

## **CNN**

**FREDDY ALEJANDRO FLOREZ BOHORQUEZ**

**ORLANDO CORREA MARTINEZ**

**LAURA LOPEZ ARBELAEZ**

**SEMESTRE 7**

**PROFESOR**

**INGENIERO ARIEL RUIZ**

**MATERIA**

**INTELIGENCIA ARTIFICIAL**

**UNIVERSIDAD SANTIAGO DE CALI**

**CALI-VALLE**

**2022**

**Taller**

* **Primero:** reconocer los tipos de kernel, sus características y los efectos que alcanza en los procesos de convolución. Por favor revise las características que propone el simulador recomendado.

**R/** Los tipos de kernel son una matriz pequeña que sirve para el reconocimiento de imágenes digitales que estas son utilizadas para el enfoque reconocimiento de bordes y más, esto se logra mediante una formulación matemática para lograr la transformación de la imagen y el kernel escogido.

Estos se caracteriza por usar funciones de kernel lineal, radial, polinomial y gaussiano

para el caso lineal se puede dar una imagen a escala de grises esto debido a que maneja un producto entre dos vectores el cual por medio de una funcion de activacion relu podemos tener con exactitud los contornos de la imagen de interés a su ves esta pueden atender esta necesidad con unos efectos especiales de matrices ya se para enfoque desenfoque o los contornos esto es algo que en escala de grises se debe de tener en cuenta que está dentro de un rango de 0 al 255 esto indicando la opacidad de la imagen y el brillo que esta llega ademas podriamos aplicar estos mismos conceptos en el caso de manejar formato RGB debido que en este tendría un cambio diferente que esté al tener tres colores como lo son rojo, verde y azul tendríamos que pensar en 3 planos y lo mismo en el kernel para cada plano dentro de este caso se realizarán producto punto y se sumarian para tener el resultado en un plano para un píxel y esto tanto para escala de grises como para RGB se va a ir desplazando una posición o depende de como lo haya configurado para reconocer la imagen si la imagen es muy complicada de reconocer no se recomienda dar salto para tener una mejor visión de lo que se quiere captar y esto está ajustado en los pesos mediante el kernel esto manejando el mismo proceso de una red neuronal este proceso se puede llevar a cabo porque se está usando un modelo supervisado.

**Blur (Desenfoque):**

Procesa la imagen digital de manera tal que el paso entre un píxel y sus antecesores sea menos notorio, creando como resultado un desenfoque en la imagen digital resultante.

**Bottom Sobel (Abajo sobel):**

En este tipo de núcleo procesa la imagen digital de tal manera que el resultado muestra que tan abruptamente o suavemente cambia una imagen digital, dando un mayor enfoque a las líneas dando esta perspectiva desde la parte inferior de la imagen, y creando distorsión en medio.

**Emboss (Relieve):**

Como principal característica tiende a crear una sensación de profundidad en la imagen digital, logra su cometido enfatizando los pixeles en determinado sentido dependiendo de la composición de la imagen digital.

**Identy (Identidad):**

Es un filtro que no hace nada, simplemente deja el resultado exactamente igual a la imagen digital original.

**Left Sobel (Izquierda sobel):**

En este kernel procesa la imagen digital de manera tal que el resultado muestra que explosiva o suavizada cambia una imagen digital, dando un mayor enfoque a las líneas dando esta perspectiva desde la parte izquierda de la imagen, contrario al (bottom sobel), crea una imagen resultante bastante diferente a este.

**Outline (contorno):**

En este kernel busca remarcar las diferencias entre píxeles que tengan como cercanos a pixeles con una diferencia de intensidad clara, creando en la imagen resultante pixeles blancos que reemplazarán a estos píxeles “diferentes”. Right sobel.

**(Derecha Sobel):**

En este kernel procesa la imagen digital de manera tal que el resultado muestra cómo de explosividad o ligeramente cambia una imagen digital, dando un mayor enfoque a las líneas dando esta perspectiva desde la parte derecha de la imagen.

**Sharpen (Nitidez):**

En este kernel enfatiza los valores de un píxel respecto a sus anteriores creando así un resultado más vivo o con más contraste.

**Top Sobel (Arribal sobel):**

En este kernel procesa la imagen digital de manera tal que el resultado muestra cómo de abruptamente o suavemente cambia una imagen, dando un mayor enfoque a las líneas dando esta perspectiva desde la parte superior de la imagen.

