

Analyse de scenes audio/video a travers le traitement du langage naturel et le traitement d'images

November 9, 2020

Contents

1	<u>Introduction</u>	1
2	<u>Description technique</u>	2
2.1	Fonctionnalites du prototype	2
2.2	Langages de programmation et outils	2
3	<u>Rendu</u>	3
3.1	Prototype (code)	3
3.2	Rapport	3
3.3	Prototype & demonstration	3
4	<u>References</u>	4
4.1	Ouvrages	4
4.2	Tutoriels en ligne	4

1 Introduction

L'objectif de ce projet est d'explorer les techniques d'inference artificielle relatives au traitement du langage naturel et au traitement de signal (audio et video).

Il faudra documenter et implementer un prototype qui permet d'evaluer certains marqueurs (nature des emotions exprimees, nature des mouvements observes, ...) presents dans une scene audio/video donnee et les attribues aux acteurs presents dans la scene.

2 Description technique

2.1 Fonctionnalites du prototype

Le prototype devra effectuer:

- La conversion d'une conversation audio en texte horodate
- L'analyse et la classification du ton de la conversation: agressif, amical, ...
- La detection des acteurs humains et objets dans un flux video
- La classification des actions/movements de chacun des acteurs detectes
- La correlation des actions/mouvements classifies pour chaque acteur detecte avec la conversation en utilisant l'horodatage
- L'evaluation de l'agressivite/passivite des acteurs de la scene analysee

Idees bonus:

- Rajouter la detection faciale lorsqu'un humain est detecte
- Prediction d'accidents: collision entre objets et humains, objet et objet, ...

2.2 Langages de programmation et outils

Il est recommande d'utiliser le langage **Python** afin de faciliter l'implementation du prototype. Il existe plusieurs librairies **Python** (TensorFlow, PyTorch, OpenCV, numpy, ...) qui permettent de deployer tres rapidement des re-seaux de neurones (a entrainer ou pre-entraines).

Pour la detection/reconnaissance d'objets, par exemple, le modele **YOLO** (You-Only-Look-Once) est disponible a travers la librairie **OpenCV**.

Pour la conversion audio vers texte, il existe plusieurs librairies comme **SpeechRecognition** qui utilise plusieurs API publiques: Google Cloud Speech, IBM Speech to Text, CMU Sphinx, ...

3 Rendu

3.1 **Prototype (code)**

Afin d'éviter les échanges de fichiers par courrier électronique et pour vous familiariser avec les outils de gestion de versions de codes, il faudra créer et travailler sur un dépôt **Git** public (i.e. **Github**). L'objectif étant aussi de faciliter les tâches de développement, de documentation et de suivi.

3.2 **Rapport**

Il faudra fournir un rapport détaillant les éléments théoriques essentiels ainsi que les principaux points relatifs à l'implémentation des modèles présentés.

Proposition de **squelette** pour le rapport:

- Introduction: l'objectif et le contexte du projet
- Historique et état de l'art: bref historique de l'évolution de l'IA et les nouvelles avancées: maths, software, hardware.
- Modèle: description du/des modèle(s) utilisé(s), des algorithmes et leur complexité, ...
- Implémentation: description des outils disponibles (bibliothèques, frameworks, réseaux de neurones pré-entraînés, ...) et des outils développés.
- Résultats: description de l'environnement d'expérimentation (OS, CPU, GPU, ...). Présentation des résultats obtenus à plusieurs étapes.
- Performances: brève analyse des performances du prototype et propositions d'optimisations (structures de données, parallélisation, ...).
- Conclusion
- References: bibliographie, webographie, vidéographie, ...
- Annexes: réserve aux longs listings de codes (de préférence références) et aux éléments complémentaires.

3.3 **Prototype & démonstration**

- Il faudra fournir un prototype fonctionnel avec les instructions et les données nécessaires pour la reproduction et la vérification des résultats présentés.

- Il faudra aussi fournir une video de 5 minutes maximum demontrant les proprietes du prototype et son fonctionnement.

4 References

4.1 Ouvrages

- Mathematics for Machine Learning, Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong

4.2 Tutoriels en ligne

- <https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech->
- <https://towardsdatascience.com/object-detection-with-less-than-10-lines-of-code-u>
- <https://www.c-sharpcorner.com/article/real-time-emotion-detection-using-python/>
- <https://www.thepythoncode.com/article/building-a-speech-emotion-recognizer-using->
- <https://www.skyfilabs.com/project-ideas/speech-emotion-recognition-using-python>