SPEECH COMMAND

BARRY Saliou & MOUBARAK Carine
Sous la supervision du Mr. Olivier SCHWANDER

17 mai 2024

Table des matières

I Introduction **3**

II Exploration des papiers **3**

III Dataset et Features **3**
III.1 Préparation du Dataset . . . 3
III.2 Extraction des caracteristiques . . . 3

IV Augmentation des données **4**

V Models **4**
V.1 Régression Logistique . . . 5
V.2 Simple Neural Network . . . 5
V.3 Convolutional Neural Network . . . 5

VI Protocole expérimental **5**

VII Résultats **6**
VII.1 Régression Logistique . . . 6
VII.2 SNN . . . 6
VII.3 CNN . . . 6

Conclusion **7**

I Introduction

L'évolution rapide des technologies de commande vocale a ouvert la voie à des avancées significatives, notamment dans le domaine de la reconnaissance de mots-clés, qui permet des interactions instantanées avec des dispositifs intelligents grâce à des commandes telles que "Hey Siri" ou "Ok Google". Notre projet s'inscrit dans cette dynamique, avec l'ambition de rapprocher les performances de notre modèle de celles des systèmes de pointe, tout en interrogeant l'une des pratiques prédominantes dans le domaine : l'augmentation des données.

Nous avons exploré la pertinence de cette méthode largement adoptée, souvent considérée comme un passage obligé pour améliorer la précision des modèles de reconnaissance de mots-clés. Notre recherche se penche sur la question cruciale de savoir si l'accumulation de données supplémentaires se traduit par une amélioration proportionnelle des performances ou si elle représente une surcharge inutile.

À travers une série d'expérimentations et d'analyses, nous avons cherché à comprendre l'impact réel de l'augmentation des données sur la reconnaissance efficace de mots-clés spécifiques. Cette introduction pose les bases d'une étude qui ne se contente pas de suivre les tendances, mais qui vise à éclairer sur l'efficacité et la nécessité de cette pratique courante dans le contexte spécifique de la reconnaissance de mots-clés

II Exploration des papiers

Notre revue de la littérature sur la reconnaissance de mots-clés (Keyword Spotting, KWS) a mis en évidence des approches variées dans l'utilisation de réseaux neuronaux et de techniques d'augmentation de données.

Streaming KWS on Mobile Devices : [1]
Cette étude évalue des modèles de réseaux neuronaux sur les datasets Google V1 et V2, entraînés pour reconnaître douze labels. Les techniques d'augmentation de données incluent des décalages temporels et l'ajout de bruit de fond. L'accuracy est utilisée comme métrique principale, reflétant une distribution équilibrée des classes.

Audio Spectrogram Transformer : [2]
L'AST, une architecture entièrement basée sur des mécanismes d'attention et dépourvue de convolutions, est appliquée directement aux spectrogrammes audio. Contrairement aux autres modèles qui utilisent des MFCC pour le prétraitement, l'AST tire parti des modèles pré-entraînés sur ImageNet, démontrant son potentiel en tant que classificateur audio générique sur le dataset Speech Commands V2.

MatchboxNet : [3]
MatchboxNet utilise des convolutions séparables en temps et en canaux pour une reconnaissance efficace des commandes vocales. Il est entraîné sur deux versions du dataset Google Speech Commands, avec des augmentations de données comprenant des perturbations temporelles et du bruit blanc. Le prétraitement des données utilise des MFCC, calculés à partir de fenêtres de 25 ms avec un chevauchement de 10 ms.

Keyword Spotting on Microcontrollers : [4]
Cette recherche se concentre sur le KWS pour les microcontrôleurs, nécessitant une haute précision et une efficacité énergétique. Les modèles sont entraînés pour classifier des mots-clés spécifiques à partir de clips audio d'une seconde, avec une augmentation de données incluant du bruit de fond et des décalages temporels aléatoires.

