Objectif

Cette expérimentation a le but d'évaluer la faisabilité de l'ajoute de l'idée de correspondance entre la vidéo et l'audio dans *liveness-control*, on voudrait refaire l'expérimentation de «*3D Convolutional Neural Networks for Cross Audio-Video Matching Recognition*» et essayer de prouver qu'elle pourra donner une bonne précision pendant la détection de correspondance de la vidéo et l'audio.

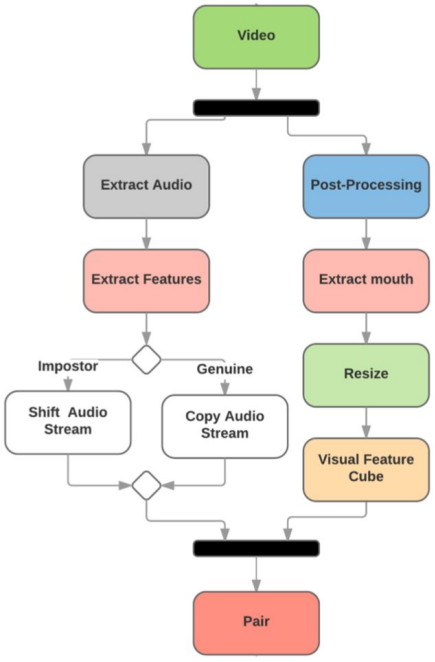
Expérimentations

Jeu de donnée

Cette expérimentation a utilisé la base *Lip Reading in the Wild*, c'est une base contenant 500 mots différents et pour chaque mot, elle offre environ 1 000 vidéos. Selon la présentation de base, *Lip Reading in the Wild* contient au total 500k vidéos et la taille de la base est environ 70 Go. De plus, le mot clé de chaque vidéo se situe toujours au milieu de la vidéo (environ 0.4s - 0.7s), la longueur de ces vidéos sont toujours 1.16s.

Les prétraitements de données

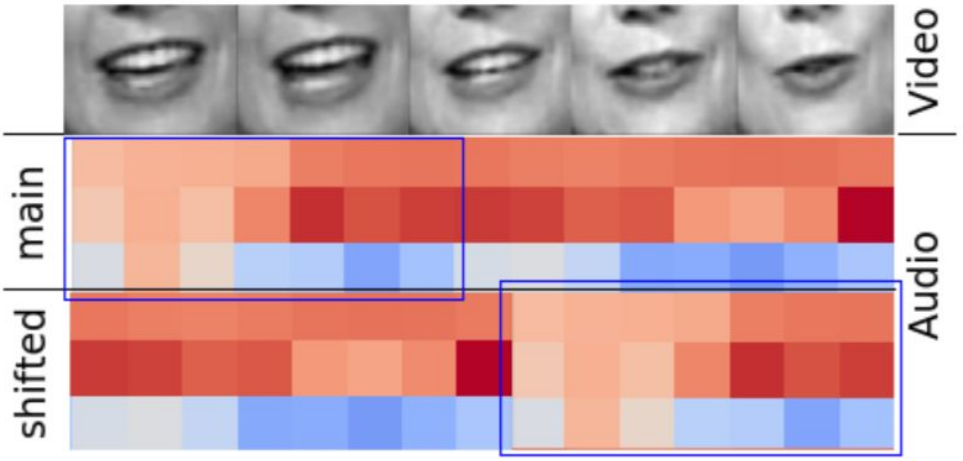
Selon l'article de M. Torfi, le processus de pré-traitements de données peut être séparé en deux partie: l'extraction de caractéristiques d'audios et de vidéos, il peut être représenté par la figure ci-dessous:



Plus précisément, pour générer les paires de vidéos et audios, notre expérimentation contient principalement les 8 étapes ci-dessous:

1) Convertir toutes les vidéos en 30 fps. Selon notre étude, toutes les vidéos dans la base *Lip Reading in the Wild* est 25fps, ici, pour avoir le même environnement que l'auteur, on doit les convertir en 30 fps.

2) Retarder 0.3s en l'axe d'audio pour chaque vidéo qui a le numéro pair. C'est une étape nécessaire pour générer les exemples négatifs, le mécanisme de ce traitement est comme la figure ci-dessous:



Comme on peut le voir, l'audio retardée ne correspond pas aux images de bouches et ce type de paire peut être utilisé comme l'exemple négatif pendant l'entraînement.

3) Segmenter toutes les vidéos en 4 parties, on utilise seulement la partie 0.4s - 0.7s qui contient généralement le mot clé. Comme on écrit dans la présentation de jeu de données ci-dessus, cette partie a toujours la voix et c'est plus facile d'obtenir les caractéristiques d'audio et de vidéo en même temps.
4) Extraire l'audio des vidéos.
5) Extraire les caractéristiques MFEC d'audio. Pour une audio avec une durée de 0.3s, on peut toujours obtenir les caractéristiques MFEC de dimension 15*40*3.
6) Extraire les régions de bouche de chaque vidéo. C'est une étape importante pour nous, les caractéristiques de vidéos sont extraites de ces images.
7) Redimensionner les images de bouche en 60*100.
8) Extraire les caractéristiques de vidéo. Parce que on peut obtenir 9 images pour une vidéo de 0.3s, donc la dimension de caractéristiques de vidéo est toujours 9*60*100.

Paramétrage des hyper-paramètres

Concernant l'architecture de CNN 3D, afin d'approcher au maximum les résultats originaux, on a utilisé les même hyper-paramètres (par exemple: batch_size = 32 et epoch = 15) que l'article, mais à cause de la date limite, on n'a pas généré 280k paires pour l'apprentissage et 70k paires pour le test. Dans notre expérimentation, la taille de la base d'apprentissage est 20k et celle de test est 5k.

Résultat

Selon la taille de notre base d'apprentissage et les hyper-paramètres, on a effectué 625 fois d'entraînement pour chaque epoch. Les résultats finals de tests sont comme la deuxième ligne dans le tableau ci-dessous:

Training Set	Test Set	Nombre d'epoch	EER	AP	AUC

20k	5k	15	0.23	0.83	0.84
280k	70k	15	0.135	0.965	0.954

Comparant aux expérimentations précédentes, la taille de la base d'apprentissage maintenant est 4 fois supérieure à celle de la dernière, et la précision de test augmente environ 10%.

Discussion

Cette expérimentation a vérifié que le fonctionnement de la partie de pré-traitement de données et la partie d'entraînement, mais parce que nous n'avons pas utilisé les GPUs, l''entraînement a pris beaucoup de temps (environ 30h). De plus, la taille de la base d'apprentissage pour le moment est juste 1/14 de la base utilisée par l'auteur, donc il reste encore environ 10% d'écart entre notre expérimentation et celle de M. Torfi (Comme la troisième ligne dans le tableau ci-dessus).

Conclusion

Selon les résultats obtenus, on a pensé que cette expérimentation a atteint les résultats prévus. On peut donc continuer la partie de génération de paires d'audio et vidéo et on voudrait générer au moins la même taille de la base d'apprentissage que l'auteur et puis commence à l'intégrer dans notre système existant.