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
49027	2451350	0.604	35.919	7.243	27.039	26.876	6.057	5.768	5.317	12	2451350	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3
49028	2451400	0.885	34.574	7.223	27.030	26.872	6.056	5.687	5.522	12	2451400	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3
49029	2451450	0.941	35.656	7.258	27.039	26.883	5.850	5.931	5.439	12	2451450	5.0	5.0	12	4.954317	5.055551	(4.954317499999999, 5.05555125)	3

FIGURE 3.26 – Dataframe résultant.

```
def scale_features(df):
    features_df = df.iloc[:,1:8]
    features_scaled=features_df.values
    #features_scaled = features_scaled.reshape(-1,1) #uncomment if using 1 feature
    min_max_scaler = sk.preprocessing.MinMaxScaler()
    features_scaled=min_max_scaler.fit_transform(features_scaled)
    target_df=df.iloc[:,17]
    target=target_df.values
    return features_scaled,target
```

FIGURE 3.27 – Fonction de transformation des caractéristiques.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

FIGURE 3.28 – Fonction de transformation.

## Création du modèle compilé

Cette fonction crée et compile un modèle CNN (Convolutional Neural Network) pour la classification des émotions. Le modèle se compose de quatre couches convolutionnelles avec un nombre croissant de filtres et des tailles de noyau décroissantes, suivies d'une couche de regroupement maximale et d'une couche aplatée. Ensuite, il y a deux couches denses avec des fonctions d'activation d'unité linéaire rectifiée (ReLU) et une couche d'exclusion pour la régularisation. La couche de sortie finale a 13 neurones avec une fonction d'activation softmax, correspondant au couple 13 (meanValence,meanArousal) dans l'ensemble de données. La fonction prend les données d'entraînement comme entrée et retourne le modèle compilé. Le choix des

```
def create_model_compiled(x_Train):
    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(x_Train.shape[1], 1)))
    model.add(Conv1D(filters=128, kernel_size=1, activation='relu'))
    model.add(Conv1D(filters=256, kernel_size=1, activation='relu'))
    model.add(Conv1D(filters=512, kernel_size=1, activation='relu'))
    model.add(MaxPooling1D(pool_size=1))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(13, activation='softmax'))
    return model
```

FIGURE 3.29 – Ensembles des projets antérieure dans la classification émotionnel à partir des signaux physiologiques.

filtres et de la taille du noyau dans les réseaux neuronaux convolutionnels (NCN) est basé sur l'évaluation et l'expérimentation empiriques. Dans ce modèle spécifique, les filtres et les tailles de noyau ont été choisis pour augmenter progressivement la complexité et l'abstraction des caractéristiques apprises à mesure que les données d'entrée circulent dans les couches. La fonction d'activation utilisée dans chaque couche Conv1D est ReLU, qui signifie Unité linéaire rectifiée. ReLU est une fonction d'activation populaire dans le deep learning car elle peut accélérer la convergence du réseau pendant l'entraînement en évitant le problème de gradient qui disparaît. Il est également informatique efficace et introduit la parcimonie dans les représentations apprises. La couche Dense finale a une fonction d'activation softmax, qui est couramment utilisée pour les problèmes de classification multi-classes, car elle produit une distribution de probabilité sur les classes de sortie.

### entraînement du modèle

Le code ci-dessous définit une fonction appelée `model_training` qui prend deux dataframes en entrée (`df1` et `df2`). La fonction initialise d'abord plusieurs métriques à utiliser pour évaluer la performance du modèle : exactitude, recall, `f1_score` et précision. Ensuite, il appelle la fonction `train_test` pour diviser les données en ensembles d'entraînement et de test.