- **Points Communs :** La plupart des études utilisent des MFCC pour le prétraitement des données audio, à l'exception de l'AST qui utilise des modèles pré-entraînés sur ImageNet.
- **Différences Notables :** Les modèles diffèrent dans le nombre de labels qu'ils sont entraînés à reconnaître, allant de douze à trente-cinq labels, et dans les versions du dataset Speech Commands utilisées.

III Dataset et Features

III.1 Préparation du Dataset

Nous avons utilisé le dataset SPEECHCOMMAND v01 [5], une collection fournie par PyTorch qui comprend 64,727 fichiers audio .wav. Chaque fichier d'une seconde contient un mot anglais unique prononcé par différents locuteurs.
Le dataset se compose de mots principaux tels que "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", et "Nine". En outre, pour aider à distinguer les mots non reconnus, il contient également dix mots auxiliaires, que la plupart des locuteurs n'ont prononcés qu'une seule fois, comme "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree", et "Wow".
Les audios ont été échantillonnés à 16000 Hz et coupés à une durée d'une seconde pour aligner la plupart des énoncés.
Pour simuler des environnements bruyants, des données de fond de bruit étaient également incluses dans le dataset.
Les audios ont été divisés en ensembles d'entraînement, de validation et de test avec des proportions de 80%, 10% et 10% respectivement, et six audios pour le bruit de fond, donnant les valeurs suivantes : entraînement : 51,088 audios, test : 6,835 audios, validation : 6,798 audios.

III.2 Extraction des caracteristiques

Nous avons opté pour une approche rigoureuse d'extraction des caractéristiques en utilisant les Coefficients Céphalo-Caustiques de Fréquence de Mel (Mel-Frequency Cepstral Coefficients, MFCC), qui sont largement reconnus pour leur efficacité dans l'analyse des signaux vocaux. Les MFCC sont particulièrement adaptés pour capturer les propriétés acoustiques pertinentes pour la reconnaissance de mots-clés.
Le schéma d'extraction des caractéristiques est illustré dans la Figure.

1. Nous avons d'abord défini une fenêtre d'analyse de 30 ms, et divisé le signal vocal en différents cadres temporels en décalant la fenêtre (décalage de 10 ms). Puisque le signal audio est de 1 seconde, nous obtiendrons (1000-30)/10+1=98 cadres temporels, comme illustré dans la Figure

2. Après la fenêtrage, la transformation de Fourier rapide (FFT) est calculée pour chaque cadre afin d'obtenir les ca-

ractéristiques de fréquence, et le banc de filtres Mel-échelonnés logarithmiques est appliqué aux cadres transformés par Fourier. La dernière étape consiste à calculer la transformation en cosinus discrète (DCT) pour obtenir le vecteur de coefficients à 40 dimensions. Dans ce projet, nous avons finalement obtenu une matrice 2D [98×40] désirée pour alimenter le réseau neuronal subséquent.

