Machine Learning aplicado para detectar tipos de productos consumibles

Eliécer Mora, Fabricio León, Sergio Guillén
Escuela de Electronica
Instituto Tecnologico
Cartago, Costa Rica
eliecer9000@gmail.com, fleonzuniga@gmail.com, sergio_guillen@hotmail.com

***Resumen*—Es difícil subestimar el impacto de la inteligencia artificial (IA) en la industria de productos de consumo. Nunca antes las empresas han podido obtener tanta información sobre sus clientes y utilizar ese conocimiento para crear soluciones inteligentes. La adopción de IA es necesaria para hacer crecer una empresa y seguir siendo relevante en la actualidad. Además, Machine learning ya no es solo un tema de actualidad entre el mundo del desarrollo científico, es una herramienta real que le permite ofrecer a sus clientes soluciones personalizadas y relevantes y, en última instancia, aumentar las ventas y la satisfacción del cliente. Dentro del ecosistema de soluciones existentes, hay una pieza que no puede faltar y se trata de la capacidad del sistema de dividir conceptualmente los productos de los que consta y agruparlos por categorías. En el presente documento se detalla la propuesta de un modelo de ML que permite tal categorización de sus datos en conjuntos de productos diferenciables. Se presenta también la implementación de dicho modelo en la tarjeta Jetson Nano con los resultados obtenidos.**

***Palabras clave*—Deep convolutional networks, VGG, ResNet.**

I. Introducción

Con el paso de los años las redes neuronales han visto su uso y popularidad incrementarse en diversas áreas. Esto por el buen desempeño que estas desarrollan. Una de las redes neuronales más usadas a la hora de clasificación de imágenes son las redes convolutivas o CNN por sus siglas en inglés. En este trabajo se hará uso de una red neuronal convolutiva para la clasificación de imágenes. Específicamente una mezcla de red VGG y una ResNet. Para este trabajo se creó un dataset de imágenes de alimentos tomados de la página de compras en línea de Walmart Centroamérica. Específicamente de bebidas y snacks con dos tipos de labels, bebidas y snack. Otro tipo de clasificación o labels dispuestas por el tipo de material. Para el trabajo en cuestión se usará la primera clasificación de labels para determinar si se tienen bebidas o snacks. Se utilizó la librería sklearn para realizar el entrenamiento del modelo así como keras para realizar las operaciones clásicas de Conv2D, ReLU, BatchNormalization, Add y MaxPooling2D entre otras.

II. Machine Learning (Background)

A. ANN

Las redes neuronales artificiales marcaron un antes y un después en el área de aprendizaje automatizado o machine learning. Dando inicio al aprendizaje profundo o deep learning. Inspirándose en las neuronas biológicas se llegó al desarrollo del perceptrón. Este modelo de neurona artificial multiplica las entradas por pesos y le suma un valor de bias $Wx + b$. Para posteriormente pasar el resultado por una función de activación no lineal. Mediante el uso de backpropagation se ajustan los pesos con respecto al error. Este backpropagation se logra mediante la gradiente descendiente.

B. CNN

Las redes neuronales convolutivas fueron propuestas por Yann LeCun en 1989. Estas redes se suelen usar para la clasificación de imágenes. La arquitectura de estas redes consta de tres etapas. La convolución, donde se obtienen los pesos mediante esta esta función matemática. El pooling, donde se extraen las características y la capa de conexión o Fully Connected Layer, donde se transforma el resultado de las etapas anteriores a un vector aplanado de una dimensión. Como la convolución es una función lineal se requiere de una función de activación no lineal igual que en las ANN. La más común es la función ReLU.