**REFERENCIAS**

<https://www.youtube.com/watch?v=ySbmdeqR0-4>

<https://aprendeia.com/kernel-maquinas-vectores-de-soporte-clasificacion-regresion/>

<https://setosa.io/ev/image-kernels/>

<https://bryanmed.github.io/kernelsConv/>

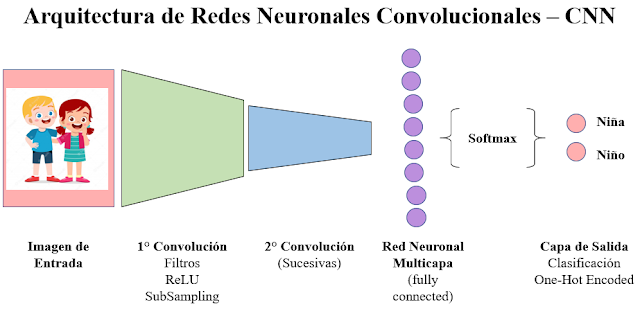
<https://es.wikipedia.org/wiki/N%C3%BAcleo_(procesamiento_digital_de_im%C3%A1genes)>

<https://poloclub.github.io/cnn-explainer/>

* **Segundo:** revisar el simulador de CNN propuesto para:

**1)** Graficar y presentar la arquitectura de una CNN

**R/**



En esta imagen se puede ver el proceso que pasa una imagen al aplicarle un a red neuronal convolucional **CNN** teniendo como entrada una imagen ya sea por los canales de escala de grises o **RGB,** luego de esto pasamos a la parte de la convolución que es allí donde aplicamos el kernel que tiene por parámetros ciertos hiperparametros a causa del entrenamiento y el sesgo obtenido del mismo de la red esto con el objetivo de poder extraer las características requeridas de dicha imagen de interés luego de esto podemos escoger cómo van a seleccionar los hiperparametros para reducir el uso de procesamiento esto partiendo de la necesidad estos son los siguientes padding qué es el tamaño o relleno de la imagen que este me sirve para obtener la imagen actual, otro es el stride, que este me sirve para los saltos que va a realizar el kernel en la imagen digital para el reconocimiento, otro es el de Max pooling que este me sirve para obtener el valor más representativo de una convolución de una imagen además y no menos importante es stacking que este me permite aplicarle a la imagen multiples filtro y así obtener múltiples imágenes resultantes esto a lo ultimo se tiene en cuenta un softmax y por último el resultado de la convolución esto teniendo en cuenta que este proceso se puede llevar a cabo gracias a que es supervisado.

<https://www.youtube.com/watch?v=QLy8v6LL_4A&list=RDCMUCFVF0MpD1lNdU5VL3Pz67Yw&index=3>

<https://www.codingninjas.com/codestudio/library/convolution-layer-padding-stride-and-pooling-in-cnn>

<https://poloclub.github.io/cnn-explainer/>

<https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>

**2)** Definir la aplicación de cada uno de los conceptos y la terminología que involucra el uso y aplicación de una CNN

**R/**

**Imagen digital:**

Una imagen digital o gráfico digital es una representación bidimensional de una imagen a partir de una matriz numérica, frecuentemente en binario. Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial o de un gráfico vectorial.

**Convolución:**

Proceso matemático el cual se obtiene las características de una imagen digital por medio de procesos matemáticos y características asociadas de una red neuronal.

**Planos:**

Estos son los que se obtienen después de realizar un proceso de convolución debido a que se obtiene una matriz resultante.

**Padding:**

Se logra que la imagen digital resultante y la original tengan el mismo tamaño. Ya que el resultado es más pequeño y este se “rellena en el borde” con valores de 0, los cuales en una imagen común y corriente serían un marco negro que no interfiere con el resultado esto para el caso del ejemplo visto en clase.

**Strides:**

Se desplaza un píxel hacia la derecha o hacia abajo durante cada iteración. A cada uno de estos desplazamientos se les conoce como Stride, en la convolución original este valor es igual a 1 y estos Strides o saltos” se pueden modificar estos para crear diferentes resultados, como por ejemplo con Strides de 1, 2, 3 o los necesarios para la función que busquemos o si la imagen es muy detallada no sería bueno darles tantos saltos.

**Max-pooling:**

Permite analizar el contenido de una imagen por regiones o secciones para de esta forma extraer la información más representativa de dicha seccion de la imagen digital. Esto se efectúa identificando el valor más alto entre los saltos.