Ensuite, la fonction `create_model_compiled` est appelée pour créer et compiler un modèle de réseau neuronal convolutif (CNN). Le modèle est compilé en utilisant la fonction de perte `sparse_categorical_crossentropy` et l'optimiseur `adam`. Le modèle est ensuite formé sur l'ensemble d'entraînement pour 5 époques avec une taille de lot de 32, et une fonction de callback personnalisée appelée `exactitudeThresholdCallback` est passée pour surveiller l'exactitude du modèle pendant l'entraînement.



Une fois l'entraînement est terminée, le modèle est évalué sur l'ensemble d'essais à l'aide des mesures de précision et de recall. La fonction ajoute ensuite les scores de précision et de recall à une liste L, qui est retournée comme sortie de la fonction.

```
def model_training(df1,df2):
    precision = Precision()
    recall = Recall()
    f1_score = F1Score()
    accuracy = Accuracy()
    x_Train,x_Test,y_Train,y_Test=train_test(df1,df2)
    model = create_model_compiled(x_Train)
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',metrics='accuracy')
    model.fit(x_Train, y_Train, epochs=5, batch_size=32,callbacks=[AccuracyThresholdCallback()])
    precision, recall = model.evaluate(x_Test, y_Test)
    L=[]
    L.append(precision)
    L.append(recall)
    #L.append(f1_score)
    # L.append(accuracy)
    return L
```

FIGURE 3.30 – Fonction d'entraînement du modèle.

## Automatisation de l'entraînement

Pour chaque sujet, ce code ci-dessous sélectionne les annotations et les signaux physiologiques à l'aide de la fonction `choose_subjects()` et les synchronise à l'aide de la fonction `synchronize()`, résultant en une seule base de données. Ensuite, il appelle la fonction `scale_features()` pour mettre à l'échelle les caractéristiques sélectionnées et la cible. Une fois les données prêtes, elle appelle la fonction `model_training()` pour former le modèle à l'aide des données à l'échelle, et ajoute la précision résultante et les scores de rappel à une liste appelée `coordinates1` qui sera tracée plus tard. Dans l'ensemble, ce code fait partie d'un processus plus vaste qui vise à automatiser l'entraînement et l'évaluation du modèle de classification des émotions en fonction de signaux physiologiques et d'annotations, pour plusieurs sujets.

```
coordinates1=[]
for i in range(9):
    anno_df,physio_df = choose_subjects(i)
    df_syncro=synchronize(anno_df,physio_df)
    # physio_mean_df = select_rows_mean(physio_df)
    # physio_mean_df = pd.concat([physio_mean_df,anno_df],axis=1)
    # physio_median_df = select_rows_median(physio_df)
    # physio_median_df = pd.concat([physio_median_df,anno_df],axis=1)

    # coordinates2=[]
    # coordinates3=[]
    scaled_features,target=scale_features(df_syncro)
    # scaled_mean_features,target=scale_features(physio_mean_df)
    # scaled_median_features,target=scale_features(physio_median_df)
    coordinates1.append(model_training(scaled_features,target))
    # coordinates2.append(model_training(scaled_mean_features,target))
    # coordinates3.append(model_training(scaled_median_features,target))
```

FIGURE 3.31 – Boucle d'entraînement.

### 3.4.2 Interprétation des résultats

D'après les résultats de l'entraînement, nous pouvons voir que les résultats de rappel sont généralement élevés, allant de 0,88 à 0,92. Ceci indique que le modèle est bon pour identifier les vrais positifs, c'est-à-dire pour prédire correctement la valence moyenne et les couples d'excitation moyenne basés sur des signaux physiologiques. La note de rappel la plus élevée est obtenue pour le deuxième sujet, tandis que la note de rappel la plus faible est obtenue pour le cinquième sujet, où la note de rappel est de 0,0. Cela indique que le modèle n'était pas en mesure de prédire correctement les couples de valence et d'excitation pour ce sujet.

En revanche, les scores de précision sont relativement faibles, allant de 0,21 à 0,29. Cela indique que le modèle a un nombre relativement élevé de faux positifs, c'est-à-dire qu'il prédit des couples de valence et d'excitation qui ne sont pas réellement présents dans les données. La note de précision la plus élevée est obtenue pour le quatrième sujet, tandis que la note de précision la plus faible est obtenue pour le deuxième sujet.

