3D Convolutional Neural Networks for LipSync avec les vidéos de 15 FPS

Kaixing Zhao, Zuheng Ming
17.07.2018

Objectif

L'objectif de cette expérimentation est de vérifier si l'algorithme proposé par [1] peut être appliqué en les vidéos de 15FPS.

Expérimentations

Dataset

Cette expérimentation a aussi utilisé la base *Lip Reading in the Wild (LRW)*, c'est un dataset qui contient 500 mots différents en anglais, et environ 1 000 vidéos pour chaque mot. Selon la présentation de base, LRW contient au total vers 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 est toujours 1.16s.

Les prétraitements de données

Selon l'article, le prétraitement de données peut être divisé en deux parties : l'extraction de caractéristiques de l'audio et de la vidéo, et la génération de paires de caractéristiques. Ce dernier est donc l'entrée des réseaux 3D CNN qui sert à entrainer et évaluer. Le protocole de prétraitement est présenté par la figure 1 ci-dessous [1] :

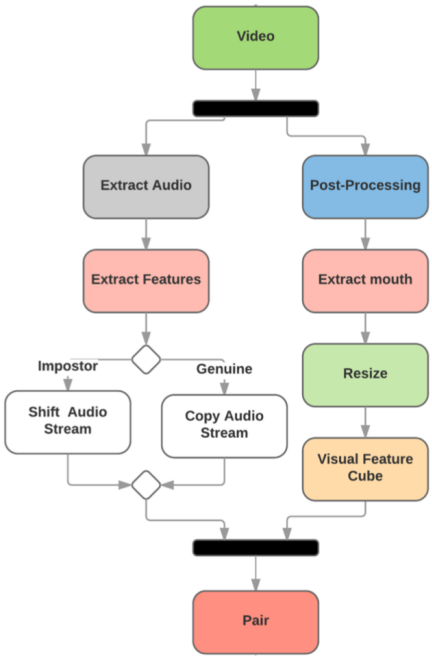

Fig. 1. Le protocole de prétraitement de données

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 15 fps. Selon notre étude, toutes les vidéos dans la base *LRW* sont 25fps, ici, pour tester si l'algorithme proposé par l'auteur peut être appliqué en les autres fréquences de vidéos, on a décidé de convertir toutes les vidéos en 15fps et ensuite répéter tous le processus suivant.
2) Retarder 0.3s en l'axe d'audio pour chaque vidéo qui possède le numéro pair. Ça c'est pour générer les paires négatives. Par contre, les vidéos sans le décalage sur l'axe temporel de l'audio sont les paires positives. Le traitement est montré dans la figure 2 ci-dessous :

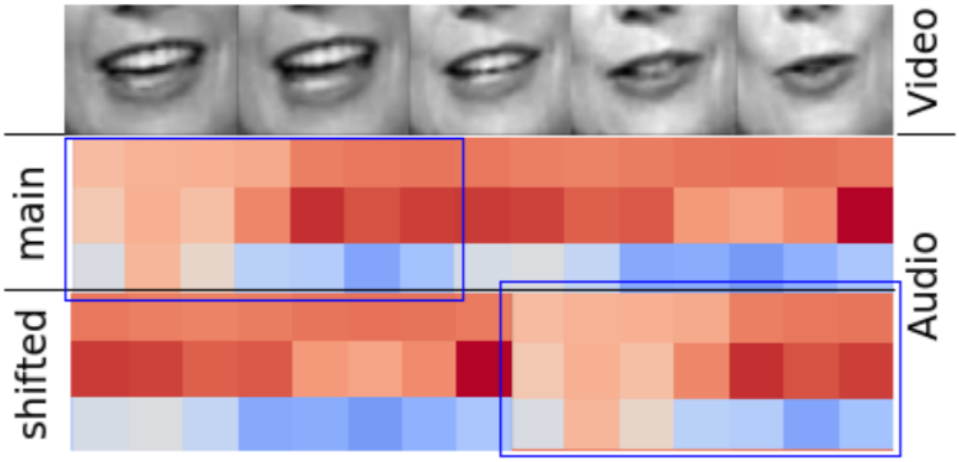

Fig. 2. Pour générer la paire négative, l'audio est retardée 0.3s qui ne correspond plus aux images de bouches.

