

```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (1.8.3)
Requirement already satisfied: black>=24.10.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (25.12.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle) (6.3.0)
Requirement already satisfied: kagglesdk<1.0,>=0.1.14 in /usr/local/lib/python3.12/dist-packages (from kaggle) (0.1.14)
Requirement already satisfied: mypy>=1.15.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.19.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from kaggle) (25.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle) (5.29.5)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.1)
Requirement already satisfied: types-requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4.20250913)
Requirement already satisfied: types-tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.0.20250809)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.5.0)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (8.3.1)
Requirement already satisfied: mypy_extensions>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (Requirement already satisfied: pathspec>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (1.0.0))
Requirement already satisfied: platformdirs>=2 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (4.5.1)
Requirement already satisfied: pytokens>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (0.3.0)
Requirement already satisfied: typing_extensions>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from mypy>=1.15.0->kaggle) (Requirement already satisfied: libert>=0.6.2 in /usr/local/lib/python3.12/dist-packages (from mypy>=1.15.0->kaggle) (0.7.7))
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.12/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (3.4)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (3.11)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (2025.11.12)
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
from google.colab import files

# 1. Upload the file
files.upload()

# 2. Make a directory named .kaggle
!mkdir -p ~/.kaggle

# 3. Move the kaggle.json file into that directory
!cp kaggle.json ~/.kaggle/

# 4. Change permission (This is required for security)
!chmod 600 ~/.kaggle/kaggle.json
```

kaggle.json(application/json) - 72 bytes, last modified: 1/5/2026 - 100% done
Saving kaggle.json to kaggle (1).json

```
!kaggle datasets list
```

ref	title	size	lastUpdated
neurocipher/heartdisease	Heart Disease	3491	2025-12-11 15:29:14.3
ahmeduzaki/wind-and-solar-energy-production-dataset	Wind & Solar Energy Production Dataset	395372	2026-01-02 21:06:22.7
kundanbedmutha/exam-score-prediction-dataset	Exam Score Prediction Dataset	325454	2025-11-28 07:29:01.0
guriya79/heart-failure-prediction-dataset	Heart Failure Clinical Records Study	4067	2025-12-27 05:13:37.7
neurocipher/student-performance	Student Performance	49705	2025-12-12 12:06:28.9
ishank2005/salary-csv	Salary.csv	392	2025-12-29 15:48:58.2
dansbecker/melbourne-housing-snapshot	Melbourne Housing Snapshot	461423	2018-06-03 12:52:24.0
datasnaek/youtube-new	Trending YouTube Video Statistics	21057546	2019-06-03 00:56:47.1
zynicide/wine-reviews	Wine Reviews	53336293	2017-11-27 17:08:04.7
datasnaek/chess	Chess Game Dataset (Lichess)	2903760	2017-09-04 03:09:09.4
dansbecker/powerlifting-database	powerlifting-database	9277600	2019-04-30 21:07:41.5
residentmario/ramen-ratings	Ramen Ratings	40762	2018-01-11 16:04:39.8
rstatman/188-million-us-wildfires	1.88 Million US Wildfires	176270559	2020-05-12 21:03:49.2
nolanbconaway/pitchfork-data	18,393 Pitchfork Reviews	34891456	2017-01-13 04:18:10.1
nasa/kepler-exoplanet-search-results	Kepler Exoplanet Search Results	1215549	2017-10-10 18:26:59.4
residentmario/things-on-reddit	Things on Reddit	16773230	2017-10-26 14:10:15.1
jpmiller/publicassistance	US Public Food Assistance 1 - WIC	304041	2023-04-17 20:01:05.9
mohitsinghrajput1307/practice-dataset1	Customer Purchase Prediction Dataset	286	2026-01-02 13:05:59.9
wardabilal spotify-global-music-dataset-20092025	Spotify Global Music Dataset (2009-2025)	1289021	2025-11-11 09:43:05.9
suvidyasonawane/student-performance-dataset	Student Performance Dataset	81679	2025-12-25 05:27:51.4

Start coding or [generate](#) with AI.

