```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import cv2
import os
from tqdm.auto import tqdm
```

```
!mkdir -p /root/.config/kaggle
```

```
!mv kaggle.json /root/.config/kaggle
```

```
! kaggle competitions download -c dogs-vs-cats
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.config/kaggle/kaggle.
Downloading dogs-vs-cats.zip to /content
 98% 793M/812M [00:03<00:00, 247MB/s]
100% 812M/812M [00:03<00:00, 270MB/s]
```

```
!unzip dogs-vs-cats.zip
```

```
Archive:  dogs-vs-cats.zip
  inflating: sampleSubmission.csv
  inflating: test1.zip
  inflating: train.zip
```

```
!rm  test1.zip sampleSubmission.csv
```

```
!unzip train.zip
```

```
       inflating: train/dog.7802.jpg
       inflating: train/dog.7803.jpg
       inflating: train/dog.7804.jpg
       inflating: train/dog.7805.jpg
       inflating: train/dog.7806.jpg
       inflating: train/dog.7807.jpg
       inflating: train/dog.7808.jpg
       inflating: train/dog.7809.jpg
       inflating: train/dog.781.jpg
       inflating: train/dog.7810.jpg
       inflating: train/dog.7811.jpg
       inflating: train/dog.7812.jpg
       inflating: train/dog.7813.jpg
       inflating: train/dog.7814.jpg
```

```
!mkdir image
!mkdir image/dog
!mkdir image/cat
```

```
import shutil
```

```
source="train/"
dest_dog="image/dog"
dest_cat="image/cat"

for i in os.listdir(source):
  if i.startswith("cat"):
    shutil.copy(source+i,dest_cat)
  elif i.startswith("dog"):
    shutil.copy(source+i,dest_dog)
```

```
len(os.listdir(dest_dog)),len(os.listdir(dest_cat))
```

```
(12500, 12500)
```

```
#idg
batch_size=64
idg = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1/255.0,
                                                      horizontal_flip=True,
                                                      rotation_range=30,
                                                      validation_split=0.1)
```

```
train_idg=idg.flow_from_directory("image",target_size=(150,150),
                                  batch_size=batch_size,
                                  subset="training")

val_idg=idg.flow_from_directory("image",target_size=(150,150),
                                batch_size=batch_size,
                                subset="validation")
```