**Stacking:**

Este es el producto de apilar múltiples imágenes resultantes, las cuales provienen de la convolución con múltiples filtros kernel obteniendo como resultado múltiples matrices resultantes de las convoluciones.

**3)** Describir con exactitud los efectos de cada capa convolucional

**R/**

**Capa de entrada(Input Layer):**

En la capa capa de entrada representa la capa de la izquierda la cual es la entrada a la CNN, esta capa tiene 3 planos ,esto se debe a que usamos imágenes RGB como entrada, los canales correspondientes son rojo, verde, azul estas estaría representado como rojo (255,0,0), verde como (0,255,0), y azul como (0,0,255) además de esto puede usarse un solo plano esto para el caso de imágenes de escala de grises.

**Capas convolucionales (convolutional layers):**

En esta la base de la red neuronal convolucional, ya que estas contienen núcleos aprendidos Pesos **W** esto por medio del proceso de entrenamiento de una red neuronal haciendo uso del backpropagation para entrenar esas características a extraer, los cuales extraen características que distinguen diferentes imágenes entre sí, esta presenta vínculos con las capas anteriores. Esta realiza un producto escalar entre un elemento con un núcleo único y la salida de la neurona, este es el resultado de todos los resultados intermedios sumados juntos con el sesgo aprendido.

**Capas de agrupación (pooling layers):**

En este parte del proceso permite disminuir o acotar la dimensión de la red, para de esta forma reducir los cálculos generales de la red neuronal.

**Capa plana (flatten layer):**

En este paso se usa softmax previamente de la convolución final convierte una capa tridimensional en la red en un vector unidimensional para adaptarse a la entrada de una capa completamente conectada para su clasificación, en esta se puede trabajar una función de predicción.

**4)** ¿Qué conclusiones tiene usted de los Hyperparameters padding, kernel size y el stride que refiere el simulador?

**R/**

Este procedimiento es necesarios para poder crear las convoluciones siendo la margen un factor que ayuda a mantener los parámetros originales de la imagen digital manejando los bordes en blanco para el resultado final, el tamaño del kernel es utilizado para la clasificación de las imágenes y este suele ser de pequeño tamaño para una mayor extracción de valores, estos pasos son también el aprendizaje de la red ya que a menor valor en el paso se logran aprender más funciones debido a que se evalúan más datos, con grandes saltos se reduce notablemente la información recolectada y esta puede ocasionar no obtener el resultado esperado por esta red.

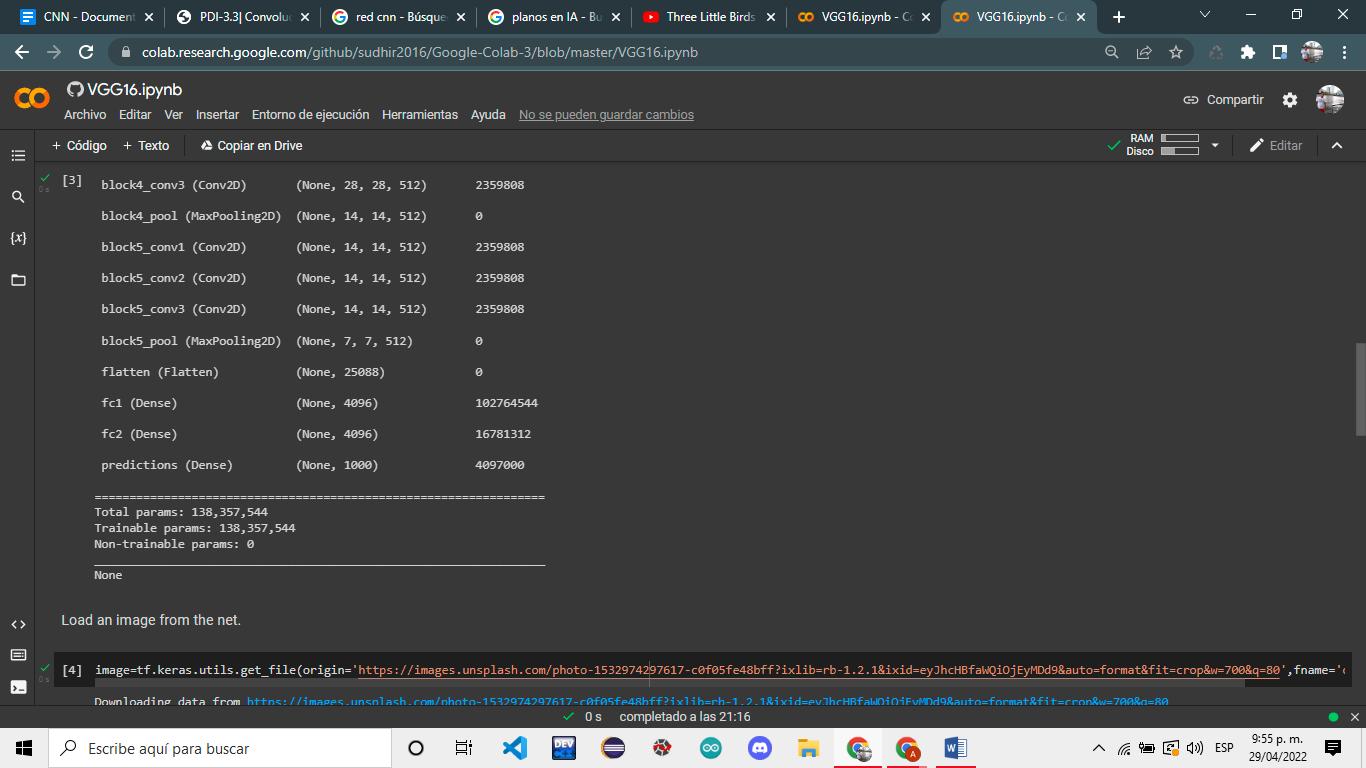
Lo anterior se puede ajustar de acorde a la imagen a detectar y su complejidad visual.

