

INTELLIGENCE ARTIFICIELLE

REFORCEMENT DE LA
COMMUNICATION : UN OUTIL
D'IA POUR LES PERSONNES
SOURDES

Réalisé par : GUENDOUL Younes - REBBAG Anass - BELKADI Hamza

Encadré par : AOUATIF AMINE

Année Universitaire : 2023 / 2024

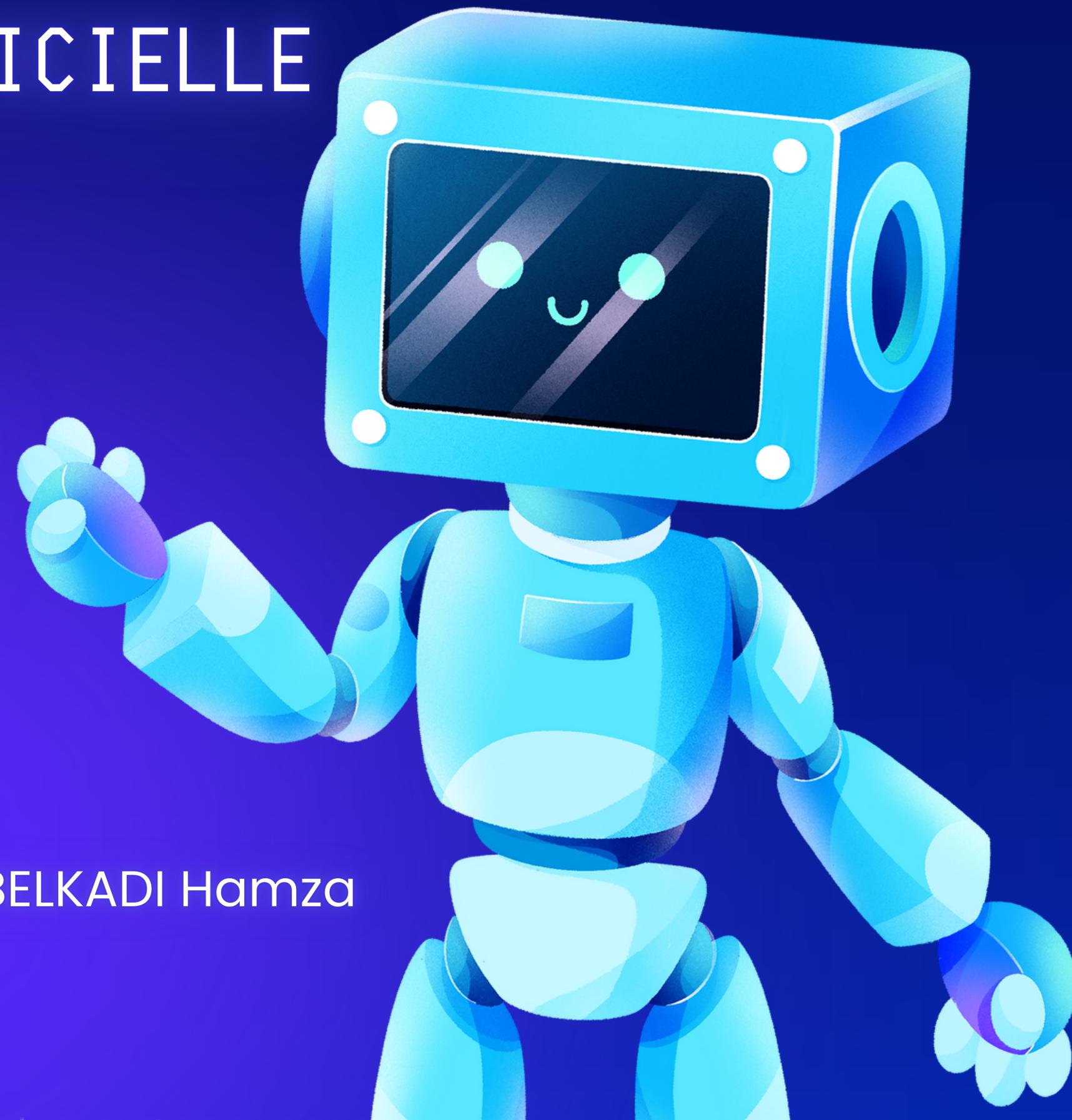




TABLE OF CONTENTS

• Introduction	01
• Outils de développement	02
• Data Preprocessing	03
• Data Pipeline	04
• Réseaux de Neurones	05
• Intégration Web	06
• Démonstration	07

