Clasificador de emojis

Daniel Sebastian Preciado Vega
Diego Gutierrez Contreras
Gerardo Garcia Valle

Propuesta

Nuestra intención es realizar un programa que al recibir una imagen de un emoji como entrada pueda **identificarlo** y muestre su nombre en pantalla.

Dataset

Nuestro dataset se compone de alrededor de **22,000 imágenes** de emojis de distintos diseños, como los de Google, Facebook, iOS, entre otros.

Modelo

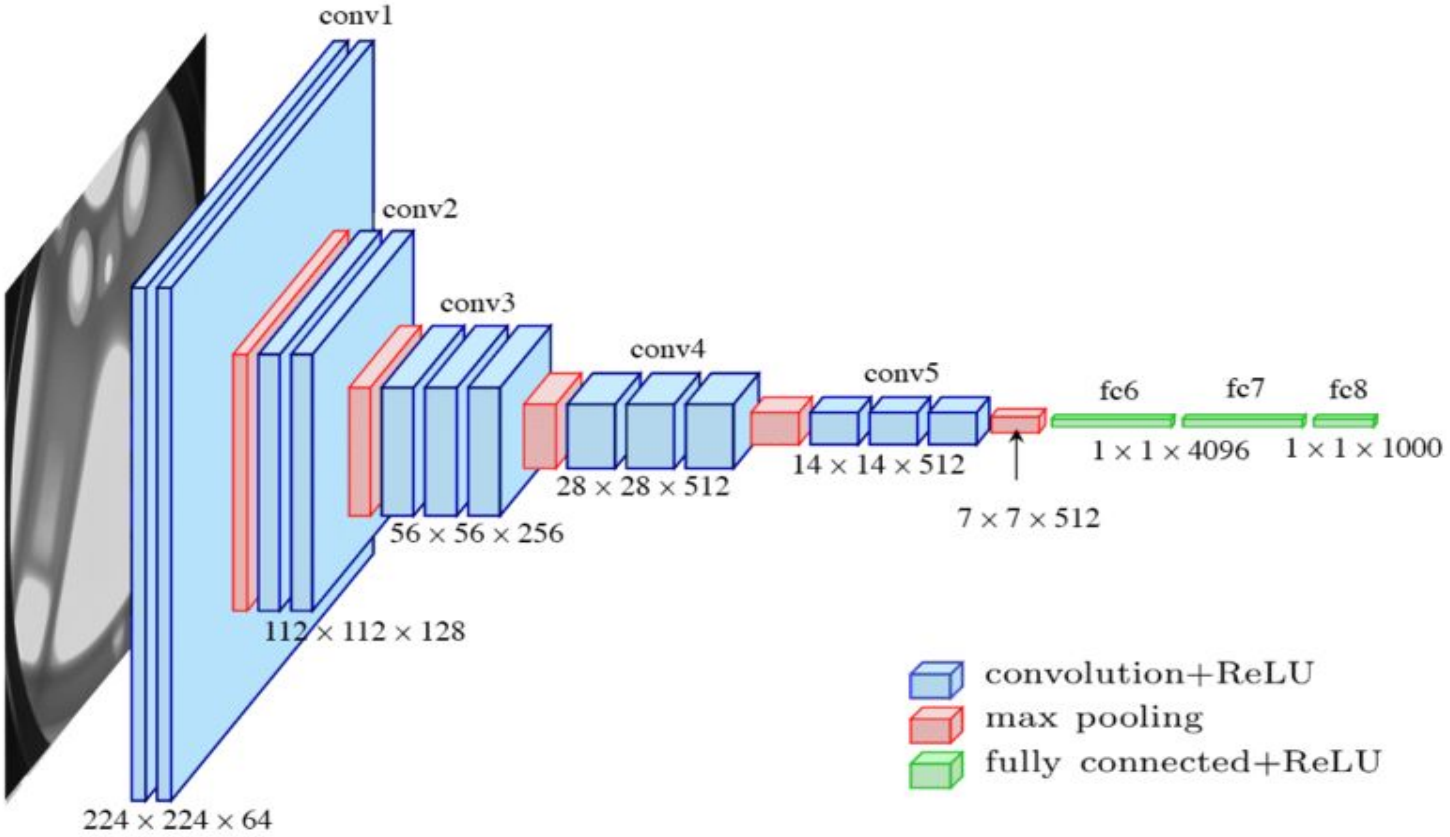
Para nuestra red neuronal optamos por un modelo **VGGNet** por las siguientes razones:

- Enfoque en el rendimiento
- Capacidad de identificar un gran número de categorías
- Corto tiempo de entrenamiento

VGG-16

Visual Geometry Group o VGG, es un modelo de red neuronal con **16 capas** convolucionales que destaca por poder clasificar objetos en un gran número de categorías.