3) Segmenter toutes les vidéos en 4 parties, on utilise seulement la partie 0.4s - 0.7s qui contient généralement le mot clé. 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 ([40 MFEC coefficients, 40 dérivées de premier ordre de MFEC, 40 dérivées de second ordre] par 20ms, 0.3s/20ms=15).
6) Extraire les régions de bouche de chaque vidéo.
7) Redimensionner les images de bouche en 60*100.
8) Extraire les caractéristiques de vidéo. Parce que on peut obtenir 5 images pour une vidéo de 0.3s (comparant à la dernière expérimentation, pour les vidéos de 15fps, on peut seulement détecter 5 images de bouche), donc la dimension de caractéristiques de vidéo est toujours 5*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, epoch = 15) que l'article, mais comme la dernière expérimentation, en raison du temps limité, 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, mais on garde le même ratio de taille des données d'apprentissage et de test, il est toujours 5/1. Dans notre cas, chaque epoch comprend 625 itérations.

Résultat

Avant de montrer les résultats, ici on voudrait déclarer que à part l'expérimentation actuelle (utilisant les vidéos de 15fps), on a aussi refait la dernière expérimentation (utilisant les vidéos de 30fps) pour que notre comparaison puisse être plus juste, et les résultats de test sont comme ci-dessous :

Training Set	Test Set	Nombre d'epoch	EER	AP	AUC
0.9k	0.1k	15	0.63	0.44	0.42
4.5k	0.5k	15	0.41	0.68	0.69
20k(30 fps)	5k(30 fps)	15	0.23	0.83	0.84
20k(15 fps)	5k(15 fps)	15	0.21	0.85	0.86

Le résultat dans l'article :

280k	70k	15	0.135	0.965	0.954

Table 1. Comparant aux expérimentations précédente, les 3 indicateurs ont amélioré légèrement

Conclusion

Selon les résultats de tests, on peut bien vérifier que le fonctionnement de l'algorithme [1] sur les vidéos de 15fps. De plus, comparant à l'expérimentation utilisant les vidéos de 30fps, on peut obtenir un meilleur résultat.

Références :

[1] Torfi, A., Iranmanesh, S. M., Nasrabadi, N., & Dawson, J. (2017). 3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition. *IEEE Access*, *5*, 22081-22091.

3D Convolutional Neural Networks for LipSync avec les vidéos personnelles

Kaixing Zhao, Zuheng Ming
18.07.2018

Objectif

L'objectif de cette expérimentation est de vérifier la généralisation de l'algorithme proposé par [1], l'idée est d'utiliser le modèle généré par l'expérimentation précédente (utilisant les vidéos de 15 fps) pour faire des tests sur des vidéos que nous avons trouvées sur Internet.

Expérimentations

Dataset

Les clips de cette expérimentation sont venu d'une vidéo qui s'appelle « Interview of first lady of President Obama » sur Youtube. Pour conformer aux conditions indiquées dans le papier, on a segmenté la vidéo originale en 50 petite vidéos et chacun a une durée de 1 second.

Les prétraitements de données

Les prétraitements de données sont comme l'expérimentation précédente, ici, on ne voudrait pas les répéter, mais à la fin, les données de test forment 3 tableaux :

1) Tableau d'images : 50*5*60*100*1
2) Tableau d'audio : 50*15*40*1*3
3) Tableau de labels : 50*1

Ici, on doit faire attention que pour toutes les vidéos de 15fps, la taille de caractéristiques est 5*60*100*1, si on voudrait faire des tests sur des vidéos de 15fps, le modèle doit aussi entraîné par des vidéos de 15fps.

Paramétrage des hyper-paramètres

La valeur batch_size de cette expérimentation a été mis en 5 car le nombre total de vidéos de test est trop petit (seulement 50 vidéos), et selon l'auteur, on a aussi utilisé Cross-Validation et le nombre d'itération égale à 10.

Résultat

Pour bien vérifier la généralisation de l'algorithme, on a fait deux types de test. Les données de premier test utilisent aussi les données d'entraînement et celles de deuxième test utilisent les 50 vidéos que nous avons téléchargés.

Training Set	Test Set	Nombre d'epoch	EER	AP	AUC

20k(15 fps)	20k(15 fps)		0.207	0.852	0.871
20k(15 fps)	50(15 fps)		0.4	0.508	0.483

Table 1. Résultats sur deux base de test

Conclusion

Selon les résultats de tests, on peut savoir que l'algorithme marche bien sur la base d'entraînement, il peut atteindre une précision similaire que l'entraînement mais sur les autres vidéos, la précision n'est pas très bonne. Selon notre expérience, peut-être la précision va augmenter si on augmente la taille de la base de test et la taille de batch_size.

Références :

[1] Torfi, A., Iranmanesh, S. M., Nasrabadi, N., & Dawson, J. (2017). 3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition. *IEEE Access*, *5*, 22081-22091.