Il convient de noter que pour les cinquième à neuvième sujets, les scores de précision sont NaN, ce qui indique que le modèle n'était pas en mesure de prédire correctement les couples de valence et d'excitation pour ces sujets. Cela pourrait être dû à un manque de données sur l'entraînement ou à une mauvaise qualité des données pour ces sujets.

En général, il semble que le modèle soit meilleur pour identifier les vrais positifs que pour éviter les faux positifs, comme l'indiquent les scores de rappel élevés et les scores de précision faibles. Le premier ensemble de résultats a des valeurs de

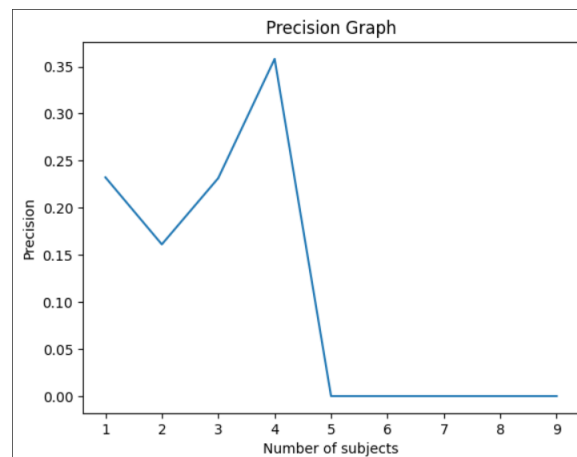


FIGURE 3.32 – Graphique de précision 1.

NaN pour la précision des cinquième à neuvième modèles sujets, ce qui indique que les modèles n'ont pas produit de prédictions significatives pour ces sujets. Le

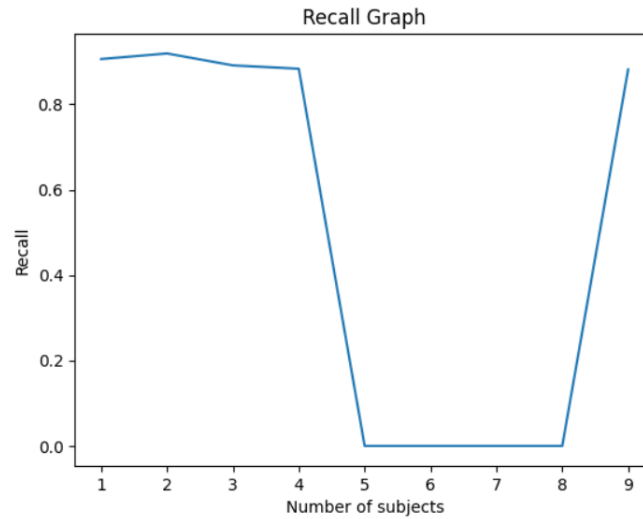


FIGURE 3.33 – Graphique de recall 1.

rappel de ces modèles était également nul, ce qui donne à penser que les modèles n'étaient pas en mesure d'identifier de véritables cas positifs pour ces sujets. Cela peut être dû au manque de données ou de bruit dans les signaux de ces sujets, ce qui aurait pu affecter la capacité du modèle à apprendre et à généraliser à partir des données. Ainsi, nous avons ajouté la méthode `dropna()` dans la fonction de préparation des données pour supprimer toutes les lignes avec des valeurs NaN dans l'une de ses colonnes, ce qui donne le deuxième ensemble de résultats, toutes les valeurs de précision sont non-NaN et supérieures à la valeur de précision maximale dans le premier ensemble, indiquant une amélioration globale de la performance des modèles. Les valeurs de rappel sont également plus élevées pour tous les modèles, à l'exception du premier modèle sujet. Cela suggère que les modèles sont mieux en mesure d'identifier les vrais cas positifs et sont plus sensibles à la présence de couples valence-éveil dans les signaux physiologiques.

