Universidad Tecnológica Nacional
ABRIL 2020 / Año 18- Nº 37

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933


Actas de las IX Jomadas Argentinas de Robótica 15-17 de noviembre, Córdoba, Argentina

# Aprendizaje profundo con imágenes RGB-D: clasificación de objetos y estimación de la pose

## Deep Learning with RGB-D Imaging: Object Classification and Pose Estimation

Presentación: 02/10/2017

Aprobación: 02/12/2017

**Juan Cruz Gassó Loncan**
Instituto de Investigación y Desarrollo en Bioingeniería y Bioinformática, Facultad de Ingeniería de la Universidad Nacional de Entre Ríos–Entre Ríos, Argentina
jcgassoloncan@ingenieria.uner.edu.ar

**Gerardo Gabriel Gentiletti**
Facultad de Ingeniería de la Universidad Nacional de Entre Ríos – Entre Ríos, Argentina
ggentiletti@ingenieria.uner.edu.ar

### Resumen

En el marco de la tesis de doctorado, se plantea como objetivo el desarrollo de una Interfaz Hombre-Máquina para comandar un brazo robótico asistencial de más de 6 grados de libertad. Se presenta el uso de técnicas de aprendizaje profundo para el reconocimiento de objetos y estimación de la pose a fin de poder interactuar con los mismos. Se implementaron 3 modelos de redes neuronales convolucionales multimodales para imágenes RGB-D de la base de datos BigBIRD, con tres salidas de clasificación: 22 Objetos - 5 Cámaras - 8 etiquetas de Rotación. Para el mejor de los modelos se alcanzaron valores de precisión de 96% para objetos, 98% para cámara y 56% para la rotación.

Palabras claves: Aprendizaje profundo, Visión robótica, RGB-D.

### Abstract

As part of the doctoral thesis, the objective is to develop a Human-Machine Interface to control a assistive robotic arm with more than 6 degrees of freedom. Deep learning techniques for object recognition and pose estimation in order to be able to interact with them is presented. Three multimodal convolutional neural network models were implemented using RGB-D images from the BigBIRD database. Each model have three classification outputs: 22 Objects - 5 Cameras - 8 Rotation. The best of model achieved 96% accuracy for objects, 98% for camera and 56% for rotation.

Keywords: Deep learning, Robotic vision, RGB-D.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

# I Introducción

Se plantea como objetivo el desarrollo de una Interfaz Hombre-Máquina (Human Machine Interface-HMI) para comandar un brazo robótico asistencial de más de 6 grados de libertad (Degree Of Freedom-DOF). El paradigma de control se basará en el concepto de "Control Compartido", con el objetivo de enlazar varias entradas de distintos tipos de sensores, para otorgarle al brazo robótico (RA) la inteligencia suficiente para adaptarse al eventual escenario de operación, e interpretar las intenciones del usuario codificadas en comandos de alto nivel (Saeedi y cols., 2013; Millán, 2016). A tal fin, es posible aplicar un sistema de control basado en visión, combinando algoritmos de procesamiento de imágenes, aprendizaje profundo (Deep Learning-DL) y técnicas avanzadas de control de manipuladores robóticos.

Con esto en mente, si lo que se desea es que el brazo robótico tenga la capacidad de interactuar con los objetos que rodean al usuario, el primer problema es identificar dichos objetos, ya que para cualquier acción que el RA sobre un objeto, se debe tener un mínimo de información de que "es" el objeto. El problema es aún más complejo, ya que para tomar el objeto se debe tener información de la posición y orientación del mismo, es decir la distancia relativa al brazo, para determinar si está en el espacio alcanzable, sumado la orientación del mismo, que determinarán los ángulos de rotación de la "mano" del RA y así encontrar la mejor manera de tomar el objeto. A esto además se le suma la acción que se va a realizar con el objeto, ya que una determinada acción puede necesitar una forma determinada de tomar el objeto, sin embargo, ese es otro problema que escapa a este trabajo.

De este modo, la información que se debería obtener a través de la visión son: (I)Identificar el objeto (Vaso, botella, caja, etc.) y (II) Parámetros de posición con 6 DOF: (II-a) Posición del objeto ($P_{obj}$ =$x$, $y$, $z$) y (II-b)Orientación del objeto($O_{obj}$= $\alpha, \beta, \gamma$).

La tarea de reconocer objetos en imágenes ampliamente estudiada, existiendo varios ejemplos populares de DL como son AlexNet,(Krizhevsky, Sutskever, y Hinton, 2012) VGGNet (Simonyan y Zisserman, 2014), ResNet (He, Zhang, Ren, y Sun, 2015), Inception (Szegedy y cols., 2015), Xception (Chollet, 2016) , etc. y todos los años se sigue avanzando en esta temática de aprendizaje maquinal (Guo y cols., 2016).

Como alternativa, las imágenes RGB-D se corresponden a los tres canales de las imágenes color estándares, más un cuarto canal "D" (depth - profundidad en inglés). Este tipo de imágenes han crecido en popularidad en los últimos años desde que salieron a la venta dispositivos XBOX Kinetic y el ASUS Xion. Este canal incorpora información de profundidad, es decir la distancia de la superficie vista de todos los objetos dentro del campo de visión, como una nube de puntos donde cada punto almacena la información de distancia en el orden de los milímetros en un rango desde que va 0.5m hasta sim4.5m (Shao, Han, Kohli, y Zhang, 2014; Cai, Han, Liu, y Shao, 2016). Además de estas posibilidades, sin ahondar en detalles, este tipo de sensores brindan la capacidad de tener una percepción tridimensional del ambiente, lo cual en robótica es de gran ayuda para detectar obstáculos.