Estructura


conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

Modelo

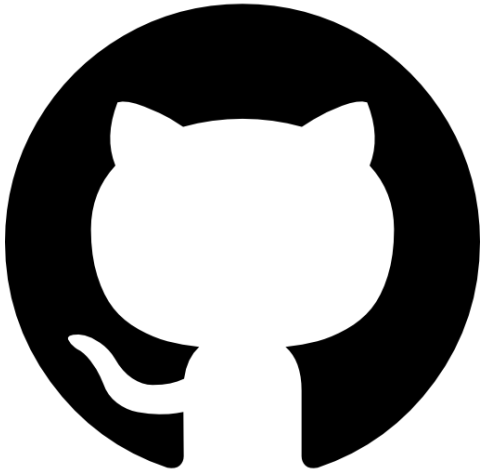
Este modelo de red neuronal se compone de la siguiente manera:

- 13 capas de convolución.
- 3 capas completamente conectadas.
- 5 capas de Max pulling.

Se utiliza la función de activación ReLU en las diferentes capas de la red.

Implementación

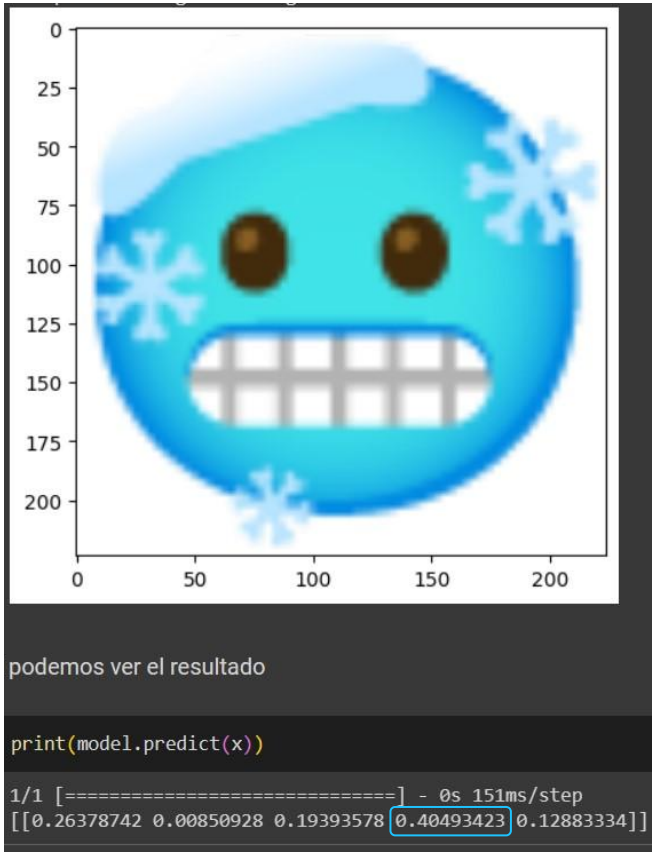
Todas las pruebas realizadas se llevaron a cabo mediante la plataforma de Google Colab, haciendo uso de un repositorio de GitHub para almacenar todos los archivos correspondientes.



Entrenamiento

Inicialmente se llevaron a cabo pruebas con entrenamientos de **15** eras.

- Se utilizaron **30 imágenes** para entrenar al modelo en la clasificación de 5 categorías.
- Obteniendo resultados de certeza entre un **30%** y **40%**.



Problemas

Gran uso de memoria con una muestra de imágenes muy pequeña.

Nuevo acercamiento

Se **modificó la propuesta original** del programa para reducir el volumen de imágenes requeridas durante el entrenamiento.

- Solo se clasificaran los emojis en 3 categorías: **Felices**, **Tristes** y **Enojados**.

