Universidad San Francisco de Quito
Maestría en Inteligencia Artificial

Visión Artificial 202430 Profesor: Ramiro Sandoval

Actividad en Clase #2

Objetivo

El objetivo de esta actividad en clase es entrenar un modelo de segmentación semántica multiclase a partir de imágenes extraídas del videojuego SuperTuxKart. Para el efecto, se puede utilizar como base el código generado en clase y/o cualquier adaptación, modificación o red entrenada desde cero o pre-entrenada que cumpla con este fin.

SuperTuxKart

Es un videojuego de carreras arcade 3D open-source.

Dataset

El dataset fue generado a partir de una interfaz de Python (pystk) que interactúa con el motor gráfico del videojuego. Las máscaras generadas a partir de la interface son una fiel representación del objeto en el juego.

La carpeta del dataset está organizado por "***tracks***", dentro de cada ***"track"*** tenemos el ***"frame"*** original, carpetas por cada una de las posibles clases y carpetas combinadas.

Dentro de las carpetas de las posibles clases se tiene una máscara binaria del objeto de interés.

Las carpetas ***combined*** contienen imágenes cuyos pixeles tienen valores desde 0 hasta N-1. Siendo N el número de clases. La carpeta ***combined_visual*** tiene la imagen $combined * 32$ para poder visualizar la combinación de las diferentes clases.

Cada máscara es una imagen de 1 canal (grises) donde el valor de cada pixel representa la clase semántica:

Clase	**Valor**
Background	0
Track	1
Kart	2
Pickup	3
Nitro	4
Bomb	5
Projectile	6

** Las carpetas combined se ha generado como una ayuda/guía para el estudiante y pueden o no usarse. combined_visual no debería usarse en el modelo, es solo una herramienta de visualización

Los frames se relacionan con las clases por su ***track*** e identificador númerico. Es decir:

track_01/frame/frame_**0128**.png tiene relación con:

track_01/bomb/mask_**0128**.png y **track_01**/kart/mask_**0128**.png

También tome en cuenta que, no hay presencia de todos los objetos en todos los *frames*. Es decir, hay *frames* donde solo hay *karts*, *background* y el *track*. Esto podría hacer que el modelo sobre aprenda estas clases. Se podría corregir con *class_weights* en el criterio de calificación del modelo.

Modelo

Cualquiera que sea capaz de realizar segmentación semántica (preferiblemente que utilice la librería torch). Por ejemplo, U-NET o FCN.

Use como métrica IoU.

Entregables

1. Train: Loop de entrenamiento
2. Eval: Una manera de ver el resultado del entrenamiento. Debe recibir alguna imagen, pasarla por el modelo entrenado y mostrar la segmentación.
3. Utils: Todo código que haya permitido la carga, manipulación, exploración del dataset, etc.
4. Modelo entrenado serializado en .th

Tips

La función de pérdida debe ser *CrossEntropyLoss*.

Reducir el tamaño de la imagen aumenta la velocidad de entrenamiento.

Agradecimientos

UT Austin. Este trabajo es una adaptación de los talleres diseñados por Philipp Krähenbühl