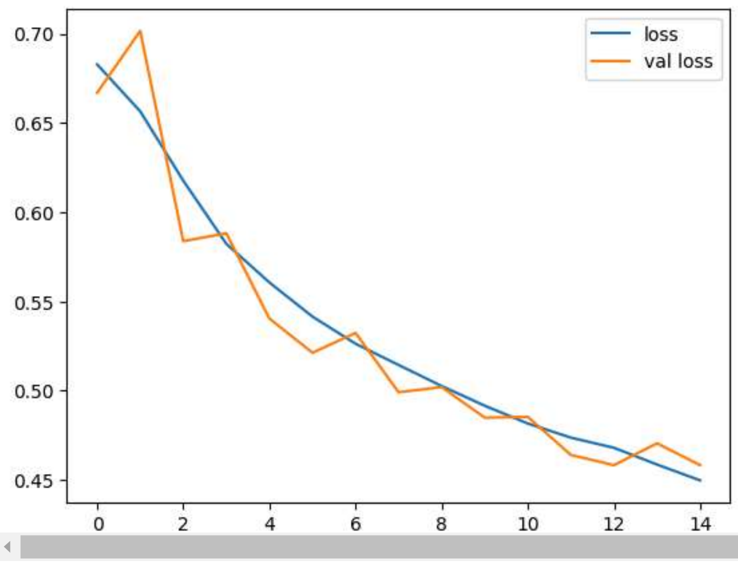
```
Found 22500 images belonging to 2 classes.
Found 2500 images belonging to 2 classes.
```

```
#MODEL BUILDING
model=tf.keras.models.Sequential()
model.add(tf.keras.layers.Input((150,150,3),name="Input layer"))
model.add(tf.keras.layers.Conv2D(filters=16,
                                 kernel_size=(3,3),
                                 padding="valid",
                                 strides=(1,1),
                                 activation="relu",
                                 name="conv1"))  # 150-3+1/1=148

model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2),
                                    strides=(2,2),
                                    padding="valid",
                                    name="pool1")) # 148-2-2/2=74

model.add(tf.keras.layers.Conv2D(filters=32,
                                 kernel_size=(3,3),
                                 strides=(1,1),
                                 padding="valid",
                                 activation="relu",
                                  name="conv2"))   # 74-3+1/1=72

model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2),
                                    strides=(2,2),
                                    padding="valid",
                                    name="pool2")) # 72-2+2//2=36

model.add(tf.keras.layers.Conv2D(filters=32,
                                 kernel_size=(3,3),
                                 strides=(1,1),
                                 padding="valid",
                                 activation="relu",
                                 name="conv3"))   # 36-3+1/1=34

model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2),
                                    strides=(1,1),
                                    padding="valid",
                                    name="pool3"))    # 34-2-2/2=17


model.add(tf.keras.layers.Flatten(name="flat"))
model.add(tf.keras.layers.Dense(128,activation="relu",name="HL1"))
model.add(tf.keras.layers.Dense(2,activation="softmax", name="Output"))



model.compile(loss=tf.keras.losses.categorical_crossentropy,
              optimizer=tf.keras.optimizers.SGD(),
              metrics=["acc"])


model.fit(train_idg,batch_size=batch_size,epochs=15,validation_data=val_idg)
```

```
Epoch 1/15
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class
  self._warn_if_super_not_called()
352/352 ━━━━━━━━━━━━━━━━━━━━ 176s 474ms/step - acc: 0.5398 - loss: 0.6882 - val_acc: 0.6060 - val_loss: 0.6670
Epoch 2/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 162s 452ms/step - acc: 0.6065 - loss: 0.6638 - val_acc: 0.5300 - val_loss: 0.7016
Epoch 3/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 157s 440ms/step - acc: 0.6417 - loss: 0.6307 - val_acc: 0.6964 - val_loss: 0.5839
Epoch 4/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 160s 442ms/step - acc: 0.6893 - loss: 0.5882 - val_acc: 0.6824 - val_loss: 0.5882
Epoch 5/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 205s 457ms/step - acc: 0.7073 - loss: 0.5664 - val_acc: 0.7284 - val_loss: 0.5405
Epoch 6/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 194s 433ms/step - acc: 0.7235 - loss: 0.5438 - val_acc: 0.7420 - val_loss: 0.5213
Epoch 7/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 165s 461ms/step - acc: 0.7439 - loss: 0.5264 - val_acc: 0.7324 - val_loss: 0.5324
Epoch 8/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 199s 449ms/step - acc: 0.7452 - loss: 0.5151 - val_acc: 0.7560 - val_loss: 0.4992
Epoch 9/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 158s 441ms/step - acc: 0.7540 - loss: 0.5012 - val_acc: 0.7564 - val_loss: 0.5021
Epoch 10/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 211s 462ms/step - acc: 0.7585 - loss: 0.4977 - val_acc: 0.7708 - val_loss: 0.4850
Epoch 11/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 160s 447ms/step - acc: 0.7658 - loss: 0.4822 - val_acc: 0.7632 - val_loss: 0.4853
Epoch 12/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 206s 455ms/step - acc: 0.7767 - loss: 0.4678 - val_acc: 0.7808 - val_loss: 0.4641
Epoch 13/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 157s 440ms/step - acc: 0.7801 - loss: 0.4679 - val_acc: 0.7776 - val_loss: 0.4583
Epoch 14/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 205s 449ms/step - acc: 0.7782 - loss: 0.4631 - val_acc: 0.7684 - val_loss: 0.4706
Epoch 15/15
352/352 ━━━━━━━━━━━━━━━━━━━━ 158s 443ms/step - acc: 0.7876 - loss: 0.4475 - val_acc: 0.7872 - val_loss: 0.4583
<keras.src.callbacks.history.History at 0x7d26a0295c60>
```

```
plt.plot(model.history.history['loss'],label="loss")
plt.plot(model.history.history['val_loss'],label="val loss")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7d26a0263820>
```



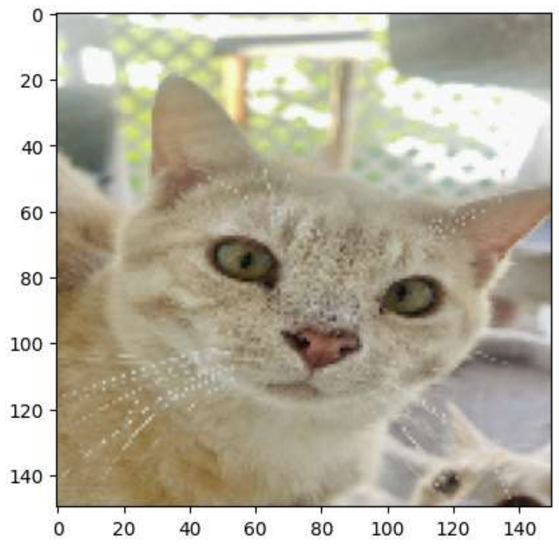
```
!wget /content/7A68FE16-3F13-40AD-A54B-C2C87194FFC8-scaled.jpeg
```

```
/content/7A68FE16-3F13-40AD-A54B-C2C87194FFC8-scaled.jpeg: Scheme missing.
```

```
#del test_image
```

```
test_image=cv2.imread("/content/7A68FE16-3F13-40AD-A54B-C2C87194FFC8-scaled.jpeg")
test_image=cv2.resize(test_image,(150,150))
test_image=cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(test_image)
```

```
<matplotlib.image.AxesImage at 0x7d26a23136d0>
```



```
test_image=test_image/255.0
test_image=np.expand_dims(test_image,axis=0)
```

```
model.predict(test_image)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 710ms/step
array([[0.680062  , 0.31993803]], dtype=float32)
```

```
train_idg.class_indices
```

```
{'cat': 0, 'dog': 1}
```

```
model.save
```

```
keras.src.models.model.Model.save
def save(filepath, overwrite=True, zipped=True, **kwargs)
```

```
- The model's configuration (architecture)
- The model's weights
- The model's optimizer's state (if any)
```

Start coding or generate with AI.