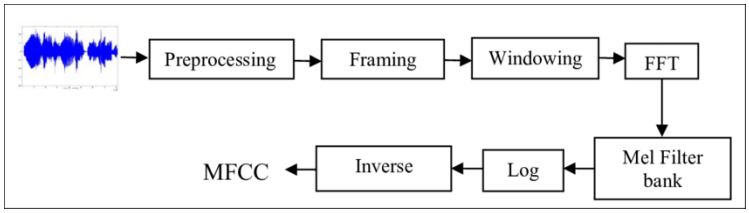

FIGURE 1 – Diagramme du processus de dérivation du MFCC

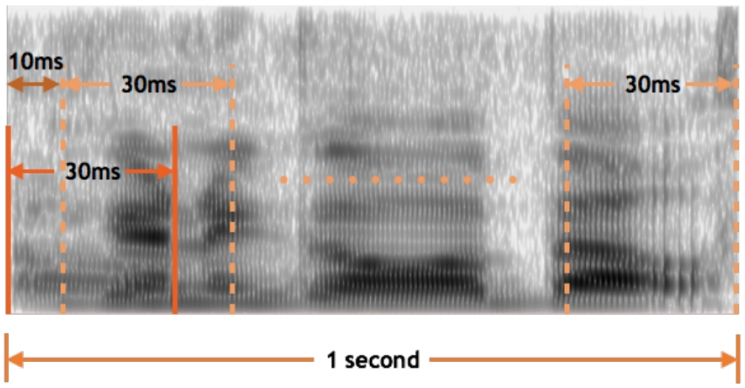

FIGURE 2 – Fenêtre d'exaction

IV Augmentation des données

[6] L'impact de l'augmentation des données sur la performance du modèle de reconnaissance de mots-clés, l'objectif est de déterminer si l'introduction de variations artificielles dans les données d'entraînement améliore la capacité du modèle à généraliser à de nouvelles données, Voici les techniques d'augmentation que nous avons utilisées :

Time Shift (Décalage Temporel) :
Valeurs : $[-100, 100]$ ms Le décalage temporel est appliqué en déplaçant aléatoirement le signal audio dans le temps de -100 à $+100$ millisecondes. Cette variation simule un début de parole anticipé ou retardé, renforçant la capacité du modèle à reconnaître les mots-clés indépendamment de leur position temporelle.

Time Masking (Masquage Temporel) :
Valeurs : $[0, 25]$ pas de temps
Le masquage temporel occulte aléatoirement une portion du signal allant jusqu'à 25 pas de temps. Cela encourage le modèle à ne pas se fier à des segments temporels spécifiques et à mieux utiliser le contexte global du signal.

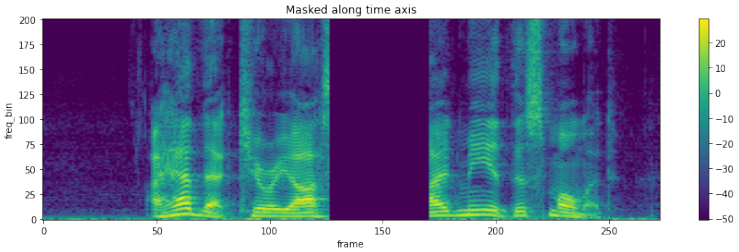

FIGURE 3 – Time Masking

Frequency Masking (Masquage Fréquentiel) :
Valeurs : $[0, 15]$ bandes de fréquence
De manière similaire au masquage temporel, le masquage fréquentiel cache des parties du signal allant jusqu'à 15 bandes de fréquence, poussant le modèle à utiliser l'ensemble du spectre pour la classification.

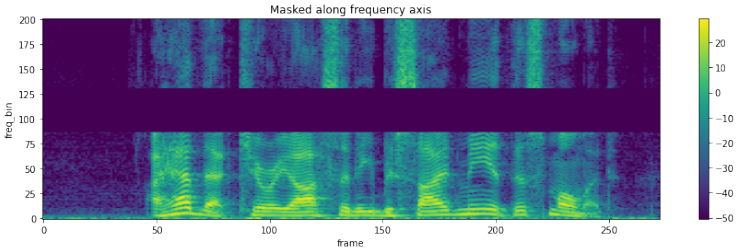

FIGURE 4 – Frequency Masking

Background Noise (Bruit de Fond) :
Valeurs : $[0.05, 0.2]$ volume relatif
L'ajout de bruit de fond avec un volume relatif variant de 5% à 20% prépare le modèle à reconnaître les mots-clés même en présence de bruit ambiant, une compétence essentielle pour les applications en environnement réel.

Stratégies d'Augmentation Testées :
Duplication : Chaque échantillon de l'ensemble d'entraînement est augmenté, permettant de tester si une plus grande variété de conditions audio améliore la performance.
Augmentation Aléatoire : Avec une probabilité de 40%, un échantillon est augmenté pendant l'entraînement, évaluant si l'introduction de variations de manière non systématique est bénéfique.

