Loading in data

In [1]:

```
import numpy as np

from tensorflow.keras.utils import image_dataset_from_directory

from PIL import Image
```

In [23]:

```
train_ds = image_dataset_from_directory(
    "train/train",
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=100,
    image_size=(500, 500),
    shuffle=True,
    seed=1,
    validation_split=.2,
    subset="training",
    interpolation="bicubic",
    follow_links=False,
    crop_to_aspect_ratio=False,
)

validation_ds = image_dataset_from_directory(
    "train/train",
    labels="inferred",
    label_mode="categorical",
    class_names=None,
    color_mode="rgb",
    batch_size=100,
    image_size=(500, 500),
    shuffle=True,
    seed=1,
    validation_split=.2,
    subset="validation",
    interpolation="bicubic",
    follow_links=False,
    crop_to_aspect_ratio=False,
)
```

```
Found 5656 files belonging to 5 classes.
Using 4525 files for training.
Found 5656 files belonging to 5 classes.
Using 1131 files for validation.
```

In [3]:

```
from sklearn.model_selection import train_test_split

from tensorflow.keras import datasets
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential

from tensorflow.keras import layers

from tensorflow.keras.layers import Dense # creates densely connected layer object
from tensorflow.keras.layers import Flatten # takes 2D input and turns into 1D array

from tensorflow.keras.layers import Conv2D # convolution layer
from tensorflow.keras.layers import MaxPooling2D # max pooling layer
```

```
from keras.utils.vis_utils import plot_model
```

In [4]:

```
type(train_ds)
```

Out[4]:

```
tensorflow.python.data.ops.dataset_ops.BatchDataset
```

Base Model

In [5]:

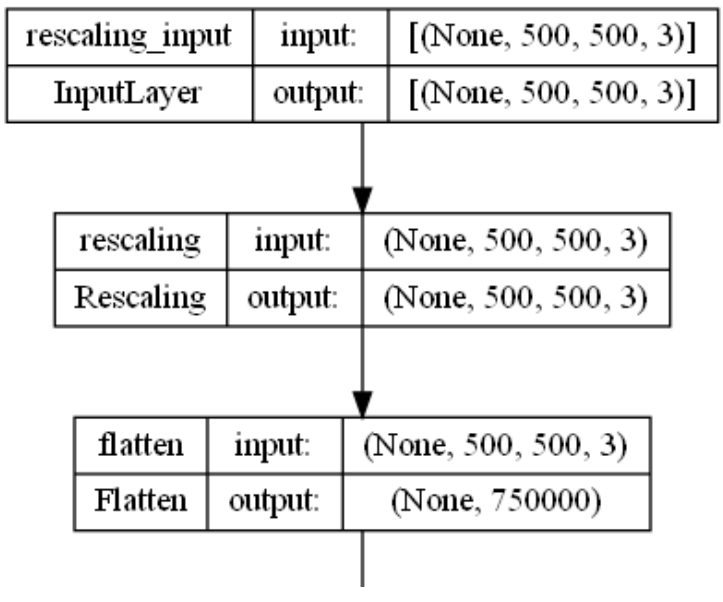
```
rescaling_layer = layers.experimental.preprocessing.Rescaling(
    scale=1. / 255,
    input_shape=(500, 500, 3)
)
```

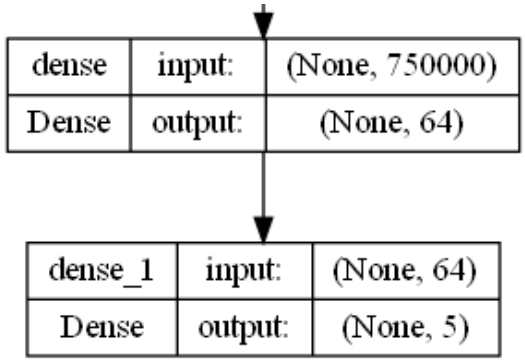
In [6]:

```
model = Sequential()

model.add(rescaling_layer)

model.add(Flatten())
model.add(Dense(64, activation="relu"))
model.add(Dense(5, activation="softmax"))

model.summary()
plot_model(model, show_shapes=True, show_layer_names=True)
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 500, 500, 3)       0

 flatten (Flatten)           (None, 750000)            0

 dense (Dense)               (None, 64)                48000064

 dense_1 (Dense)             (None, 5)                 325

=================================================================
Total params: 48,000,389
Trainable params: 48,000,389
Non-trainable params: 0
_________________________________________________________________
```

Out[6]:





In [8]:

```
model.compile(optimizer='adam', loss='categorical_crossentropy',  metrics=['accuracy'])
```

In [20]:

```
history_cnn = model.fit(train_ds, validation_data=validation_ds, epochs= 40,
                        validation_split = 0.2)
```

```
Epoch 1/40
71/71 [==============================] - 21s 293ms/step - loss: 1.4068 - accuracy: 0.4707
- val_loss: 1.4098 - val_accuracy: 0.4668
Epoch 2/40
71/71 [==============================] - 25s 340ms/step - loss: 1.3965 - accuracy: 0.4707
- val_loss: 1.4011 - val_accuracy: 0.4668
Epoch 3/40
71/71 [==============================] - 25s 334ms/step - loss: 1.3877 - accuracy: 0.4707
- val_loss: 1.3934 - val_accuracy: 0.4668
Epoch 4/40
71/71 [==============================] - 25s 331ms/step - loss: 1.3801 - accuracy: 0.4707
- val_loss: 1.3869 - val_accuracy: 0.4668
Epoch 5/40
71/71 [==============================] - 25s 335ms/step - loss: 1.3736 - accuracy: 0.4707
- val_loss: 1.3815 - val_accuracy: 0.4668
Epoch 6/40
71/71 [==============================] - 25s 339ms/step - loss: 1.3680 - accuracy: 0.4707
- val_loss: 1.3770 - val_accuracy: 0.4668
Epoch 7/40
71/71 [==============================] - 29s 391ms/step - loss: 1.3633 - accuracy: 0.4707
- val_loss: 1.3731 - val_accuracy: 0.4668
Epoch 8/40
71/71 [==============================] - 27s 366ms/step - loss: 1.3593 - accuracy: 0.4707
- val_loss: 1.3699 - val_accuracy: 0.4668
Epoch 9/40
71/71 [==============================] - 26s 353ms/step - loss: 1.3559 - accuracy: 0.4707
- val_loss: 1.3672 - val_accuracy: 0.4668
Epoch 10/40
71/71 [==============================] - 27s 361ms/step - loss: 1.3531 - accuracy: 0.4707
- val_loss: 1.3650 - val_accuracy: 0.4668
Epoch 11/40
71/71 [==============================] - 26s 356ms/step - loss: 1.3507 - accuracy: 0.4707
- val_loss: 1.3632 - val_accuracy: 0.4668
Epoch 12/40
71/71 [==============================] - 27s 362ms/step - loss: 1.3486 - accuracy: 0.4707
- val_loss: 1.3617 - val_accuracy: 0.4668
Epoch 13/40
71/71 [==============================] - 29s 391ms/step - loss: 1.3469 - accuracy: 0.4707
- val_loss: 1.3605 - val_accuracy: 0.4668
Epoch 14/40
71/71 [==============================] - 28s 376ms/step - loss: 1.3455 - accuracy: 0.4707
- val_loss: 1.3596 - val_accuracy: 0.4668
Epoch 15/40
71/71 [==============================] - 27s 364ms/step - loss: 1.3444 - accuracy: 0.4707
- val_loss: 1.3588 - val_accuracy: 0.4668
Epoch 16/40
24/71 [=========>....................] - ETA: 15s - loss: 1.3332 - accuracy: 0.4753

---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
```

```
Input In [20], in <cell line: 1>()
----> 1 history_cnn = model.fit(train_ds, validation_data=validation_ds, epochs= 40,
      2                         validation_split = 0.2)

File ~\anaconda3\envs\Mercury\lib\site-packages\keras\utils\traceback_utils.py:64, in filter_traceback.<locals>.error_handler(*args, **kwargs)
     62 filtered_tb = None
     63 try:
---> 64   return fn(*args, **kwargs)
     65 except Exception as e:  # pylint: disable=broad-except
     66   filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\anaconda3\envs\Mercury\lib\site-packages\keras\engine\training.py:1409, in Model.fit(self, x, y, batch_size, epochs, verbose, callbacks, validation_split, validation_data, shuffle, class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_steps, validation_batch_size, validation_freq, max_queue_size, workers, use_multiprocessing)
   1402 with tf.profiler.experimental.Trace(
   1403     'train',
   1404     epoch_num=epoch,
   1405     step_num=step,
   1406     batch_size=batch_size,
   1407     _r=1):
   1408   callbacks.on_train_batch_begin(step)
