Clasificación y manipulación de basura doméstica utilizando deep-learning

Santiago Puente and Pablo Gil and Victor de Gea
AUROVA Lab, Computer Science Research Institute,
University of Alicante, Alicante 03690, SPAIN
{santiago.puente,pablo.gil,victor.degea}@ua.es
http://www.aurova.ua.es

Resumen

Este artículo presenta una aplicación de reconocimiento mediante el uso de redes de aprendizaje profundo para llevar a cabo la clasificación de basura en el ámbito doméstico. Así mismo, una vez realizado el reconocimiento se determina su localización, para poder obtener los puntos de agarre para que un brazo robot dotado de una pinza de dedos paralelos pueda hacerlo de manera automática. Se presenta el algoritmo utilizado, así como, los resultados experimentales que permiten comprobar la bondad de la propuesta.

Palabras clave: Deep Learning, Grasping, Perception for Grasping.

1 Introducción

El uso de sistemas robotizados y de inteligencia artificial, permite llevar a cabo tareas de clasificación de residuos, mediante el uso de aprendizaje profundo es posible que un sistema automático sea capaz de localizar en el entorno los residuos, así como, su naturaleza y mediante un sistema robótico autónomo proceder a su recogida.

La primera consideración necesaria es el reconocer y localizar los objetos en un escenario desconocido [11]. Lo que implica sistemas de percepción visual. Estos sistemas están compuestos por diferentes tecnologías de procesamiento y captura de imágenes: como cámaras estereoscópicas, RGBD, ToF, etc. La reciente explosión de técnicas de aprendizaje profundo facilita los procesos de segmentación y detección para encontrar la posición de los objetos en la imagen, incluso cuando se presenta una escena compleja y desestructurada. Los modelos de redes neuronales más comunes para estas tareas se basan en capas convolucionales, tal y como plantean [3] y [8].

Una vez que el objeto es reconocido y localizado en la imagen, se requiere calcular la ubicación del mismo en el mundo real para permitir su manipulación por el robot. En [10] y [6] se plantea el cálculo de zonas de agarre en objetos previamente reconocidos, reconstruyendo la malla de la superficie a partir de modelos CAD almacenados. Otros autores estiman los agarres usando aprendizaje profundo como en [7] que parte de vistas RGBD de la escena que contiene objetos, o desde una nube de puntos como en [9]. Ambos métodos no requieren que los objetos tengan que ser reconocidos y/o reconstruidos previamente.

En este trabajo, hemos creado un conjunto de muestras de objetos para acometer las tareas de clasificación y manipulación de objetos para su posterior reciclaje. Para ello se utiliza un modelo de red neuronal [4] al que se le aplica un reajuste hiperparamétrico para el conjunto de muestras. Obteniendo la localización en la imagen de los objetos a manipular, posteriormente, utilizando la información de profundidad proporcionada por el sensor RGBD se calcula la posición en el espacio 3D, para utilizando GeoGrasp [12] determinar los puntos de agarre para manipular el objeto.

Este artículo está estructurado de la siguiente manera. Primero, se describe la arquitectura del sistema. Posteriormente el conjunto de muestras generado, para seguir con la experimentación y terminar con las conclusiones.

2 Arquitectura del sistema

El objetivo es realizar una manipulación robótica de objetos considerados residuos o basura, tal y como se ha planteado en la introducción. Para lograrlo se utiliza una cámara RealSense d435i que proporciona datos RGBD de escenarios interiores y exteriores con los objetos a manipular. Las imágenes RGB tienen una resolución espacial de 640x480 píxeles para cada uno de los tres canales, se realiza un corte de las mismas a 512x512 para utilizarlas como entrada para la red Mask-RCNN que realiza la segmentación de instancias de objetos en la escena [2]. Aquí se utiliza la implementación planteada en [1] de Mask-RCNN [4] para detectar instancias de objetos diferentes al nuestro, esta red se re-entrena con el conjunto de muestras propio para ajustar los pesos de la misma. Este tipo de red permite distinguir qué píxeles pertenecen al objeto detectado y cuáles no. De esta forma, es posible realizar una detección

https://doi.org/10.17979/spudc.9788497498043.695

precisa de los objetos en imágenes 2D. La figura 1 muestra la arquitectura del modelo Mask-RCNN usado, que se describe en detalle en [5].