V Models

Nous avons exploré plusieurs modèles pour le traitement du signal audio dans le but de les comparer et de mieux comprendre leurs performances respectives. Nous avons débuté avec la Régression Logistique. Ensuite, nous avons introduit un Réseau de Neurones Simple, fréquemment choisi comme point de départ pour des tâches de classification. Pour aller plus loin, nous nous sommes penchés sur le Réseau de Neurones Convolutif (CNN), reconnu pour son efficacité dans l'extraction des caractéristiques spatiales et temporelles des données audio.

V.1 Régression Logistique

La Régression Logistique est souvent choisie comme point de départ pour des tâches de classification en raison de sa simplicité et de sa facilité de mise en œuvre. Bien qu'elle soit un modèle linéaire simple, elle peut servir de référence pour évaluer les performances des modèles plus complexes dans le cadre de notre étude.

V.2 Simple Neural Network

Notre modèle est un réseau neuronal entièrement connecté avec deux couches cachées, chaque couche contenant 256 neurones cachés, comme illustré dans la Figure 5. Nous avons choisi d'utiliser deux couches cachées car, en pratique, un réseau neuronal entièrement connecté à deux couches cachées surpasse généralement les réseaux avec une couche cachée, mais est légèrement moins performant que ceux avec quatre couches cachées ou plus. Pour les couches cachées, nous avons utilisé la fonction d'activation ReLU (Rectified Linear Unit) pour calculer la réduction du réseau, c'est-à-dire la somme pondérée des sorties de la couche précédente. Comparé à une regression logistique, ce modèle devrait fournir des résultats plus précis au prix d'une empreinte mémoire plus importante et d'un coût computationnel plus élevé.

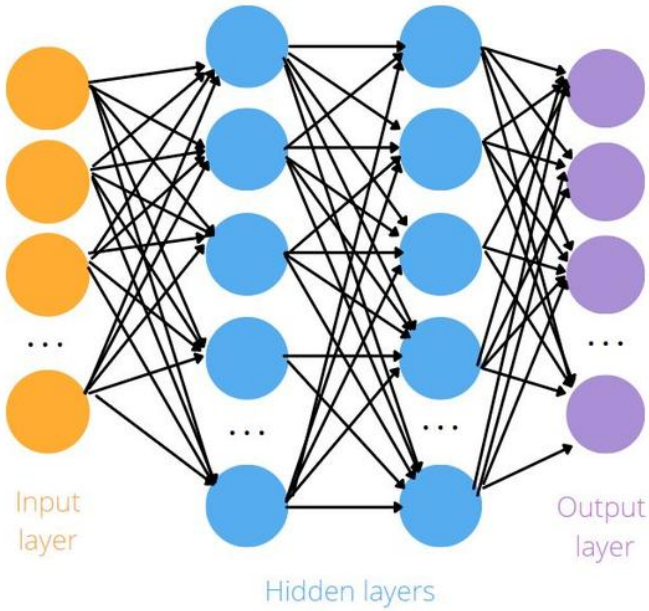

FIGURE 5 – Architecture SNN

V.3 Convolutional Neural Network

Les réseaux de neurones convolutionnels (CNNs) sont attractifs pour la tâche de détection de mots-clés, c'est pourquoi nous avons implémenté une architecture convolutionnelle avec deux couches convolutionnelles. La raison pour laquelle nous avons choisi d'appliquer uniquement deux couches convolutionnelles au lieu d'utiliser des CNNs très profonds et volumineux à la pointe de la technologie est de limiter le nombre de paramètres au détriment de la précision. Notre modèle CNN est agencé en cascade comme illustré dans la Figure 6. Voici une description détaillée de ses composants principaux :

Couches Convolutives : Le modèle débute par une couche de convolution (`conv1`) avec 128 filtres de taille 7×7. Cette couche est suivie d'une normalisation par lots (`bn1`) pour stabiliser l'entraînement et d'une fonction d'activation ReLU pour introduire de la non-linéarité. Ensuite, une couche de max pooling (`maxpool`) est utilisée pour réduire la dimension spatiale de la sortie, favorisant ainsi l'extraction des caractéristiques les plus importantes.