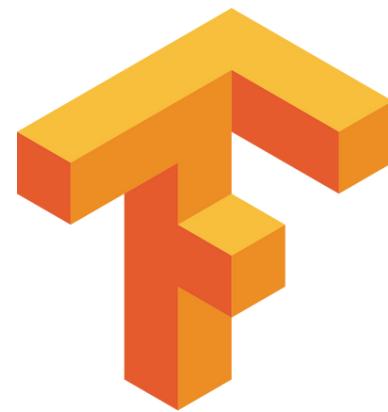


INTRODUCTION

Lip reading technology peut être utilisée pour améliorer l'accessibilité des personnes malentendantes. Elle peut contribuer à la traduction en temps réel du langage parlé en texte, permettant ainsi aux personnes sourdes ou malentendantes de comprendre les conversations et de participer plus pleinement à diverses activités. Elle pourrait être aussi utilisée dans des applications de sécurité et de surveillance, où elle pourrait aider à comprendre les paroles dans des situations où l'information audio n'est pas claire ou n'est pas disponible.



Outils de Développement

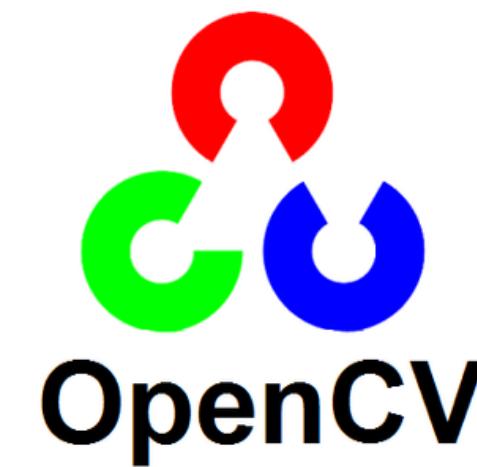


matplotlib



Streamlit

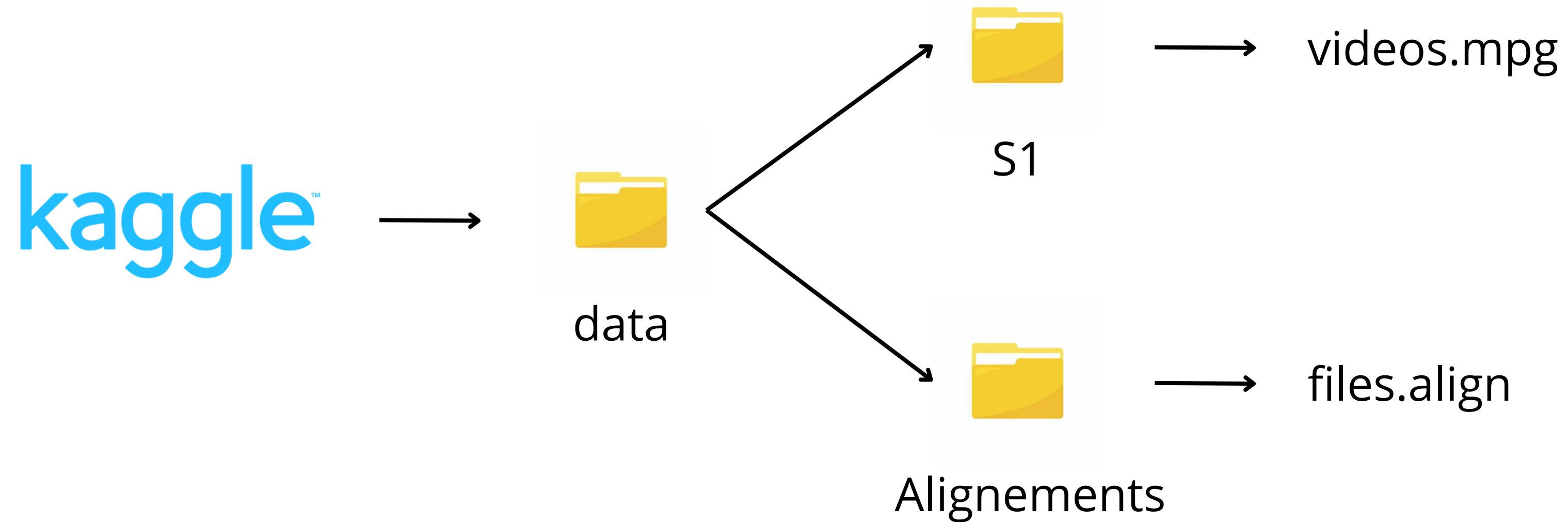
gdown



imageio



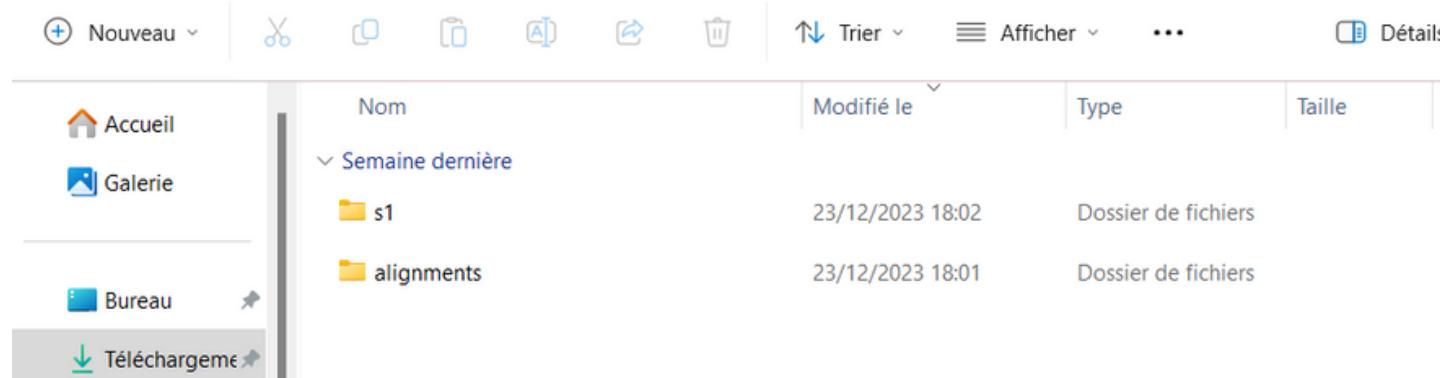
Dataset



Aperçu sur Dataset

S1

data



Nouveau			
Trier			
Afficher			
Nom	Modifié le	Type	Taille
Il y a longtemps			
Thumbs	31/10/2006 10:00	Data Base File	658 Ko
swwv9a	30/10/2006 18:15	Fichier MPG	436 Ko
swwv8p	30/10/2006 18:15	Fichier MPG	404 Ko
swwv7s	30/10/2006 18:15	Fichier MPG	396 Ko