C. Entrenamiento

Cada capa transforma los datos de entrada en una representación más abstracta y el modelo aprende a elegir las mejores características que pueden mejorar el rendimiento. En el aprendizaje supervisado se tiene una referencia que le indica al modelo cuál resultado es el mejor para poder complir con la referencia, que no es más que la salida esperada del modelo. Una forma de determinar el error del modelo es mediante la entropía cruzada o cross entropy. $\text{CE} = -\sum_x p(x)\log(q(x))$, donde x es una muestra en el set de entrenamiento, $p(x)$ es la probabilidad verdadera de la distribución de la variable dependiente Y $q(x)$ es la distribución de probabilidad predicha. Otra función usada es el error cuadrático medio $\text{MSE} = \frac{1}{|n|}\sum_x (y(x) - \tilde{y}(x))^2$, donde n es el número de muestras set de entrenamiento, y es la salida real y $\tilde{y}$ es la salida del modelo.

D. Conjunto de datos

El conjunto de datos con el que se estará trabajando es creado para este trabajo de imágenes de alimentos tomados de la página de compras en línea de Walmart Centroamérica. Estas imágenes son fotografías de los productos de esta cadena

	image name	type	Descripción	Category	Material	Precio (Colones)
0	744100161906	jpg	Jugo Dos Pinos Naranja 100% Natural - 1800 ml	Bebibas	Botella de plástico	2595
1	744107412727	jpg	Agua De Pipa – 1800 ml	Bebibas	Botella de plástico	4100
2	1630016574	jpg	Jugo Florida Natural Naranja No Pulp – 2630 ml	Bebibas	Botella de plástico	6650
3	4850020277	jpg	Jugo Naranja Con Calcio Tropicana – 1530 ml	Bebibas	Botella de plástico	3550
4	744107411249	jpg	Agua De Pipa – 8 oz	Bebibas	Botella de plástico	900

Fig. 1. Conjunto de datos. Tomado de [1].

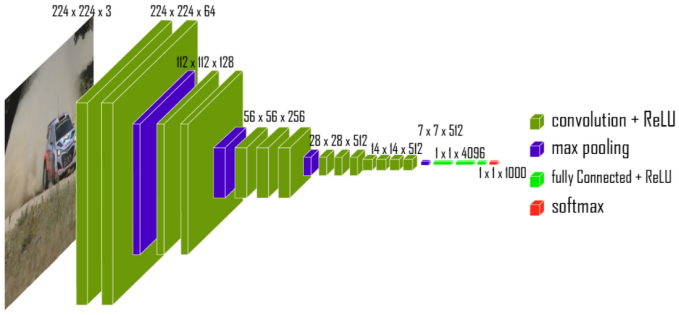

Fig. 2. Arquitectura de VGG. Tomado de [2].

de supermercados. Por lo que la imagen en cuestión consta del producto frente a un fondo blanco. Es por esta razón que al realizar la posterior prueba se debe tomar esto en cuenta ya que podría llevar a que el fondo de las imágenes de prueba, de no tener un fondo blanco lleven al modelo a incurrir en un error. La figura 1 presenta un resumen de las características del conjunto de datos que se analizarán:

III. Modelo

A. VGG (Visual Geometry Group)

Las redes VGG fueron propuestas por Karen Simonyan y Andrew Zisserman en el artículo Very Deep convolutional networks for large-scale image recognition [2] en 2014 para el ILSVRC challenge. Las VGG se forman mediante grupos de bloques compuestos por 2 o 3 capas convolutivas y una capa de pooling. La red se forma colocando un bloque después de otro para lograr una arquitectura más profunda. Todas las capas utilizan ReLU como función de activación. La función de activación ReLU es más eficiente computacionalmente porque da como resultado un aprendizaje más rápido y también disminuye la probabilidad de que desaparezca el problema del gradiente. La arquitectura se muestra en la figura 2.

B. ResNet (Residual Network)

Las redes ResNet fueron propuestas por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en el artículo Deep Residual Learning for Image Recognition en 2015 [3]. Para lograr mejores modelos se recomienda diseñar el modelo con más capa o lo que es lo mismo hacerlo más profundo. Pero para los modelos de arquitectura tradicionales se ha visto que tienen un umbral de profundidad, como se ve en la figura 3. Esto por la desaparición del gradiente.