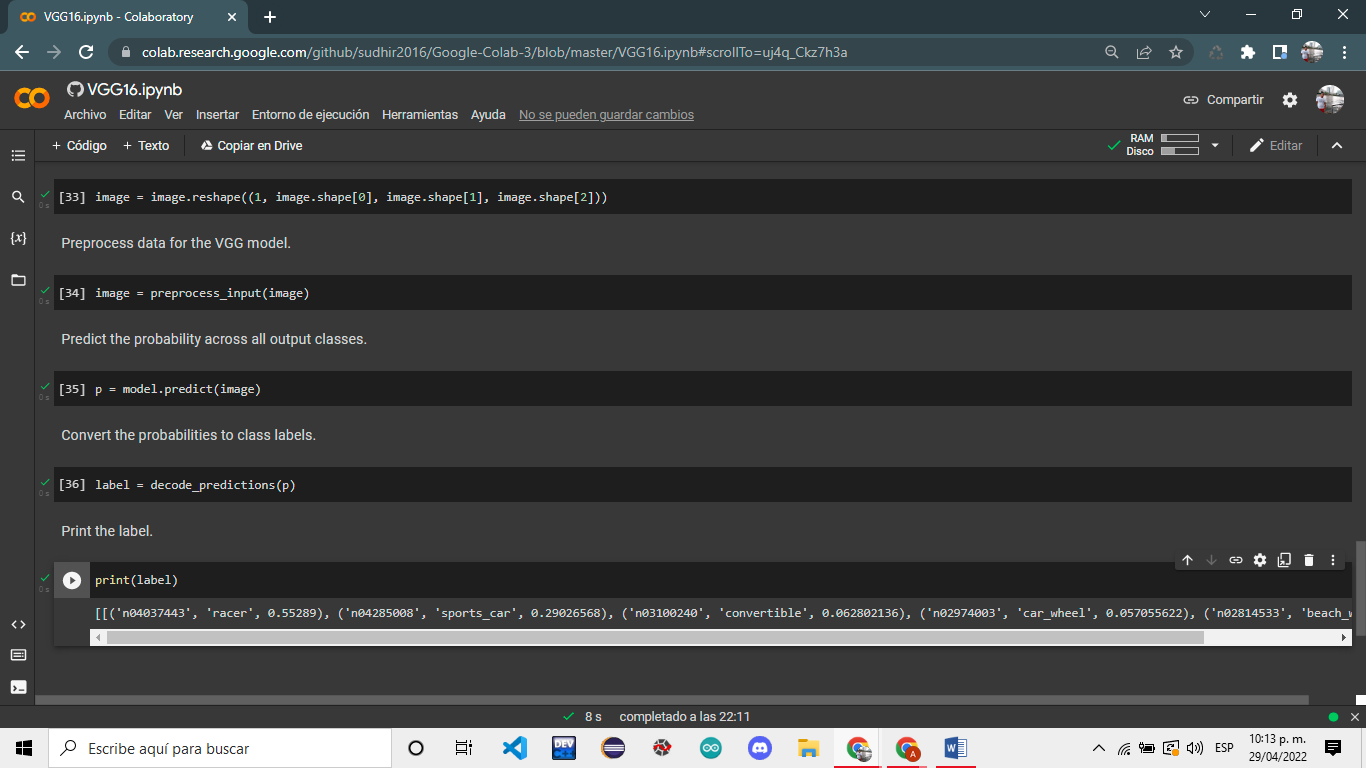
* **Tercero:** utilizar CNN tipo VGG16 o VGG19 para aplicar el reconocimiento de una imagen (propia o del contexto), obtenga resultados del modelo, interprete la salida del modelo e imprima el resultado.Ver ejemplo:

**R/**

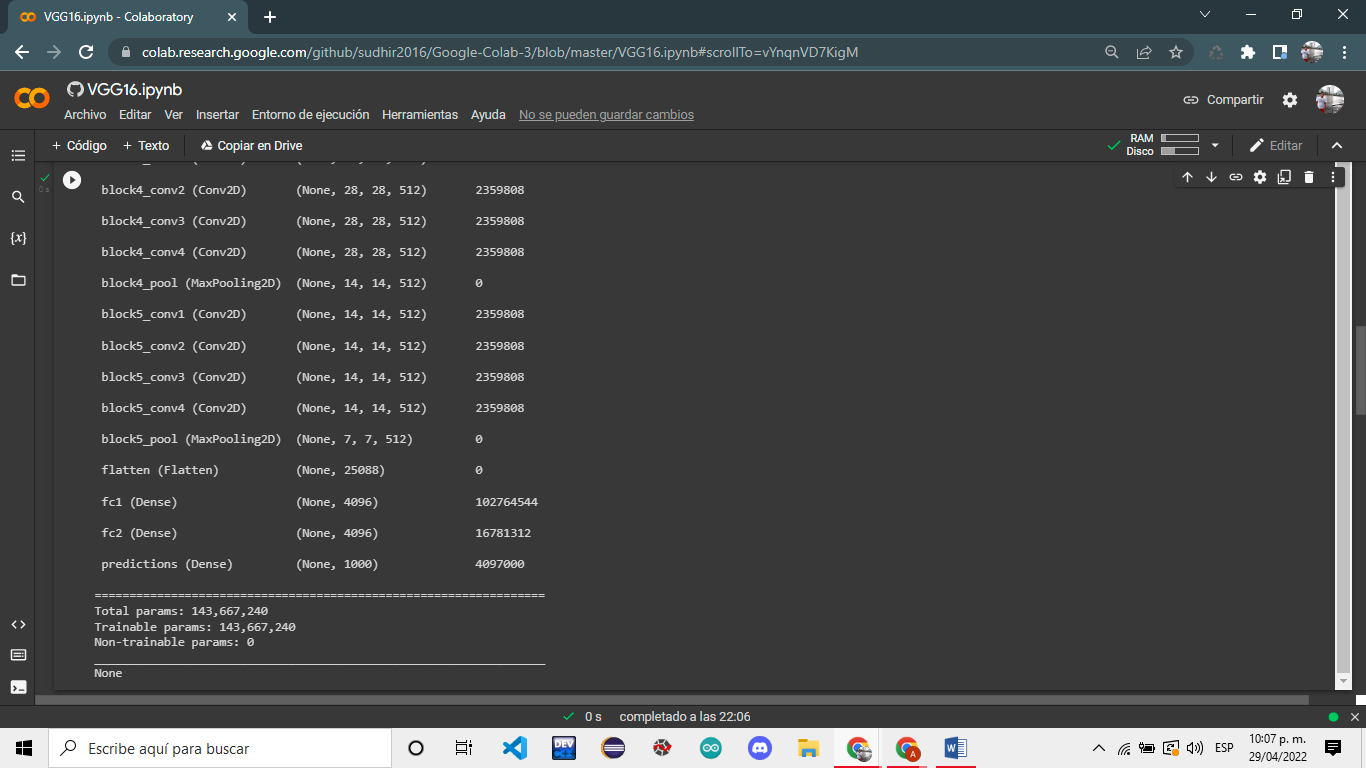
Los modelos VGG-16 y VGG-19 son redes neuronales convolucionales las cuales se encargan de procesar millones de imágenes, estas cumplen con la misma función, pero se diferencian en algunos aspectos como por ejemplo: su estructura, el modelo VGG-16 tiene 16 capas de profundidad, a diferencia del modelo VGG-19 que tiene 19 capas de profundidad, no obstante, también se diferencian en la cantidad de parámetros que se pueden entrenar, dependiendo de los tiempos de entrenamiento.