swwv6n	30/10/2006 18:15	Fichier MPG	402 Ko
swwp5a	30/10/2006 18:14	Fichier MPG	422 Ko
swwp4p	30/10/2006 18:14	Fichier MPG	420 Ko

Alignements

Nouveau			
Trier			
Afficher			
Nom	Modifié le	Type	Taille
Il y a longtemps			
bbaf2n.align	03/03/2005 11:41	Fichier ALIGN	1 Ko
bbaf3s.align	03/03/2005 11:41	Fichier ALIGN	1 Ko
bbaf4p.align	03/03/2005 11:41	Fichier ALIGN	1 Ko
bbaf5a.align	03/03/2005 11:41	Fichier ALIGN	1 Ko
bbaf6n.align	03/03/2005 11:41	Fichier ALIGN	1 Ko
bbaf7s.align	03/03/2005 11:41	Fichier ALIGN	1 Ko

examp : bbaf2n.align

bbaf2n.align		
Fichier	Modifier	Affichage
0 23750 sil 23750 29500 bin 29500 34000 blue 34000 35500 at 35500 41000 f 41000 47250 two 47250 53000 now 53000 74500 sil		

DATA PREPROCESSING

On va se servir de 2 fonctions pour télécharger les données

01

- load_video : cette fonction prend en entrée un chemin vers un fichier d'alignements, charge une vidéo, convertit chaque image en niveaux de gris, sélectionne une région d'intérêt spécifique à savoir la bouche et normalise les valeurs des pixels sur toutes les images en utilisant des opérations TensorFlow.