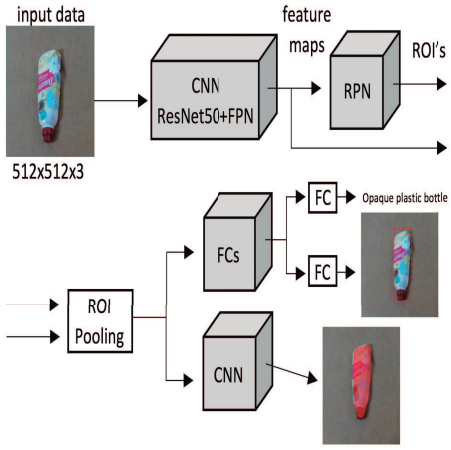

Figura 1: Modelo de la red Mask-RCNN

El enfoque realizado consiste en cargar los pesos previamente entrenados para realizar un ajuste fino mediante nuestro conjunto de muestras y ajustar el modelo al problema objetivo. El ajuste fino se realiza mediante un ajuste hiperparamétrico. Para realizar la evaluación de los hiperparámetros, se ha dividido el conjunto de datos en tres subconjuntos: un conjunto de entrenamiento, otro de validación y otro de test.

La tarea de manipulación robótica de objetos, basada en las predicciones del modelo que proporciona la red neuronal, se ha realizado teniendo en cuenta el uso de una pinza de dos dedos paralelos para llevar a cabo la tarea de manipulación. El sistema planteado se divide en los siguientes pasos (Figura 2):

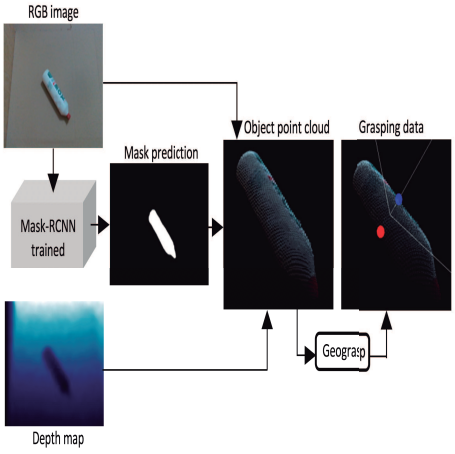

Figura 2: Arquitectura del sistema

- **Capturar datos RGBD**. Se captura la información RGBD de la escena, proporcionada por la cámara Realsense d435i. De esta forma, se obtienen imágenes RGB y mapas de profundidad de forma simultánea y sincronizada. Las predicciones del modelo se ejecutan sobre las imágenes en color capturadas, mientras que la información del canal D se utiliza para conocer los valores de profundidad en los que se ubican los posibles objetos detectados.

- **Leer predicciones de modelos entrenados**. Una vez que se han ejecutado las predicciones en imágenes RGB, se obtiene una máscara binaria para cada objeto detectado. Esta máscara permite distinguir qué píxeles, tanto en imagen RGB como en mapa de profundidad, pertenecen al objeto y cuáles pertenecen al fondo.

- **Reconstrucción 3D de objetos detectados**. Con los siguientes datos: imágenes RGB, D y de máscara binaria de la escena, se procesa la reconstrucción tridimensional de cada objeto detectado. Para ello, se aplican las expresiones 2, 3 y 1 a cada píxel que pertenece al objeto.

$$Z = D(u, v) \tag{1}$$

$$X = Z * \frac{(u - cx)}{fx} \tag{2}$$

$$Y = Z * \frac{(v - cy)}{fy} \tag{3}$$

donde (cx, cy) es el centro óptico, (fx, fy) es la distancia focal y (u, v) son las coordenadas de la imagen en píxeles.

Realizando este procesamiento se obtiene el conjunto de puntos (X, Y, Z) que componen la nube de puntos de cada objeto detectado. La máscara binaria obtenida por predicción de red permite conocer las coordenadas de los píxeles que pertenecen al objeto (u, v).

