Departamento de Ingeniería Eléctrica
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile
EL7021 - Seminario de Robótica y
Sistemas Autónomos

Profesor: Javier Ruiz-del-Solar
Auxiliares: Isao Parra
Kenzo Lobos
Mauricio Correa
Paul Vallejos

Tarea 2: Deep Drive

En esta actividad se estudiarán aplicaciones de visión computacional para problemas de distinta naturaleza dentro del contexto de autos autónomos, utilizando para ello técnicas modernas provenientes del aprendizaje de máquinas. En particular, se estudiarán sistemas del estado del arte relacionados al reconocimiento de objetos, segmentación semántica y navegacion visual, utilizando para ello la base de datos *Deep Drive (Berkley)* y *Deep Drive (Unreal)*.

Para la realización de esta actividad se dispone de un computador con GPU en las dependencias del *AMTC*. Se dispondrá de cuentas individuales, y de un sistema de uso compartido, para evitar problemas de administración de recursos. Sin embargo, se recomienda desarrollar lo más posible las actividades en sus computadores personales, debido su extensión.

DeepDrive (Berkley)

BDD100K[1] es una base de datos dedicada a aplicaciones de autos autónomos realizada por *UC Berkley*. Esta contiene detallada información tanto de múltiples sensores, como etiquetas de objetos y de segmentación semántica, y se diferencia de bases de datos anteriores principalmente debido a su cantidad de imagenes (100.000), y su diversidad de escenarios.

Esta base de datos puede descargarse directamente desde la página oficial[1] (requiere registro), o se puede obtener directamente desde el cuerpo docente.

Para la realización de las actividades que utilizan esta base de datos, solo se utilizará el conjunto de validación de la versión 10k de la base de datos (1000 imagenes). Se debe tener en consideración que las etiquetas de reconocimiento de objetos están en formato *json*, por lo que se debe utilizar un *parser* o bien exportar dichas etiquetas a otro formato, y que las etiquetas disponibles en *BDD100K* no coinciden con las utilizadas por otros sistemas, por lo que se debe realizar un mapeo entre las etiquetas que sean compatibles e ignorar las etiquetas que no correspondan.

DeepDrive (Unreal)

DeepDrive 2.0[2] es un simulador implementado en *Unreal Engine* con el objetivo de facilitar el desarrollo e investigación de temas relacionados a autos autónomos. Sus principales características son ambientes realistas, cámaras tanto RGB como de profundidad, y un ambiente de pruebas con una interfaz del tipo *OpenAI* para poder desarrollar agentes de navegación autónoma.

Si bien las instrucciones de instalación del desarrollador[2] recomiendan el uso de GPU, es posible realizar la experiencia en computadores que no posean GPU, lo cual sin embargo aumenta considerablemente el tiempo de los experimentos involucrados.

1. Reconocimiento de objetos

En esta actividad se estudiará el rendimiento del sistema de reconocimiento de objetos *YOLO*[3, 4, 5] en la base de datos *BDD100K*. En particular particular, se validará el rendimiento del sistema en términos de *mAP* e *IOU* utilizando la implementación original del autor utilizando para ello la librería *Darknet*[6].

(a) Descargue y compile la librería *Darknet*[3]. Modifique el *Makefile* según corresponda a su computador y a las librerías disponibles.

[1] http://bdd-data.berkeley.edu/
[2] https://deepdrive.io/
[3] https://pjreddie.com/darknet/

(b) Descargue el modelo pre-entrenado de *YOLO v3* y realice el tutorial correspondiente. Además de utilizar las imagenes de prueba disponibles, pruebe imagenes de la base de datos *BDD100K*.

(c) Adapte las etiquetas de *BDD100K* al formato utilizado por *Darknet*. El modelo disponible de *YOLO* fue entrenado con la base de datos *COCO*, por lo que detecta las 80 clases de dicha base de datos.