02

- load_alignments : cette fonction prend en entrée un chemin vers un fichier d'alignements, lit le fichier, extrait certains tokens (en excluant ceux égaux à 'sil' "silence"), les traite avec TensorFlow pour les convertir en nombres, puis retourne le résultat.

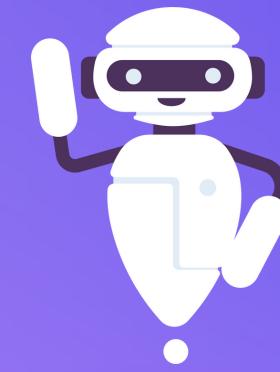
Création de Pipeline de Données

La création de pipeline de données va nous servir à déterminer les paramètres de notre réseau de neurone, à savoir la taille d'entrée (input_size) ainsi de diviser le dataset en partie train et test.

On a 1000 vidéos triées par ordre alphabétique.
Il est nécessaire de mélanger l'ordre pour éviter l'Overfitting.

On va prendre 70% Train et 30% Test.
Ce qui fait 700 vidéos pour entraînement et 300 vidéos pour tester.

RÉSEAUX DE NEURONES



LAYERS

input layer
5x hidden layers
output layer



ARCHITECTURE

CNN + LSTM



FNC ACTIVATION

ReLU & Softmax

Réseaux de Neurones

```
model = Sequential()
model.add(Conv3D(128, 3, input_shape=(75,46,140,1), padding='same'))
model.add(Activation('relu'))
model.add(MaxPool3D((1,2,2)))

model.add(Conv3D(256, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPool3D((1,2,2)))

model.add(Conv3D(75, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPool3D((1,2,2)))

model.add(TimeDistributed(Flatten()))

model.add(Bidirectional(LSTM(128, kernel_initializer='Orthogonal', return_sequences=True)))
model.add(Dropout(.5))

model.add(Bidirectional(LSTM(128, kernel_initializer='Orthogonal', return_sequences=True)))
model.add(Dropout(.5))

model.add(Dense(char_to_num.vocabulary_size()+1, kernel_initializer='he_normal', activation='softmax'))
```

Réseaux de Neurones

- **Conv3D(128, 3, input_shape=(75, 46, 140, 1), padding='same')** : Cette couche Conv3D est une couche de convolution tridimensionnelle avec 128 filtres, une taille de noyau de 3x3x3, une taille d'entrée de (75, 46, 140, 1) et un remplissage ('same' pour conserver la taille de l'entrée). Elle est suivie d'une activation ReLU.
- **MaxPool3D((1, 2, 2))** : Cette couche de max pooling tridimensionnelle avec une fenêtre de pooling de taille (1, 2, 2) effectue une réduction de la dimensionnalité en prenant la valeur maximale dans chaque sous-région.
- **TimeDistributed(Flatten())** : Cette couche applique une opération de "Flatten" (aplatir) à chaque étape temporelle des données en entrée. Elle est souvent utilisée pour traiter des séquences de données tridimensionnelles avant de les passer à des couches récurrentes.
- **Bidirectional(LSTM(128, kernel_initializer='Orthogonal', return_sequences=True))** : Une couche bidirectionnelle LSTM avec 128 unités dans chaque direction, une initialisation orthogonale des poids et la sortie de séquence activée (return_sequences=True). Les LSTMs bidirectionnels permettent au modèle de capturer des motifs dans les deux sens temporels.
- **Dropout(0.5)** : Une couche de dropout qui désactive aléatoirement 50% des neurones pendant l'entraînement. Cela aide à prévenir le surajustement (**overfitting**) du modèle en introduisant une régularisation.
- Les deux couches LSTM bidirectionnelles avec dropout sont répétées pour augmenter la capacité du modèle et améliorer sa capacité à capturer des motifs complexes dans les séquences.
- **Dense(char_to_num.vocabulary_size()+1, kernel_initializer='he_normal', activation='softmax')** : Une couche dense avec une sortie égale à la taille du vocabulaire plus un . Elle utilise une initialisation he_normal des poids et une activation softmax pour générer des probabilités de classe pour chaque classe possible dans le vocabulaire.

