

Propuestas Proyectos EL7008 Primavera 2016

1. Tracking de robots para un sistema de ground truth aplicado a fútbol robótico

Tutor: Pablo Cano. **Estudiante:** Rodrigo Pérez.

Implementar algoritmos de detección y tracking de robots para un *ground truth* aplicado a fútbol robótico. La solución debe permitir detectar la pose de los robots (posición + orientación en 2D) en imágenes obtenidas mediante dos cámaras *wide angle* ubicadas sobre la cancha de fútbol robótico en el laboratorio de robótica.

2. Mejoramiento en la detección de robots en contexto de fútbol robótico basado en puntos de interés

Tutor: Pablo Cano. **Estudiante:** Constanza Villegas.

Implementar un sistema para eliminar falsos positivos en la detección de robots causados por la percepción de postes de los arcos, malla y brillos en la cancha causados por luz natural. Para ello se propone analizar las regiones generadas por el actual detector de robots utilizando por los robots NAO y aplicar técnicas de extracción automática de características para descartar los casos incorrectamente clasificados.

3. Evaluación de XNOR-NETs aplicadas a fútbol robótico

Tutor: Matías Mattamala. **Estudiante:** Kenzo Lobos.

Las redes *XNOR* (2016) corresponden a un tipo de red convolucional binaria que reemplaza las operaciones de punto flotante por operaciones lógicas. Esto permite no sólo acelerar el cómputo, sino también acercar las redes convolucionales a la CPU, haciéndolas ideales para su uso con robots de recursos limitados como el NAO. En este trabajo se pretende implementar y evaluar su uso en el contexto de fútbol robótico, utilizando la base de datos *SPQR Team NAO image dataset*.

4. Evaluación de SqueezeNets aplicadas a fútbol robótico (tutor: Matías Mattamala)

De modo análogo a las *XNOR-NETs*, otra alternativa utilizada para reducir la complejidad de cómputo de las redes convolucionales es la técnica *model compression*. Esta técnica se basa en la reducción de las conexiones, la simplificación de los cálculos y la compresión de los pesos de la red, disminuyendo los tiempos de cálculo y la memoria requerida para ejecutarla. En este trabajo se busca evaluar el uso de *SqueezeNet* (2016) en contexto de fútbol robótico con la base de datos *SPQR Team NAO image dataset*.

5. Implementación de aplicaciones de Deep Learning en un robot NAO utilizando procesamiento en hardware externo (tutor: Matías Mattamala)

* Proyecto para 2 personas

Implementar sistemas actuales de visión computacional basados en *deep learning* para resolver problemas como detección, reconocimiento, segmentación semántica, entre otras aplicaciones. Cada alumno deberá implementar y evaluar una solución a uno de estos problemas utilizando las imágenes generadas en un robot NAO, las cuales serán procesadas en un computador externo ODROID XU4 ubicado en una mochila que usará

el robot. El proyecto requiere que los alumnos diseñen la mochila y evalúen algoritmos en la ODROID.

6. Detección de Pelota con Método HOG-Adaboost (tutor: Pablo Cano)

Diseñar e implementar un método de detección de pelota para aplicaciones de fútbol robótico. El método implementará y mejorará la propuesta descrita en “Robust Game Perception and Action: Winners NimbRo TeenSize” (RoboCup 2016), basada en el uso de HOG-Adaboost.

7. Deep Faces (tutor: Luz Martínez)

Implementar un sistema reconocimiento de cara a distintas distancias utilizando Deep Neural Networks. Se debe realizar el entrenamiento de la persona de prueba a 1 metro aproximadamente y después el programa debe buscarlos hasta 4 metros. El sistema deberá ser evaluado con distintas personas a distintas distancias gracias a una base de datos entregada.