**VGG-16**

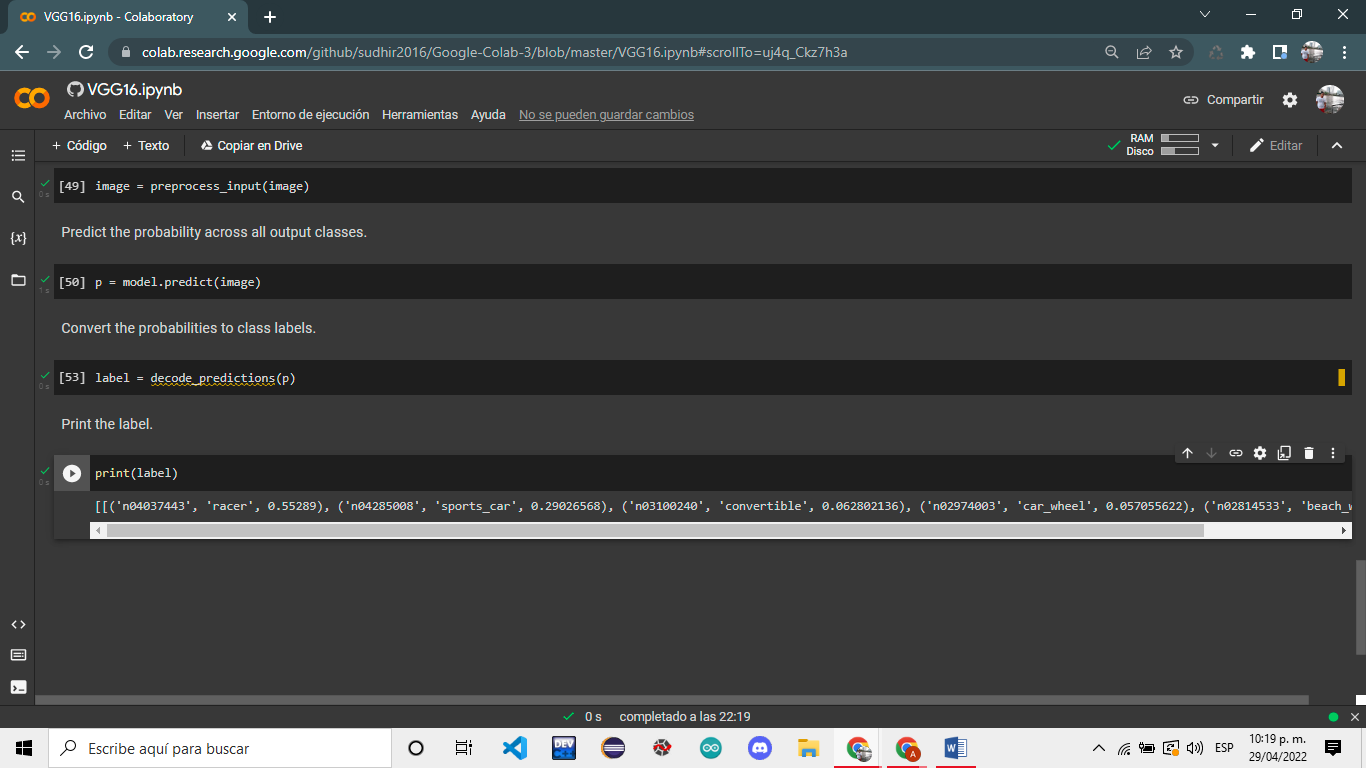


****

**VGG-19**

****

Al imprimir el Label de ambos modelos se observa que ambos es distinta, ya que el modelo VGG-19 requiere más parámetros para poder filtrar las imágenes digitales.

****

**Ejemplo:** <https://colab.research.google.com/drive/1wtxr755-ZTVd31ukJ3G6qT6RMxvxPGu5?usp=sharing>

**Keras:**

<https://keras.io/api/applications/vgg/>

**¿Qué es una CNN VGG16?**

su estructura, el modelo VGG-16 tiene 16 capas de profundidad.

**Arquitectura:** <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>

**Interpretación de una VGG16:** [https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16- f2d63849f615#:~:text=VGG16%2C%20as%20the%20name%20implieslarge%20number%20of%20hyper%2Dparameters](https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-) .