-> 1409   tmp_logs = self.train_function(iterator)
   1410   if data_handler.should_sync:
   1411     context.async_wait()

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\util\traceback_utils.py:150, in filter_traceback.<locals>.error_handler(*args, **kwargs)
    148 filtered_tb = None
    149 try:
--> 150   return fn(*args, **kwargs)
    151 except Exception as e:
    152   filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\def_function.py:915, in Function.__call__(self, *args, **kwds)
    912 compiler = "xla" if self._jit_compile else "nonXla"
    914 with OptionalXlaContext(self._jit_compile):
--> 915   result = self._call(*args, **kwds)
    917 new_tracing_count = self.experimental_get_tracing_count()
    918 without_tracing = (tracing_count == new_tracing_count)

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\def_function.py:947, in Function._call(self, *args, **kwds)
    944   self._lock.release()
    945   # In this case we have created variables on the first call, so we run the
    946   # defunned version which is guaranteed to never create variables.
--> 947   return self._stateless_fn(*args, **kwds)  # pylint: disable=not-callable
    948 elif self._stateful_fn is not None:
    949   # Release the lock early so that multiple threads can perform the call
    950   # in parallel.
    951   self._lock.release()

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\function.py:2453, in Function.__call__(self, *args, **kwargs)
   2450 with self._lock:
   2451   (graph_function,
   2452    filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 2453 return graph_function._call_flat(
   2454     filtered_flat_args, captured_inputs=graph_function.captured_inputs)

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\function.py:1860, in ConcreteFunction._call_flat(self, args, captured_inputs, cancellation_manager)
   1856 possible_gradient_type = gradients_util.PossibleTapeGradientTypes(args)
   1857 if (possible_gradient_type == gradients_util.POSSIBLE_GRADIENT_TYPES_NONE
   1858     and executing_eagerly):
   1859   # No tape is watching; skip to running the function.
-> 1860   return self._build_call_outputs(self._inference_function.call(
   1861       ctx, args, cancellation_manager=cancellation_manager))
   1862 forward_backward = self._select_forward_and_backward_functions(
   1863     args,
```

```
   1864     possible_gradient_type,
   1865     executing_eagerly)
   1866 forward_function, args_with_tangents = forward_backward.forward()

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\function.py:497,
in _EagerDefinedFunction.call(self, ctx, args, cancellation_manager)
    495 with _InterpolateFunctionError(self):
    496   if cancellation_manager is None:
--> 497     outputs = execute.execute(
    498         str(self.signature.name),
    499         num_outputs=self._num_outputs,
    500         inputs=args,
    501         attrs=attrs,
    502         ctx=ctx)
    503   else:
    504     outputs = execute.execute_with_cancellation(
    505         str(self.signature.name),
    506         num_outputs=self._num_outputs,
   (...)
    509         ctx=ctx,
    510         cancellation_manager=cancellation_manager)

File ~\anaconda3\envs\Mercury\lib\site-packages\tensorflow\python\eager\execute.py:54, in
quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     52 try:
     53   ctx.ensure_initialized()
---> 54   tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
     55                                       inputs, attrs, num_outputs)
     56 except core._NotOkStatusException as e:
     57   if name is not None:

KeyboardInterrupt:
```

