Projet Fil Rouge VitCow: Analyse de l'article N. Ravi et al., "SAM 2: Segment Anything in Images and Videos"

Charlotte TIBI Damien Glaizal Mafalda FRERE Marie MEIER
Marius CHARLES
Sous la direction de
Joseph Allyndree

Agroparistech
Institut national des sciences et industries du vivant et de l'environnement

Contents

Introduction

Contexte

- Existence d'un foundation models pour la segmentation promptable d'image (SA, Kirillov et al, 2023) mais pas pour la vidéo.
- De plus en plus des contenus multi-médias inclut une dimension temporelle (càd ce sont des vidéos).
- La segmentation de vidéo est essentielle dans de nombreux domaines: VR, Robotique, Véhicules autonomes...
- La vidéo entraine des problématiques inexistantes pour la segmentation d'image: une qualité d'image souvent plus faible, des objets qui subissent de grands changements (changement de forme, occlusion, changement d'éclairage...), les données sont beaucoup plus volumineuses...

NB: Il y a eu des essais pour adapter SAM (un modèle conçu à l'origine pour de la segmentation d'image) à la segmentation de vidéos en essayant de faire du tracking de prompt mais cela ne marche pas bien.

SAM 2

Meta propose un modèle capable de segmenter des objets très divers sur des vidéos. L'idée est de créer un foundation model appliqué à la tâche de classification promptable de vidéo.

SAM 2

C'est un modèle de segmentation promptable de vidéos.
Particularités:

- En entrée reçoit une frame composée de plusieurs images pas juste une image.
- Reçoit des prompts de natures diverses (clics positifs ou négatifs, masque complet ou bounding box).
- Architecture de transformers avec une attention temporelle.
- Peut recevoir plusieurs prompts (correction par un nouveau prompt si la segmentation n'est pas bonne sur une frame).

SAM 2, un modèle mais pas que...

En plus de son modèle, meta propose un dataset (SA-V dataset) issu du travail d'un data engine fonctionnant avec une boucle interactive.

SA-V dataset

Pas limité à une classe ou un type d'objet.
Les objets ne sont pas forcément entièrement représentés dans l'image, ils peuvent sortir du cadre puis éventuellement y revenir.
Dataset mixte: vidéos mais aussi images (ici une image = une vidéo avec une unique frame).

Quelques chiffres sur SA-V

642,6 K masklets (masques spatio-temporels)
36,5 M mask
196 h de vidéos

Architecture

Apparté sur l'attention

L'attention intuitivement

Le modèle apprend à se concentrer en priorité sur les parties importantes d'une entrée pour produire la sortie. Les éléments qui ont une importance importante reçoivent un poids plus important que les autres.
Avec la phrase: "Le chat que le chien poursuivait a sauté".
Pour apprendre qui a sauté le modèle apprend qu'il doit faire attention à "chat". Le modèle regarde tous les mots et les mots importants se voient attribuer un poids important. Ici le mot "chat" obtient un poids fort par rapport au mot "chien".

Apparté sur l'attention

Mathématiquement mais simplement

Pour chaque élément de l'input, le modèle calcule les matrices/ vecteurs suivants:

- Query (Q): ce que je cherche
- Key (K): ce que je suis
- Values (V): l'information que je porte

La similarité entre Q et K est calculée et des poids sont attribués aux différents éléments. Un softmax est appliqué. La moyenne pondérée des values est ensuite calculée afin de changer la représentation. En somme on multiplie chaque patch de l'input par un poids qui informe sur l'importance de chacun des autres patches.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

Les différents blocs

Le modèle SAM 2 est construit comme une succession de transformers suivis d'un MLP. Les différents "étages" de transformers permettent de prendre en compte différents niveaux de détails.

Pour chaque "étage" il y a d'abord un calcul d'une "self-attention" puis d'une "cross-attention" vis-à-vis des représentations en mémoire. Les différents éléments de

l'architecture que nous allons détailler sont les suivants:

- l'encodeur des prompts
- le décodeur des masques
- l'encodeur de mémoire
- la banque de mémoire

L'encodeur des prompts

Les prompts peuvent être de deux types et seront encodés différemment.

- Sparse prompts (clics ou box): encodés sous la forme des positions du prompt dans l'image et d'un embedding propre à chaque type.
- Masques: encodés par une convolution

Les embeddings des différents types de prompt sont ensuite sommés.

Le décodeur de masque

Prend en entrée les embeddings issus de l'encodage des prompts, de la frame actuelle, et de la mémoire pour générer des masques de segmentation.

En cas de prompts ambigus, plusieurs masques sont conçus. Si aucune image de la frame considérée ne résout l'ambiguïté alors c'est le masque qui coïncide le plus (du point de vue du IoU) qui est conservé.

Puisque l'objet que le modèle cherche à segmenter peut être absent de l'image, le modèle calcule aussi une probabilité de présence de l'objet d'intérêt au sein de l'image.

L'encodeur de mémoire et la banque associée

Le masque prédit précédemment est réduit par un downsampling par convolution puis sommé avec un embedding de la frame.

Des convolutions sont ensuite effectuées pour réduire la dimensionalité et créer une mémoire légère.

La banque de mémoire:

- Queue FIFO des N dernières frames non promptées
- Queue FIFO des M dernières frames promptées
- Des vecteurs d'objet: garde des informations sémantiques sur l'objet segmenté.

Entrainement et data engine

Le data engine

Le dataset a été construit en 3 phases d'interaction entre les annotateurs et un modèle les aidant à la tâche. Les différents niveaux de la collecte de données ont été faits pour différents niveaux d'aide aux annotateurs. Les phases sont:

- Phase 1: utilisation de SAM et annotation de chaque image
- Phase 2: SAM pour créer le masque initial et utilisation de SAM 2 pour propager le masque. En cas d'erreur l'annotateur corrige en fournissant un nouveau masque généré par SAM.
- Phase 3: utilisation unique de SAM 2 avec correction par l'annotateur en utilisant tous les types de prompt.

NB: les utilisateurs ont tendance à annoter les objets centraux et de grande taille. Pour varier les objets, des exemples sont créés automatiquement en fournissant une grille de points à SAM 2. Les segmentations ainsi obtenues sont ensuite corrigées par les annotateurs humains.

Les métriques utilisées

L'évaluation des performances de tels modèles nécessite des métriques adaptées aux tâches de segmentation.

Le score J&F

Cette métrique fait la moyenne de deux mesures spécifiques à la segmentation (du moins adapté au cas de la segmentation):

$$J = \frac{|M_{\text{pred}} \cap M_{\text{gt}}|}{|M_{\text{pred}} \cup M_{\text{gt}}|}$$

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$J\&F = \frac{J + F}{2}$$

Principaux résultats

Résultats

Sur toutes les tâches évaluées – segmentation promptable de vidéo (en online ou offline), segmentation semi-supervisée (fournit uniquement un prompt sur la première image) et en segmentation d'image – SAM 2 est meilleur que les stats of arts (XMem++ et Cutie combiné à SAM pour les prompts).

Il faut retenir que le modèle est non seulement plus performant mais aussi plus rapide.

Que retenir ?

Points clés

Les métriques à utiliser en segmentation d'image sont spécifiques et bonnes à connaître.

Si on veut utiliser SAM 2 doit-on lui fournir un seul prompt par frame ou plusieurs ? Il

existe différents encodeurs (Hiera S/B/L) qui sont adaptés à différents formats d'images et tailles d'image. Ils ont aussi des performances différentes d'un point de vue de l'accuracy mais aussi de la vitesse d'exécution.