Para solucionar este problema los autores propusieron un bloque residual que se ilustra en la figura 4. Este bloque

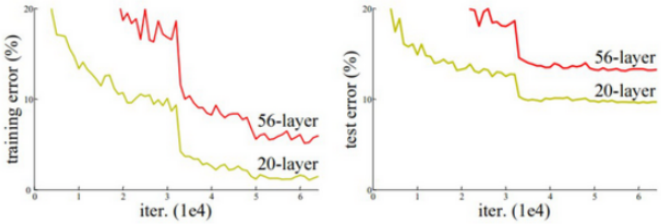

Fig. 3. Comparación del error por capas. Tomado de [3].

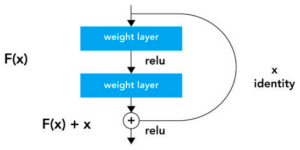

Fig. 4. Bloque residual. Tomado de [3].

consiste en una conexión o salto de una capa inicial a otra capa. Matemáticamente se tiene

$$H(x) = f(wx + b)$$

para un bloque tradicional. En el caso del bloque residual se tendría

$$H(x) = f(wx + b) + x$$

. Mediante la suma de x, el salto o conexión de capas no adyacentes se logra resolver el problema de la desaparición del gradiente, al permitir esta ruta de acceso directo alternativo para que fluya el gradiente

IV. Propuesta

El modelo propuesto es una arquitectura que mezcla las redes VGG [4] y las redes ResNet [5]. Se hará uso de dos bloques VGG compuestos por tres capas convolutivas cada uno. De la primera capa del bloque VGG se da un salto o conección a la tercera capa. Creando una conexión residual propia de las ResNet. Con esto se busca hacer uso de las características de ambas arquitecturas para reducir la desaparición del gradiente descendiente y así obtener mejores resultados. Luego de estas capas convolutivas con su conección residual correspondiente se coloca una capa de activación ReLU y una capa de Pooling. Esto siguiendo la arquitectura VGG, donde se usa la función de activación ReLU por su eficiencia computacional y la capa MaxPooling para realizar un downsampling o submuestreo. Cada nuevo bloque VGG que se coloque aumentará al doble su número de filtros. En este caso se hará uso de dos bloques VGG, el primero con 64 filtros y el segundo en consecuencia con lo antes establecido de 128 filtros. Finalmente se cuenta con la etapa Fully Connected Layer donde se realizará la clasificación en base a los pesos suministrados por las capas convolutivas anteriores. Inicialmente se realiza el aplanamiento de los datos mediante la capa de Flatten. Donde se pasa a una dimensión. Luego una capa densa para reducir las salidas. Posteriormente una capa de regularización Dropout. Y finalmente la última

capa densa cuya salida son las categorías en las que se estará trabajando.

Con un tamaño de batch de 128 y 200 epoch's ejecutados en el entrenamiento se alcanza una precisión del 90% tras 61 minutos de tiempo consumido para el entrenamiento. En la figura 7 se muestran las gráficas de precisión y pérdidas obtenidas:

V. Resultados Obtenidos

Un detalle observado al momento de poner a funcionar el modelo, es que la precisión de la predicción aumenta al acercar el producto a la cámara. En el siguiente ejemplo se observa que un paquete de galleta soda es detectado como bebida (figura 8 al estar más alejado, mientras que al colocarlo como un objeto que totaliza el encuadre sobre el eje vertical, su detección como *snack* es correcta (figura 9):

Dicho resultado es consistente con muchas pruebas, como se muestra en las figuras 10 y 11:

El detalle completo se ha puesto a disposición en el video subido a youtube [6].

VI. Análisis de resultados

Al observar la figura 7 se observa que la precisión del modelo para los datos de prueba, así como para los datos de validación anda en un rango cercano. El mismo comportamiento se observa para las pérdidas.