Importing dog vs cat

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

401 Client Error: Unauthorized for url: <https://api.kaggle.com/v1/competitions.Competition ApiService/DownloadDataFiles>

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

401 Client Error: Unauthorized for url: <https://api.kaggle.com/v1/competitions.Competition ApiService/DownloadDataFiles>

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

401 Client Error: Unauthorized for url: <https://api.kaggle.com/v1/competitions.Competition ApiService/DownloadDataFiles>

```
!cat ~/.kaggle/kaggle.json
```

```
{"username": "anshraninga", "key": "KGAT_94a2b629c40a00542ae20fb9a5c24ef1"}
```

```
!kaggle competitions list --search "dogs"
```

401 Client Error: Unauthorized for url: <https://api.kaggle.com/v1/competitions.Competition ApiService/ListCompetitions>

```
import os
```

```
# FORCE the system to use your new key
os.environ['KAGGLE_USERNAME'] = "anshraninga"
os.environ['KAGGLE_KEY'] = "KGAT_94a2b629c40a00542ae20fb9a5c24ef1"
```

```
# Check if it works now
```

```
!kaggle competitions list --search "dogs"
```

401 Client Error: Unauthorized for url: <https://api.kaggle.com/v1/competitions.Competition ApiService/ListCompetitions>

```
# 1. Uninstall the old version
!pip uninstall -y kaggle
```

```
# 2. Install the latest version
!pip install kaggle==1.6.14
```

```
# 3. FORCE the environment variables again (just to be safe)
```

```
import os
os.environ['KAGGLE_USERNAME'] = "anshraninga"
os.environ['KAGGLE_KEY'] = "KGAT_94a2b629c40a00542ae20fb9a5c24ef1" # Ensure NO spaces at the end!
```

```
# 4. Try the list command
```

```
!kaggle competitions list --search "dogs"
```

```
Found existing installation: kaggle 1.8.3
```

```
Uninstalling kaggle-1.8.3:
```

```
Successfully uninstalled kaggle-1.8.3
```

```
Collecting kaggle==1.6.14
```

```
Using cached kaggle-1.6.14-py3-none-any.whl
```

```
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (1.17.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (2025.11.12)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (2.9.0.post0)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (2.32.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (4.67.1)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (2.5.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle==1.6.14) (6.3.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from bleach->kaggle==1.6.14) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.12/dist-packages (from python-slugify->kaggle==1.6)
Requirement already satisfied: charset_normalizer<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle==1.6.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle==1.6.14) (3.11)
Installing collected packages: kaggle
Successfully installed kaggle-1.6.14
401 - Unauthorized - Unauthenticated
```

```
# 1. Install the LATEST version of Kaggle (to support KGAT keys)
!pip install --upgrade kaggle

# 2. FORCE the credentials (using your latest ...ef1 key)
import os
os.environ['KAGGLE_USERNAME'] = "anshraninga"
# I copied this exactly from your screenshot:
os.environ['KAGGLE_KEY'] = "KGAT_94a2b629c40a00542ae20fb9a5c24ef1"

# 3. Test it
!kaggle competitions list --search "dogs"

Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (1.6.14)
Collecting kaggle
  Using cached kaggle-1.8.3-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: black>=24.10.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (25.12.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle) (6.3.0)
Requirement already satisfied: kagglestdk<1.0,>=0.1.14 in /usr/local/lib/python3.12/dist-packages (from kaggle) (0.1.14)
Requirement already satisfied: mypy>=1.15.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.19.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from kaggle) (25.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle) (5.29.5)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.1)
Requirement already satisfied: types-requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4.20250913)
Requirement already satisfied: types-tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.0.20250809)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.5.0)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (8.3.1)
Requirement already satisfied: mypy-extensions>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (Requirement already satisfied: pathspec>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (1.0.0)
Requirement already satisfied: platformdirs>=2 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (4.5.1)
Requirement already satisfied: pytokens>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from black>=24.10.0->kaggle) (0.3.0)
Requirement already satisfied: typing_extensions>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from mypy>=1.15.0->kaggle) (Requirement already satisfied: librt>=0.6.2 in /usr/local/lib/python3.12/dist-packages (from mypy>=1.15.0->kaggle) (0.7.7)
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.12/dist-packages (from python-slugify->kaggle) (1.3
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (3.4.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (3.11)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->kaggle) (2025.11.12
Using cached kaggle-1.8.3-py3-none-any.whl (102 kB)
Installing collected packages: kaggle
  Attempting uninstall: kaggle
    Found existing installation: kaggle 1.6.14
    Uninstalling kaggle-1.6.14...
      Successfully uninstalled kaggle-1.6.14
Successfully installed kaggle-1.8.3
401 Client Error: Unauthorized for url: https://api.kaggle.com/v1/competitions.Competition ApiService>ListCompetitions
```

```
import zipfile
print("Extracting images...")
with zipfile.ZipFile("train.zip", 'r') as zip_ref:
    zip_ref.extractall("dataset")
print("Done! Your images are in the 'dataset' folder.")

Extracting images...
Done! Your images are in the 'dataset' folder.
```

```
!ls
dataset 'kaggle (1).json' kaggle.json sample_data train.zip
```

```
import os

#counting the number of files in train folder

path, dirs, files = next(os.walk("/content/dataset/train"))
file_count = len(files)
print(file_count)

25000
```