Código : <https://github.com/cmusatyalab/openface>

Lenguaje y herramienta : Python, Torch

Paper : “OpenFace: A general-purpose face recognition library with mobile applications”

*Base de datos disponible para evaluar

8. Automatización de segmentación de ropa en movimiento continuo (tutor: Luz Martínez)

Automatizar la segmentación de ropa mientras es manipulada desde uno o dos puntos. Además, se deberá automatizar el filtro necesario para no obtener imágenes similares. Las imágenes serán obtenidas por un sensor RGBD, que permitirá que puedan utilizar la imagen RGB y la imagen de profundidad para realizar el procesamiento. Las clases de ropa a trabajar son: camisa, polera, pantalón y toalla de mano.

Lenguaje y herramientas : Python o C++, OpenCV, Caffe

9. Deep Q-learning para juegos de Atari (requiere usar Ubuntu, GPU)

Tutor: Patricio Loncomilla. Estudiante Carlos Tampier.

Q-learning es una técnica que permite resolver problemas de aprendizaje reforzado, en los cuales los pares de características/etiquetas no están disponibles, por lo cual el sistema debe aprender realizando acciones en su entorno. En este proyecto se evaluará un sistema de Deep Reinforcement Learning que es capaz de aprender a jugar una familia de juegos de Atari a partir de las imágenes generadas por el juego y el puntaje alcanzado en el mismo, sin ningún tipo de preprocesamiento o intervención humana. La cantidad y tipo de juegos a probar será determinada con el tutor. El código se encuentra disponible en varias plataformas. Se recomienda el uso de Neon como base, ya que un proyecto contiene redes ya entrenadas. Si se usa otro sistema, hay que entrenar la red desde cero, lo que puede tomar varios días.

https://github.com/tambetm/simple_dqn

10. (v1) Reconocimiento de signos de manos RGB (Requiere usar Ubuntu, Caffe, idealmente GPU) (tutor: Patricio Loncomilla)

El objetivo de este trabajo es probar un reconocedor de signos de manos basado en redes convolucionales, usando imágenes RGB, presentado en CVPR 2016. El sistema es

capaz de reconocer signos directamente sobre la imagen, sin la necesidad de segmentar las manos. Se debe usar transfer learning para entrenarlo en una base de datos de signos distinta a la usada en el entrenamiento original, idealmente una base de datos con signos en español. Se debe reentrenar, o bien usar SVM como última capa si hay limitaciones de hardware. Requiere instalación de Caffe.

<http://www-i6.informatik.rwth-aachen.de/~koller/1miohands/>

11. (v2) Reconocimiento de gestos de manos Kinect (Requiere usar Ubuntu/Win 64 bits, GPU) (tutor: Patricio Loncomilla)

El objetivo de este trabajo es implementar un reconocedor de gestos de manos basado en redes deep o bien otro clasificador, usando como base un modelo articulado de manos detectadas por un método de tracking de manos. Requiere generar una base de datos de gestos de manos usando Kinect. Requiere instalación de CUDA.

<https://github.com/FORTH-ModelBasedTracker/HandTracker>

12. Tracking de personas multi-hipótesis (tutor: Patricio Loncomilla)

El objetivo de este trabajo es implementar un sistema de tracking multi-hipótesis de personas. Se debe integrar un tracker de objetos (idealmente en C/C++) con un detector de personas. Se propone adaptar el siguiente tracker:

<https://github.com/WeatherGod/MHT/tree/master/tracking>

El reconocedor de personas debe ser seleccionado por el alumno.

13. Reconocimiento de voz usando FFT + CNN (Opción 1 – Requiere Ubuntu / GPU / caffe) (tutor: Patricio Loncomilla)

El objetivo de este trabajo es implementar un sistema de reconocimiento de voz (dígitos), usando Caffe. Se debe calcular el espectrograma de una señal de voz, y usarlo como entrada a una red convolucional. Un proyecto de ejemplo (que se está migrando a Tensorflow) se encuentra en la siguiente página web:

Código: <https://github.com/pannous/caffe-speech-recognition>

Espectrogramas dígitos (451 MB 512x512): http://pannous.net/spoken_numbers.tar

OBSERVACION: Si no se usa la referencia al trabajo indicado, se podría hacer incluso en Matlab (con Matconvnet), como se hace en la tarea 5 que se dio.