Por otro lado, al observar los resultados obtenidos en las figuras 8 y 9, se nota que la distancia entre los productos y la cámara afecta considerablemente el resultado. Esto se comprende porque al observar el conjunto de prueba, se puede notar que los *snacks* se encuentran a lo alto de toda la imagen. Esto afectará la capacidad de inferencia del modelo con objetos a distintas distancias. El comportamiento observado se corrobora con las figuras 10 y 11.

Además, las imágenes de prueba no contienen un fondo blanco, si no que tienen varios elementos que las diferencias notablemente del conjunto de prueba. Aún más, el vídeo de demostración [6] presenta varios casos en los que el ambiente y la rotación de los objetos es distinta a la del conjunto de datos.

El vídeo permite demostrar que efectivamente se ejecutó el modelo previamente entrenado en un sistema embebido de uso optimizado para aplicaciones de aprendizaje automático. Sin embargo, en el vídeo se observa que existe un retraso perceptible entre la captura de imagen y el procesamiento de esta.

VII. Conclusiones y recomendaciones

Se diseñó una red neuronal capaz de discriminar objetos capturados por medio de una cámara que permite distinguir entre productos alimenticios que son bebidas y *snacks* con una precisión cercana al 85%. Donde dicha red no se encuentra sobre ajustada con respecto a los datos de entrenamiento.

Se ejecutó un modelo de aprendizaje automático basado en una red neuronal convolutiva en una Jetson Nano de 4 GiB. A pesar de que este sistema está optimizado para aplicaciones de