Figura 1. Muestras de imágenes de entrenamiento de la base de datos BigBIRD. (CF:Con Fondo - SF:Sin Fondo). En las imágenes SF-4 y SF-8 se observan los artefactos causados por las máscaras de segmentación incorrectas.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

La base de datos a partir de la cual se va a entrenar cualquier modelo de aprendizaje maquinal juega un papel fundamental, y en especial en DL. Gracias los trabajos realizados por (Firman, 2016) y (Cai y cols., 2016), es posible tener una noción del estado del arte relacionado a las bases de datos RGB-D. En estos trabajos se cuenta con un listado de los diferentes bases de datos disponibles, acompañado de una descripción de sus características particulares, referencias y enlaces. Dentro de la clasificación que plantea Firman, encontramos "object pose estimation" (estimación de la pose de objetos). En esta sección Firman, comenta alguno de los conceptos introducidos en la sección 2, como el uso de modelos 3D de un objeto para inferir los parámetros de 6-DOF, pero también introduce otros parámetros, como son la oclusión, los cambios de iluminación y variación en ladistancia objeto. Todos estos parámetros también son parte del problema ya que en cualquier ambiente real del cual se quiera extraer información, probablemente no se tenga control sobre estas variables.

Se optó por la base de datos BigBIRD(Singh, Sha, Narayan, Achim, y Abbeel, 2014) por la buena cantidad de imágenes para el mismo objeto, el etiquetado de las imágenes más la disponibilidad de las matrices de posición intrínsecas de las cámaras y datos de calibración. Mediante la combinación de las matrices de transformación de calibración con las matrices transformaciones respecto al mesa, se puede obtener información de la pose para cada muestra. Además, dado que cada imagen está identificada con la cámara de la cual se tomó y el ángulo de rotación de la mesa rotante, se simplifica la generación de etiquetas. En la Fig.1 se pueden ver algunos ejemplos de los objetos y tipos de imágenes de la base de datos.

## A.Antecedentes

Ya que una revisión exhaustiva de la literatura en reconocimiento de objetos e inteligencia artificial con este tipo de datos RGB-D se escapa de los lineamientos de este trabajo, limita-remos la problemática a técnicas de aprendizaje profundo y se hará un breve comentario de algunos trabajos, destacando las conexiones con lo que se desarrollará más adelante.

Los trabajos de (Eitel y cols., 2015) y (Schwarz y cols., 2015) fueron los que más se asimilaron a la idea inicial y se tomaron como base. Ambos trabajos hacen uso de redes neuronales convolucionales (Convolutional Neural Networks - CNN) e imágenes RGB-D como entradas.

(Schwarz y cols., 2015) implementaron un modelo haciendo uso de una versión pre-entrenada de la CNN desarrollada para ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2011 por (Krizhevsky y cols., 2012). Para hacer uso de esta red con datos RGB-D aplicaron un preprocesamiento de las imágenes de la base de datos (Lai, Bo, Ren, y Fox, 2011) para adaptarlas a esta red, codificando en colores las imágenes de profundidad. Su modelo es de doble entrada, una para las imágenes RGB y otra para las deprofundidad, para finalmente hacer uso de las características extraídas de la CNN y alimentar máquinas de soporte vectorial (Support Vector Machines - SVM) sucesivas para determinar la categoría del objeto en primer nivel, la instancia del objeto en segundo y la estimación de la poción en último nivel. La estimación de la posición se basa en detectar el ángulo de rotación del objeto (Eitel y cols., 2015).

En el trabajo de (Eitel y cols., 2015), también hacen uso de una red de dos columnas separadas y se basan en la CNN pre-entrenada de (Krizhevsky y cols., 2012), y además coinciden en la coloración de las imágenes de profundidad. Por último, también coincidieron en la base de datos usada (Lai y cols., 2011). El contraste con el trabajo anterior está en que Eitel et al. dedicaron más trabajo al ajuste fino de los parámetros de la red en tres etapas, realizando entrenamiento con el modelo completo. Primero trabajaron ajustando los parámetros de la columna de imágenes RGB, luego la columna de profundidad y por último con ajuste con la red completa. Otro de sus aportes fue trabajar con dos técnicas de colorización de las imágenes de profundidad y además implementaron un proceso de "data augmentation" novedosa para las imágenes de profundidad, agregando patrones comunes de ruido (Eitel y cols., 2015; Schwarz y cols., 2015).

Una característica común en estos trabajos, es la arquitectura del modelo, donde la extracción de características de cada entrada, son procesadas por separado y luego se fusionan en capas superiores. Esta arquitectura se la suele denominar "fusión tardía". Existe otra arquitectura habitual, conocida como "fusión temprana" que consiste en unificar la entrada concatenando los canales RGB y D para ser procesados juntos. Se pueden encontrar otras arquitecturas, y si bien no está del todo clara cuál es mejor, en los trabajos de Lenz et al. y Shao et al. sen presenta un análisis de los diferentes enfoques (Lenz, Lee, y Saxena, 2015; Shao, Cai, Liu, y Lu, 2017).

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

# II Implementación

## A.Materiales

La implementación de software para este trabajo se realizó en lenguaje Python 3.5 corriendo tanto en los sistemas operativos Ubuntu 16.04LTS como en Windows 10. La librería base de aprendizaje maquinal usada es Keras 2.0 con backend de Tensorflow. Otras librerías que vale la pena mencionar son OpenCV-3.3, scikit-learn, scikit-image, y numpy. El hardware utilizado es una notebook equipada con un procesador Intel Core i7-7700HQ, 8GB memoria RAM DDR4-2400Mhz y una GPU NVIDIA GeForce GTX 1060-3GB móvil.

