07MIAR29 - Redes Neuronales y Deep Learning

Proyecto de programación "Deep Vision in classification tasks"

Integrantes: Anthony Playmith Sanchez, Steven Mena Chavez y David Pozo Spin

Estrategia 1: Entrenar desde cero o *from scratch*

1. Cargar del dataset

In [ ]:

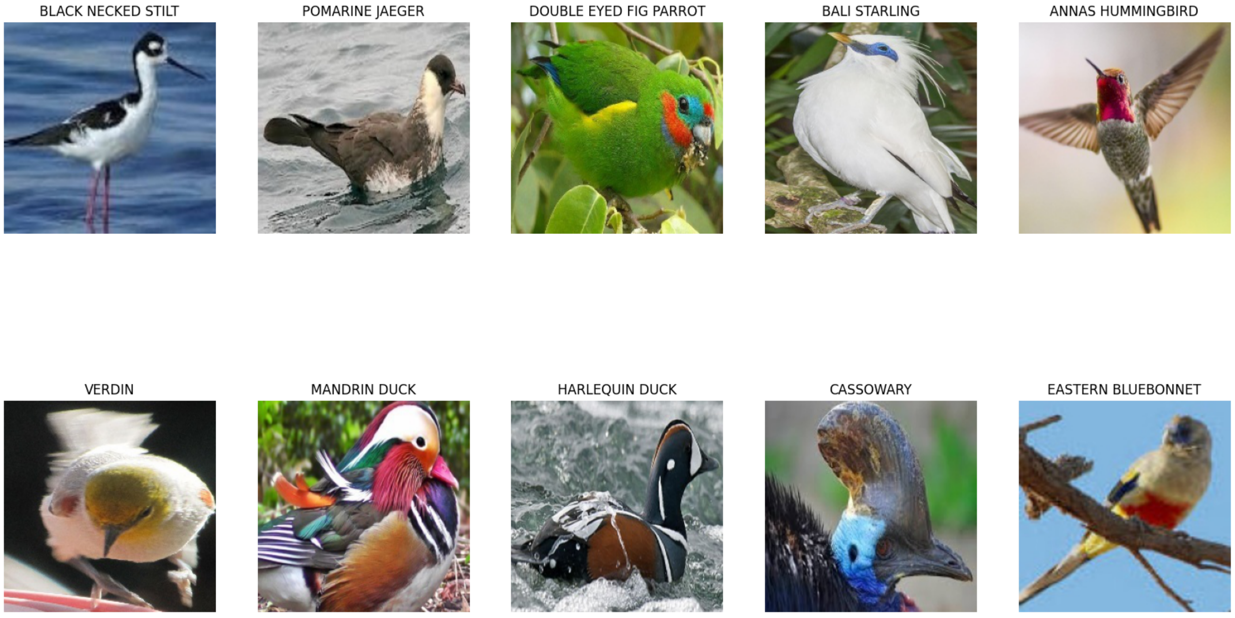
```
!kaggle datasets download -d gpiosenka/100-bird-species
```

2. Inspección del conjunto de datos

In [ ]:

```
data_generator = ImageDataGenerator( )
train_images = data_generator.flow_from_directory(BASE_FOLDER+'/train')
train_files = train_images.filepaths
train_labels = train_images.classes

class_name = list(train_images.class_indices.keys()) # se obtiene los nombres d
print("Exiten: {0} clases".format(len(class_name)))  # corroboramos la cantidad

length_images = len(train_labels)
print("Exiten: {0} elementos de entrenamiento".format(length_images))  # corrob
sample_size = min(length_images, 10) #se escoge mostrar un número menor o igual
sample_images = random.sample(range(length_images), sample_size)

for i in range(sample_size):
    plt.rcParams['figure.figsize'] = (20, 30)
    img = plt.imread(train_files[sample_images[i]])
    plt.subplot(5, 5, i+1)
    plt.title(class_name[train_labels[sample_images[i]]]) # se obtiene el nombr
    plt.imshow(img)
    plt.axis('off')
plt.show()
```

```
Found 84635 images belonging to 525 classes.
Exiten: 525 clases
Exiten: 84635 elementos de entrenamiento
```



3. Acondicionamiento del conjunto de datos

- Realizaremos un escalamiento de las imágenes (0-1)
- Se realiza un redimensionamiento de las imágenes a 150x150x3

- Debido a que le dataset ya nos entrega agrupemientos de datos de train, test y valid no es necesario realizar un proceso de HoldOut (partición interna de entrenamiento y validación)
- Se utilizarán lotes de 1024 imágenes para el poder realizar el entrenamiento de la red neuronal.
- Se utilizarán lotes de 64 imágenes para validación
- Los lotes se escogieron debido a la cantidad de datos correspondiente
- Para todo esto usaremos un Generator de Imágenes
- En primera instancia no usaemos (Data Augmentation) para ver como se comporta la red que entrenaremos, en caso de existir overfitting se realizar'a una nueva prueba con data augmentation.

In [ ]:

```
IMG_WIDTH = 150 # 224 original
IMG_HEIGHT = 150 # 224 original
BATCH_SIZE_TRAIN = 1024
BATCH_SIZE_VALID=64
```

In [ ]:

```
#Declaración de rutas relativas de los folders donde se encuentran las imagenes
DIRECTORY_TRAIN = BASE_FOLDER+'/train/'
DIRECTORY_VALID = BASE_FOLDER+'/valid/'
DIRECTORY_TEST = BASE_FOLDER+'/test/'
```

In [ ]:

```
datagen = ImageDataGenerator(rescale=1./255)
train_generator = datagen.flow_from_directory(directory=DIRECTORY_TRAIN,
                                                     target_size=(IMG_WIDTH, IMG
                                                     batch_size=BATCH_SIZE_TRAIN
                                                     class_mode='categorical')
validation_generator = datagen.flow_from_directory(directory=DIRECTORY_VALID,
                                                     target_size=(IMG_WIDTH, IMG
                                                     batch_size=BATCH_SIZE_VALID
                                                     class_mode='categorical')
```

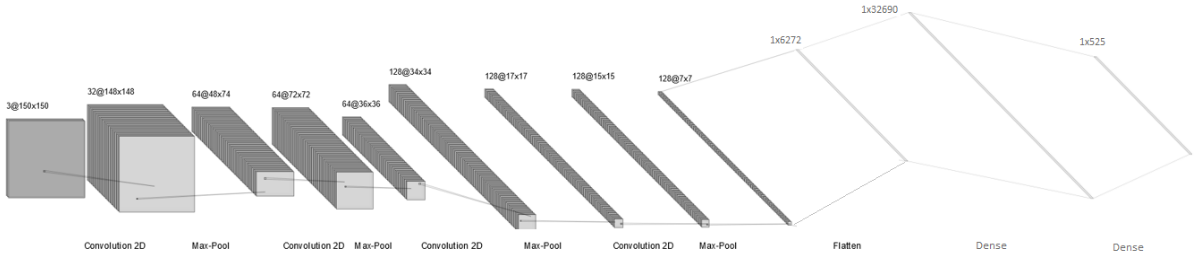
4. Desarrollo de la arquitectura de red neuronal y entrenamiento de la red

Esta es la primera red CNN, en la cual en el diseño inicial se han tomado las siguientes consideraciones.

1. Un Base Model de 4 Capas convolucionales
2. En cada capa se usa filtros de 3x3 y función de activación ReLU
3. En cada capa se aplica un maxpooling de 2, con el objetivo de reducir los parámetros entrenables en cada capa y obtener con esto un menor coste computacional.
4. En el Top Model se decidio por tener una capa oculta de 32960 neuronas
5. En la salida se tienen 525 neuronas correspondientes a las 525 especies.
6. La función de perdida es SoftMax, la cual cálcula las probabilidas de cada una de las 525 posibles salidas.

In [ ]:

```
model = models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',
                        input_shape=(150,150,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))

#TOP MODEL
model.add(layers.Flatten())
model.add(layers.Dense(32960,activation='relu'))
model.add(layers.Dense(525,activation='softmax'))
```



In [ ]:

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 148, 148, 32)      896

 max_pooling2d_4 (MaxPooling  (None, 74, 74, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_6 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_6 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 conv2d_7 (Conv2D)           (None, 15, 15, 128)       147584

 max_pooling2d_7 (MaxPooling  (None, 7, 7, 128)        0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 32960)             206758080

 dense_3 (Dense)             (None, 525)               17304525

=================================================================
Total params: 224,303,437
Trainable params: 224,303,437
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
#Realizamos el compilado de nuestra primera red
model.compile(loss='categorical_crossentropy',
              optimizer=optimizers.RMSprop(learning_rate=1e-4),
              metrics=['acc'])
```

In [ ]:

```
#Entrenando el primer modelo
history = model.fit(
    train_generator,
    steps_per_epoch = train_generator.n//train_generator.batch_size,
    epochs = 40,
    validation_data = validation_generator,
    validation_steps = validation_generator.n//validation_generator.batch_size
)
```

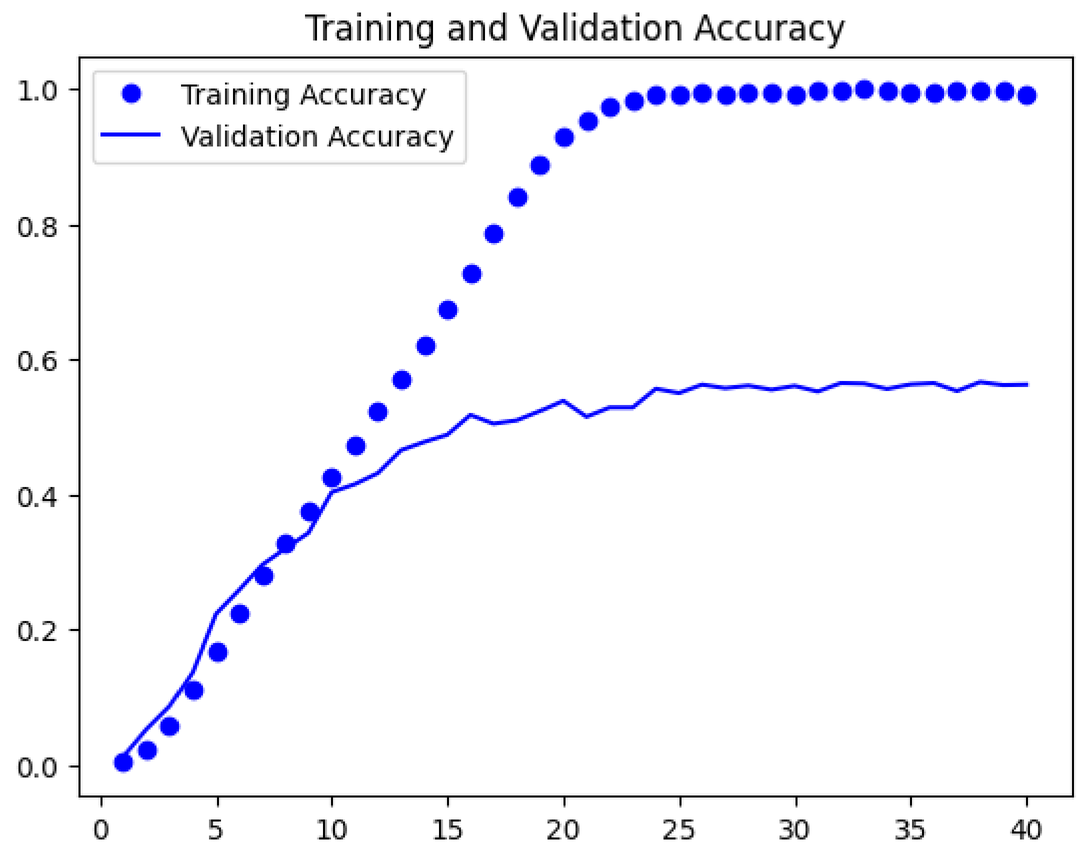
```
Epoch 1/40
82/82 [==============================] - 241s 3s/step - loss: 6.2275 - acc: 0.
0041 - val_loss: 6.0808 - val_acc: 0.0122
Epoch 2/40
82/82 [==============================] - 207s 3s/step - loss: 5.7634 - acc: 0.
0228 - val_loss: 5.3953 - val_acc: 0.0530
Epoch 3/40
82/82 [==============================] - 195s 2s/step - loss: 5.2404 - acc: 0.
0594 - val_loss: 4.9319 - val_acc: 0.0873
Epoch 4/40
82/82 [==============================] - 193s 2s/step - loss: 4.7933 - acc: 0.
1110 - val_loss: 4.5733 - val_acc: 0.1376
Epoch 5/40
82/82 [==============================] - 198s 2s/step - loss: 4.3545 - acc: 0.
1671 - val_loss: 3.9882 - val_acc: 0.2233
Epoch 6/40
82/82 [==============================] - 191s 2s/step - loss: 3.9705 - acc: 0.
2236 - val_loss: 3.6803 - val_acc: 0.2588
Epoch 7/40
82/82 [==============================] - 191s 2s/step - loss: 3.6172 - acc: 0.
2798 - val_loss: 3.4543 - val_acc: 0.2961
Epoch 8/40
82/82 [==============================] - 190s 2s/step - loss: 3.3097 - acc: 0.
3287 - val_loss: 3.3178 - val_acc: 0.3201
Epoch 9/40
82/82 [==============================] - 195s 2s/step - loss: 3.0200 - acc: 0.
3769 - val_loss: 3.1237 - val_acc: 0.3434
Epoch 10/40
82/82 [==============================] - 200s 2s/step - loss: 2.7383 - acc: 0.
4253 - val_loss: 2.8546 - val_acc: 0.4036
Epoch 11/40
82/82 [==============================] - 196s 2s/step - loss: 2.4709 - acc: 0.
4726 - val_loss: 2.7672 - val_acc: 0.4154
Epoch 12/40
82/82 [==============================] - 191s 2s/step - loss: 2.2007 - acc: 0.
5226 - val_loss: 2.7385 - val_acc: 0.4318
Epoch 13/40
82/82 [==============================] - 195s 2s/step - loss: 1.9464 - acc: 0.
5704 - val_loss: 2.5620 - val_acc: 0.4657
Epoch 14/40
82/82 [==============================] - 191s 2s/step - loss: 1.6818 - acc: 0.
6228 - val_loss: 2.5016 - val_acc: 0.4779
Epoch 15/40
82/82 [==============================] - 196s 2s/step - loss: 1.4232 - acc: 0.
6743 - val_loss: 2.4174 - val_acc: 0.4886
Epoch 16/40
82/82 [==============================] - 196s 2s/step - loss: 1.1733 - acc: 0.
7290 - val_loss: 2.3119 - val_acc: 0.5179
Epoch 17/40
82/82 [==============================] - 195s 2s/step - loss: 0.9120 - acc: 0.
7867 - val_loss: 2.4681 - val_acc: 0.5050
Epoch 18/40
82/82 [==============================] - 198s 2s/step - loss: 0.6874 - acc: 0.
8401 - val_loss: 2.5150 - val_acc: 0.5099
Epoch 19/40
82/82 [==============================] - 198s 2s/step - loss: 0.4820 - acc: 0.
8891 - val_loss: 2.4432 - val_acc: 0.5236
Epoch 20/40
82/82 [==============================] - 200s 2s/step - loss: 0.3177 - acc: 0.
9308 - val_loss: 2.5085 - val_acc: 0.5389
Epoch 21/40
82/82 [==============================] - 200s 2s/step - loss: 0.2128 - acc: 0.
9539 - val_loss: 2.7357 - val_acc: 0.5152
Epoch 22/40
82/82 [==============================] - 199s 2s/step - loss: 0.1313 - acc: 0.
9735 - val_loss: 2.8764 - val_acc: 0.5290
Epoch 23/40
82/82 [==============================] - 195s 2s/step - loss: 0.1007 - acc: 0.
9813 - val_loss: 2.9224 - val_acc: 0.5290
Epoch 24/40
82/82 [==============================] - 194s 2s/step - loss: 0.0531 - acc: 0.
9906 - val_loss: 2.7585 - val_acc: 0.5568
Epoch 25/40
82/82 [==============================] - 194s 2s/step - loss: 0.0521 - acc: 0.
9903 - val_loss: 2.8952 - val_acc: 0.5503
Epoch 26/40
82/82 [==============================] - 195s 2s/step - loss: 0.0245 - acc: 0.
9958 - val_loss: 2.9577 - val_acc: 0.5629
```

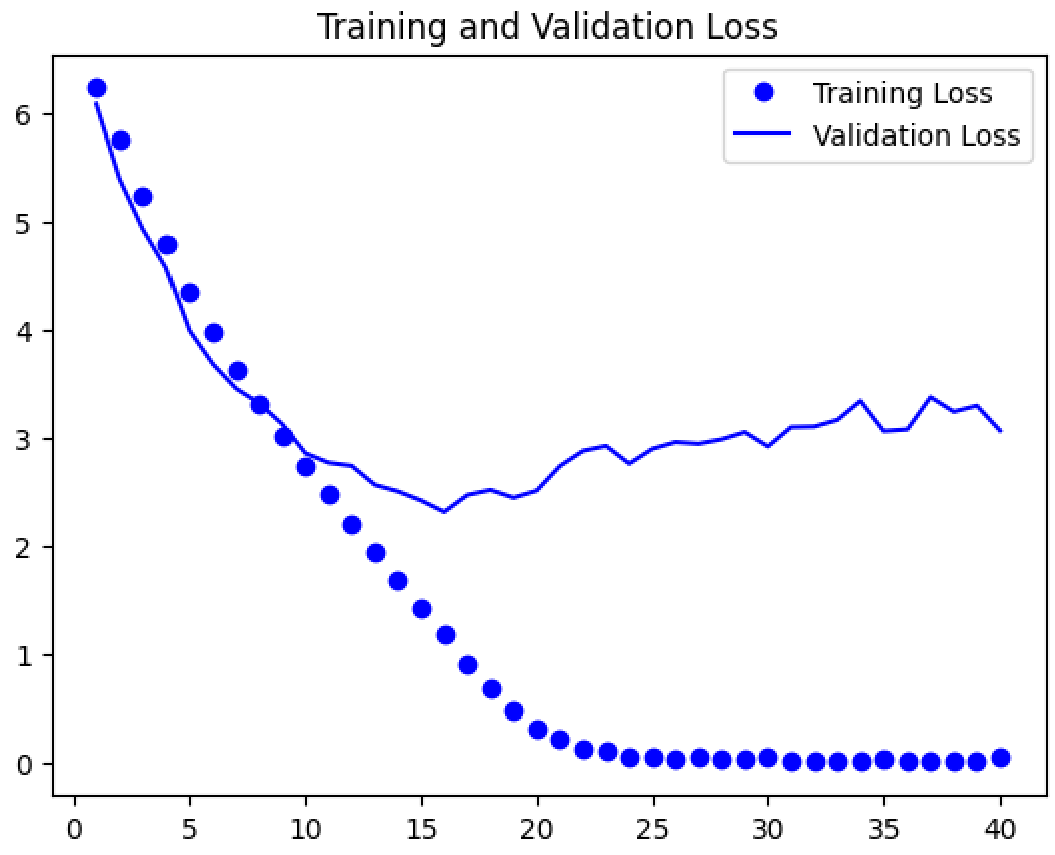
```
Epoch 27/40
82/82 [==============================] - 198s 2s/step - loss: 0.0480 - acc: 0.
9912 - val_loss: 2.9425 - val_acc: 0.5575
Epoch 28/40
82/82 [==============================] - 198s 2s/step - loss: 0.0324 - acc: 0.
9946 - val_loss: 2.9837 - val_acc: 0.5614
Epoch 29/40
82/82 [==============================] - 198s 2s/step - loss: 0.0281 - acc: 0.
9940 - val_loss: 3.0502 - val_acc: 0.5553
Epoch 30/40
82/82 [==============================] - 198s 2s/step - loss: 0.0433 - acc: 0.
9915 - val_loss: 2.9172 - val_acc: 0.5610
Epoch 31/40
82/82 [==============================] - 198s 2s/step - loss: 0.0124 - acc: 0.
9981 - val_loss: 3.0966 - val_acc: 0.5526
Epoch 32/40
82/82 [==============================] - 194s 2s/step - loss: 0.0125 - acc: 0.
9981 - val_loss: 3.1014 - val_acc: 0.5652
Epoch 33/40
82/82 [==============================] - 194s 2s/step - loss: 0.0069 - acc: 0.
9991 - val_loss: 3.1636 - val_acc: 0.5644
Epoch 34/40
82/82 [==============================] - 194s 2s/step - loss: 0.0132 - acc: 0.
9976 - val_loss: 3.3397 - val_acc: 0.5564
Epoch 35/40
82/82 [==============================] - 194s 2s/step - loss: 0.0264 - acc: 0.
9954 - val_loss: 3.0601 - val_acc: 0.5633
Epoch 36/40
82/82 [==============================] - 194s 2s/step - loss: 0.0217 - acc: 0.
9959 - val_loss: 3.0758 - val_acc: 0.5652
Epoch 37/40
82/82 [==============================] - 197s 2s/step - loss: 0.0057 - acc: 0.
9989 - val_loss: 3.3747 - val_acc: 0.5534
Epoch 38/40
82/82 [==============================] - 196s 2s/step - loss: 0.0063 - acc: 0.
9990 - val_loss: 3.2393 - val_acc: 0.5667
Epoch 39/40
82/82 [==============================] - 202s 2s/step - loss: 0.0188 - acc: 0.
9969 - val_loss: 3.2948 - val_acc: 0.5621
Epoch 40/40
82/82 [==============================] - 198s 2s/step - loss: 0.0519 - acc: 0.
9917 - val_loss: 3.0644 - val_acc: 0.5629
```

5. Monitorización del proceso de entrenamiento para la toma de decisiones

In [ ]:

```python
import matplotlib.pyplot as plt

acc = history['acc']
val_acc = history['val_acc']
loss = history['loss']
val_loss = history['val_loss']
epochs = range(1,len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training Accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training Loss')
plt.plot(epochs, val_loss, 'b', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.legend()
plt.rcParams['figure.figsize'] = (20, 30)
plt.show()
```





Como se puede apreciar existe un over-fiting en el modelamiento.

Debido a que la gráfica correspondiente al los datos de validación no sigue el compartamiento de la gráfica de los datos de entrenamiento.

Para poder asegurar que nuestro entrenamiento del modelo es apropiado los valores de precisión y función de perdida deben ser similares para los dataset entrenamiento y validación. Además la función de perdida debe tener un valor muy cercano a cero y la precisión deberá tener un valor muy cercano a 1.

6. Evaluación del modelo predictivo y planteamiento de la siguiente prueba experimental

- Como se puede observar en los gráficos de pérdidas y exactitud, el modelo se encuentra sobreajustado. El modelo no esta generalizando bien.
- Evaluaremos el comportamiento del modelo más afondo, con los datos de test.

In [ ]:

```
# evaluamos y observamos el comportamiento de todos los datos de test
# predichos por el modelo generado
DIRECTORY_TEST = BASE_FOLDER+'/test/'
datagen = ImageDataGenerator(rescale=1./255)
test_generator = datagen.flow_from_directory(directory=DIRECTORY_TEST,
                                             target_size=(IMG_WIDTH, IMG
                                             batch_size=1,
                                             class_mode='categorical',
                                             shuffle=False)


test_labels = test_generator.classes
test_class_name = list(test_generator.class_indices.keys()) # se obtiene los no
```

In [ ]:

```
Batch_test=8
test_generator = datagen.flow_from_directory(directory=DIRECTORY_TEST,
                                             target_size=(IMG_WIDTH, IMG
                                             batch_size=Batch_test,
                                             class_mode='categorical',
                                             shuffle=True)
test_class_name = list(test_generator.class_indices.keys()) # se obtiene los no
def predict_one(model):
    image_batch, classes_batch = next(test_generator)
    predicted_batch = model.predict(image_batch)
    for k in range(0,image_batch.shape[0]):
      image = image_batch[k]
      pred = predicted_batch[k] # 525 valores de predicción
      the_pred = np.argmax(pred) # se busca el ín dice de la predicción con el
      predicted = files[the_pred]
      val_pred = max(pred) #se tiene el valor más alto de la predicción
      the_class = np.argmax(classes_batch[k])
      value = files[np.argmax(classes_batch[k])]
      plt.rcParams['figure.figsize'] = (20, 30)
      plt.subplot(4, 4, k+1)
      isTrue = (the_pred == the_class)
      plt.title(str(isTrue) + ' - class: ' + value + ' - ' + '\n predicted: ' +
      plt.imshow(image)
      plt.tight_layout()
predict_one(model)
```

7. Re-acondicionamiento del conjunto de datos

- Luego de visualizar en nuestro primer modelo Over-fitting, usaremos varias técnicas para reducirlo.

- Para evitar el sobreajuste se usará como técnica principal el **data augmentation** para los datos de entrada. El cual nos permitirá crear imagenes sintéticas para que nuestro modelo tenga mayor datos de entrada.

In [ ]:

```
DIRECTORY_TEST = BASE_FOLDER+'/test/'
BATCH_SIZE_TRAIN = 256  # para esta nueva arquitectura se disminuye el batch, p
                        # size debido a que hay problemas de memoria con el bat
BATCH_SIZE_VALID=64
IMG_WIDTH = 150 # 224 original
IMG_HEIGHT = 150 # 224 original
```

In [ ]:

```
#Declarando la clase que nos permitirá realizar el data augmentation
datagen_2 = ImageDataGenerator(rescale=1./255,
                                    rotation_range = 15,
                                    width_shift_range = 0.2,
                                    height_shift_range = 0.2,
                                    shear_range = 0.2,
                                    zoom_range = 0.2,
                                    horizontal_flip = True,
                                    fill_mode = 'nearest')


train_generator = datagen_2.flow_from_directory(directory=DIRECTORY_TRAIN,
                                             target_size=(IMG_WIDTH, IMG
                                             batch_size=BATCH_SIZE_TRAIN
                                             class_mode='categorical')

validation_generator = datagen_2.flow_from_directory(directory=DIRECTORY_VALID,
                                             target_size=(IMG_WIDTH, IMG
                                             batch_size=BATCH_SIZE_VALID
                                             class_mode='categorical')
test_generator = datagen_2.flow_from_directory(directory=DIRECTORY_TEST,
                                             target_size=(IMG_WIDTH, IMG
                                             batch_size=1,
                                             class_mode='categorical')
```

8. Desarrollo de la nueva arquitectura de red neuronal y entrenamiento de la solución

Para reducir el over-fitting detectado, además de Data augmentation, se trabajará sobre la arquitectura de red.

- Para evitar sobreajuste en la arquitectura del modelo se utilizará **regularización L1, L2 y drop out.**
- Por otra parte, se detectó que a determinadas épocas ya no existía una mejora sustancial en las pérdidas de validación, razón por la cual para ahorrar tiempo y además obetener el mejor modelo antes de que pueda producirse overfiting se procederá a usar técnica de **early stop**.
- Se realizará un **early stop** con **val_loss** debido a que el probelma más crítico que se tiene al momento es el overfitting, ya que el training si llega a valores de pérdidas cercanos a cero y una exactitud cercana a 0.99

In [ ]:

```
#BASE MODEL
model_2 = models.Sequential()
model_2.add(layers.Conv2D(32,(3,3),activation='relu',
                        input_shape=(150,150,3)))
model_2.add(layers.MaxPooling2D((2,2)))
model_2.add(layers.Conv2D(64,(3,3),activation='relu',kernel_regularizer=regular
model_2.add(layers.MaxPooling2D((2,2)))
model_2.add(layers.Conv2D(128,(3,3),activation='relu'))
model_2.add(layers.MaxPooling2D((2,2)))
model_2.add(layers.Conv2D(128,(3,3),activation='relu'))
model_2.add(layers.MaxPooling2D((2,2)))
model_2.add(layers.Flatten())
model_2.add(layers.Dropout(0.5))
model_2.add(layers.Dense(3000,activation='relu'))
model_2.add(layers.Dropout(0.5))
model_2.add(layers.Dense(525,activation='softmax'))
```

In [ ]:

```
model_2.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_8 (Conv2D)           (None, 148, 148, 32)      896       
                                                                 
 max_pooling2d_8 (MaxPooling  (None, 74, 74, 32)       0         
 2D)                                                             
                                                                 
 conv2d_9 (Conv2D)           (None, 72, 72, 64)        18496     
                                                                 
 max_pooling2d_9 (MaxPooling  (None, 36, 36, 64)       0         
 2D)                                                             
                                                                 
 conv2d_10 (Conv2D)          (None, 34, 34, 128)       73856     
                                                                 
 max_pooling2d_10 (MaxPoolin  (None, 17, 17, 128)      0         
 g2D)                                                            
                                                                 
 conv2d_11 (Conv2D)          (None, 15, 15, 128)       147584    
                                                                 
 max_pooling2d_11 (MaxPoolin  (None, 7, 7, 128)        0         
 g2D)                                                            
                                                                 
 flatten_2 (Flatten)         (None, 6272)              0         
                                                                 
 dropout (Dropout)           (None, 6272)              0         
                                                                 
 dense_4 (Dense)             (None, 3000)              18819000  
                                                                 
 dropout_1 (Dropout)         (None, 3000)              0         
                                                                 
 dense_5 (Dense)             (None, 525)               1575525   
                                                                 
=================================================================
Total params: 20,635,357
Trainable params: 20,635,357
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```python
model_2.compile(loss='categorical_crossentropy',
              optimizer=optimizers.RMSprop(learning_rate=1e-4),
              metrics=['acc'])
```

In [ ]:

```python
from keras import callbacks
callback=callbacks.EarlyStopping(
    monitor="val_loss",
    min_delta=0,
    patience=3,
    verbose=0, # se desea conocer en qué momento se produce el callback
    mode="auto",
    baseline=None,
    restore_best_weights=True, # se tomarán en cuenta los mejores valores obten
    start_from_epoch=0,
)
```

In [ ]:

```python
history_2 = model_2.fit(
    train_generator,
    steps_per_epoch = train_generator.n//train_generator.batch_size,
    epochs = 20,
    validation_data = validation_generator,
    validation_steps = validation_generator.n//validation_generator.batch_size,
    callbacks=[callback,cp_callback])
```

```
Epoch 1/20
330/330 [==============================] - ETA: 0s - loss: 13.9869 - acc: 0.00
58
Epoch 1: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 641s 2s/step - loss: 13.9869 - acc:
0.0058 - val_loss: 10.4982 - val_acc: 0.0191
Epoch 2/20
330/330 [==============================] - ETA: 0s - loss: 8.7578 - acc: 0.022
4
Epoch 2: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 638s 2s/step - loss: 8.7578 - acc:
0.0224 - val_loss: 7.1805 - val_acc: 0.0537
Epoch 3/20
330/330 [==============================] - ETA: 0s - loss: 6.8032 - acc: 0.040
8
Epoch 3: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 616s 2s/step - loss: 6.8032 - acc:
0.0408 - val_loss: 6.1929 - val_acc: 0.0694
Epoch 4/20
330/330 [==============================] - ETA: 0s - loss: 6.1750 - acc: 0.056
9
Epoch 4: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 613s 2s/step - loss: 6.1750 - acc:
0.0569 - val_loss: 5.8366 - val_acc: 0.0774
Epoch 5/20
330/330 [==============================] - ETA: 0s - loss: 5.8012 - acc: 0.072
7
Epoch 5: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 615s 2s/step - loss: 5.8012 - acc:
0.0727 - val_loss: 5.4644 - val_acc: 0.1002
Epoch 6/20
330/330 [==============================] - ETA: 0s - loss: 5.5140 - acc: 0.088
9
Epoch 6: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 619s 2s/step - loss: 5.5140 - acc:
0.0889 - val_loss: 5.1388 - val_acc: 0.1288
Epoch 7/20
330/330 [==============================] - ETA: 0s - loss: 5.2846 - acc: 0.102
9
Epoch 7: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 606s 2s/step - loss: 5.2846 - acc:
0.1029 - val_loss: 4.9332 - val_acc: 0.1341
Epoch 8/20
330/330 [==============================] - ETA: 0s - loss: 5.0964 - acc: 0.115
6
Epoch 8: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 613s 2s/step - loss: 5.0964 - acc:
0.1156 - val_loss: 4.7464 - val_acc: 0.1566
Epoch 9/20
330/330 [==============================] - ETA: 0s - loss: 4.9287 - acc: 0.129
7
Epoch 9: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoints/
cp.ckpt
330/330 [==============================] - 614s 2s/step - loss: 4.9287 - acc:
0.1297 - val_loss: 4.5584 - val_acc: 0.1795
Epoch 10/20
330/330 [==============================] - ETA: 0s - loss: 4.7820 - acc: 0.142
5
Epoch 10: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 603s 2s/step - loss: 4.7820 - acc:
0.1425 - val_loss: 4.3965 - val_acc: 0.2058
Epoch 11/20
330/330 [==============================] - ETA: 0s - loss: 4.6518 - acc: 0.156
2
Epoch 11: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 616s 2s/step - loss: 4.6518 - acc:
0.1562 - val_loss: 4.3359 - val_acc: 0.2050
Epoch 12/20
```