PARAMETRES DE NN



OPTIMIZER

ADAM



LOSS

CTC



EPOCHS

200

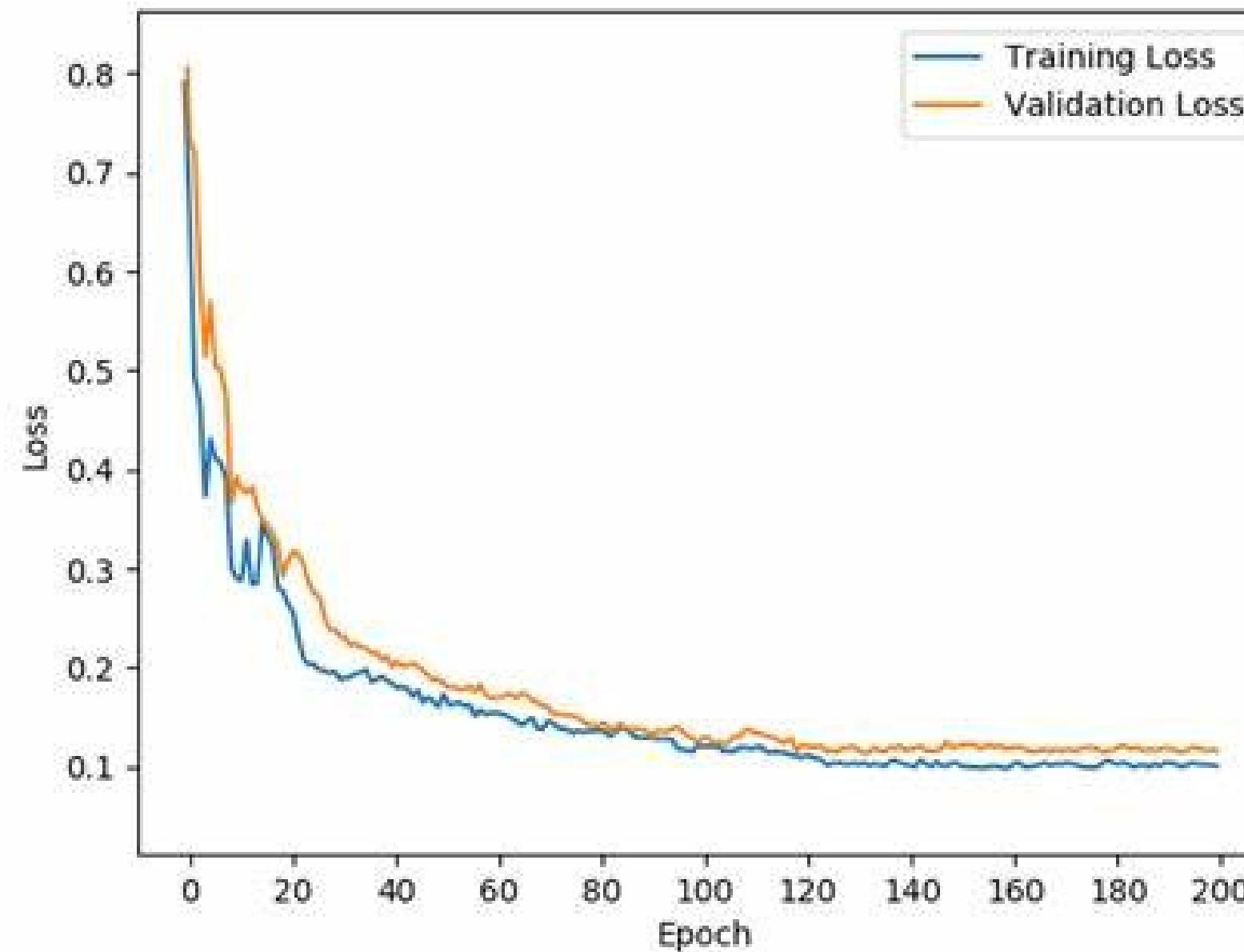


LEARNING RATE

$Lr = 0.0001$

Connectionist
Temporal Classification

Results



Accuracy: 94%

283 / 300 correct in the testing Dataset

Intégration Web

localhost:8501



>

≡

Renforcement de la communication : un outil d'IA pour les personnes sourdes

Choose video

bbaf2n.mpg

▼

This is all the machine learning model sees when making a prediction



This is the output of the machine learning model as tokens

```
[[ 2  9 14 39  2 12 21  5 39  1 20 39  6 39 20 23 15 39 14 15 23  0  0  0  
  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1 -1  
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  
-1 -1 -1]]
```

Decode the raw tokens into words

bin blue at f two now

THANK YOU!

TIME FOR DEMO