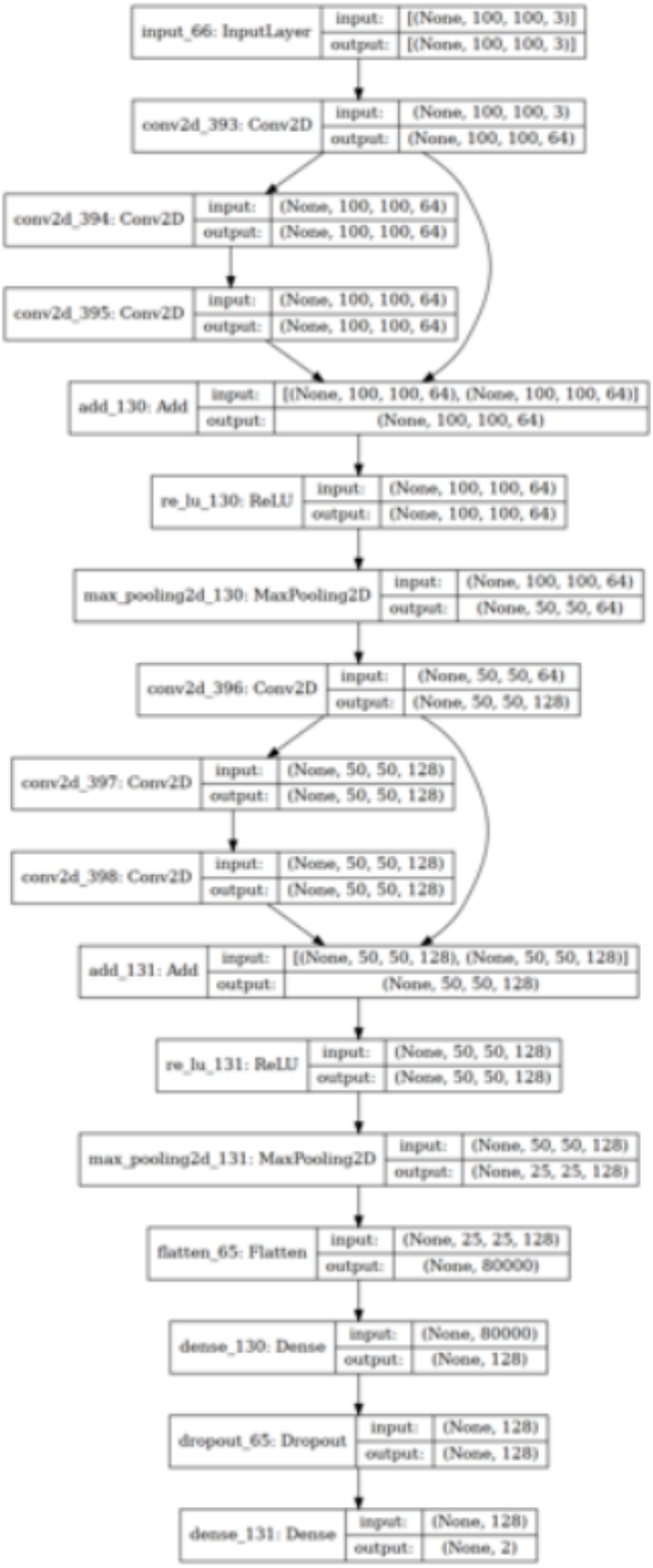

Fig. 5. Arquitectura del modelo.

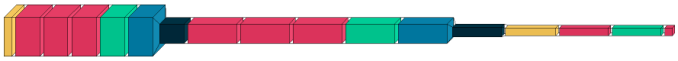

Fig. 6. Las capas del modelo.

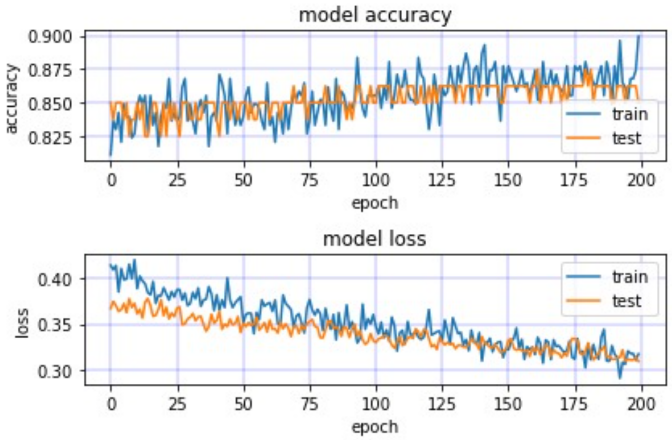

Fig. 7. Precisión y Pérdidas.

Fig. 8. *Snack* detectado como bebida

Fig. 9. *Snack* detectado correctamente

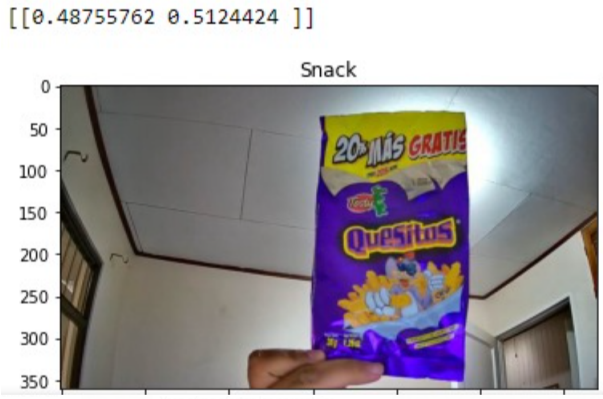

Fig. 10. Quesitos con un margen de detección muy cercano

Fig. 11. Quesitos con un margen de detección considerable

aprendizaje automático, comparte la RAM con el GPU, por lo que se recomienda implementar una partición de intercambio (*swap*)

Referencias

[1] walmart centroamerica. (2021) Entrega domicilio walmart. [Online]. Available: https://centroamerica.walmart.com/1/browse/items

[2] A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[4] geeksforgeeks. (2021) Vgg-16 — cnn model. [Online]. Available: https://www.geeksforgeeks.org/vgg-16-cnn-model/

[5] H. Mujtaba. (2021) Introduction to resnet or residual network. [Online]. Available: https://www.mygreatlearning.com/blog/resnet/

[6] S. Guillén, F. León, and E. Mora. (2021) Clasificación de alimentos - demostración. [Online]. Available: https://www.youtube.com/watch?v=n2-q3D62fkw