Start coding or generate with AI.

printing the name of images

```
file_names = os.listdir("/content/dataset/train")
print(file_names)

['dog.6388.jpg', 'dog.1027.jpg', 'dog.9850.jpg', 'dog.10135.jpg', 'dog.3417.jpg', 'dog.1854.jpg', 'cat.4544.jpg', 'dog.2576.jpg']
```

Start coding or [generate](#) with AI.

Importing libraries

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.model_selection import train_test_split
from google.colab.patches import cv2_imshow
```

```
#displaying dog image

img = mpimg.imread('/content/dataset/train/dog.8298.jpg')
imgplot = plt.imshow(img)
plt.show()
```



```
#displaying cat image

img = mpimg.imread('/content/dataset/train/cat.1000.jpg')
imgplot = plt.imshow(img)
plt.show()
```



```
file_names = os.listdir("/content/dataset/train")
```

```
for i in range(10):
    name = file_names[i]
```

```
print(name[0:9])
```

```
dog.6388.
dog.1027.
dog.9850.
dog.10135
dog.3417.
dog.1854.
cat.4544.
dog.2576.
cat.743.j
cat.11755
```

```
file_names = os.listdir("/content/dataset/train")
```

```
dog_count = 0
cat_count = 0
```

```
for img_file in file_names:
```

```
    name = img_file[0:3]
    if img_file[0:3] == 'dog':
        dog_count += 1
    elif img_file[0:3] == 'cat':
        cat_count += 1
```

```
print(dog_count)
print(cat_count)
```

```
12500
12500
```

Start coding or [generate](#) with AI.

Resizing all images

```
#creating a directory for resized images
```

```
import os
os.makedirs('/content/dataset/resized', exist_ok=True)
```

```
original_folder = '/content/dataset/train/'
resized_folder = '/content/dataset/resized/'
```

```

for i in range(2000):

    filename = os.listdir(original_folder)[i]
    img_path = os.path.join(original_folder, filename)
    img = Image.open(img_path)
    img = img.resize((224, 224))
    img.save(os.path.join(resized_folder, filename))
    img = img.convert('RGB')
    img.save(os.path.join(resized_folder, filename))

    newImgPath = os.path.join(resized_folder, filename)
    newImg = Image.open(newImgPath)
    newImg.show()

