

# Procesamiento HPC en SmartTrashCan

Mariana Ojeda, Gabriel Flores, Micaela Ramirez, Luis Ferruffino, Brian Ponce

Universidad Nacional de La Matanza,  
Departamento de Ingeniería e Investigaciones Tecnológicas,  
Florencio Varela 1903 - San Justo, Argentina  
ojeda.e.mariana@gmail.com, gabriel.1991.flores@gmail.com, ramirezmibe@gmail.com,  
luis.a.ferruffino@gmail.com, poncec.briannicolas@gmail.com

**Resumen.** El objetivo de esta investigación es analizar los beneficios de incorporar HPC al proyecto SmartTrashCan. Dicha incorporación añadirá la posibilidad de identificar el tipo de residuos arrojados, creando una solución que automatice completamente la gestión de residuos y su clasificación. Para esto, utilizaremos algoritmos de reconocimiento de imágenes, utilizando la API Rekognition de Amazon.

**Palabras claves:** HPC, API, Rekognition, Amazon

## 1 Introducción

La siguiente investigación está aplicada al sistema SmartTrashCan, el cual es un contenedor de residuos inteligente. Dicho contenedor cuenta con varias funciones, tales como apertura de la tapa mediante detección de proximidad, peso del contenido e informar del nivel de temperatura y humedad en el interior. Estas últimas mediciones ayudan al usuario a identificar qué tipo de residuos contiene. Por ejemplo, si es un elemento de gran tamaño, poco peso y baja humedad, se podría deducir que el elemento es una botella de plástico y por lo tanto, es un elemento reciclable.

Debido a que dicho análisis del elemento no es 100% exacto y está basado en deducciones, nuestro objetivo es introducir un método de identificación de residuos mejorado. Lo propuesto es introducir al sistema una cámara de manera tal que, se identifique de forma gráfica el tipo de residuo, comparando la foto del residuo actual contra las almacenadas en una base de datos en la nube. Una vez identificado el tipo de residuo, podrá ser clasificado en “Orgánico”, “Vidrio”, “Plástico”, “Papel” o “Metal”.

Esto es similar a lo ya implementado en apps como Aipoly [1] y TapTapSee [2], las cuales son utilizadas para ayudar a personas con problemas de visión, identificando los objetos que se apuntan desde el celular. Estas aplicaciones utilizan Deep Learning y Fog Computing, ya que parte del procesamiento ocurre dentro del mismo celular, a diferencia de nuestra solución propuesta donde todo el procesamiento ocurre únicamente en la nube.

## 2 Desarrollo

Nuestra investigación tiene como origen la necesidad de no sólo proveer un artículo que ayude con la gestión de residuos a nivel personal, sino ayudar a colaborar con la clasificación de los mismos, para poder reciclarlos después. Esto lo vemos por los valores expuestos por Will Steffen, Johan Rockström, entre otros, en [3], donde establecen que la temperatura global continúa aumentando y sin una notable intervención que reduzca las emisiones, el calentamiento global entrará en un peligroso bucle de retroalimentación. En concreto, si el aumento global de la temperatura fuera mayor de 2 grados Celsius, el nivel del mar incrementaría hasta niveles nunca vistos desde el Holoceno. Asimismo, el termómetro alcanzaría los valores más altos de los últimos 1,2 millones de años.

Para lograr esto, necesitamos detectar en tiempo real qué tipo de residuo se está arrojando y la mejor forma, es utilizando HPC y el poder de procesamiento de GPU [4]. En este caso, utilizaríamos la solución brindada por AWS, llamada Amazon Rekognition. Esta solución utiliza algoritmos de Deep Learning, mediante el uso de clústeres de GPU y CPU para realizar operaciones de matrices complejas en tareas con uso intensivo. Mediante el entrenamiento de algoritmos con millones de imágenes etiquetadas, las redes neuronales de aprendizaje profundo (utilizan Convolutional Deep Neural Networks, también conocidas como CNNs) pueden identificar correctamente objetos o sujetos. Para esta implementación, únicamente utilizaremos el servicio para objetos. Esto se hará en conjunto con una Instancia P2 de Amazon. Las instancias P2 proporcionan hasta 16 GPU NVIDIA K80, 64 vCPU y 732 GiB de memoria del host, con 192 GB de memoria GPU en total, 40 000 núcleos de procesamiento paralelo, 70 teraflops de desempeño de puntos flotantes de precisión sencilla y más de 23 teraflops de desempeño de puntos flotantes de precisión doble. También ofrecen capacidades GPUDirect (comunicación GPU de punto a punto) para hasta 16 GPU, de modo que varias GPU pueden trabajar juntas con un solo host. Por ejemplo, con una instancia p2.16xlarge (16 GPU, 64 vCPU, 732 RAM GiB) para el uso de la aplicación de recupero de contraseñas Hashcat, se logró obtener 12275.6MH/s (millones de hashes por segundo) [5] mientras que la misma aplicación sin el uso de GPU, obtiene aproximadamente 2MH/s.