- **Calcular datos de agarre**. Una vez que se obtienen las representaciones tridimensionales de los objetos, se utiliza al algoritmo Geograsp para calcular los puntos de manipulación para el objeto. Geograsp permite calcular dos puntos de agarre en la superficie del objeto y la orientación de la pinza para lograr la manipulación. [13].

3 Conjunto de muestras

En este apartado se describe el conjunto de muestras generado para llevar a cabo las pruebas del sistema.

https://doi.org/10.17979/spudc.9788497498043.695

Para el conjunto de datos de entrenamiento se han tomado desde diferentes perspectivas las imágenes, es decir, los objetos aparecen en diferentes entornos y con diferentes posiciones, orientaciones, distancias y visibilidad. Se ha creado un conjunto de muestras con 1434 imágenes de 27 objetos diferentes clasificados en 5 clases 3, cada imagen se ha etiquetado manualmente de acuerdo con las siguientes clases: botella de plástico opaco (C1), caja de cartón (C2), botella de plástico transparente (C3), lata de bebida (C4) y recipiente de plástico opaco (C5). La tabla 1 muestra la distribución de muestras de cada categoría de objeto. Para realizar la experimentación, hemos creado un conjunto de datos con una representación equilibrada de clases, el número de muestras para cada clase es muy similar para el entrenamiento y el test, el conjunto de validación queda un poco escaso en muestras para la clase C2 respecto al número de instancias en el resto de clases. Este aspecto habría que mejorarlo en posteriores análisis para trabajar con un conjunto de objetos que tengan igual número de instancias para cada categoría..

Figura 3: Imagen de los objetos en las cinco categorías

Tabla 1: Distribución de instancias de objetos para cada categoría.

Categoría	Entre	Vali	Test
C1	157	42	117
C2	131	24	93
C3	145	36	109
C4	152	41	95
C5	163	35	94

4 Experimentación

En este apartado se describen los resultados experimentales obtenidos. Se han dividido en dos partes diferenciadas: la detección de objetos 4.1 y la estimación del agarre una vez detectados los objetos 4.2.

4.1 Detección de objetos

En esta sección, se ha utilizado el modelo Mask-RCNN para probar la detección de objetos una vez que fue entrenada con nuestro conjunto de muestras y se seleccionaron los hiperparámetros óptimos para el entrenamiento (tabla 2).

Tabla 2: Hyperparámetros seleccionados para el entrenamiento.

Batch size	1
Numero de clases	1+5
Tamaño imagen	512x512 pixels
Arquitectura neuronal base	resnet50
Ratio de aprendizaje	0.006
ROIs por imagen	200
Factor ROI positivo	0.23
Number of anchors	512
Anchor scales	(64,128,256 ,512,1024)
Anchor aspects	[0.5 1 2]
Umbral	0.9

De cara a evaluar el rendimiento del modelo propuesto, se ha utilizado un conjunto de datos de validación y prueba. El rendimiento en términos de MS COCO Average Precision (AP) se ha probado con diferentes valores para el índice Jaccard (IoU) (Tabla 3) donde AP50 se corresponde con un umbral de 0,5 y AP75 con un umbral de 0,75. Además, también se ha realizado para diversas escalas del objeto (Tabla 4), donde APm se corresponde con el AP para objetos pequeños, APm con el AP para objetos medianos y APl el AP para objetos grandes.

Tabla 3: Diferentes valores del umbral IoU.

Evaluation data	AP	AP50	AP75
Test data set	0.370	0.556	0.417
Validation subset	0.703	0.896	0.851

Tabla 4: Diferentes escalas de los objetos.

Evaluation data	APs	APm	APl
Test data set	0.059	0.374	0.511
Validation subset	0.289	0.720	0.766

El modelo entrenado tiene una mayor precisión en imágenes similares a las que se utilizan para entrenar el modelo, subconjunto de validación, mientras que al evaluar el modelo con imágenes menos

https://doi.org/10.17979/spudc.9788497498043.695

similares, conjunto de datos de prueba, el sistema devuelve un menor nivel de precisión.