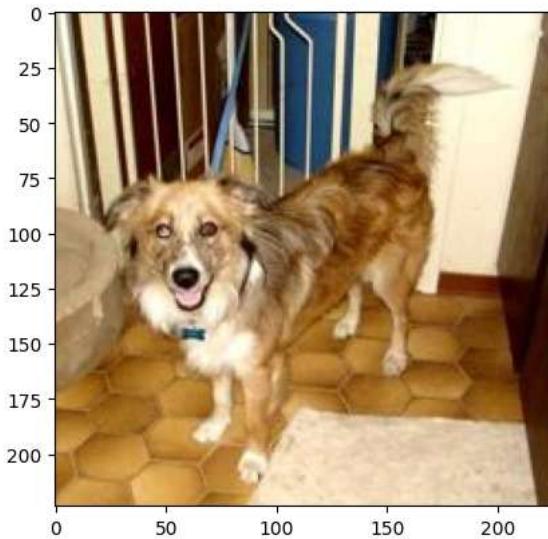
```

#displaying resized dog image

```

# The file 'dog.8298.jpg' was not among the first 2000 images processed by the previous cell.
# Changing to 'dog.4293.jpg', which was the last image processed according to the kernel state.
img = mpimg.imread('/content/dataset/resized/dog.4293.jpg')
imgplot = plt.imshow(img)
plt.show()

```



Start coding or [generate](#) with AI.

Creating labels for resized images of dogs and cats

```

#Cat--0, Dog--1
#creating a for loop to assign labels

file_names = os.listdir("/content/dataset/resized")

labels = []
for img_file in file_names:
    if img_file[0:3] == 'dog':
        labels.append(1)
    elif img_file[0:3] == 'cat':
        labels.append(0)

```

```

print(file_names[0:10])
print(len(file_names))

```

```

['dog.6388.jpg', 'dog.1027.jpg', 'dog.9850.jpg', 'dog.10135.jpg', 'dog.3417.jpg', 'dog.1854.jpg', 'cat.4544.jpg', 'dog.2576.jpg'
200

```

```

print(labels[0:10])
print(len(labels))

```

```
[1, 1, 1, 1, 1, 1, 0, 1, 0, 0]  
2000
```

```
#counting the images of dogs and cats out of 2000 images  
  
values, counts = np.unique(labels, return_counts=True)  
print(values)  
print(counts)
```

```
[0 1]  
[ 955 1045]
```

Start coding or generate with AI.

```
#converting all the resized images to numpy arrays
```

```
import cv2  
import glob
```

```
image_directory = '/content/dataset/resized'  
image_extension = ['png', 'jpg']
```

```
files = []  
  
# Correcting the glob pattern to look inside the directory  
[files.append(glob.glob(os.path.join(image_directory, '*' + e))) for e in image_extension]  
  
dog_cat_images = np.array([cv2.imread(file) for file in files])
```

```
print(dog_cat_images)
```

```
[ 85 165 210]  
...  
[ 18  55  81]  
[ 25  65  90]  
[ 16  56  81]]  
  
[[ 61 141 188]  
[ 77 157 204]  
[ 87 167 214]  
...  
[ 18  55  81]  
[ 24  64  89]  
[ 14  54  79]]
```