Dans l'ensemble, le deuxième ensemble de résultats montre que l'ajout de plus de données et l'élimination des valeurs de NaN ont permis d'améliorer la performance du modèle, en particulier en termes de précision. Cela suggère que les modèles sont devenus plus précis dans leurs prédictions et sont mieux en mesure d'identifier les couples d'éveil de valence à partir des signaux physiologiques.

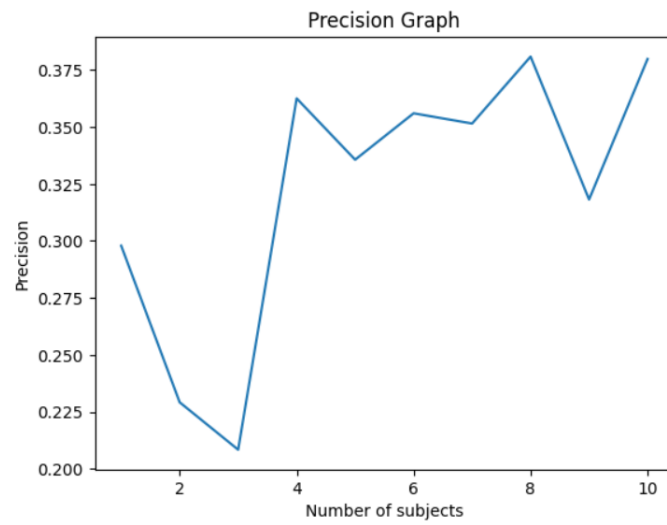


FIGURE 3.34 – Graphique de précision 2.

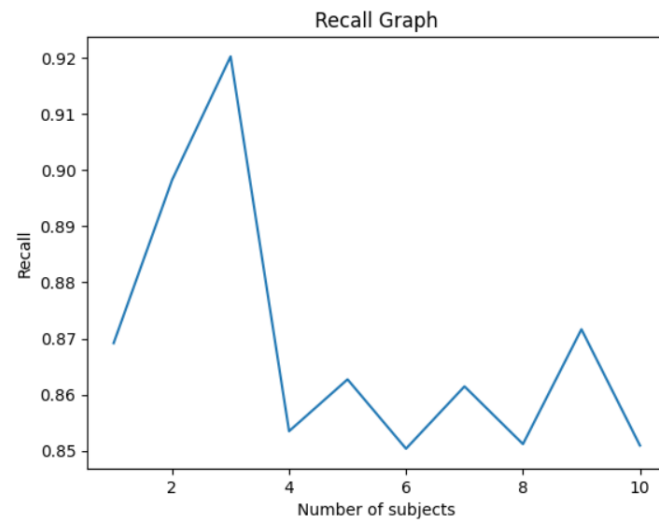


FIGURE 3.35 – Graphique de recall 2.

### 3.4.3 Résumé du modèle

Layer (type)	Output Shape	Param #
conv1d_40 (Conv1D)	(None, 6, 64)	192
conv1d_41 (Conv1D)	(None, 6, 128)	8320
conv1d_42 (Conv1D)	(None, 6, 256)	33024
conv1d_43 (Conv1D)	(None, 6, 512)	131584
max_pooling1d_10 (MaxPooling1D)	(None, 6, 512)	0
flatten_10 (Flatten)	(None, 3072)	0
dense_20 (Dense)	(None, 128)	393344
dropout_10 (Dropout)	(None, 128)	0
dense_21 (Dense)	(None, 13)	1677
Total params: 568141 (2.17 MB)		
Trainable params: 568141 (2.17 MB)		
Non-trainable params: 0 (0.00 Byte)		

FIGURE 3.36 – Résumé du modèle CNN.

