Ingeniería Eléctrica
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

EL5206-1 Laboratorio de Inteligencia Computacional y Robótica
Unidad: Procesamiento de Imágenes

Actividad de Laboratorio 3

Profesor: Carlos Navarro C.
Auxiliar: Jorge Zambrano I
Ayudantes: Vanessa González A. y Manuel Zamorano C.

Parte 1: Detección de movimiento con diferencia de frames y modelo de fondo

El objetivo de esta Actividad de Laboratorio es implementar y analizar algoritmos simples de detección de movimiento y tracking de objetos. Debe descargar el archivo Lab-Mov.rar que contiene las secuencias de video para trabajar sobre ellas. Para esta actividad deberá convertir las imágenes de color de entrada en imágenes en escala de grises

1. Implemente un detector de movimiento por diferencia de cuadros y aplíquelo sobre la secuencia de imágenes (carpetas “mora_mov”, “sephanoides_mov” y “vigilancia_mov”). Muestre ejemplos de detección de movimiento y explique la forma que adoptan los pixeles de primer plano.

2. Utilizando la secuencia de imágenes de la carpeta “sephanoides_fondo” y “vigilancia_fondo”, genere un modelo de fondo de dos matrices calculando el promedio de los cuadros (matriz 1) y la desviación estándar (matriz 2). Elija un umbral apropiado y aplíquelo para detectar movimiento sobre las secuencias de imágenes respectivas (“sephanoides_mov” y “vigilancia_mov”). Entregue las matrices del modelo de fondo. Muestre ejemplos de detección de movimiento y explique la forma que adoptan los pixeles de primer plano.

3. Usando la detección de movimiento anterior, calcule el histograma proyectado por columnas (suma de pixeles por columna) y el histograma proyectado por filas (suma de pixeles por fila) Con esta información, encierre en una caja de dimensiones adecuadas el blob de la detección de movimiento.

4. Utilizando la información de la posición de la caja en el cuadro actual y la de los cuadros anteriores, estime la posición del objeto para el cuadro siguiente. Dibuje la caja estimada y compárela con la obtenida usando el procedimiento descrito en 3. Comente.

5. Utilice el detector por diferencia de cuadros implementado y aplíquelo sobre la secuencia de imágenes "estacionamiento". Muestre ejemplos de detección de movimiento y explique la forma que adoptan los pixeles de primer plano. Obtenga los bounding boxes de cada blob. Identifique los principales problemas de utilizar este método.

Parte 2: Tracking de personas basado en Deep Learning.

Para poder realizar esta parte es necesario utilizar el entorno virtual de Google Colaboratory.

Prueba con CNN para Detectar de objetos

6. Utilice la red neuronal convolucional entrenada con la base de datos MS-COCO en las imágenes de la carpeta examples y visualice los resultados. Para esto siga los pasos del notebook Colab_CNN.ipynb.

Detector + Tracking

7. Utilice el método de tracking de personas basado en Deep Learning, sobre las imágenes de las carpetas "estacionamiento", "mora_desestabilizado" y "mora_mov". Para esto, genere videos en formato MP4 de con una tasa de 25 fps y siga los pasos del notebook DeepLearning_Tracking.ipynb.

8. Puesto que el método de tracking es de personas, si usted genera un video con "sephanoides_mov" no encontrará detecciones. Modifique la línea 91 de yolov3_deepsort.py, cambiando su valor de 0 (clase persona) a 14 (clase pájaro) y genere un video como el de la parte anterior. Haga un cambio similar para cambiar la clase buscada a 32 (clase pelota deportiva) y modifique los valores del archivo deep_sort.yaml. de manera de generar un video a partir de "mora_mov" cuyo tracking no sean las personas sino el balón de futbol.

9. Obtenga y almacene las detecciones de cada frame y haga una comparación con las obtenidas en el punto 5.

Se trabajará en grupos de 2 alumnos(as) y se entregará 1 informe por grupo. El informe debe incluir los códigos creados.

Fecha de entrega informe: Jueves 07 de septiembre de 2023, 23:59 hrs. por u-cursos.

Anexos

BUG reportado:
Las nuevas versiones de la librería pyyaml necesitan otro argumento llamado Loader. Tras hacer el clon de la parte 7, en la fila 23 del archivo /content/deep_sort_pytorch/utils/parser.py se debe reemplazar:

```
        yaml.load(fo.read()),
por     yaml.load(fo.read(), Loader=yaml.FullLoader)
```

Con eso se incluye el argumento Loader.

Acá está la explicación de algo similar.

Acerca de las imágenes: los frames de 'vigilancia' fueron capturados por una cámara estática ubicada en el tercer piso del DIE. Las imágenes de 'estacionamiento' son parte de la base LASIESTA (Labeled and Annotated Sequences for Integral Evaluation of SegmenTation Algorithms), desarrollada por la Universidad Politécnica de Madrid. Por último, las imágenes 'mora' y ' sephanoides' se obtuvieron de una cámara no estática, y se estabilizaron utilizando un plugin de ImageJ.