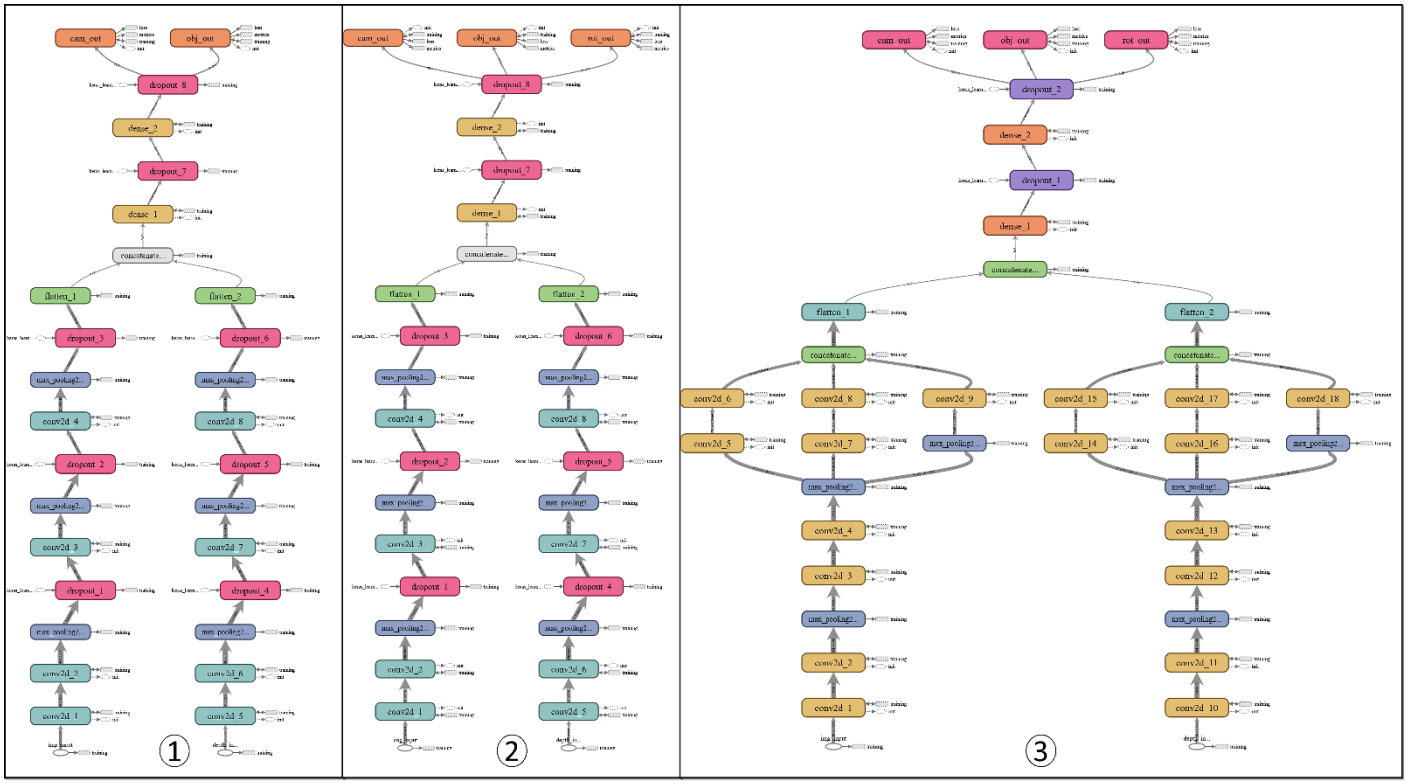


Figura 2. Arquitecturas de los tres modelos de CNN.

## B. Procesamiento común datos

Se dedicó un buen tiempo para encontrar la forma de leer los datos de la base de datos locales de un modo que se puedan agregar o quitar objetos sin problemas. Alguno de los problemas fueron el contar con 1800 imágenes por objeto sumando las RGB, D y las máscaras, por lo cual cargar los datos en memoria RAM no era posible, al menos sin redimensionarlas drásticamente. Por otro lado, no estánseparadas en sets de entrenamiento, test y validación.Se tuvieron que generar las etiquetas y tener cuidado de mantener correlación entre las imágenes RGB y las D. Estas últimas hay que obtenerlas de matrices binarias en archivos HFD5, por lo que se debe realizar un paso más para convertirlas a imágenes.

| Tipo | Cantidad | Dimensión | Tamaño | Total |
|---|---|---|---|---|
| RGB-uint8 | 600 | 1280x1024 | 950KB | 570MB |
| D-unit16 | 600 | 480x640 | 602KB | 601.2MB |
| Máscara-uint8 | 600 | 1280x1024 | 160KB | 96MB |
| Total | 1800 | - | - | 1267.2MB |

Tabla I. Base de datos bigBIRD.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

Para resolver el problema de cómo leer las imágenes, se trabajó directamente con las direcciones absolutas de las imágenes, organizándolos en listas ordenadas y dado que todos los nombres de todos los archivos tienen en común el nombre de la cámara correspondiente y el ángulo de rotación de la mesa, el orden de las listas de imágenes RGB y D, era correspondiente, por los que se las asoció. Esto permitió en forma sencilla aleatorizar las listas manteniendo la correspondencia y a su vez facilitando la separación en sets de entrenamiento, validación y test. La proporción seleccionada fue de 70-20-10, por lo que se cuenta con 420 para entrenamiento, 120 para validación y 60 para test, por objeto.

