

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
IMG_SIZE = 244
BATCH_SIZE = 32
```

```
train_datagen =
ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/archive (1)/dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)
```

```
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/archive (1)/dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 312 images belonging to 4 classes.
Found 77 images belonging to 4 classes.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Define the model
```

```
model = keras.Sequential([
    layers.Conv2D(32,
(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid') #output layer
])
```

```

#compile the model
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['ac
curacy'])

model.fit(train_generator,validation_data=val_generator,epochs=1)

10/10 [=====] - 50s 5s/step - loss: 0.5714 -
accuracy: 0.7500 - val_loss: 0.5706 - val_accuracy: 0.7500

<keras.src.callbacks.History at 0x7ebe182e2170>

model.save("Model.h5","label.txt")

/usr/local/lib/python3.10/dist-packages/keras/src/engine/
training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(

model=load_model('/content/Model.h5')

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('/content/Model.h5')
test_image_path = '/content/drive/MyDrive/archive
(1)/dataset/normal/1034_left.jpg'
img = image.load_img(test_image_path, target_size=(244,244))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array,axis=0)

img_array /=244.
prediction = model.predict(img_array)
print(prediction)

1/1 [=====] - 0s 130ms/step
[[0.28648528]]

if prediction < 0.5:
    print("it is the eye disease")
else:
    print("it is not a eye disease")

it is the eye disease

```