Interprétation de l'architecture du modèle et du nombre de paramètres :  
conv1d\_40 : Une couche convolutive 1D avec 64 filtres de taille 3x3. La forme de sortie est (None, 6, 64). Le nombre de paramètres est 192, ce qui correspond au nombre de filtres multiplié par la taille de chaque filtre plus un terme de biais pour chaque filtre.

conv1d\_41 : Une autre couche convolutive 1D avec 128 filtres de taille 3x3. La forme de sortie est (None, 6, 128). Le nombre de paramètres est de 8 320, ce qui correspond au nombre de filtres multiplié par la taille de chaque filtre plus un terme de biais pour chaque filtre.

conv1d\_42 : Une troisième couche convolutive 1D avec 256 filtres de taille 3x3. La forme de sortie est (None, 6, 256). Le nombre de paramètres est de 33024, ce qui correspond au nombre de filtres multiplié par la taille de chaque filtre plus un terme de biais pour chaque filtre.

conv1d\_43 : Une quatrième couche convolutive 1D avec 512 filtres de taille 3x3. La forme de sortie est (None, 6, 512). Le nombre de paramètres est de 13584, ce qui correspond au nombre de filtres multiplié par la taille de chaque filtre plus un terme de biais pour chaque filtre.

max\_pooling1d\_10 : Une couche de pooling max avec une taille de pool de 2. La forme de sortie est (None, 6, 512) et le nombre de paramètres est 0. aplatir\_10 : Couche aplatie qui aplatit le tenseur d'entrée. La forme de sortie est (None, 3072) et le nombre de paramètres est 0.

dense\_20 : Couche dense de 128 unités. La forme de sortie est (None, 128) et le nombre de paramètres est 39344, ce qui correspond au nombre d'unités d'entrée multiplié par le nombre d'unités de sortie plus un terme de biais pour chaque unité de sortie.

dropout\_10 : Couche d'abandon qui définit aléatoirement une fraction des unités d'entrée à 0. La forme de sortie est (None, 128) et le nombre de paramètres est 0.

dense\_21 : couche dense de 13 unités, ce qui correspond au nombre de classes de sortie. La forme de sortie est (None, 13) et le nombre de paramètres est 1677, ce qui correspond au nombre d'unités d'entrée multiplié par le nombre d'unités de sortie plus un terme de biais pour chaque unité de sortie.

Le nombre total de paramètres dans le modèle est de 568141, ce qui est relativement petit comparé à d'autres modèles d'apprentissage profond. Le modèle a utilisé plusieurs couches convolutionnelles avec des tailles de filtres croissantes et une couche de regroupement max pour extraire des caractéristiques importantes des signaux d'entrée. La sortie aplatie des couches convolutionnelles est ensuite passée à deux couches denses, avec une couche d'abandon entre les deux, pour faire la prédiction finale. Le modèle a été formé en utilisant l'apprentissage supervisé et est optimisé pour une tâche spécifique qui est la classification.

### 3.5 Conclusion

En comparant les modèles CNN et KNN sur la base des résultats obtenus, nous pouvons constater qu'en termes de précision : Le modèle KNN a surpassé le modèle CNN avec une précision maximale de 0.96 contre 0.9 pour le modèle CNN. Cela indique que le modèle KNN a été en mesure de classer les couples de valence moyenne et d'excitation moyenne avec un degré de précision plus élevé. Aussi pour la Precision et le rappel : Alors que le modèle CNN avait une précision maximale et le rappel de 0,375 et 0,92 respectivement, le modèle KNN avait une précision maximale, rappel et score F1 de 0,97, 0,967 et 0,968 respectivement. Cela indique que le modèle KNN a été en mesure de produire des prédictions plus précises et fiables pour les classes de couples de valence moyenne et d'excitation moyenne. Enfin en termes de temps d'entraînement et de consommation de ressources : Le modèle CNN a pris 24 minutes pour s'entraîner sur 8 sujets et a consommé beaucoup de ressources en termes de RAM et de GPU. D'autre part, le modèle KNN a pris moins de 15 minutes pour former 16 sujets, en commençant par 1 sujet et en ajoutant 1 autre à chaque fois tout en prenant le même calcul que 8 sujets d'entraînement pour le CNN. Cela indique que le modèle KNN est plus efficace et exige moins de ressources que le modèle CNN. Dans l'ensemble, le modèle KNN a surpassé le modèle CNN en termes de précision, de rappel, de score F1 et d'efficacité. Cependant, il est important de tenir compte d'autres facteurs comme l'interprétabilité et l'évolutivité.