Por otro lado, dado que la parte de información de interés de las imágenes son aquellos píxeles propios al objeto, el cual para todas las imágenes se encuentra en el aproximadamente en el centro de las mismas, una posibilidad era recortar las imágenes de forma de que contenga menos píxeles correspondientes al fondo o background. Para tomar la misma modalidad en las imágenes D, primero había que unificar las dimensiones de

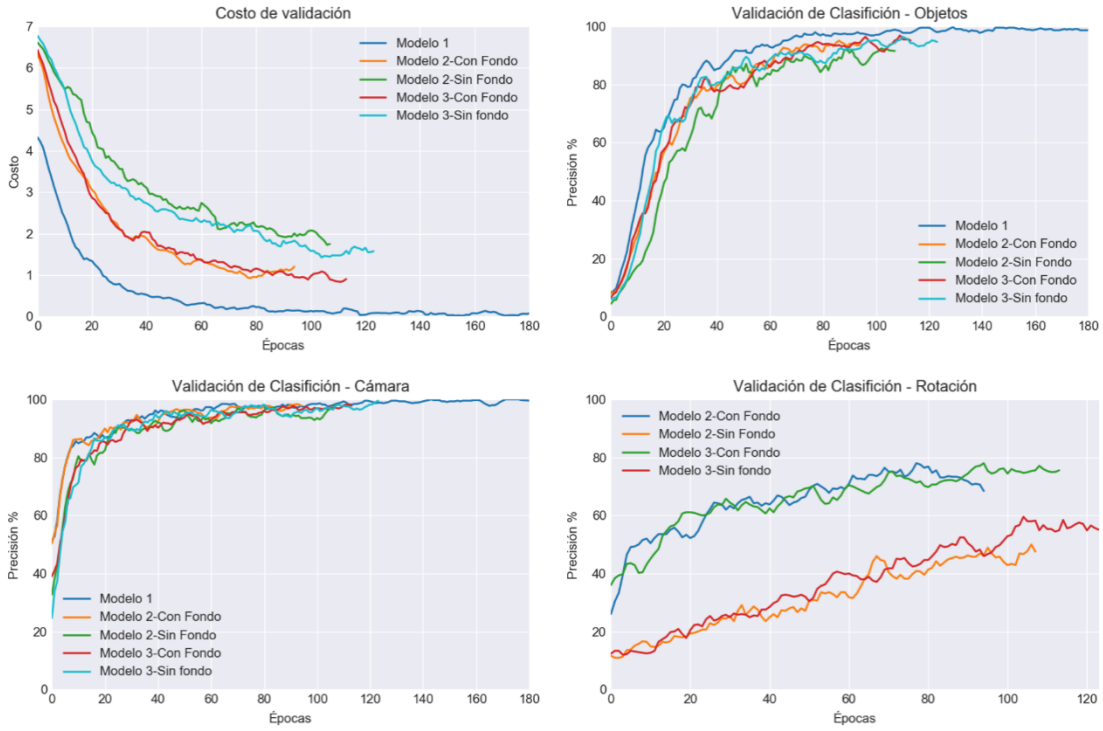


Figura 3. Gráficos de resultados de validación durante el entrenamiento. Las gráficas fueron suavizadas por promediación con ventana deslizante de 3 muestras para facilitar la visualización.

todas las imágenes, ya que son aproximadamente del doble de tamaño como se muestra en la Tabla I. Para esto, primero se redujo la dimensión de las RGB a 640x512, para luego recortar dos franjas verticales de 16 píxeles a ambos lados y de este modo alcanzar la resolución de 640x480 de las imágenes D. Finalmente, para cada imagen se tomó el centroide de la máscara, a partir del cual se recortaron a una dimensión de 240x240. A partir de aquí, cualquier cambio de tamaño se hizo redimensionando las imágenes.

Por último, para obtener el centroide de las máscaras se realizó un proceso de suavizado por nacionalización con un kernel grande sobre las mismas para suavizarlas. Esto fue necesario porque para algunos objetos, las máscaras no están bien determinadas y poseen huecos o no son continuas. Entonces, muchas veces se calculaba el centroide a partir de alguna de las manchas y no del objeto completo o simplemente no se encontraba.

Este problema se resolvía al suavizarlas el detector de contornos era más robusto a las discontinuidades, y mediante un umbral se siempre se selecciona la isla más grande a partir de cual calcular el centroide. De este modo, la mayoría de las imágenes eran correctamente centradas antes de ser recortadas. En todo caso de no encontrarse un centroide, se usaba el centro de la imagen.

## C. Modelos

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

Se probaron un total de 3 modelos diferentes utilizando el mismo set de entrenamiento, validación y test. Todos los modelos fueron entrenados usando los mismos parámetros. Para la función de costo se usó la *categorical crossentropy* (entropía cruzada categórica) y como optimizador *Adadelta* y no se usaron técnicas de regularización. Cada época constaba de 100 aleatorias, y la validación 75 muestras. Po último se configuró la detención temprana monitoreando el costo de validación con una paciencia de 15 épocas. En la Fig.3 se presentan la evolución de la validación durante el entrenamiento. A continuación, se detallan las diferencias y los resultados de cada uno.