14. Reconocimiento de voz usando DNN (Opción 2 – Requiere GPU / Tensorflow) (tutor: Patricio Loncomilla)

El objetivo de este proyecto es probar un reconocedor de voz (dígitos) que está en desarrollo, el cual está basado en redes deep (no necesariamente CNNs). Se debe probar en la misma base de datos de dígitos usada en el proyecto anterior.

Código: <https://github.com/pannous/tensorflow-speech-recognition>

Espectrogramas dígitos (451 MB 512x512): http://pannous.net/spoken_numbers.tar

15. Reconocimiento de signos de tráfico (tutor: Patricio Loncomilla)

El objetivo de este trabajo es implementar un sistema que permita reconocer signos de tráfico. El método que se usará en el reconocimiento es libre, y las opciones deben ser discutidas con el tutor. Dentro de las posibilidades está usar features de CNNs con SVM, o directamente CNNs. La base de datos que se utilizará es la siguiente:

<http://www.cvl.isy.liu.se/research/datasets/traffic-signs-dataset/>

16. Sistema de localización y clasificación de objetos usando Resnet más Faster R-CNN (tutor: Diego Campanini)

Las redes Deep Residual Networks [1] posibilitan entrenar CNN ultra profundas de una forma clara y simple, logrando el estado del arte en rendimiento en tareas tales como clasificación de imágenes, detección de objetos y segmentación semántica.

En este proyecto se busca implementar la arquitectura resnet más el sistema Faster R-CNN [2] que permite clasificar y localizar objetos en una imagen, esta implementación fue la ganadora de la competencia de detección de ImageNet 2015. Un ejemplo de su funcionamiento se puede encontrar en [3].

Las tareas para este proyecto son fundamentalmente entrenar con Caffe el sistema Resnet+Faster R-CNN inicializando los pesos del nuevo sistema a partir de las implementaciones disponibles de la red resnet [4].

[1] KaimingHe, XiangyuZhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016

[2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real Time Object Detection with Region Proposal Networks. NIPS 2015.

[3] <https://youtu.be/WZmSMkK9VuA>

[4] <https://github.com/KaimingHe/deep-residual-networks>

17. Clasificación de género usando redes neuronales convolucionales (tutor: Diego Campanini)

La clasificación automática de género ha concentrado gran atención particularmente desde el surgimiento de las redes sociales. Para este proyecto se plantea reconocer género a partir de imágenes de rostros de personas, usando la arquitectura CNN empleada en [1]. Las principales tareas a efectuar en este proyecto es desarrollar un código utilizando el framework Caffe para usar la arquitectura y pesos de la red convolucional disponibles en [2]. Además se debe caracterizar el rendimiento del sistema en la base de datos evaluada en [1].

[1] Gil Levi and Tal Hassner. Age and Gender Classification using Convolutional Neural Networks. IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015.

[2] http://www.openu.ac.il/home/hassner/projects/cnn_agegender/
<https://gist.github.com/GilLevi/c9e99062283c719c03de>

18. Clasificación de edad usando redes neuronales convolucionales (tutor: Diego Campanini)

Los rostros humanos constantemente transmiten información tales como emociones, género, edad, entre otras. En este proyecto se aborda una de las tareas más desafiantes en el campo de análisis facial y es la clasificación automática de la edad de personas a partir de imágenes de su rostro [1]. Las tareas a efectuar son generar un código en caffe que permita usar la implementación disponible en [2], tal que un computador pueda procesar una imagen y entregar el rango de edad en el que se encuentra una persona, además se debe obtener una matriz de confusión que muestre los resultados de la clasificación en la base de datos usada en [1].

[1] Gil Levi and Tal Hassner. Age and Gender Classification using Convolutional Neural Networks. IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015.

[2] http://www.openu.ac.il/home/hassner/projects/cnn_agegender/
<https://gist.github.com/GilLevi/c9e99062283c719c03de>