```
330/330 [==============================] - ETA: 0s - loss: 4.5344 - acc: 0.168
3
Epoch 12: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 614s 2s/step - loss: 4.5344 - acc:
0.1683 - val_loss: 4.2870 - val_acc: 0.2081
Epoch 13/20
330/330 [==============================] - ETA: 0s - loss: 4.4229 - acc: 0.179
6
Epoch 13: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 614s 2s/step - loss: 4.4229 - acc:
0.1796 - val_loss: 4.1540 - val_acc: 0.2127
Epoch 14/20
330/330 [==============================] - ETA: 0s - loss: 4.3226 - acc: 0.189
3
Epoch 14: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 614s 2s/step - loss: 4.3226 - acc:
0.1893 - val_loss: 4.1035 - val_acc: 0.2248
Epoch 15/20
330/330 [==============================] - ETA: 0s - loss: 4.2247 - acc: 0.200
8
Epoch 15: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 615s 2s/step - loss: 4.2247 - acc:
0.2008 - val_loss: 3.9258 - val_acc: 0.2553
Epoch 16/20
330/330 [==============================] - ETA: 0s - loss: 4.1441 - acc: 0.212
5
Epoch 16: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 616s 2s/step - loss: 4.1441 - acc:
0.2125 - val_loss: 3.8632 - val_acc: 0.2504
Epoch 17/20
330/330 [==============================] - ETA: 0s - loss: 4.0697 - acc: 0.221
4
Epoch 17: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 611s 2s/step - loss: 4.0697 - acc:
0.2214 - val_loss: 3.7421 - val_acc: 0.2793
Epoch 18/20
330/330 [==============================] - ETA: 0s - loss: 3.9911 - acc: 0.228
3
Epoch 18: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 613s 2s/step - loss: 3.9911 - acc:
0.2283 - val_loss: 3.5852 - val_acc: 0.3148
Epoch 19/20
330/330 [==============================] - ETA: 0s - loss: 3.9197 - acc: 0.242
3
Epoch 19: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 618s 2s/step - loss: 3.9197 - acc:
0.2423 - val_loss: 3.5162 - val_acc: 0.3148
Epoch 20/20
330/330 [==============================] - ETA: 0s - loss: 3.8522 - acc: 0.249
4
Epoch 20: saving model to /content/gdrive/MyDrive/descargas_kaggle/checkpoint
s/cp.ckpt
330/330 [==============================] - 614s 2s/step - loss: 3.8522 - acc:
0.2494 - val_loss: 3.4787 - val_acc: 0.3205
```

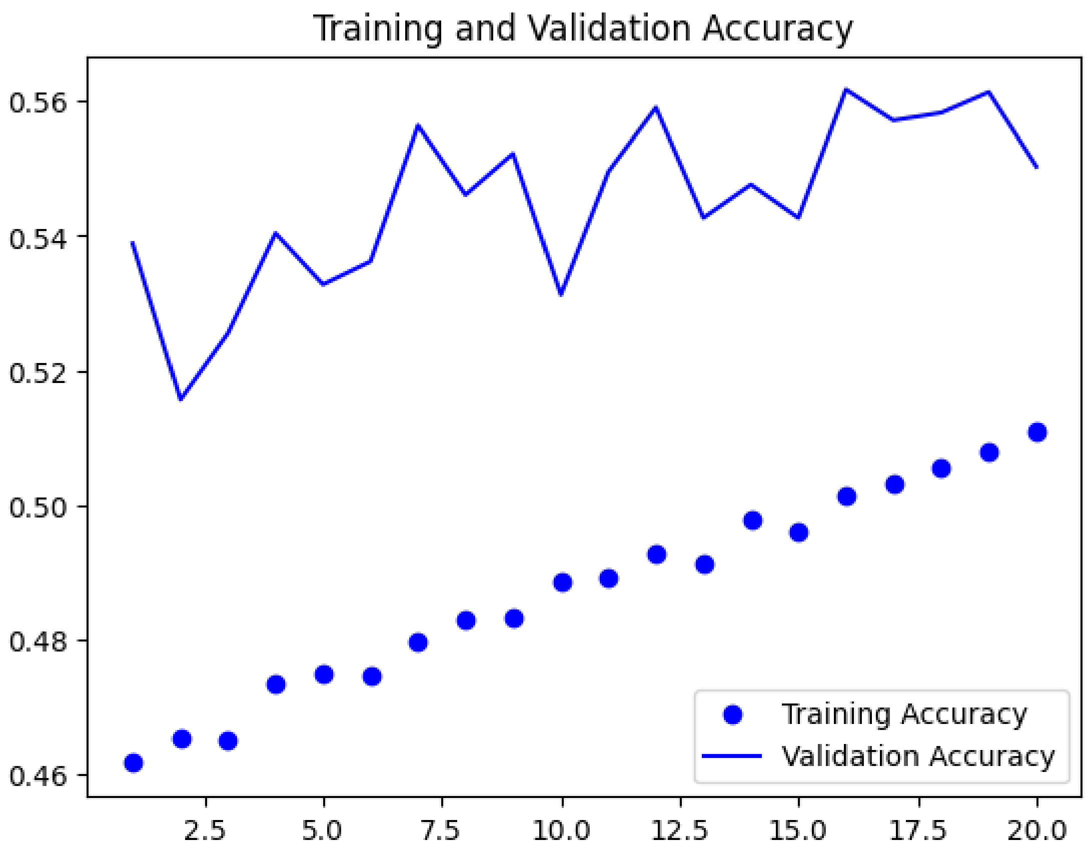
9. Monitorización del proceso de entrenamiento del modelo #2

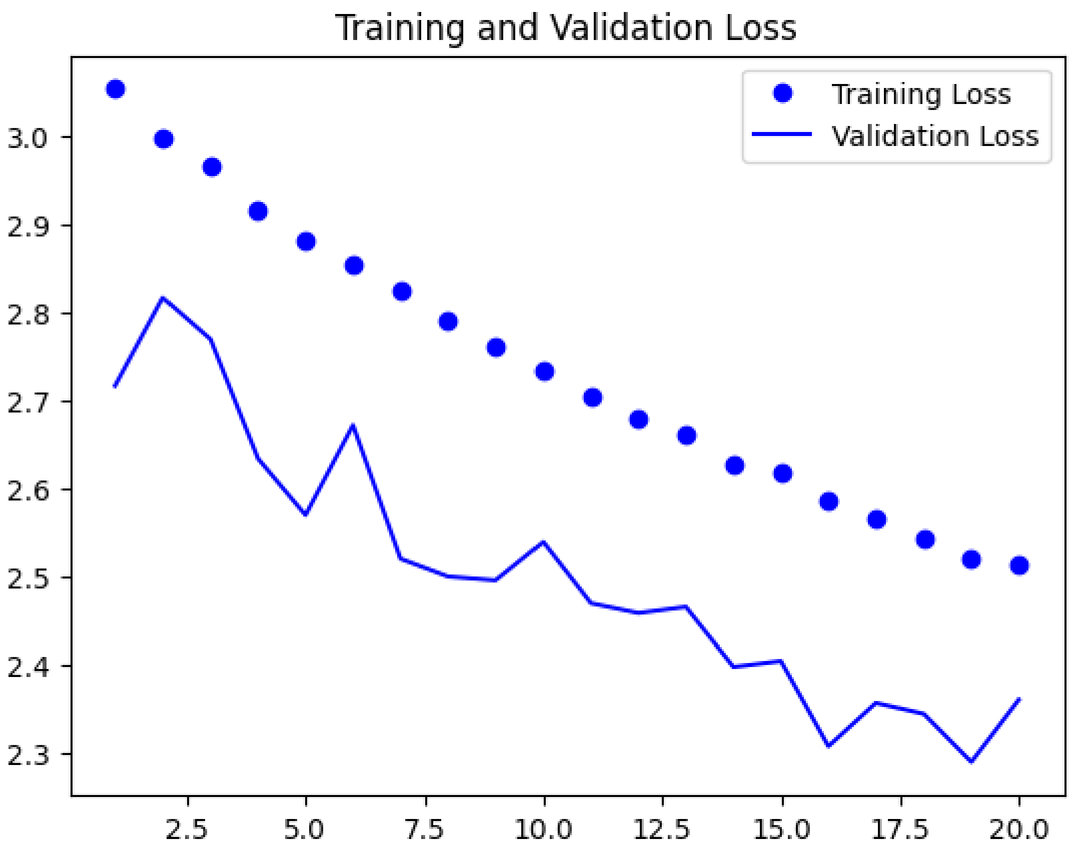
Debido al tiempo que demora el entrenamiento de la nueva red, se ha realizado en varias partes, aprovechando el manejo del callback de model checkpoint. El modelo almacenado así como su historial corresponden al último proceso de entrenamiento realizado, razón por la cual ya se encuentra con valores de loss bajos.

Si se realizaran más epocas de entrenamiento, la exactitud del modelo podría mejorar. Sin embargo, una mejor opción sería hacer cambios en la arquitectura del modelo.

In [ ]:

```
acc_2 = history_2['acc']
val_acc_2 = history_2['val_acc']
loss_2 = history_2['loss']
val_loss_2 = history_2['val_loss']

epochs = range(1,len(acc_2) + 1)

plt.plot(epochs, acc_2, 'bo', label='Training Accuracy')
plt.plot(epochs, val_acc_2, 'b', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss_2, 'bo', label='Training Loss')
plt.plot(epochs, val_loss_2, 'b', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.legend()

plt.show()
```





10. Evaluación del modelo predictivo #2

- Como se puede observar en los gráficos de pérdidas y exactitud ya no se encuentra en sobreajuste. Sin embargo, sería necesario incrementar el número de épocas de análisis o hacer un cambio en la arquitectura de la red entrenada.
- Evaluaremos el comportamiento del modelo más afondo, con los datos de test.

In [ ]:

```
# evaluamos y observamos el comportamiento de todos los datos de test
# predichos por el modelo generado
from tensorflow.keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(rescale=1./255)
test_generator = datagen.flow_from_directory(directory=DIRECTORY_TEST,
                                                    target_size=(IMG_WIDTH, IMG
                                                    batch_size=1,
                                                    class_mode='categorical',
                                                    shuffle=False)
test_labels = test_generator.classes
test_class_name = list(test_generator.class_indices.keys()) # se obtiene los no
```

In [ ]:

```
# Evaluamos de manera general el loss y el acuracy en la data de test
score=model.evaluate(test_generator, steps=None, max_queue_size=10, workers=1,
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
2625/2625 [==============================] - 29s 8ms/step - loss: 1.9502 - ac
c: 0.6545
Test loss: 1.9501781463623047
Test accuracy: 0.6544761657714844
```

Estrategia 2: Red pre-entrenada

3. Acondicionamiento del conjunto de datos

Realizaremos un redimensionamiento de las imágenes y lotes para el poder realizar el entrenamiento de la red neuronal. Para todo esto usaremos un Generator de Imágenes.

In [ ]:

```
BASE_FOLDER = "my_dataset"

DIRECTORY_TRAIN = BASE_FOLDER+'/train/'
DIRECTORY_VALID = BASE_FOLDER+'/valid/'
DIRECTORY_TEST = BASE_FOLDER+'/test/'

train_datagen_3 = ImageDataGenerator(rescale=1./255.)

train_generator_3 = train_datagen_3.flow_from_directory(directory=DIRECTORY_TRA
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    class_mode='categorical',
                                                    seed=42)

validation_generator_3 = train_datagen_3.flow_from_directory(directory=DIRECTOR
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    class_mode='categorical',
                                                    seed=42)

test_labels_3 = validation_generator_3.classes
test_class_name_3 = list(validation_generator_3.class_indices.keys()) # se obt
```

Utilizaremos en hub de tensorflow para carga la red pre-entrenada ResNetV2

In [ ]:

```
conv_base = hub.KerasLayer("https://tfhub.dev/google/imagenet/resnet_v2_50/feat
                          trainable=False,
                          name='feature_extraction_layer',
                          input_shape=(224, 224, 3))
```

Definimos el modelo sequencial

In [ ]:

```
model_3 = tf.keras.Sequential([
    conv_base,
    tf.keras.layers.Dropout(.2),
    tf.keras.layers.Dense(512,  activation="relu"),
    tf.keras.layers.Dense(256,  activation="relu"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(.2),
    tf.keras.layers.Dense(128,  activation="relu"),
    layers.Dense(525, activation='softmax', name='output_layer')
  ])
```

Vemos el resumen de las capas y parametros

In [ ]:

```
model_3.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 feature_extraction_layer (K  (None, 2048)             23564800  
 erasLayer)                                                      
                                                                 
 dropout_6 (Dropout)         (None, 2048)              0         
                                                                 
 dense_9 (Dense)             (None, 512)               1049088   
                                                                 
 dense_10 (Dense)            (None, 256)               131328    
                                                                 
 batch_normalization_3 (Batc  (None, 256)              1024      
 hNormalization)                                                 
                                                                 
 dropout_7 (Dropout)         (None, 256)               0         
                                                                 
 dense_11 (Dense)            (None, 128)               32896     
                                                                 
 output_layer (Dense)        (None, 525)               67725     
                                                                 
=================================================================
Total params: 24,846,861
Trainable params: 1,281,549
Non-trainable params: 23,565,312
_________________________________________________________________
```

Compilamos el modelo

In [ ]:

```
model_3.compile(loss='categorical_crossentropy',
                     optimizer=tf.keras.optimizers.Adam(),
                     metrics=['accuracy'])
```

Fine tuning

In [ ]:

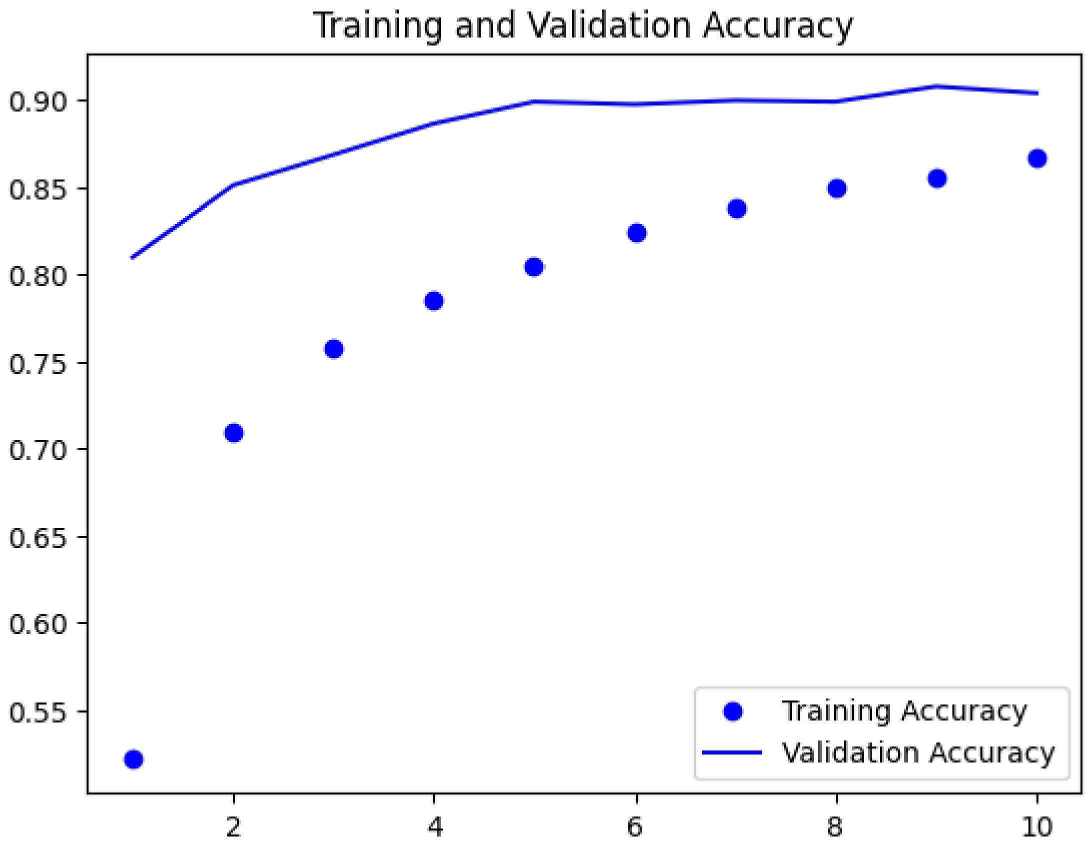
```
history_3 = model_3.fit(train_generator_3,
                        epochs=10,
                        steps_per_epoch=len(train_generator_3),
                        validation_data=validation_generator_3)
```

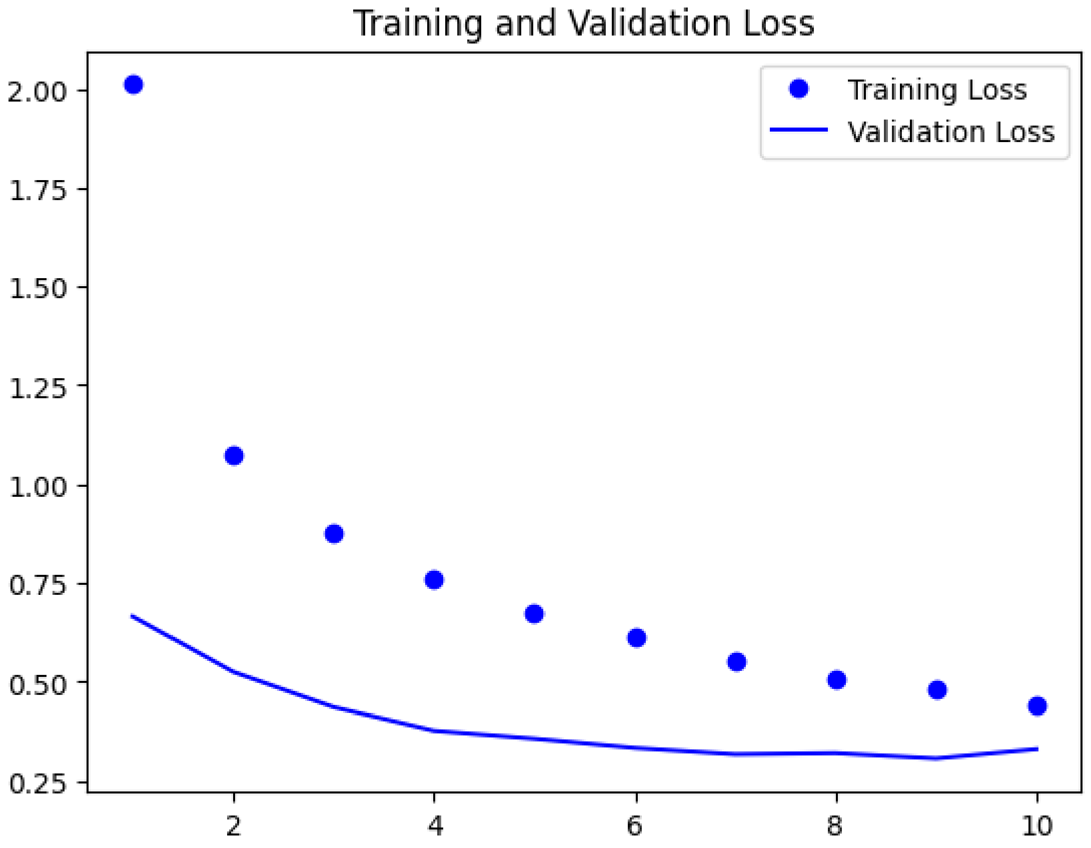
```
Epoch 1/10
2645/2645 [==============================] - 296s 106ms/step - loss: 2.0107 -
accuracy: 0.5215 - val_loss: 0.6656 - val_accuracy: 0.8099
Epoch 2/10
2645/2645 [==============================] - 282s 106ms/step - loss: 1.0744 -
accuracy: 0.7089 - val_loss: 0.5251 - val_accuracy: 0.8510
Epoch 3/10
2645/2645 [==============================] - 262s 99ms/step - loss: 0.8775 - a
ccuracy: 0.7573 - val_loss: 0.4363 - val_accuracy: 0.8686
Epoch 4/10
2645/2645 [==============================] - 279s 105ms/step - loss: 0.7618 -
accuracy: 0.7853 - val_loss: 0.3766 - val_accuracy: 0.8865
Epoch 5/10
2645/2645 [==============================] - 269s 102ms/step - loss: 0.6760 -
accuracy: 0.8049 - val_loss: 0.3568 - val_accuracy: 0.8990
Epoch 6/10
2645/2645 [==============================] - 276s 104ms/step - loss: 0.6129 -
accuracy: 0.8240 - val_loss: 0.3335 - val_accuracy: 0.8975
Epoch 7/10
2645/2645 [==============================] - 273s 103ms/step - loss: 0.5530 -
accuracy: 0.8378 - val_loss: 0.3175 - val_accuracy: 0.8998
Epoch 8/10
2645/2645 [==============================] - 272s 103ms/step - loss: 0.5088 -
accuracy: 0.8495 - val_loss: 0.3205 - val_accuracy: 0.8990
Epoch 9/10
2645/2645 [==============================] - 307s 116ms/step - loss: 0.4796 -
accuracy: 0.8559 - val_loss: 0.3069 - val_accuracy: 0.9078
Epoch 10/10
2645/2645 [==============================] - 292s 110ms/step - loss: 0.4390 -
accuracy: 0.8666 - val_loss: 0.3306 - val_accuracy: 0.9040
```

Grafico para ver la precisión de entrenamiento y validación y funcion de pérdida del modelo

In [ ]:

```
import matplotlib.pyplot as plt

acc_3 = history_3.history['accuracy']
val_acc_3 = history_3.history['val_accuracy']
loss_3 = history_3.history['loss']
val_loss_3 = history_3.history['val_loss']

epochs = range(1,len(acc_3) + 1)

plt.plot(epochs, acc_3, 'bo', label='Training Accuracy')
plt.plot(epochs, val_acc_3, 'b', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss_3, 'bo', label='Training Loss')
plt.plot(epochs, val_loss_3, 'b', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.legend()

plt.show()
```





Arquitectura #2

7. Re-acondicionamiento del conjunto de datos

Usaremos otro tipo de arquitectura y luego visualizar su desenvolvimiento.

In [ ]:

```
train_datagen_4 = ImageDataGenerator(rescale=1./255.,
                                     zoom_range=0.2,
                                     width_shift_range=0.2,
                                      height_shift_range=0.2)

val_datagen_4 = ImageDataGenerator(rescale=1/255.)

test_datagen_4 = ImageDataGenerator(rescale=1/255.)

train_generator_4 = train_datagen_4.flow_from_directory(directory=DIRECTORY_TRA
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    class_mode='categorical',
                                                    shuffle=True)

validation_generator_4 = val_datagen_4.flow_from_directory(directory=DIRECTORY_
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    class_mode='categorical',
                                                    shuffle=False)

test_generator_4 = test_datagen_4.flow_from_directory(DIRECTORY_TEST,
                                                    target_size=(224, 224),
                                                    batch_size=32,
                                                    shuffle=False,
                                                    class_mode='categorical')
```

In [ ]:

```
conv_base_4 = MobileNetV2(include_top = False,
                       weights = 'imagenet',
                       input_shape = (224,224,3))
```

In [ ]:

```
conv_base_4.summary()
```

```
Model: "mobilenetv2_1.00_224"
______________________________________________________________________________
____________________
 Layer (type)                   Output Shape         Param #     Connected to
==============================================================================
====================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864         ['input_1[0]
[0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0]
[0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0           ['bn_Conv1[0]
[0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288        ['Conv1_relu
[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128        ['expanded_co
nv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0          ['expanded_co
nv_depthwise_BN[0][0
 ReLU)                          )                                ]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512        ['expanded_co
nv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64         ['expanded_co
nv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536        ['expanded_co
nv_project_BN[0][0]'
                                )                                ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384        ['block_1_exp
and[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0           ['block_1_exp
and_BN[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0           ['block_1_exp
and_relu[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864         ['block_1_pad
[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384         ['block_1_dep
thwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0           ['block_1_dep
thwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304        ['block_1_dep
thwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_1_pro
ject[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_1_pro
ject_BN[0][0]']
```

```
 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576         ['block_2_exp
and[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_2_exp
and_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296        ['block_2_exp
and_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576         ['block_2_dep
thwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0           ['block_2_dep
thwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456        ['block_2_dep
thwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_2_pro
ject[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0           ['block_1_pro
ject_BN[0][0]',
                                                                  'block_2_pro
ject_BN[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_2_add
[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576         ['block_3_exp
and[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_3_exp
and_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0           ['block_3_exp
and_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296        ['block_3_pad
[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576         ['block_3_dep
thwise[0][0]']
 malization)

 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0           ['block_3_dep
thwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608        ['block_3_dep
thwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_3_pro
ject[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_3_pro
ject_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768         ['block_4_exp
and[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_4_exp
and_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728        ['block_4_exp
and_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768         ['block_4_dep
thwise[0][0]']
 malization)
 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_4_dep
thwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_4_dep
thwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_4_pro
ject[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0           ['block_3_pro
ject_BN[0][0]',
                                                                  'block_4_pro
ject_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_4_add
[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_5_exp
and[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_5_exp
and_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728       ['block_5_exp
and_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768        ['block_5_dep
thwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_5_dep
thwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_5_dep
thwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_5_pro
ject[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0           ['block_4_add
[0][0]',
                                                                  'block_5_pro
ject_BN[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_5_add
[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_6_exp
and[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_6_exp
and_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0           ['block_6_exp
and_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728       ['block_6_pad
[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768        ['block_6_dep
thwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0           ['block_6_dep
thwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288       ['block_6_dep
thwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_6_pro
ject[0][0]']
```

```
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_6_pro
ject_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_7_exp
and[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_7_exp
and_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_7_exp
and_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_7_dep
thwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_7_dep
thwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_7_dep
thwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_7_pro
ject[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0           ['block_6_pro
ject_BN[0][0]',
                                                                  'block_7_pro
ject_BN[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_7_add
[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_8_exp
and[0][0]']
 ization)

 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_8_exp
and_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_8_exp
and_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_8_dep
thwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_8_dep
thwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_8_dep
thwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_8_pro
ject[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0           ['block_7_add
[0][0]',
                                                                  'block_8_pro
ject_BN[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_8_add
[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_9_exp
and[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_9_exp
and_BN[0][0]']
```

```
 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_9_exp
and_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_9_dep
thwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0          ['block_9_dep
thwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576      ['block_9_dep
thwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256        ['block_9_pro
ject[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0          ['block_8_add
[0][0]',
                                                                 'block_9_pro
ject_BN[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576      ['block_9_add
[0][0]']

 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536       ['block_10_ex
pand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0          ['block_10_ex
pand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456       ['block_10_ex
pand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536       ['block_10_de
pthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0          ['block_10_de
pthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864      ['block_10_de
pthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384        ['block_10_pr
oject[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296      ['block_10_pr
oject_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_11_ex
pand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0          ['block_11_ex
pand_BN[0][0]']

 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184       ['block_11_ex
pand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304       ['block_11_de
pthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0          ['block_11_de
pthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296      ['block_11_de
pthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384        ['block_11_pr
oject[0][0]']
 alization)
 block_11_add (Add)             (None, 14, 14, 96)   0           ['block_10_pr
oject_BN[0][0]',
                                                                  'block_11_pr
oject_BN[0][0]']

 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_11_ad
d[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_12_ex
pand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_12_ex
pand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184       ['block_12_ex
pand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304       ['block_12_de
pthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0          ['block_12_de
pthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 14, 14, 96)   55296       ['block_12_de
pthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384         ['block_12_pr
oject[0][0]']
 alization)

 block_12_add (Add)             (None, 14, 14, 96)   0           ['block_11_ad
d[0][0]',
                                                                  'block_12_pr
oject_BN[0][0]']

 block_13_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_12_ad
d[0][0]']

 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_13_ex
pand[0][0]']
 lization)

 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_13_ex
pand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0           ['block_13_ex
pand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184        ['block_13_pa
d[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304        ['block_13_de
pthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0           ['block_13_de
pthwise_BN[0][0]']

 block_13_project (Conv2D)      (None, 7, 7, 160)    92160       ['block_13_de
pthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_13_pr
oject[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_13_pr
oject_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_14_ex
pand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_14_ex
pand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_14_ex
pand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_14_de
pthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_14_de
pthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_14_de
pthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_14_pr
oject[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0           ['block_13_pr
oject_BN[0][0]',
                                                                  'block_14_pr
oject_BN[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_14_ad
d[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_15_ex
pand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_15_ex
pand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_15_ex
pand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_15_de
pthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_15_de
pthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_15_de
pthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_15_pr
oject[0][0]']
 alization)

 block_15_add (Add)             (None, 7, 7, 160)    0           ['block_14_ad
d[0][0]',
                                                                  'block_15_pr
oject_BN[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_15_ad
d[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_16_ex
pand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_16_ex
pand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_16_ex
pand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_16_de
pthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_16_de
pthwise_BN[0][0]']
```

```
 block_16_project (Conv2D)      (None, 7, 7, 320)    307200      ['block_16_de
pthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280        ['block_16_pr
oject[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600      ['block_16_pr
oject_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0]
[0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0           ['Conv_1_bn
[0][0]']

==============================================================================
====================
Total params: 2,257,984
Trainable params: 2,223,872
Non-trainable params: 34,112
______________________________________________________________________________
____________________
```

In [ ]:

```
# Congelamiento de la primera mitad de redes
num_layers = len(conv_base_4.layers)
for layer in conv_base_4.layers[:num_layers//2]:
    layer.trainable = False
```

In [ ]:

```
conv_base_4.summary()
```

```
Model: "mobilenetv2_1.00_224"
______________________________________________________________________________
____________________
 Layer (type)                   Output Shape         Param #     Connected to
==============================================================================
====================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864         ['input_1[0]
[0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0]
[0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0           ['bn_Conv1[0]
[0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288        ['Conv1_relu
[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128        ['expanded_co
nv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0          ['expanded_co
nv_depthwise_BN[0][0
 ReLU)                          )                                ]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512        ['expanded_co
nv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64         ['expanded_co
nv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536        ['expanded_co
nv_project_BN[0][0]'
                                )                                ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384        ['block_1_exp
and[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0           ['block_1_exp
and_BN[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0           ['block_1_exp
and_relu[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864         ['block_1_pad
[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384         ['block_1_dep
thwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0           ['block_1_dep
thwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304        ['block_1_dep
thwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_1_pro
ject[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_1_pro
ject_BN[0][0]']
```

```
 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576        ['block_2_exp
and[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_2_exp
and_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296       ['block_2_exp
and_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576        ['block_2_dep
thwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0           ['block_2_dep
thwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456        ['block_2_dep
thwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96          ['block_2_pro
ject[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0           ['block_1_pro
ject_BN[0][0]',
                                                                  'block_2_pro

ject_BN[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456        ['block_2_add
[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576        ['block_3_exp
and[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0           ['block_3_exp
and_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0           ['block_3_exp
and_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296       ['block_3_pad
[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576        ['block_3_dep
thwise[0][0]']
 malization)