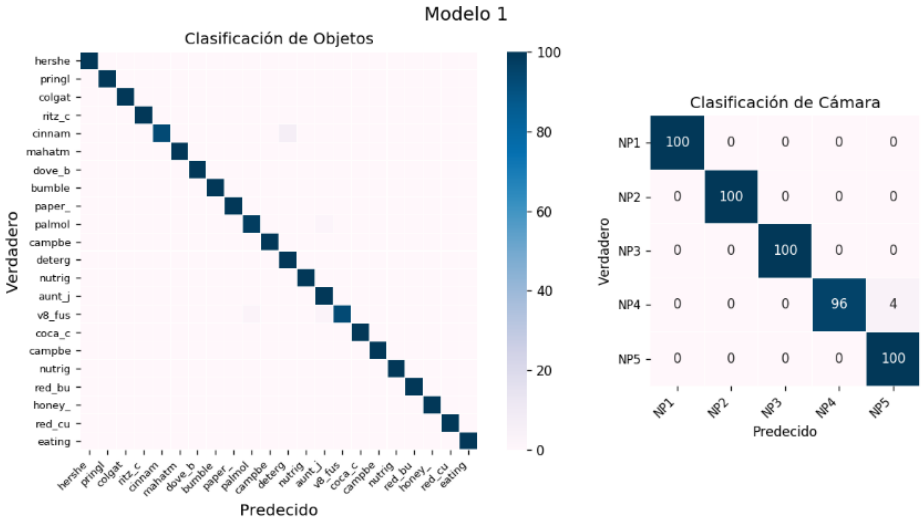


Figura 4. Matriz de confusión normalizadas para la clasificación de objetos y cámara para el Modelo 1.

*Primer Modelo*

Para la arquitectura del primer modelo, como se ve en la Fig.2-1 se optó por la arquitectura de fusión tardía, con dos columnas independientes. Ambas columnas son simétricas, con la diferencia en número de entradas. Dado que una columna, procesa las imágenes color, su entrada es de dimensiones 96x96x3, para el alto y ancho de la imagen, más los 3 canales correspondientes a cada color RGB. La segunda columna procesa las imágenes D, en escalas de grises (monocanal).

Cada columna de capas convolucionales está igualmente compuesta. La primera capa es una convolución2D (conv2d) de 10 capas (c) y un kernel (k) de 5x5 conv2d(10c-5x5k). Las siguientes capas convolucionales fueron conv2d(32c-3x3), conv2d(64c-3x3) y conv2d(128c-3x3). Se intercalaron operaciones de MaxPoolings2D y dropOut(10 %) en colores Azul y Fucsia respectivamente, en la Fig.2.

De cada una de estas columnas se toman las salidas y se reestructuran en forma de vector fila, para luego concatenarse en un único vector de características. Este vector pasa a ser la entrada a un par de capas densas comunes de 512 y 256 neuronas, intercalando dropOut(20 %). Finalmente, dela última capa densa se desprenden dos más, una para la clasificación de objetos y otra para la clasificación de la cámara, por consiguiente, el número de neuronas de estas capas queda definido por el número de clases en cuestión, para cada caso. En particular, se usaron 22 objetos y el arreglo de 5 cámaras usado para la base de datos BigBIRD(Singh y cols., 2014). A lo largo de todo el modelo se usó la función de activación ReLu para las capas intermedias y la Softmax para las salidas. El número total de parámetros de este modelo es de 19.204.695 de los cuales cerca de 19 millones corresponden a las capas densas.

*Resultados Modelo 1*

Como se puede ver en la en las matrices de confusión de la Fig.4 modelo puedo clasificar los objetos y detectar la cámara con una tasa de aciertos casi perfecta. La precisión de clasificación alcanzada fue de 98.40 % para objetos y de un 98.78 % para la cámara.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

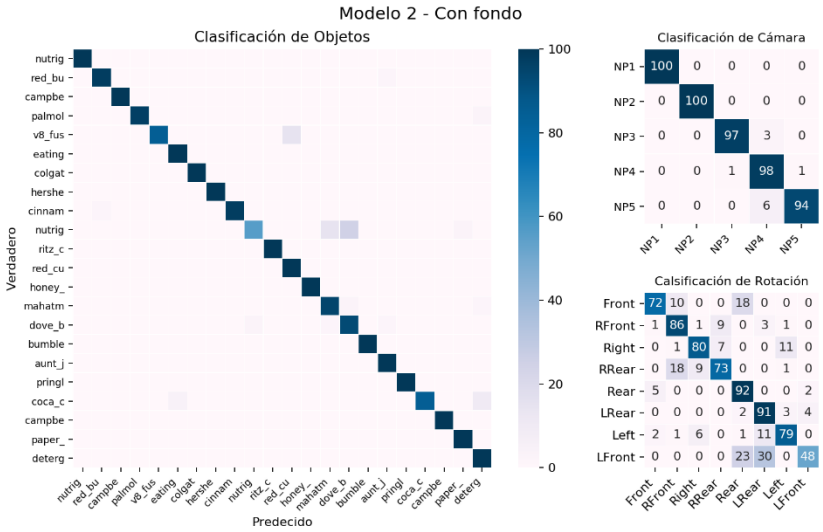


Figura 5. Matrices de confusión del Modelo 2-CF para las tres salidas.