Autres Couches Convolutives : Une deuxième couche de convolution (`conv2`) avec 64 filtres de taille 7×7 est ensuite appliquée, suivie de normalisations par lots (`bn2`) et de fonctions d'activation ReLU. Ces couches convolutives permettent au modèle d'apprendre des caractéristiques de plus haut niveau.

Couches Entièrement Connectées : Après les couches convolutives, la sortie est aplatie (`view`) et propagée à travers une couche entièrement connectée (`fc1`) avec 500 neurones, suivie d'une activation ReLU. Enfin, deux autres couches entièrement connectées (`fc2` et `fc3`) sont utilisées pour la classification finale, produisant les logits de sortie correspondant aux différentes classes de l'ensemble de données.

Couches de Dropout : Afin de prévenir le surapprentissage et améliorer la généralisation du modèle, des couches de dropout (`dropout1`, `dropout2`) sont intégrées après les opérations de max pooling et une autre couche (`dropout3`) est intégrée après la première couche entièrement connectée. Ces couches éteignent aléatoirement une fraction des neurones lors de l'entraînement, forçant ainsi le réseau à apprendre des représentations plus robustes et générales.

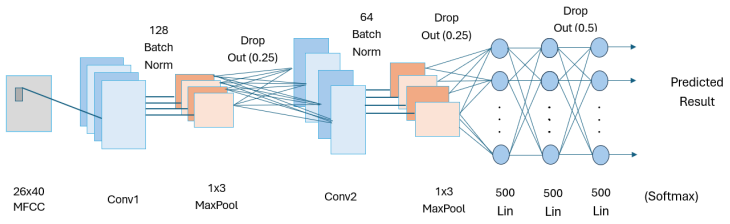

FIGURE 6 – Architecture du Modèle CNN

VI Protocole expérimental

Nous avons mené des expériences sur chaque modèle (régression logistique, SNN (Simple Neural Network) , CNN) en suivant un protocole rigoureux. Tout d'abord, nous avons défini les paramètres d'entraînement de nos modèles. Le choix de la fonction de perte a été fixé à la **cross-entropy**, adaptée à la nature de notre tâche de classification de mots-clés. De plus, nous avons utilisé un taux d'apprentissage de **0.001** pour régler la vitesse à laquelle notre modèle apprend à partir des données.

En ce qui concerne la taille du lot (batch size), nous avons fait un compromis entre la prévention du surapprentissage et les limitations matérielles. Nous avons choisi une taille de lot de **256** pour l'entraînement des modèles. Cette valeur nous permet de bénéficier des avantages de la mise à jour des poids du modèle avec un ensemble représentatif d'échantillons tout en évitant les inefficacités liées à un batch size trop petit. Le choix d'un batch

size différent de 1 nous a permis de prévenir le surapprentissage en introduisant une certaine régularisation dans le processus d'entraînement.

Nous avons ensuite spécifié le type d'augmentation des données à utiliser, avec des options telles que "random" pour une augmentation aléatoire et "duplicate" pour une duplication des données.
Les techniques d'augmentation comprenaient le "time shit", "time masking", "frequency masking" et "background noise". Nous avons ensuite entraîné chaque modèle sans l'utilisation des paramètres d'augmentation spécifiques, puis avec chaque technique d'augmentation individuellement activée, et enfin avec toutes les techniques d'augmentation combinées. Cette approche nous a permis d'analyser l'effet de chaque technique d'augmentation sur les performances des modèles et de déterminer l'impact global de l'augmentation des données sur les résultats finaux.