Model 1

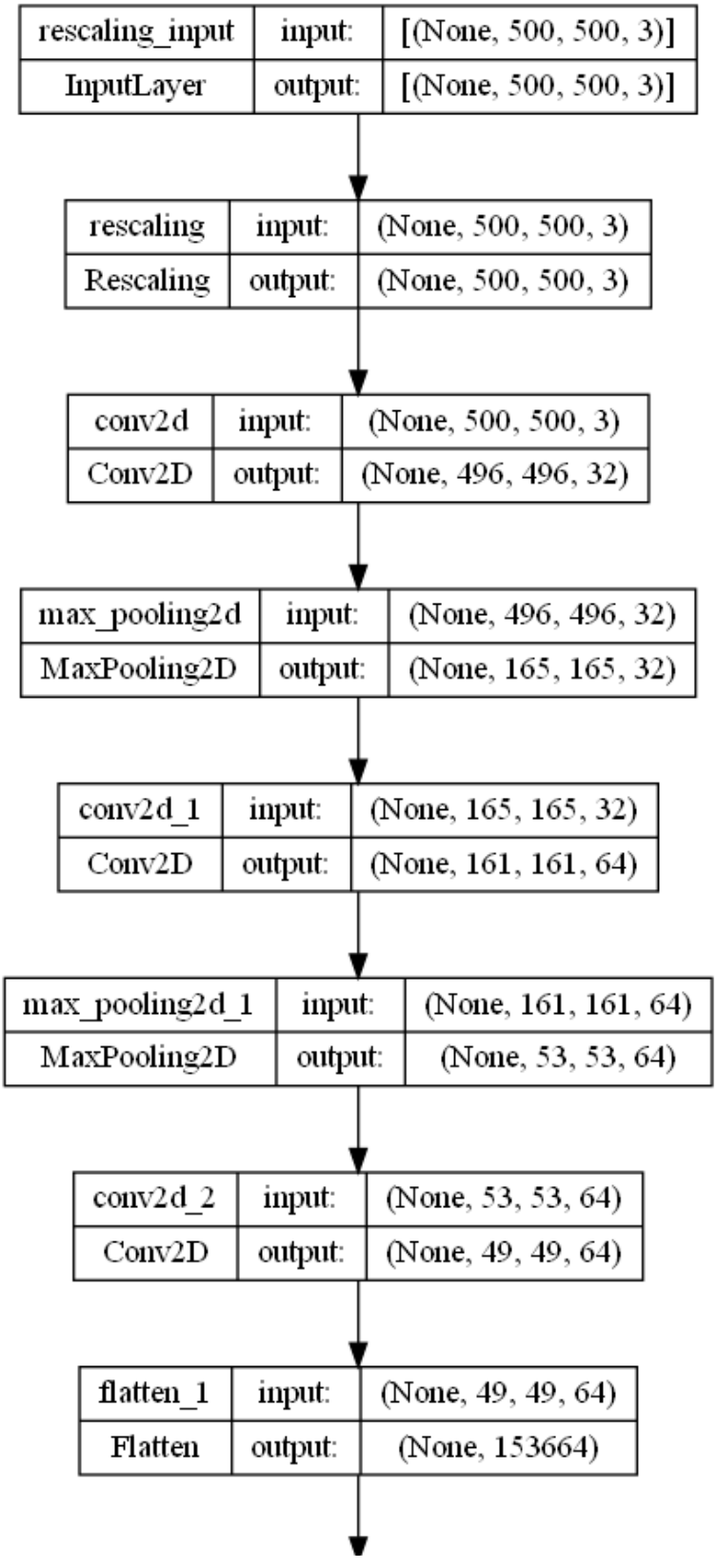
In [13]:

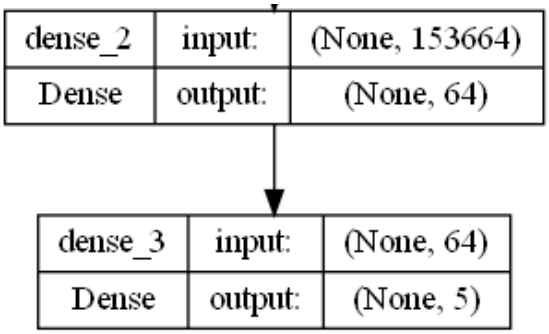
```
model1 = Sequential()

model1.add(rescaling_layer)

model1.add(Conv2D(
filters= 32, kernel_size = (5, 5), activation = "relu",
input_shape = (500, 500, 3)))

model1.add(MaxPooling2D(pool_size = (3, 3)))

model1.add(Conv2D(64, (5, 5), activation = "relu"))
model1.add(MaxPooling2D(pool_size = (3, 3)))
model1.add(Conv2D(64, (5, 5), activation = "relu"))

model1.add(Flatten())
model1.add(Dense(64, activation="relu"))
model1.add(Dense(5, activation="softmax"))

model1.summary()
plot_model(model1, show_shapes=True, show_layer_names=True)
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 500, 500, 3)       0

 conv2d (Conv2D)             (None, 496, 496, 32)      2432

 max_pooling2d (MaxPooling2D  (None, 165, 165, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 161, 161, 64)      51264
```

```
 max_pooling2d_1 (MaxPooling  (None, 53, 53, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 49, 49, 64)        102464

 flatten_1 (Flatten)         (None, 153664)            0

 dense_2 (Dense)             (None, 64)                9834560

 dense_3 (Dense)             (None, 5)                 325

=================================================================
Total params: 9,991,045
Trainable params: 9,991,045
Non-trainable params: 0
_________________________________________________________________
```

Out[13]:





In [21]:

```
model1.compile(optimizer='adam', loss='categorical_crossentropy',  metrics=['accuracy'])
```

In [25]:

```
history_cnn = model1.fit(train_ds, validation_data= validation_ds, epochs= 1,
                         validation_split = 0.2)
```

```
46/46 [==============================] - 710s 14s/step - loss: 0.0396 - accuracy: 0.9947
- val_loss: 3.6043 - val_accuracy: 0.5261
```

In [ ]:

```

```