 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0           ['block_3_dep
thwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608        ['block_3_dep
thwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_3_pro
ject[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_3_pro
ject_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768        ['block_4_exp
and[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_4_exp
and_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728       ['block_4_exp
and_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768        ['block_4_dep
thwise[0][0]']
 malization)
 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_4_dep
thwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_4_dep
thwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_4_pro
ject[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0           ['block_3_pro
ject_BN[0][0]',

                                                                  'block_4_pro

ject_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_4_add
[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768         ['block_5_exp
and[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_5_exp
and_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728        ['block_5_exp
and_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768         ['block_5_dep
thwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_5_dep
thwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_5_dep
thwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_5_pro
ject[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0           ['block_4_add
[0][0]',

                                                                  'block_5_pro

ject_BN[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_5_add
[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768         ['block_6_exp
and[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_6_exp
and_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0           ['block_6_exp
and_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728        ['block_6_pad
[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768         ['block_6_dep
thwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0           ['block_6_dep
thwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288       ['block_6_dep
thwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_6_pro
ject[0][0]']
```

```
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_6_pro
ject_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_7_exp
and[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_7_exp
and_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_7_exp
and_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_7_dep
thwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_7_dep
thwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_7_dep
thwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_7_pro
ject[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0           ['block_6_pro
ject_BN[0][0]',
                                                                  'block_7_pro
ject_BN[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_7_add
[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_8_exp
and[0][0]']
 ization)

 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_8_exp
and_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_8_exp
and_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_8_dep
thwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_8_dep
thwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_8_dep
thwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256         ['block_8_pro
ject[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0           ['block_7_add
[0][0]',
                                                                  'block_8_pro
ject_BN[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576       ['block_8_add
[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536       ['block_9_exp
and[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0           ['block_9_exp
and_BN[0][0]']
```

```
 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456       ['block_9_exp
and_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536       ['block_9_dep
thwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0           ['block_9_dep
thwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576       ['block_9_dep
thwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 14, 14, 64)   256        ['block_9_pro
ject[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0           ['block_8_add
[0][0]',
                                                                  'block_9_pro
ject_BN[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576       ['block_9_add
[0][0]']

 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536       ['block_10_ex
pand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0           ['block_10_ex
pand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456       ['block_10_ex
pand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536       ['block_10_de
pthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0          ['block_10_de
pthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864       ['block_10_de
pthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)   384        ['block_10_pr
oject[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296       ['block_10_pr
oject_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304       ['block_11_ex
pand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0           ['block_11_ex
pand_BN[0][0]']

 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184       ['block_11_ex
pand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304       ['block_11_de
pthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0          ['block_11_de
pthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296       ['block_11_de
pthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 14, 14, 96)   384        ['block_11_pr
oject[0][0]']
 alization)
 block_11_add (Add)            (None, 14, 14, 96)   0           ['block_10_pr
oject_BN[0][0]',
                                                                  'block_11_pr
oject_BN[0][0]']

 block_12_expand (Conv2D)      (None, 14, 14, 576)  55296       ['block_11_ad
d[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304      ['block_12_ex
pand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0          ['block_12_ex
pand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184      ['block_12_ex
pand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304      ['block_12_de
pthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0         ['block_12_de
pthwise_BN[0][0]']

 block_12_project (Conv2D)     (None, 14, 14, 96)   55296       ['block_12_de
pthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384        ['block_12_pr
oject[0][0]']
 alization)

 block_12_add (Add)            (None, 14, 14, 96)   0           ['block_11_ad
d[0][0]',
                                                                  'block_12_pr
oject_BN[0][0]']

 block_13_expand (Conv2D)      (None, 14, 14, 576)  55296       ['block_12_ad
d[0][0]']

 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304      ['block_13_ex
pand[0][0]']
 lization)

 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0          ['block_13_ex
pand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0          ['block_13_ex
pand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184       ['block_13_pa
d[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304       ['block_13_de
pthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0          ['block_13_de
pthwise_BN[0][0]']

 block_13_project (Conv2D)     (None, 7, 7, 160)    92160       ['block_13_de
pthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640        ['block_13_pr
oject[0][0]']
 alization)

 block_14_expand (Conv2D)      (None, 7, 7, 960)    153600      ['block_13_pr
oject_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840       ['block_14_ex
pand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0          ['block_14_ex
pand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_14_ex
pand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_14_de
pthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_14_de
pthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_14_de
pthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_14_pr
oject[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0           ['block_13_pr
oject_BN[0][0]',
                                                                  'block_14_pr
oject_BN[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_14_ad
d[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_15_ex
pand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_15_ex
pand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_15_ex
pand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_15_de
pthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_15_de
pthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600      ['block_15_de
pthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640         ['block_15_pr
oject[0][0]']
 alization)

 block_15_add (Add)             (None, 7, 7, 160)    0           ['block_14_ad
d[0][0]',
                                                                  'block_15_pr
oject_BN[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600      ['block_15_ad
d[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840        ['block_16_ex
pand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0           ['block_16_ex
pand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640        ['block_16_ex
pand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840        ['block_16_de
pthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0           ['block_16_de
pthwise_BN[0][0]']
```

```
 block_16_project (Conv2D)      (None, 7, 7, 320)    307200      ['block_16_de
pthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280        ['block_16_pr
oject[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600      ['block_16_pr
oject_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0]
[0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0           ['Conv_1_bn
[0][0]']

==============================================================================
====================
Total params: 2,257,984
Trainable params: 2,063,488
Non-trainable params: 194,496
______________________________________________________________________________
____________________
```

In [ ]:

```python
from keras import models

model_4 = models.Sequential()
model_4.add(conv_base_4)
model_4.add(layers.Flatten())
model_4.add(layers.Dense(512, activation='relu'))
model_4.add(Dropout(0.5))
model_4.add(layers.Dense(525, activation='softmax'))
```

In [ ]:

```python
model_4.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984   
 ional)                                                          

 flatten (Flatten)           (None, 62720)             0         

 dense_12 (Dense)            (None, 512)               32113152  

 dropout_8 (Dropout)         (None, 512)               0         

 dense_13 (Dense)            (None, 525)               269325    

=================================================================
Total params: 34,640,461
Trainable params: 34,445,965
Non-trainable params: 194,496
_________________________________________________________________
```

In [ ]:

```python
model_4.compile(loss='categorical_crossentropy',
                     optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001),
                     metrics=['accuracy'])
```

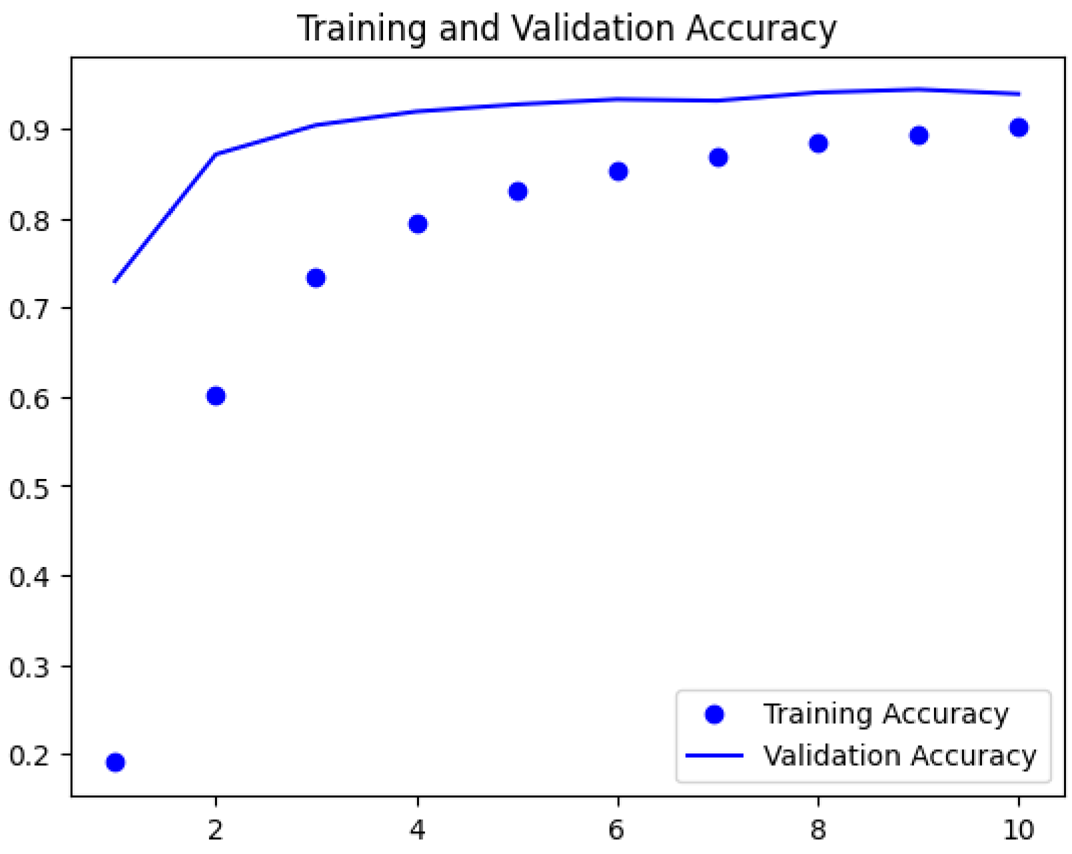
In [ ]:

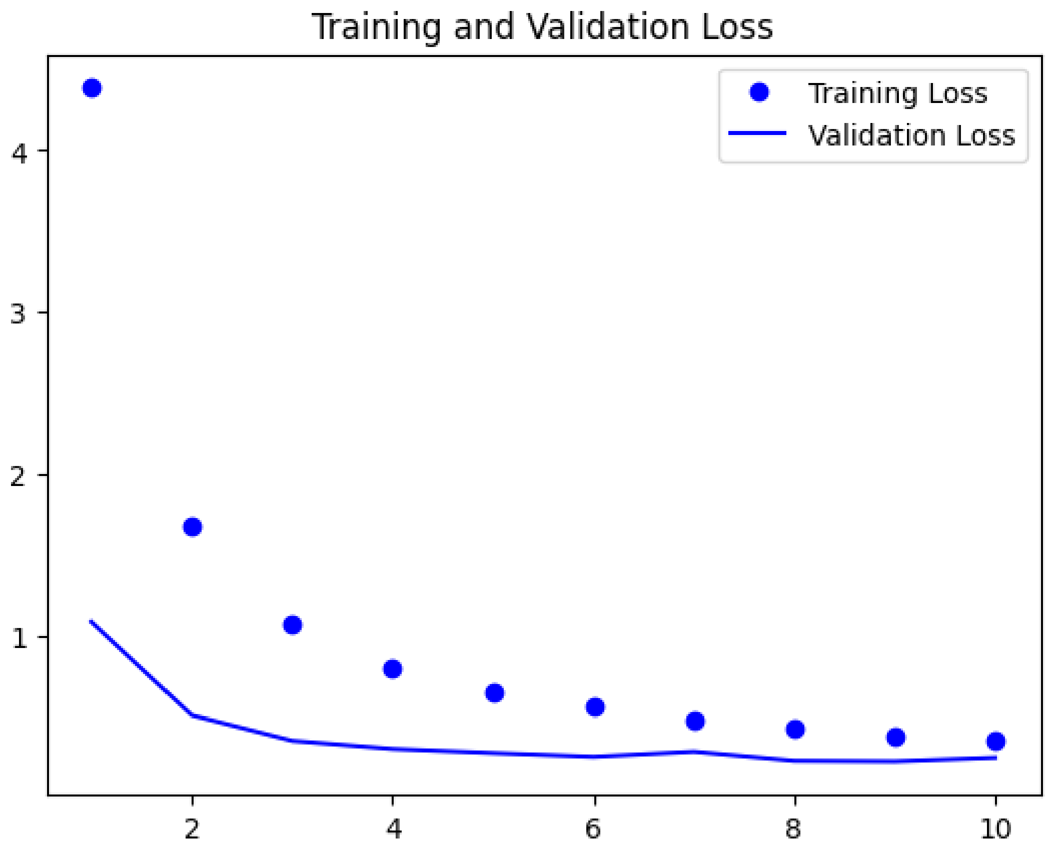
```python
history_4 = model_4.fit(train_generator_4,
                        epochs=10,
                        validation_data=validation_generator_4)
```