```
type(dog_cat_images)
```

```
numpy.ndarray
```

```
print(dog_cat_images.shape)
```

```
(2000, 224, 224, 3)
```

```
X = dog_cat_images  
y = np.asarray(labels)
```

```
Start coding or generate with AI.
```

Train test split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)
```

```
print(X_train.shape)  
print(X_test.shape)  
print(y_train.shape)  
print(y_test.shape)
```

```
(1600, 224, 224, 3)  
(400, 224, 224, 3)  
(1600,)  
(400,)
```

```
#scalling the data
```

```
X_train_scaled = X_train / 255  
X_test_scaled = X_test / 255
```

```
print(X_train_scaled)
```

```

...
[[0.41568627 0.54901961 0.63921569]
[0.35294118 0.50196078 0.59607843]
[0.4       0.55686275 0.65490196]]]

...
[[0.41176471 0.52941176 0.63529412]
[0.30980392 0.42745098 0.53333333]
[0.43921569 0.55686275 0.6627451 ]]

...
[[0.46666667 0.50980392 0.5254902 ]
[0.54117647 0.58431373 0.6       ]
[0.46666667 0.50980392 0.5254902 ]]

[[0.26666667 0.39215686 0.50588235]
[0.3372549 0.4627451 0.57647059]
[0.63137255 0.74901961 0.85490196]]

...
[[0.49803922 0.54117647 0.55686275]
[0.54901961 0.59215686 0.60784314]
[0.56862745 0.61176471 0.62745098]]]

[[0.41176471 0.5372549 0.65098039]
[0.51372549 0.63921569 0.75294118]
[0.61176471 0.72941176 0.83529412]]

...
[[0.50980392 0.55294118 0.56862745]
[0.50196078 0.54509804 0.56078431]
[0.50784311 0.55089020 0.56666667]]

```

Start coding or [generate](#) with AI.

Building the neural network

```

import os
# Force TensorFlow to use the legacy Keras 2
os.environ["TF_USE_LEGACY_KERAS"] = "1"

import tensorflow as tf
import tensorflow_hub as hub
import tf_keras as keras

mobilenet_model = 'https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4'

pretrained_model = hub.KerasLayer(mobilenet_model, input_shape=(224, 224, 3), trainable=False)

```

Start coding or [generate](#) with AI.

```

import tensorflow as tf

# Load MobileNetV2 directly from Keras applications
# include_top=False removes the final classification layer (acting as a feature vector)
# pooling='avg' adds a GlobalAveragePooling layer so the output is a 1D vector
base_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg'
)

base_model.trainable = False
num_of_classes = 2

model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.Dense(num_of_classes)
])

model.summary()

```

Model: "sequential_10"

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 1280)	2,257,984
dense_10 (Dense)	(None, 2)	2,562

Total params: 2,260,546 (8.62 MB)
Trainable params: 2,562 (10.01 KB)
Non-trainable params: 2,257,984 (8.61 MB)

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['acc']
)
```

```
model.fit(X_train_scaled, y_train, epochs=5)
```

```
Epoch 1/5
50/50 73s 1s/step - acc: 0.7057 - loss: 0.5590
Epoch 2/5
50/50 74s 1s/step - acc: 0.9698 - loss: 0.0915
Epoch 3/5
50/50 58s 1s/step - acc: 0.9768 - loss: 0.0688
Epoch 4/5
50/50 83s 1s/step - acc: 0.9858 - loss: 0.0555
Epoch 5/5
50/50 58s 1s/step - acc: 0.9838 - loss: 0.0471
<keras.src.callbacks.history.History at 0x7c71ca8fe180>
```

```
score, acc = model.evaluate(X_test_scaled, y_test)
print('Test score:', score)
print('Test accuracy:', acc)

13/13 19s 1s/step - acc: 0.9774 - loss: 0.0816
Test score: 0.06759966909885406
Test accuracy: 0.9800000190734863
```

Start coding or [generate](#) with AI.

Predictive System

```
input_image_path = input("Enter the path to the input image: ")

input_image = cv2.imread(input_image_path)

cv2.imshow(input_image)

input_image_resized = cv2.resize(input_image, (224, 224))

input_image_scaled = input_image_resized / 255

input_image_reshaped = np.reshape(input_image_scaled, [1, 224, 224, 3])
input_prediction = model.predict(input_image_reshaped)
input_pred_label = np.argmax(input_prediction)

if input_pred_label == 0:
    print("The image is a cat")
else:
    print("The image is a dog")
```

Enter the path to the input image: /dog.jpg