VII Résultats

Nous présentons ici les résultats de nos expériences sur les trois modèles principaux : régression logistique, SNN et CNN, en utilisant différentes techniques d'augmentation des données. Les performances sont évaluées en termes de l'accuracy sur l'ensemble de validation et de test pour chaque modèle.

VII.1 Régression Logistique

En raison de sa nature simpliste, le modèle de régression logistique a affiché des performances relativement modestes, attribuables à son incapacité à appréhender la complexité inhérente aux données.

Lr	**Acc Validation**	**Acc Test**	**Type**
No Augment	31.9%	30.3%	-
Time Shift	32%	30.2%	random
Time Masking	31.7%	30%	random
Frequency Mask	30%	29.6%	random
Background Noise	33%	30.3%	random
Combined Aug	31.9%	30%	random

Table 1 – Résultats d'expérimentation

VII.2 SNN

Le réseau neuronal simple a montré des performances supérieures à la régression logistique.

SNN	**Acc Validation**	**Acc Test**	**Type**
No Augment	73.7%	73%	-
Time Shift	76.9%	76.2%	random
Time Masking	75.6%	75%	random
Frequency Mask	74.6%	74.1%	random
Background Noise	75.9%	74.5%	random
Combined Aug	76.6%	75.4%	random

Table 2 – Résultats d'expérimentation

VII.3 CNN

Le CNN a produit les meilleures performances parmi les trois modèles.

CNN	**Acc Validation**	**Acc Test**	**Type**
No Augment	90.1%	90%	-
Time Shift	90.5%	90.8%	random
Frequency Mask	87.8	87	random
Background Noise	88.3%	87.2%	random
Time Masking	89.2	88.7	random
Combined Aug	90.4%	90%	random

Table 3 – Résultats d'expérimentation

Nous affichons ci-dessous le classification report pour le CNN, fournissant une analyse détaillée des performances du modèle.

Table 4: Résultats des métriques (CNN - Time Shift)

Classe	Précision	Rappel	F1-score	Support
bed	0.84	0.91	0.87	176
bird	0.92	0.87	0.89	158
cat	0.93	0.93	0.93	166
dog	0.94	0.86	0.90	180
down	0.88	0.83	0.86	253
eight	0.88	0.94	0.91	257
go	0.78	0.78	0.78	251
five	0.93	0.89	0.91	271
four	0.95	0.92	0.94	253
happy	0.98	0.96	0.97	180
house	0.97	0.93	0.95	150
left	0.90	0.94	0.92	267
marvin	0.90	0.89	0.89	162
nine	0.89	0.95	0.92	259
no	0.80	0.92	0.86	252
off	0.92	0.94	0.93	262
on	0.92	0.94	0.93	246
one	0.94	0.90	0.92	248
right	0.95	0.86	0.90	259
seven	0.95	0.97	0.96	239
sheila	0.96	0.95	0.95	186
six	0.97	0.93	0.95	244
stop	0.98	0.93	0.96	249
three	0.85	0.86	0.86	267
tree	0.91	0.84	0.87	193
two	0.84	0.89	0.86	264
up	0.81	0.93	0.86	272
wow	0.98	0.93	0.96	165
yes	0.98	0.95	0.96	256
zero	0.97	0.91	0.94	250
Précision globale			0.91	6835
Moyenne pondérée	0.91	0.91	0.91	6835
weighted avg	0.91	0.91	0.91	6835

