If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

* Zero configuration required
* Access to GPUs free of charge
* Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](https://www.youtube.com/watch?v=inN8seMm7UI) to learn more, or just get started below!

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

seconds\_in\_a\_day = 24 \* 60 \* 60  
seconds\_in\_a\_day

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

seconds\_in\_a\_week = 7 \* seconds\_in\_a\_day  
seconds\_in\_a\_week

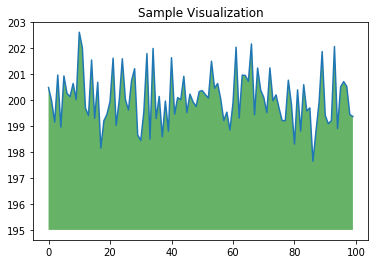
604800

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To learn more, see [Overview of Colab](/notebooks/basic_features_overview.ipynb). To create a new Colab notebook you can use the File menu above, or use the following link: [create a new Colab notebook](http://colab.research.google.com#create=true).

Colab notebooks are Jupyter notebooks that are hosted by Colab. To learn more about the Jupyter project, see [jupyter.org](https://www.jupyter.org).

With Colab you can harness the full power of popular Python libraries to analyze and visualize data. The code cell below uses **numpy** to generate some random data, and uses **matplotlib** to visualize it. To edit the code, just click the cell and start editing.

import numpy as np  
from matplotlib import pyplot as plt  
  
ys = 200 + np.random.randn(100)  
x = [x for x in range(len(ys))]  
  
plt.plot(x, ys, '-')  
plt.fill\_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)  
  
plt.title("Sample Visualization")  
plt.show()



You can import your own data into Colab notebooks from your Google Drive account, including from spreadsheets, as well as from Github and many other sources. To learn more about importing data, and how Colab can be used for data science, see the links below under [Working with Data](#working-with-data).

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just [a few lines of code](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb). Colab notebooks execute code on Google's cloud servers, meaning you can leverage the power of Google hardware, including [GPUs and TPUs](#using-accelerated-hardware), regardless of the power of your machine. All you need is a browser.

Colab is used extensively in the machine learning community with applications including:

* Getting started with TensorFlow
* Developing and training neural networks
* Experimenting with TPUs
* Disseminating AI research
* Creating tutorials

To see sample Colab notebooks that demonstrate machine learning applications, see the [machine learning examples](#machine-learning-examples) below.

* [Overview of Colaboratory](/notebooks/basic_features_overview.ipynb)
* [Guide to Markdown](/notebooks/markdown_guide.ipynb)
* [Importing libraries and installing dependencies](/notebooks/snippets/importing_libraries.ipynb)
* [Saving and loading notebooks in GitHub](https://colab.research.google.com/github/googlecolab/colabtools/blob/main/notebooks/colab-github-demo.ipynb)
* [Interactive forms](/notebooks/forms.ipynb)
* [Interactive widgets](/notebooks/widgets.ipynb)
* [Loading data: Drive, Sheets, and Google Cloud Storage](/notebooks/io.ipynb)
* [Charts: visualizing data](/notebooks/charts.ipynb)
* [Getting started with BigQuery](/notebooks/bigquery.ipynb)

### Machine Learning Crash Course

These are a few of the notebooks from Google's online Machine Learning course. See the [full course website](https://developers.google.com/machine-learning/crash-course/) for more.

* [Intro to Pandas DataFrame](https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/pandas_dataframe_ultraquick_tutorial.ipynb)
* [Linear regression with tf.keras using synthetic data](https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/linear_regression_with_synthetic_data.ipynb)
* [TensorFlow with GPUs](/notebooks/gpu.ipynb)
* [TensorFlow with TPUs](/notebooks/tpu.ipynb)
* [NeMo Voice Swap](https://colab.research.google.com/github/NVIDIA/NeMo/blob/stable/tutorials/VoiceSwapSample.ipynb): Use Nvidia's NeMo conversational AI Toolkit to swap a voice in an audio fragment with a computer generated one.
* [Retraining an Image Classifier](https://tensorflow.org/hub/tutorials/tf2_image_retraining): Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
* [Text Classification](https://tensorflow.org/hub/tutorials/tf2_text_classification): Classify IMDB movie reviews as either *positive* or *negative*.
* [Style Transfer](https://tensorflow.org/hub/tutorials/tf2_arbitrary_image_stylization): Use deep learning to transfer style between images.
* [Multilingual Universal Sentence Encoder Q&A](https://tensorflow.org/hub/tutorials/retrieval_with_tf_hub_universal_encoder_qa): Use a machine learning model to answer questions from the SQuAD dataset.
* [Video Interpolation](https://tensorflow.org/hub/tutorials/tweening_conv3d): Predict what happened in a video between the first and the last frame.