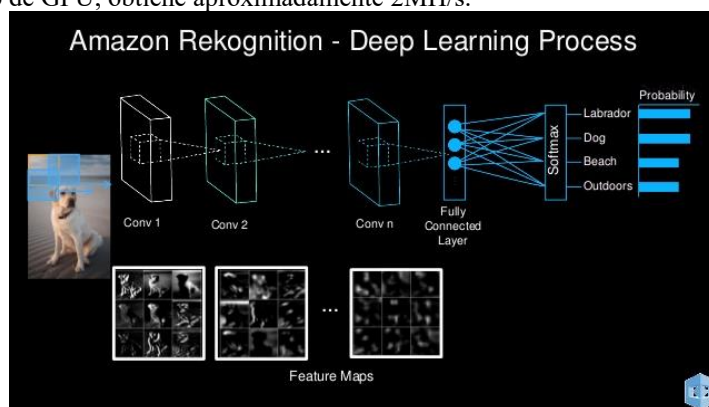


Ilustración 1: Deep Learning en Amazon Rekognition

Estando SmartTrashCan conectado a la nube, cada vez que se ingrese un residuo dentro del dispositivo, se tomará una foto del mismo y se la analizará desde dicha API para obtener como resultado a qué residuo corresponde.



**Ilustración 2:** diagrama de comunicación

La API es invocada desde la placa Arduino instalada en SmartTrashCan. El resultado del reconocimiento de imagen devuelto por dicha API, es recibido nuevamente en la placa Arduino para poder clasificar el residuo. Este resultado también será enviado a la aplicación mobile, para poder informarle al usuario qué tipo de residuos están siendo ingresados.

### 3 Explicación del algoritmo

```
//Creamos un request con la imagen tomada por la cámara, de forma secuencial por cada residuo
DetectLabelsRequest request = new DetectLabelsRequest()
.withImage(camara.tomarFoto());
try {
    //Usamos la funcion brindada por la API, esta detección de etiquetas la hace AWS de forma paralela
    DetectLabelsResult result = rekognitionClient.detectLabels(request);
    List <Label> labels = result.getLabels();
    //Una vez obtenidas las etiquetas para la foto, la clasificamos
    //para saber que tipo de residuo es
    for (Label label: labels) {
        clasificarResiduo(label.getName());
    }
} catch (AmazonRekognitionException e) {
    e.printStackTrace();
}
```

Para analizar el residuo en cuestión, lo que haremos es primero llamar desde el Arduino a una Lambda function la cual invoca a la API.

Parte secuencial: Crearemos un request enviando la foto sacada por SmartTrashCan para luego obtener las etiquetas que correspondan con dicha imagen. Estos requests se realizarán por cada residuo ingresado.

Parte paralela: La detección de los labels que coinciden con la foto obtenida, lo realiza de forma paralela AWS, aprovechando el poder de procesamiento de GPU.

Finalmente, al tener el listado de etiquetas, podremos clasificar el residuo en “Orgánico”, “Vidrio”, “Plástico”, “Papel” o “Metal”.

## 4 Pruebas que pueden realizarse

Para poder probar la eficacia de la mejora y que la implementación fue correcta, se pueden arrojar residuos de diferentes tipos al interior de SmartTrashCan.

## 5 Conclusiones

En resumen, hemos planteado una propuesta en la que implementando una cámara en el interior de SmartTrashCan e integrando el sistema a una solución provista por AWS como Amazon Rekognition, se puede mejorar de manera significativa el análisis de los residuos arrojados, lo que permitirá su clasificación para su posterior reciclaje y colaborar así con el conservamiento del medio ambiente.

Para futuros trabajos se podrían hacer análisis del tipo de residuos que mayormente desecha el usuario de SmartTrashCan, para de esta forma identificar qué tipo de usuario es. En caso de ser un usuario que desecha pocos materiales reciclables, poder recomendarle sustitutos reciclables para su estilo de consumo.

## 6 Referencias

1. <https://www.aipoly.com/>
2. <https://taptapseeapp.com/>
3. Will Steffen, Johan Rockström, Katherine Richardson, Timothy M. Lenton, Carl Folke, Diana Liverman, Colin P. Summerhayes, Anthony D. Barnosky, Sarah E. Cornell, Michel Crucifix, Jonathan F. Donges, Ingo Fetzer, Steven J. Lade, Marten Scheffer, Ricarda Winkelmann, and Hans Joachim Schellnhuber: Trajectories of the Earth System in the Anthropocene. PNAS Agosto 14, 2018 115 (33) 8252-8259; primera publicación Agosto 6, 2018 <https://doi.org/10.1073/pnas.1810141115>. Editado por William C. Clark, Harvard University, Cambridge, MA, y aprobado Julio 6, 2018
4. Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng \*. “Deep learning with COTS HPC systems” Internet: <http://proceedings.mlr.press/v28/coates13.pdf>, 2013 [Junio de 2018]
5. <https://medium.com/@iraklis/running-hashcat-in-amazons-aws-new-16-gpu-p2-16xlarge-instance-9963f607164c>