Resultados

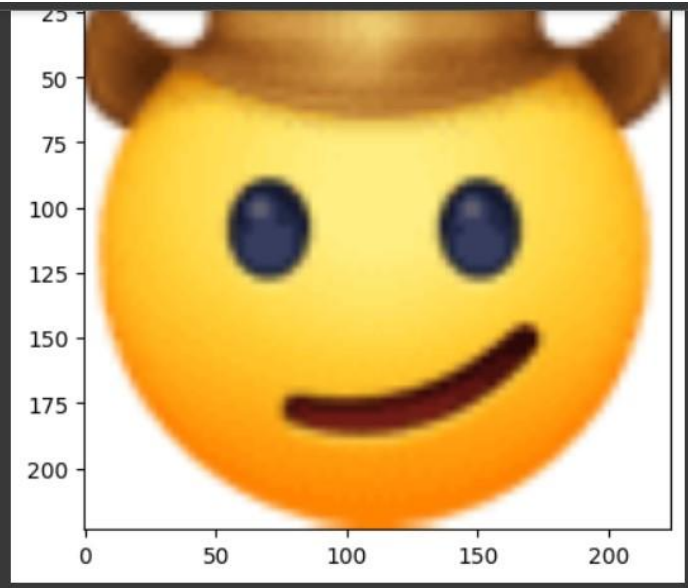
Tras realizar los cambios mencionados anteriormente, se obtuvieron **resultados variados** al modificar parámetros como el número de imágenes por lote o la división de imágenes para entrenar/evaluar.

Aqui podemos ver el resultado

```
[16] resultado = model.predict(x)

     print(tipos_emoji[1] + ": \t" + str(format(resultado[0][1],
     print(tipos_emoji[2] + ": \t" + str(format(resultado[0][2],
     print(tipos_emoji[0] + ": \t" + str(format(resultado[0][0],
```

```
1/1 [==============================] - 1s 842ms/step
1_Felicidad:    0.24%
2_Tristeza:     0.18%
3_Enojo:        0.58%
```

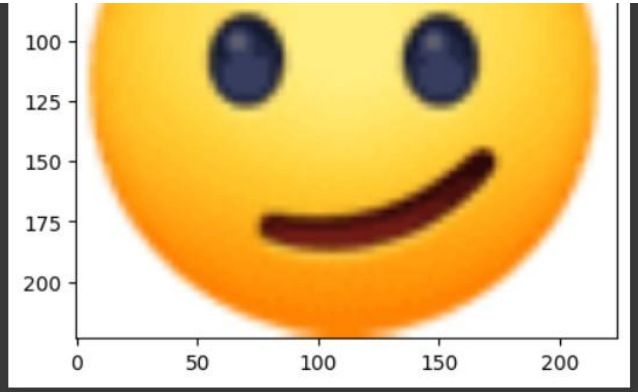


Aqui podemos ver el resultado

```
[16] resultado = model.predict(x)

     print(tipos_emoji[1] + ": \t" + str(format(resultado[0][1],
     print(tipos_emoji[2] + ": \t" + str(format(resultado[0][2],
     print(tipos_emoji[0] + ": \t" + str(format(resultado[0][0],
```

```
1/1 [==============================] - 1s 661ms/step
1_Felicidad:    0.60%
2_Tristeza:     0.17%
3_Enojo:        0.23%
```



Aqui podemos ver el resultado

```
[16] resultado = model.predict(x)

     print(tipos_emoji[1] + ": \t" + str(format(resultado[0][1],'
     print(tipos_emoji[2] + ": \t" + str(format(resultado[0][2],'
     print(tipos_emoji[0] + ": \t" + str(format(resultado[0][0],'

     1/1 [==============================] - 1s 864ms/step
     1_Felicidad:    0.98%
     2_Tristeza:     0.00%
     3_Enojo:        0.02%
```

Hasta que finalmente se obtuvieron resultados aceptables.

CRONOGRAMA DE ACTIVIDADES

Actividades	03-may	06-may	08-may	10-may	12-may	17-may	18-may	23-may
Elección del tema	Todos							
Creación del repositorio		Daniel						
Búsqueda del dataSet			Daniel					
Definición del logo				Todos				
Elección de la arquitectura				Todos	Todos			
Presentación de avances						Gerardo, Daniel		
Limpieza del dataSet						Gerardo	Diego	
Codificación						Daniel	Todos	
Elaboración del reporte								Gerado, Diego
Elaboración de las diapositivas								Daniel