#install kaggle  
!pip install -q kaggle

#upload the json file   
from google.colab import files  
files.upload()

#upload the json file   
from google.colab import files  
files.upload()

<IPython.core.display.HTML object>

Saving kaggle.json to kaggle.json

{'kaggle.json': b'{"username":"priyavj12","key":"cbe838aead4fb406ad2c3653204672b0"}'}

#make directory for kaggle  
!mkdir ~/.kaggle

#copy the kaggle.json to folder created  
!cp kaggle.json ~/.kaggle/

#permission for the json to act  
! chmod 600 ~/.kaggle/kaggle.json

#to list all datasets in the kaggle  
!kaggle datasets list

ref title size lastUpdated downloadCount voteCount usabilityRating   
-------------------------------------------------------------------- -------------------------------------------------- ----- ------------------- ------------- --------- ---------------   
arnabchaki/data-science-salaries-2023 Data Science Salaries 2023 💸 25KB 2023-04-13 09:55:16 25604 709 1.0   
tawfikelmetwally/automobile-dataset Car information dataset 6KB 2023-05-28 18:26:48 1081 32 0.9411765   
fatihb/coffee-quality-data-cqi Coffee Quality Data (CQI May-2023) 22KB 2023-05-12 13:06:39 3818 82 1.0   
mohithsairamreddy/salary-data Salary\_Data 17KB 2023-05-18 14:05:19 3086 55 0.88235295   
mauryansshivam/netflix-ott-revenue-and-subscribers-csv-file Netflix OTT Revenue and Subscribers (CSV File) 2KB 2023-05-13 17:40:23 1871 47 1.0   
omarsobhy14/mcdonalds-revenue 🍟💰From Flipping Burgers to Billions: McDonald's 565B 2023-06-01 23:22:49 354 22 1.0   
zsinghrahulk/rice-pest-and-diseases Rice - Pest and Diseases 312KB 2023-06-01 08:57:29 292 22 1.0   
iammustafatz/diabetes-prediction-dataset Diabetes prediction dataset 734KB 2023-04-08 06:11:45 13382 194 1.0   
vstacknocopyright/fruit-and-vegetable-prices Fruit and Vegetable Prices 1KB 2023-06-02 06:17:43 444 25 0.7647059   
bilalwaseer/microsoft-stocks-from-1986-to-2023 Microsoft Stocks from 1986 to 2023 120KB 2023-05-16 10:07:28 592 21 0.9411765   
aryansingh0909/earthquakes-alerts-tweets-dataset-daily-updated Earthquakes Alerts Tweets Dataset (Daily Updated) 436KB 2023-06-03 23:04:47 359 25 1.0   
darshanprabhu09/stock-prices-for Stock prices of Amazon , Microsoft , Google, Apple 85KB 2023-05-16 15:17:16 1608 40 1.0   
rajkumarpandey02/2023-world-population-by-country World Population by Country 38KB 2023-06-01 06:10:41 644 23 1.0   
danishjmeo/karachi-housing-prices-2023 Karachi\_Housing\_Prices\_2023 1MB 2023-06-01 07:08:13 310 21 0.88235295   
adityaramachandran27/world-air-quality-index-by-city-and-coordinates World Air Quality Index by City and Coordinates 372KB 2023-05-07 07:29:26 1547 38 1.0   
dansbecker/melbourne-housing-snapshot Melbourne Housing Snapshot 451KB 2018-06-05 12:52:24 116992 1277 0.7058824   
pushpakhinglaspure/oscar-dataset Oscar Academy Award-winning films 1927-2022 161KB 2023-05-21 18:14:44 393 23 1.0   
aryansingh0909/weekly-patent-application-granted Patent Application Granted Dataset 6MB 2023-06-01 19:04:40 117 23 1.0   
utkarshx27/heart-disease-diagnosis-dataset Heart Disease Prediction Dataset 3KB 2023-05-26 09:33:13 866 27 1.0   
shreyanshverma27/water-quality-testing Water Quality Testing 4KB 2023-05-16 05:22:17 915 27 0.9411765

#download the dataset using api command  
! kaggle datasets download -d akash2907/bird-species-classification

Downloading bird-species-classification.zip to /content  
100% 1.37G/1.37G [00:47<00:00, 33.0MB/s]  
100% 1.37G/1.37G [00:47<00:00, 31.2MB/s]