Además, la precisión del modelo entrenado se evalúa en cada categoría de objeto en los datos de prueba (Figura 4).

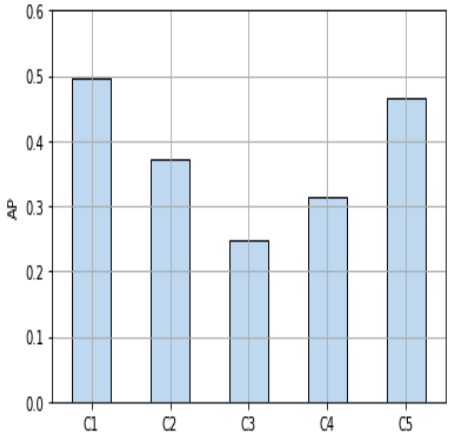

Figura 4: Resultados de AP por categoría

Según los resultados (Figura 4), el modelo entrenado tiene una precisión AP más alta en botellas de plástico opacas (C1) y recipientes de plástico opacos (C5). Sin embargo, los objetos con una menor precisión AP son las botellas de plástico transparente (C3). Para observar el comportamiento del modelo entrenado desde otro punto de vista, se visualizan algunos ejemplos de predicciones que el modelo calcula en imágenes del conjunto de datos de prueba. En algunas pruebas, se puede ver cómo las predicciones son incorrectas (Figura 5), porque las sombras proyectadas por los objetos pueden confundirse con píxeles pertenecientes al mismo, e incluso el reflejo del objeto en el terreno. Mientras que en otras pruebas, las predicciones de clase y máscara se ejecutan correctamente, donde se puede observar como en otros casos la detección no se ve afectada por sombras.

4.2 Estimación del agarre

Este apartado evalúa los agarres de las predicciones de objetos obtenidas según 4.1. Los experimentos se llevan a cabo a partir de la detección correcta de los objetos, con el objetivo de evaluar únicamente el procesamiento de los datos de agarre, ya que los objetos incorrectamente detectados no podría ser manipulados por el brazo robot.

La figura 6 muestra el resultado de las nubes de puntos para diferentes obejtos detectados. Analizando los resultados, el hecho de utilizar una sola cámara en el procesamiento limita la reconstrucción del área del objeto visible por el sensor. En la prueba (c), se observa que la nube de puntos no tiene suficiente información 3D sobre el objeto para calcular un buen agarre. Esta limitación depende del posicionamiento de los objetos en relación con la cámara y la forma de los objetos. Para objetos claros, los datos de profundidad registrados no son del todo correctos debido a la transparencia. Esto provoca que las reconstrucciones de este tipo de objetos no sean tan precisas como en los objetos opacos. Se obtienen buenos puntos de contacto en el objeto probado (f), pero estas limitaciones podrían afectar al cálculo de los agarres. Por otro lado, la prueba (e) muestra que los datos de profundidad de algunos píxeles pertenecientes a la lata de bebida no son correctos, lo que también podría afectar el cálculo del agarre. Las pruebas (a, b, d) muestran un cálculo de agarre correcto.