*Segundo modelo - Con fondo*

A partir de los buenos resultados obtenidos con el modelo anterior, lo que se hizo fue solo agregar una nueva salida al modelo correspondiente al ángulo de rotación del objeto. Para esto, en lugar de tomar 120 etiquetas para cada rotación, se redujo el número a 8 etiquetas correspondientes a las vistas de Frente, Frente izquierdo, Izquierda, Trasera izquierda, Trasera, Trasera derecha, Derecha y Frente derecha. Para esto se tomaron intervalos de 45°. La etiqueta Frente, se centró en 0° y dado que las rotaciones son múltiplos de 3, el intervalo fue desde 339° hasta 21°. Es decir, la imagen en 0°, siete imágenes para un lado (3° - 21°) y siete para el otro (357° - 339°); así sucesivamente para cada etiqueta. En este modelo el número de parámetros se incrementó en 2021 por la nueva capa densa para la salida de rotación, como se aprecia en la Fig.2-2.

La hipótesis de este modelo, fue que se iban a mantener las tasas de acierto de clasificación de objetos y cámaras, pero se esperaban resultados relativamente pobres respeto a la rotación. Esto se atribuye principalmente a dos cuestiones,por un lado, existen objetos que presentan simetría radial por lo cual no existe diferencias significativas entre los lados del objeto en cuanto a morfología del objeto, quedando solo disponible la información relativa la textura o colores del mismo. Por otro lado, los objetos vistos desde la vista superior, prácticamente vertical, es aún más difícil determinar el ángulo de rotación ya que se pierde cualquier información relativa a los lados del objeto.

*Resultados Modelo 2-CF*

Los resultados mantuvieron una tasa de acierto aceptable respeto a la clasificación de objetos (95.15 %) y cámara (96.15 %). Contrariamente a lo esperado, se obtuvo una clasificación aceptable en cuanto a la rotación (73.63 %) como se puede apreciar en la matriz de confusión en la 5.

*Segundo modelo - Sin fondo*

A partir de ver en los resultados del modelo anterior, que la hipótesis no fue correcta, dando evidencia que el modelo estaba tomando información del fondo para determinar al menos la rotación. Una posible fuente de información respecto al ángulo de rotación, evidentemente podía ser es la posición del tablero de calibración de las cámaras. Este tablero, es muy fácil de detectar ya que presenta un patrón geométrico tipo tablero de ajedrez y además está presente en todas las imágenes. Se había pensado que, a partir de recortar las imágenes, la información del tablero se perdía, pero luego se vio que en muchas de las imágenes alguna parte del mismo aparecía, como se puede ver en los ejemplos de la Fig.1.

Para comprobar esta nueva hipótesis, se implementó un algoritmo para quitar el fondo de las imágenes RGB. En principio, se pensó simplemente usar las máscaras de segmentación para retirar el fondo, pero como se mencionó anterior-mente muchas de estas máscaras estaban mal determinadas. Por lo que de nuevo se trabajó con las máscaras haciendo un suavizado con un kernel rectangular (7x15) y de este modo la máscara de segmentación pasó de ser binaria y bien definida, a ser una mancha en tonos de grises.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

Finalmente se usa esta nueva máscara como entrada al algoritmo de grabCut disponible en OpenCV. Para esto, seetiquetaron los píxeles entre Fondo seguro-FS, Primer plano seguro-PPS, Fondo probable-FP y Primer plano probable-PPP. En base a prueba y error el criterio de selección de píxeles fue tomar como FS aquellos píxeles iguales al máximo, FP a aquellos píxeles mayores iguales al 99 % del máximo, PPS aquellos píxeles menores iguales al percentil 70 de los valores únicos de la máscara y finalmente como PPP a los píxeles restantes. Con esta información el algoritmo de grabCut genera una nueva máscara final, a partir de la cual se extrae el fondo. En reemplazo del fondo, se decidió dar valor de color promedio de la imagen original, por lo que el fondo de las imágenes procesadas serán diferentes entre imágenes, pero siempre en tonos de similar. A pesar de este procedimiento, no se tuvieron buenos resultados con algunas mascaras que eran muy pobres. En la Fig.1 se pueden ver un par de ejemplos.

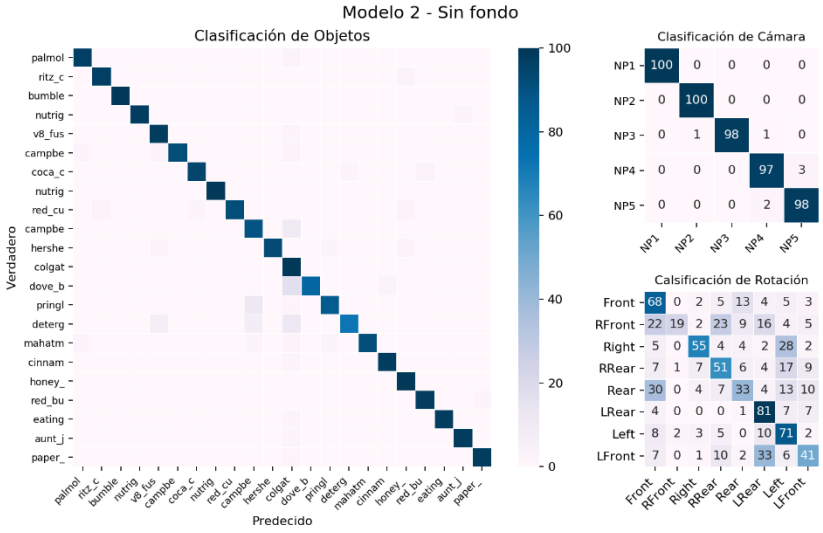

Figura 6. Matrices de confusión del Modelo 2-SF para las tres salidas.