!unzip /content/bird-species-classification.zip

Archive: /content/bird-species-classification.zip  
 inflating: test\_data/test\_data/blasti/DSC\_6396.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6397.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6398.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6399.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6400.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6401.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6402.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6403.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6405.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6406.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6407.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6408.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6409.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6410.jpg   
 inflating: test\_data/test\_data/blasti/DSC\_6411.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4587.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4588.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4589.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4590.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4591.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4592.jpg   
 inflating: test\_data/test\_data/bonegl/DSC\_4593.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0473.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0474.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0475.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0477.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0478.jpg   
 inflating: test\_data/test\_data/brhkyt/D72\_0479.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10310.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10311.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10312.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10313.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10314.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10317.jpg   
 inflating: test\_data/test\_data/cbrtsh/\_D32\_10318.jpg   
 inflating: test\_data/test\_data/cmnmyn/DSC\_2443.jpg   
 inflating: test\_data/test\_data/cmnmyn/DSC\_4681.jpg   
 inflating: test\_data/test\_data/cmnmyn/DSC\_5137.jpg   
 inflating: test\_data/test\_data/cmnmyn/DSC\_7625.jpg   
 inflating: test\_data/test\_data/cmnmyn/P1050277.jpg   
 inflating: test\_data/test\_data/cmnmyn/\_D32\_12426.jpg   
 inflating: test\_data/test\_data/cmnmyn/\_D32\_12427.jpg   
 inflating: test\_data/test\_data/cmnmyn/\_D32\_12428.jpg   
 inflating: test\_data/test\_data/gretit/11620454726\_31a35c26da\_o.jpg   
 inflating: test\_data/test\_data/gretit/11776135285\_ccf938fa2e\_o.jpg   
 inflating: test\_data/test\_data/gretit/11905645146\_6a5d4ff9f9\_o.jpg   
 inflating: test\_data/test\_data/gretit/8537646712\_0b282c4c6a\_o.jpg   
 inflating: test\_data/test\_data/gretit/D72\_0693.jpg   
 inflating: test\_data/test\_data/gretit/D72\_0694.jpg   
 inflating: test\_data/test\_data/gretit/D72\_0695.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6359.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6362.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6364.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6368.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6403.jpg   
 inflating: test\_data/test\_data/hilpig/DSC\_6404.jpg   
 inflating: test\_data/test\_data/hilpig/P1000319.jpg   
 inflating: test\_data/test\_data/hilpig/P1000323.jpg   
 inflating: test\_data/test\_data/hilpig/P1050044.jpg   
 inflating: test\_data/test\_data/hilpig/P1050045.jpg   
 inflating: test\_data/test\_data/himbul/11602064885\_3826df4500\_o.jpg   
 inflating: test\_data/test\_data/himbul/11619942574\_42f23a7b05\_o.jpg   
 inflating: test\_data/test\_data/himbul/11620097904\_33fa8ee878\_o.jpg   
 inflating: test\_data/test\_data/himbul/5857371972\_ed4f88b275\_o.jpg   
 inflating: test\_data/test\_data/himbul/6154954471\_eefe6e00d1\_o.jpg   
 inflating: test\_data/test\_data/himbul/D72\_0688.jpg   
 inflating: test\_data/test\_data/himbul/D72\_0691.jpg   
 inflating: test\_data/test\_data/himbul/DSC08357.jpg   
 inflating: test\_data/test\_data/himbul/DSC\_3230.jpg   
 inflating: test\_data/test\_data/himbul/DSC\_7439.jpg   
 inflating: test\_data/test\_data/himbul/DSC\_7602.jpg   
 inflating: test\_data/test\_data/himbul/P1060368.jpg   
 inflating: test\_data/test\_data/himbul/P1060369.jpg   
 inflating: test\_data/test\_data/himbul/\_D32\_10242.jpg   
 inflating: test\_data/test\_data/himbul/\_D32\_10245.jpg   
 inflating: test\_data/test\_data/himbul/\_D32\_10251.jpg   
 inflating: test\_data/test\_data/himbul/\_D32\_10319.jpg   
 inflating: test\_data/test\_data/himbul/\_D32\_10320.jpg   
 inflating: test\_data/test\_data/himbul/\_DSC9446.jpg   
 inflating: test\_data/test\_data/himgri/IMG\_5384.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5385.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5386.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5387.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5459.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5460.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5487.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5488.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5489.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5490.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5491.JPG   
 inflating: test\_data/test\_data/himgri/IMG\_5492.JPG   