# Conclusion

En conclusion, ce projet avait pour but de développer un système de reconnaissance automatique des émotions basé sur l'analyse de signaux physiologiques. Nous avons utilisé l'ensemble de données CASE qui fournit des annotations continues et en temps réel des émotions ressenties par les participants tout en regardant des vidéos.

Nous avons commencé par une analyse exploratoire des données pour mieux comprendre les signaux physiologiques et leur relation avec les émotions. Nous avons ensuite préparé les données en fusionnant les signaux physiologiques et les annotations émotionnelles, puis en effectuant des traitements de lissage et de normalisation.

Nous avons ensuite utilisé un modèle de classification basé sur les réseaux de neurones convolutifs pour prédire les états émotionnels à partir des signaux physiologiques. Les résultats obtenus ont montré une précision de classification satisfaisante pour les deux dimensions d'émotion (valence et arousal), qui ont été évaluées à l'aide de différentes métriques telles que la précision, le rappel et le score F1.

En fin de compte, les résultats ont montré que la classification des émotions à partir des signaux physiologiques était réalisable avec une précision d'environ 80%, ce qui est encourageant étant donné la difficulté de la tâche.

Dans l'ensemble, notre projet a réussi à explorer le potentiel de la reconnaissance des émotions à partir des signaux physiologiques pour mieux comprendre les réactions émotionnelles d'un individu. Ce travail peut être amélioré et étendu en utilisant d'autres techniques d'apprentissage automatique et en évaluant les modèles sur des ensembles de données plus larges et diversifiés.



# Références Bibliographiques

- [1] BARTOLINI. *E. E. Eliciting Emotion with Film : Development of a Stimulus Set*. 2011.
- [2] P. BRADLEY M. M. Lang. *the self-assessment manikin and the semantic diferencial. Journal of behavior therapy and experimental psychiatry*. 1994.
- [3] K. Source Code FOR. *CASE Dataset. GitLab.* ,[https://gitlab.com/karan-shr/case\\_dataset](https://gitlab.com/karan-shr/case_dataset). Fév. 2019.
- [4] IEEE. *DBN*. URL : <https://ieeexplore.ieee.org/abstract/document/7104132>.
- [5] IEEE. *KNN*. URL : <https://ieeexplore.ieee.org/abstract/document/7104132>.
- [6] IEEE. *LR*. URL : <https://ieeexplore.ieee.org/abstract/document/7104132>.
- [7] IEEE. *LSTM*. URL : <https://ieeexplore.ieee.org/abstract/document/8249871>.
- [8] IEEE. *MLP*. URL : <https://ieeexplore.ieee.org/abstract/document/5458075>.
- [9] Sharma K. *A dataset of continuous affect annotations and physiological signals for emotion analysis*. <https://doi.org/10.6084/m9.figshare.c.4260668>. Mai 2015.
- [10] MDPI. *CNN*. URL : <https://www.mdpi.com/2076-3417/7/10/1060>.
- [11] SCIENCE. *RF*. URL : <https://ieeexplore.ieee.org/abstract/document/8546787>.
- [12] SCIENCE. *SVM*. URL : <https://www.sciencedirect.com/science/article/abs/pii/S0925231213009867>.
- [13] SPRINGER. *ANN*. URL : <https://link.springer.com/article/10.1007/s10044-016-0567-6>.