*Resultados Modelo 2-SF*

Los resultados de este modelo respaldan la idea de que la red estaba tomando información del fondo, conservando una buena tasa de aciertos para objetos (93.33 %) y las cámaras (98.40 %), pero una marcada diferencia en cuanto a la clasificación de rotación con un porcentaje de aciertos de 41.69 %. Estos resultados se ven reflejados en las matrices de confusión en la Fig.6.

*Tercer modelo*

Para el último modelo, se buscó implementar una arquitectura más compleja (Ver Fig.2-3), de mayor cantidad de parámetros (más del doble 51.352.512) buscando ver si hay cambios significativos en los resultados. Luego de revisar algunas de las arquitecturas novedosas las redes citadas anteriormente, se decidió implementar un módulo similar al Inception (Szegedy y cols., 2015), a cada columna. Este módulo se caracteriza por tres ramas paralelas de procesamiento diferente. En dos de ellas, la operación es una conv2D(64c-1x1k) seguida de un conv2D(64c-3x3k) por un lado y de conv2D(64c-5x5k) por el otro. La ultima rama, realiza un MaxPooling2D(3x3) seguida de una conv2D(64c-1x1k).

Sin entrar en detalles, de los fundamentos teórico de este módulo combina características extraídas desde diferentes tamaños de kernel, para luego combinarlas en un único vector de características, cuidando el uso de memoria a través de reducciones de dimensionalidad de las conv2D(1x1k). Según los autores, los diferentes tamaños de kernel podrían captar patrones con diferentes capos visuales, por lo que podría detectar detalles máspequeños sin la necesidad de apilar tantas capas convolucionales. Adicionalmente, para este modelo se usaron tamaños de imagen más grandes de 128x128 a la entrada de la red.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

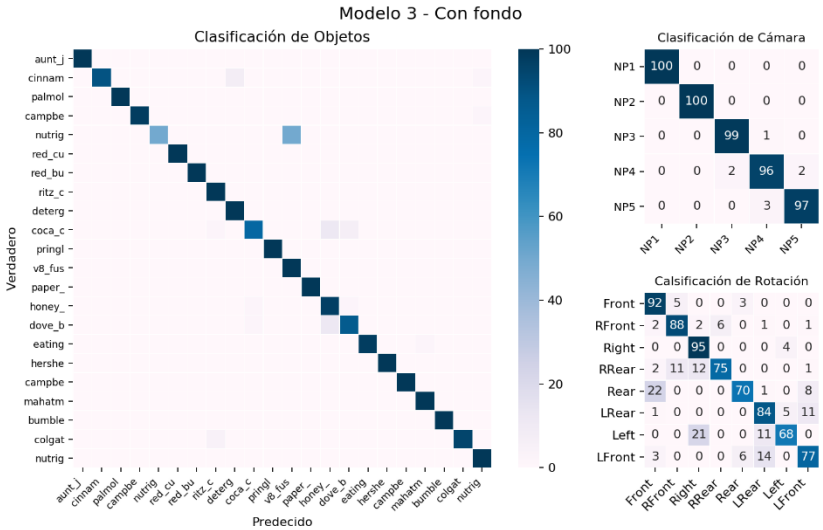


Figura 7. Matrices de confusión del Modelo 3-CF para las tres salidas.

Por último, vale mencionar que el cambio en número de parámetros es debido a, por un lado, el aumento del tamaño de las imágenes de entrada y por otro debido a un número considerable de capas convolucionales en cada columna, ya que las capas densas finales son las mismas que el modelo anterior. La secuencia inicial de capas es conv2D(32c-7x7k) y MaxPooling(3x3) ambas con paso de 2x2, conv2D(64-5x5k), MaxPooling(2x2), conv2D(128c-3x3k), conv2D(128c-3x3k) y MaxPooling(2x2), todas con paso simple. Esta estructura también está inspirada en la red Inception (Szegedy y cols., 2015).

*Resultados Modelo 3*

Este modelo presentó mejores resultados que el anterior, pero sin ser superlativas. la tasa de acierto para objetos fue de 95.15 %-CF y 96.81 %-SF, para la cámara 97.5 %-CF y 98.56 %-SF, finalmente para la rotación 79.84 %-CF y 56.59 %-SF. Lo llamativo en en los resultados fue que al contrario que el modelo anterior, el entrenamiento sin fondo presento mejores resultados para la clasificación de objetos, que con fondo.

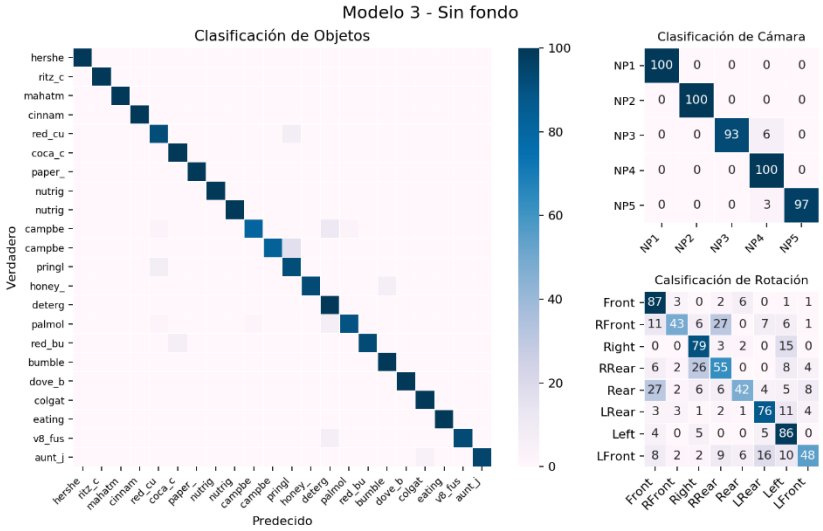


Figura 8. Matrices de confusión del Modelo 3-SF para las tres salidas.