```
Epoch 1/10
2645/2645 [==============================] - 1229s 456ms/step - loss: 4.3815 -
accuracy: 0.1905 - val_loss: 1.0909 - val_accuracy: 0.7295
Epoch 2/10
2645/2645 [==============================] - 1182s 447ms/step - loss: 1.6825 -
accuracy: 0.6023 - val_loss: 0.5143 - val_accuracy: 0.8712
Epoch 3/10
2645/2645 [==============================] - 1182s 447ms/step - loss: 1.0729 -
accuracy: 0.7346 - val_loss: 0.3575 - val_accuracy: 0.9040
Epoch 4/10
2645/2645 [==============================] - 1168s 442ms/step - loss: 0.8097 -
accuracy: 0.7944 - val_loss: 0.3076 - val_accuracy: 0.9192
Epoch 5/10
2645/2645 [==============================] - 1137s 430ms/step - loss: 0.6567 -
accuracy: 0.8296 - val_loss: 0.2824 - val_accuracy: 0.9272
Epoch 6/10
2645/2645 [==============================] - 1121s 424ms/step - loss: 0.5657 -
accuracy: 0.8535 - val_loss: 0.2594 - val_accuracy: 0.9330
Epoch 7/10
2645/2645 [==============================] - 1121s 424ms/step - loss: 0.4900 -
accuracy: 0.8698 - val_loss: 0.2907 - val_accuracy: 0.9314
Epoch 8/10
2645/2645 [==============================] - 1108s 419ms/step - loss: 0.4345 -
accuracy: 0.8848 - val_loss: 0.2349 - val_accuracy: 0.9406
Epoch 9/10
2645/2645 [==============================] - 1107s 418ms/step - loss: 0.3887 -
accuracy: 0.8946 - val_loss: 0.2314 - val_accuracy: 0.9440
Epoch 10/10
2571/2645 [============================>.] - ETA: 30s - loss: 0.3614 - accurac
y: 0.9020
```

In [ ]:

```
import matplotlib.pyplot as plt

acc_4 = history_4.history['accuracy']
val_acc_4 = history_4.history['val_accuracy']
loss_4 = history_4.history['loss']
val_loss_4 = history_4.history['val_loss']

epochs = range(1,len(acc_4) + 1)

plt.plot(epochs, acc_4, 'bo', label='Training Accuracy')
plt.plot(epochs, val_acc_4, 'b', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss_4, 'bo', label='Training Loss')
plt.plot(epochs, val_loss_4, 'b', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.legend()

plt.show()
```





In [ ]:

```
model_4.evaluate(validation_generator_4, steps=None, max_queue_size=10, workers
```

Comparación de los modelos

Estrategia 1: CNN from scracth El modelo 1 ha sido un modelo de 4 capas en el base model y con 2 capas en el top model, se ocupo ningún método de regularización y data augmentation y con 40 épocas no se pudo llegar a una precisión de más del 60% y se observo un overfitting marcado en el modelo 1.

En comparación al modelo 1, el modelo 2 se ocupo la misma arquitectura de red que el modelo incorporando métodos regularizadores como Dropout, L1, L2 y early stopping, así como también incorporando data augmentation para lograr una mejoría en el entrenamiento del modelo 1, el resultado del modelo 2 fue mejor con menos épocas ya que la presición subió y la pérdida disminuyo y también se redujo el overfitting, aún el modelo 2 no se puede tipificar como óptimo debido a que no se obtiene una precisión mayor o igual que el 80% con los datos de prueba.

Estrategia 2: RED PRE-ENTRENADA Los modelos pre-entrenados Resnetv2 y MobileNetV2 nos permitieron mejorar sustancialmente el problema de clasificación de imagenes, lo que se tuvo que adecuar fue el top model para que nos permitiera poder clasificar las imagenes según nuestro problema de 525 especies de aves, la precisión de estos dos modelos 3 y 4 es por encima del 80% con una perdida cercana a 0, por lo cual podemos afirmar que con estos dos modelos son óptimos para realizar inferencias de nuestro problema en cuestión.

Conclusión: Se puede evidenciar que por un lado ocupar redes pre-entrenadas nos proporciona una ventaja sustancial a la hora de resolver un problema, ya que la reutilización de los pesos de esas redes y sus arquitecturas que han sido probadas por multiples dataset da como resultado mayor precisión en la inferencia y la red entrena en un tiempo mucho menor que cuando no hacemos uso de las redes pre-entrenadas.

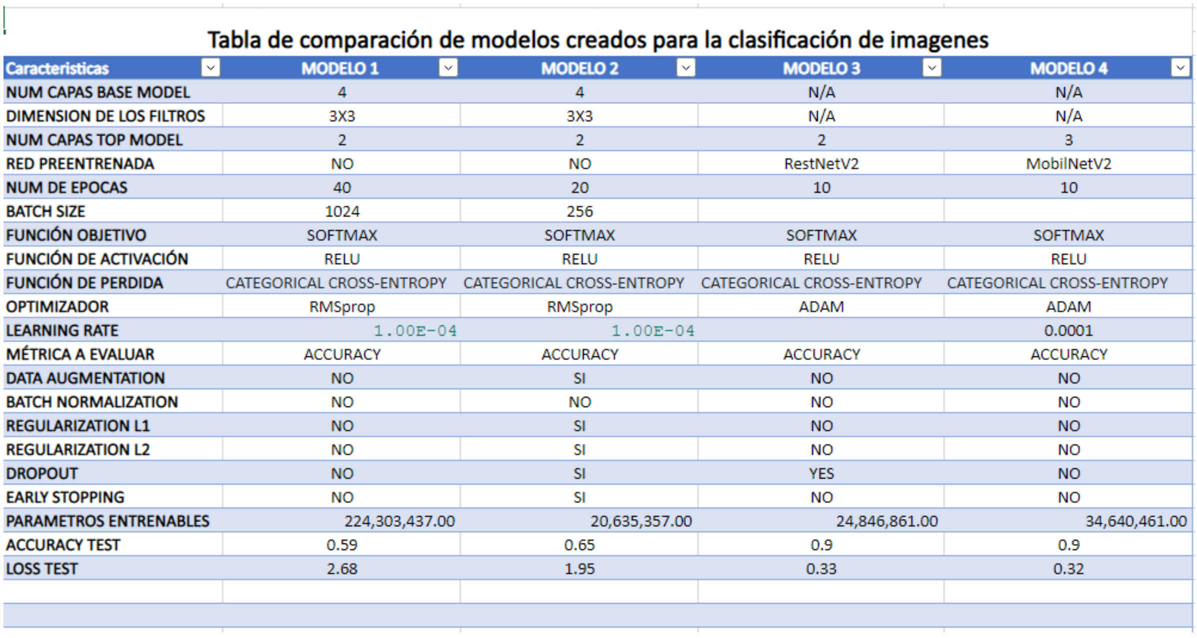


Tabla de comparación de modelos creados para la clasificación de imagenes

Caracteristicas	MODELO 1	MODELO 2	MODELO 3	MODELO 4
NUM CAPAS BASE MODEL	4	4	N/A	N/A
DIMENSION DE LOS FILTROS	3X3	3X3	N/A	N/A
NUM CAPAS TOP MODEL	2	2	2	3
RED PREENTRENADA	NO	NO	RestNetV2	MobilNetV2
NUM DE EPOCAS	40	20	10	10
BATCH SIZE	1024	256		
FUNCIÓN OBJETIVO	SOFTMAX	SOFTMAX	SOFTMAX	SOFTMAX
FUNCIÓN DE ACTIVACIÓN	RELU	RELU	RELU	RELU
FUNCIÓN DE PERDIDA	CATEGORICAL CROSS-ENTROPY	CATEGORICAL CROSS-ENTROPY	CATEGORICAL CROSS-ENTROPY	CATEGORICAL CROSS-ENTROPY
OPTIMIZADOR	RMSprop	RMSprop	ADAM	ADAM
LEARNING RATE	1.00E-04	1.00E-04		0.0001
MÉTRICA A EVALUAR	ACCURACY	ACCURACY	ACCURACY	ACCURACY
DATA AUGMENTATION	NO	SI	NO	NO
BATCH NORMALIZATION	NO	NO	NO	NO
REGULARIZATION L1	NO	SI	NO	NO
REGULARIZATION L2	NO	SI	NO	NO
DROPOUT	NO	SI	YES	NO
EARLY STOPPING	NO	SI	NO	NO
PARAMETROS ENTRENABLES	224,303,437.00	20,635,357.00	24,846,861.00	34,640,461.00
ACCURACY TEST	0.59	0.65	0.9	0.9
LOSS TEST	2.68	1.95	0.33	0.32

In [ ]:

```
#%%shell
!jupyter nbconvert '/content/Copy_of_Copy_of_Proyecto.ipynb' --to html
```

```
[NbConvertApp] Converting notebook /content/Copy_of_Copy_of_Proyecto.ipynb to
html
[NbConvertApp] Writing 2903931 bytes to /content/Copy_of_Copy_of_Proyecto.html
```