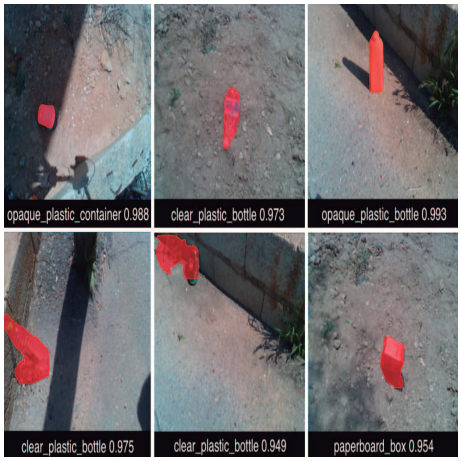

Figura 5: Predicciones correctas e incorrectas

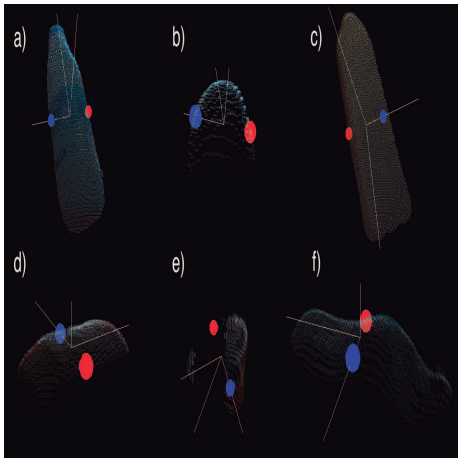

Figura 6: Datos de agarre

https://doi.org/10.17979/spudc.9788497498043.695

5 Conclusiones

Para concluir, el presente trabajo permite conocer una buena configuración paramétrica para los datos propios, con los que re-entrenar el modelo original. En general, de acuerdo con las pruebas realizadas, el procesamiento propuesto para calcular los datos de agarre a partir de las predicciones del modelo entrenado Mask-RCNN permite obtener agarres correctos. Un punto a destacar es que el sistema se puede extender a un gran número de clases para clasificar los objetos con criterios de reciclaje. Ahora el trabajo se está instalando en una plataforma móvil para probar en campo el algoritmo propuesto. Además, esta extensión a un entorno de campo nos permitirá mejorar el conjunto de datos y por tanto mejorar el método de clasificación.

Agradecimientos

Este trabajo ha sido financiado por la Comisión Europea y los fondos FEDER por medio del proyecto COMMANDIA (SOE2/P1/F0638), dentro del programa Interreg-V Sudoe. Para la fase de entrenamiento se ha usado la infraestructura computacional financiada por la Generalitat Valenciana y los fondos FEDER en el proyecto IDIFEDER/2020/003.

English summary

Classification and pick-up of domestic waste using deep-learning

Abstract

This paper presents an application of recognition through the use of deep learning networks to carry out domestic waste classification. Likewise, once the recognition is carried out, it is used to determine its location, in order to obtain gripping points so that a robot arm equipped with a two-finger parallel gripper performs it automatically. Initially, the algorithm used is explained, as well as the experimental results that allow to verify the goodness of the proposal.

Keywords: Deep Learning, Grasping, Perception for Grasping.

Referencias

[1] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[2] H. Azizpour, A. Razavian, and J. Sullivan. Factors of transferability for a generic convnet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38, 2016.

[3] D. Chaves, S. Saikia, L. Fernandez-Robles, E. Alegre, and M. Trujillo. A systematic review on object localisation methods in images. *Revista Iberoamericana de Automática e Informática industrial*, 15(3):231–242, 2018.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[5] J. Hui. Image segmentation with mask r-cnn. https://jonathan-hui.medium.com/image-segmentation-with-mask-r-cnn-ebe6d793272, 2018.

[6] S. Jain and B. Argall. Grasp detection for assistive robotic manipulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 2015–2021, 2016.

[7] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The Int. J. of Robotics Research*, 34(4-5):705–724, 2015.

[8] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *Int J Computer Vision*, (2):261–318, 2020.

[9] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt. Grasp pose detection in point clouds. *The Int. J. of Robotics Research*, 36(13-14):1455–1473, 2017.

[10] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour. Part-based grasp planning for familiar objects. In *16th Int. Conf. on Humanoid Robots*, pages 919–925, 2016.

[11] A. Wei and B. Chen. Robotic object recognition and grasping with a natural background. *Int. J. of Advanced Robotic Systems*, 17(2), 2020.

https://doi.org/10.17979/spudc.9788497498043.695

[12] B. Zapata-Impata, C. Mateo, P. Gil, and J. Pomares. Using geometry to detect grasping points on 3d unknown point cloud. In *14th Int. Conf. on Informatics in Control, Automation and Robotics*, volume 2, pages 154–161, 2017.

[13] B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres. Fast geometry-based computation of grasping points on three-dimensional point clouds. *Int. J. of Advanced Robotic Systems*, 16(1):1–18, 2019.

https://doi.org/10.17979/spudc.9788497498043.695