(d) Evalue el rendimiento de *YOLO v3* sobre la versión *10K* de la base de datos *BDD100K*. Considere los criterios *mAP* para los *IOU* 0.25, 0.5 y 0.75, el *mIOU* predicho y el real. Recuerde solo considerar las clases de *COCO* que también estén presentes en *BDD100K*.

(e) Realice el proceso anterior para la versión *Tiny-YOLO*, y comente en términos cualitativos, de criterios de rendimiento y su aplicabilidad en aplicaciones reales (e.g recursos computacionales).

2. Segmentación Semántica

En esta actividad se evalúa el rendimiento del sistema de segmentación semántica *ESPNet*[7]. El objetivo de esta actividad es entender los conceptos involucrados y conocer el rendimiento de sistemas del estado del arte.

(a) Descargue e instale la implementación de *ESPNet*[4] realizada por los autores.

(b) Adapte las etiquetas de *BDD100K* para poder compararlas con las de la base de datos *CityScape* (recuerde que las clases pueden diferir).

(c) Mida el rendimiento del sistema sobre la base de datos *BDD100K* en terminos de *mIOU* y rendimiento de clasificación. Obtenga la matriz de confusión

(d) Comente acerca del rendimiento del sistema tanto en términos cuantitaticos como cualitativos. Compare visualmente el rendimiento del sistema sobre el conjunto de validación de la base de datos *CityScape* y comente cuales son las limitaciones actuales de los sistemas de segmentación semántica, y cual es el *roadmap* a seguir en esta área.

3. Navegación Visual

Esta última actividad consiste en entrenar un agente de navegación autónoma mediante información visual utilizando para ello el simulador *DeepDrive*[2]. Mediante la realización de esta actividad se busca adquirir conocimientos prácticos acerca del diseño y entrenamiento de un agente, además de los alcances y limitaciones del aprendizaje supervisado para este tipo de problemas.

(a) Descargue e instale el simulador *DeepDrive*[5].

(b) Ejecute el agente de prueba *path follower*, el cual está basado en el seguimiento de *splines*. Evalúe cualitativamente su desempeño e identifique sus limitaciones.

(c) Guarde datos del agente de pruebas para distintos períodos de tiempo (e.g 30 minutos, 1 hora, 4 horas).

(d) Modele un agente de navegación visual (clasificador o regresor). Puede inspirarse en el modelo que propone el autor, o en otros papers relacionados.

(e) Entrene y evalue el rendimiento del agente tanto en términos de error como de recompensa. Cómo varía el rendimiento del agente con respecto a la cantidad de datos sobre los cuales se entrena? Comente con respecto al rendimiento de la política original.

(f) Implemente un sistema de *data aggregation*[8] para el entrenamiento del agente. Realice varias iteraciones tomando periodos de 1 hora.

[4] https://github.com/sacmehta/ESPNet
[5] https://github.com/deepdrive/deepdrive

(g) Comente acerca de los resultados utilizando *data aggregation*. Que estrategía propondría para entrenar agentes de navegación mediante aprendizaje supervisado?

Referencias

[1] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling," *CoRR*, vol. abs/1805.0, 2018. [Online]. Available: http://arxiv.org/abs/1805.04687

[2] C. Quiter and M. Ernst, "Deepdrive: 2.0," 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1248998

[3] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *CoRR*, vol. abs/1506.0, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *CoRR*, vol. abs/1612.0, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[5] ——, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.0, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[6] J. Redmon, "Darknet: Open Source Neural Networks in C," http://pjreddie.com/darknet/, 2013.

[7] S. Mehta, M. Rastegari, A. Caspi, L. G. Shapiro, and H. Hajishirzi, "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation," *CoRR*, vol. abs/1803.0, 2018. [Online]. Available: http://arxiv.org/abs/1803.06815

[8] S. Ross, G. J. Gordon, and J. A. Bagnell, "No-Regret Reductions for Imitation Learning and Structured Prediction," *CoRR*, vol. abs/1011.0, 2010. [Online]. Available: http://arxiv.org/abs/1011.0686