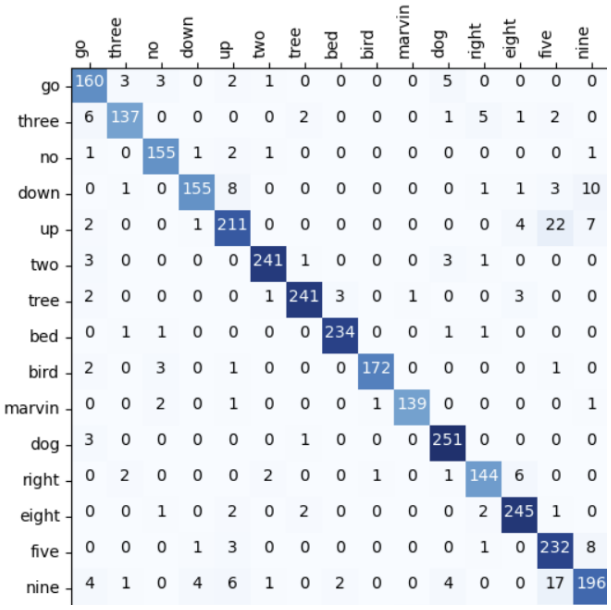

FIGURE 7 – Matrice de confusion

Il apparaît que le réseau éprouve des difficultés à classifier correctement certains mots qui présentent des sonorités similaires, tels que "go", "up", "no", entre autres. La figure 7 illustre la matrice de confusion pour les 15 étiquettes les plus fréquemment mal classées, révélant des confusions notables entre des mots comme "go" et "no", ainsi que "nine" et "five".

Remarque : Il convient de noter que le type d'augmentation utilisé pour les trois modèles est l'augmentation aléatoire ("random"), après des tests approfondis, il a été déterminé que l'augmentation de type aléatoire offre un compromis optimal entre rapidité et performance. Cette méthode, qui consiste à augmenter une donnée dans 40% des cas et à la laisser inchangée dans les 60% restants, a été privilégiée. En effet, elle permet d'introduire une diversité suffisante dans l'ensemble de données pour améliorer la généralisation du modèle, tout en limitant le temps de calcul. Par rapport à l'augmentation par duplication, qui multiplie toutes les données mais se révèle plus gourmande en temps de calcul, l'approche aléatoire semble être le meilleur compromis.

TABLE 5 – Récapitulatif des performances de chaque model

	CNN	**SNN**	**LR**
Accuracy	90,8%	74.5%	30,1%
Precision	91%	75.4	34,7%
Recall	90,8%	75.1	30,5%
F1-Score	91%	74.9	30,1%

Conclusion

Nous avons employé trois modèles pour la tâche de détection de mots-clés : la Régression Logistique, les Réseaux de Neurones Simples (SNN) et les Réseaux de Neurones Convolutifs (CNN). Le modèle CNN, avec une accuracy de 90%, se distingue sans nécessiter l'emploi de techniques d'augmentation des données telles que le décalage temporel, le masquage temporel, le masquage fréquentiel et l'ajout de bruit de fond. Les accuracies de la Régression Logistique et du SNN sont respectivement de 30.3% et 76%, ce qui met en exergue l'efficacité du CNN.
De manière plus spécifique, nous avons constaté que seul le "time shift" induit une légère amélioration de performance, d'environ ± 0.2, par rapport aux modèles sans augmentation. Cela remet en question la valeur ajoutée de l'augmentation des données sur le dataset SPEECHCOMMAND et suggère qu'une approche plus simplifiée pourrait être tout aussi efficace.

Références

[1] Oleg Rybakov, Natasha Kononenko, Niranjan Subrahmanya, Mirkó Visontai, and Stella Laurenzo. Streaming keyword spotting on mobile devices. In *Interspeech 2020*, interspeech_2020. ISCA, October 2020.

[2] Yuan Gong, Yu-An Chung, and James Glass. Ast : Audio spectrogram transformer, 2021.

[3] Somshubra Majumdar and Boris Ginsburg. Matchboxnet : 1d time-channel separable convolutional neural network architecture for speech commands recognition. In *Interspeech 2020*, interspeech_2020. ISCA, October 2020.

[4] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. Hello edge : Keyword spotting on microcontrollers, 2018.

[5] Pete Warden. Speech commands : A dataset for limited-vocabulary speech recognition, 2018.

[6] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment : A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, interspeech_2019. ISCA, September 2019.