## III Conclusiones

Se implementaron y compararon diferentes modelos de redes neuronales convolucionales multimodales a partir de imágenes RGB-D. Uno de los aspectos más interesantes de los resultados, es que en todos los casos se generalizo el concepto de cámara. Al mismo tiempo, el reconocimiento de objetos desde cualquier perspectiva

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

también es prometedor. En cuanto a la clasificación de la rotación, pese a no ser buenas tasa de acierto, fue importante reconocer el factor fondo como baias, por lo que se implementó un algoritmo para quitar el fondo.

Al mismo tiempo se observó que la clasificación sin fondo, de objetos y cámara, no se ve notoriamente afectada. El tercer modelo parece evidenciar el papel de las capas convolucionales en la extracción de características, facilitando la tarea de las capas densas finales.

## IV Discusión

Las diferencias de precisión entre el primer modelo y los siguientes, puede apreciarse en el costo de validación de la Fig.3, ya que al no lograr que costo introducido por la clasificación de la rotación hace el costo general se vea afectado, por lo que el entrenamiento termine antes por la detención temprana. Por otro lado, se puede ver que no hay grandes diferencias entre el modelo 2 y 3 durante el entrenamiento, por lo que se debe tener en cuenta el costo beneficio de un modelo más complejo.

Si bien los resultados de rotación no son buenos, se puede ver que las etiquetas que se confunden tienen relación. Por ejemplo, en la Fig.8 se puede ver que hay una confusión entre Atras con Adelante (Rear - Front) y entre Izquierda y Derecha (Left - Right). Además, no hay que dejar de tener en cuenta que hay objetos que poseen forma cilíndrica y textura casi uniforme (por ejemplo, red cup Fig.1 SF-2 y 3). Para evaluar esto, se debería realizar un revelamiento del error de clasificación en la rotación por objeto individual.

Se debe trabajar para resolver los problemas en la extracción del fondo. El problema con las máscaras, se puede atribuir a que fueron extraídas automáticamente de la información de profundidad, el cual tiene problemas con cuerpos trasparentes como el vidrio (Ejemplo en la Fig.1 SF-4).

Por último, hay que tener en cuenta que, para el caso de la clasificación de la cámara, con el uso de las matrices de T se puede estimar indirectamente la pose del objeto, pero en este caso solo si esta en la misma posición que en la base de datos, es decir vertical. Por lo que está claro que se debe tener en cuenta que sucederá sí se quiere reconocer un objeto en otra posición, por ejemplo, la un vaso acostado o una caja sobre otra de sus caras.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.37.146-156.2020 - ISSN 1666-6933

## Referencias

Cai, Z., Han, J., Liu, L., y Shao, L. (2016). RGB-D datasets using microsoft kinect or similar sensors: a survey.*Multimed. Tools Appl.*, *76*(3), 4313–4355.

Chollet, F. (2016, 7 octubre). Xception: Deep learning with depthwise separable convolutions.

Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M., y Burgard, W. (2015). Multimodal deep learning for robust RGB-D object recognition. En *2015 IEEE/RSJ international conference on intelligent robots and sys-tems(IROS)*.

Firman, M. (2016). RGBD datasets: Past, present and future. En *2016 IEEE conference on computer visionand pattern recognition workshops(CVPRW)*.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., y Lew, M. S. (2016, 26 abril). Deep learning for visualunderstanding: A review. *Neurocomputing*, *187*(Supplement C),27–48.

He, K., Zhang, X., Ren, S., y Sun, J. (2015, 10 diciembre).Deep residual learning for image recognition.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En F. Pereira, C. J. C. Burges, L. Bottou, y K. Q. Weinberger (Eds.), *Advances in neural infor- mation processing systems 25* (pp. 1097–1105). CurranAssociates,Inc.

Lai, K., Bo, L., Ren, X., y Fox, D. (2011). A large- scale hierarchical multi-view RGB-D object dataset. En *2011 IEEE international conference on robotics andautomation.*

Lenz, I., Lee, H., y Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Rob. Res.*, *34*(4-5), 705– 724.

Millán, J. D. R. (2016, febrero). Brain-controlled devices: the perception-action closed loop. En *2016 4th international winter conference on Brain-Computer interface (BCI)*(pp. 1–2). IEEE.

Saeedi, S., Carlson, T., Chavarriaga, R., y del R. Millán, J. (2013, junio). Making the most of context-awareness in brain-computer interfaces. En *2013 IEEE international conference on cybernetics (CYBCO)* (pp. 68–73).

Schwarz, M., Schulz, H., y Behnke, S. (2015). RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. En *2015 IEEE international conference on robotics and automation (ICRA)*.

Shao, L., Cai, Z., Liu, L., y Lu, K. (2017). Performance eva- luation of deep feature learning for RGB-D image/video classification. *Inf. Sci.*, *385-386*, 266–283.

Shao, L., Han, J., Kohli, P., y Zhang, Z. (Eds.). (2014). *Computer vision and machine learning with RGB-D sensors:*. Springer International Publishing.

Simonyan, K., y Zisserman, A. (2014, 4 septiembre). Very deep convolutional networks for Large-Scale image recognition.

Singh, A., Sha, J., Narayan, K. S., Achim, T., y Abbeel, P. (2014). BigBIRD: A large-scale 3D database of object instances. En *2014 IEEE international conference on robotics and automation (ICRA)*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.,Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp.1–9).