 inflating: test\_data/test\_data/himgri/P1020742.jpg   
 inflating: test\_data/test\_data/himgri/P1020743.jpg   
 inflating: test\_data/test\_data/himgri/P1020746.jpg   
 inflating: test\_data/test\_data/himgri/P1020748.jpg   
 inflating: test\_data/test\_data/himgri/P1030055.jpg   
 inflating: test\_data/test\_data/himgri/P1060776.jpg   
 inflating: test\_data/test\_data/himgri/P1060777.jpg   
 inflating: test\_data/test\_data/himgri/P1060778.jpg   
 inflating: test\_data/test\_data/hsparo/100\_5573.JPG   
 inflating: test\_data/test\_data/hsparo/100\_5585.JPG   
 inflating: test\_data/test\_data/hsparo/100\_5586.JPG   
 inflating: test\_data/test\_data/hsparo/100\_5742.JPG   
 inflating: test\_data/test\_data/hsparo/12265273154\_48e0fefd42\_o.jpg   
 inflating: test\_data/test\_data/hsparo/DSC\_4521.jpg   
 inflating: test\_data/test\_data/hsparo/DSC\_7619.jpg   
 inflating: test\_data/test\_data/hsparo/\_DSC\_11244.jpg   
 inflating: test\_data/test\_data/hsparo/\_DSC\_11246.jpg   
 inflating: test\_data/test\_data/indvul/DSC\_1263.jpg   
 inflating: test\_data/test\_data/indvul/DSC\_1273.jpg   
 inflating: test\_data/test\_data/indvul/DSC\_1274.jpg   
 inflating: test\_data/test\_data/indvul/DSC\_1275.jpg   
 inflating: test\_data/test\_data/indvul/DSC\_1277.jpg   
 inflating: test\_data/test\_data/indvul/\_DSC8754.jpg   
 inflating: test\_data/test\_data/indvul/\_DSC9520.jpg   
 inflating: test\_data/test\_data/indvul/\_DSC9521.jpg   
 inflating: test\_data/test\_data/indvul/\_DSC9523.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_10584.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_10587.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_10593.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_13515.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_13516.jpg   
 inflating: test\_data/test\_data/jglowl/\_D32\_13517.jpg   
 inflating: test\_data/test\_data/lbicrw/100\_4917.JPG   
 inflating: test\_data/test\_data/lbicrw/IMG\_5538.JPG   
 inflating: test\_data/test\_data/lbicrw/IMG\_5541.JPG   
 inflating: test\_data/test\_data/lbicrw/IMG\_5543.JPG   
 inflating: test\_data/test\_data/lbicrw/IMG\_5544.JPG   
 inflating: test\_data/test\_data/lbicrw/P1030030.jpg   
 inflating: test\_data/test\_data/mgprob/100\_5811.JPG   
 inflating: test\_data/test\_data/mgprob/100\_5812.JPG   
 inflating: test\_data/test\_data/mgprob/100\_5813.JPG   
 inflating: test\_data/test\_data/mgprob/DSC\_3233.jpg   
 inflating: test\_data/test\_data/mgprob/\_DSC0085.jpg   
 inflating: test\_data/test\_data/mgprob/\_DSC0133.jpg   
 inflating: test\_data/test\_data/mgprob/\_DSC0135.jpg   
 inflating: test\_data/test\_data/rebimg/100\_5756.JPG   
 inflating: test\_data/test\_data/rebimg/100\_5757.JPG   
 inflating: test\_data/test\_data/rebimg/100\_5758.JPG   
 inflating: test\_data/test\_data/rebimg/100\_5759.JPG   
 inflating: test\_data/test\_data/rebimg/100\_5760.JPG   
 inflating: test\_data/test\_data/rebimg/13009302395\_5d88a9d62b\_o.jpg   
 inflating: test\_data/test\_data/rebimg/13009315305\_f1cbfb6767\_o.jpg   
 inflating: test\_data/test\_data/rebimg/13009451823\_8eb502581f\_o.jpg   
 inflating: test\_data/test\_data/rebimg/DSC\_7148.jpg   
 inflating: test\_data/test\_data/rebimg/IMG\_5251.JPG   
 inflating: test\_data/test\_data/rebimg/IMG\_5252.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4462.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4463.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4464.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4466.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4467.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_4468.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_5103.JPG   
 inflating: test\_data/test\_data/wcrsrt/100\_5105.JPG   
 inflating: test\_data/test\_data/wcrsrt/11382677585\_ef6570a041\_o.jpg   
 inflating: test\_data/test\_data/wcrsrt/11382746854\_7ce73793b3\_o.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6382.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6383.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6384-2.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6384.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6385.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6386.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6387.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6388.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6389.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6390.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6391.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6392.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6393.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6394.jpg   
 inflating: train\_data/train\_data/blasti/DSC\_6395.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4570.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4571.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4583.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4584.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4585.jpg   
 inflating: train\_data/train\_data/bonegl/DSC\_4586.jpg   
 inflating: train\_data/train\_data/brhkyt/D72\_0400.jpg   
 inflating: train\_data/train\_data/brhkyt/D72\_0401.jpg   
 inflating: train\_data/train\_data/brhkyt/D72\_0470.jpg   
 inflating: train\_data/train\_data/brhkyt/D72\_0471.jpg   
 inflating: train\_data/train\_data/brhkyt/D72\_0472.jpg   
 inflating: train\_data/train\_data/cbrtsh/100\_5097.JPG   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10303.jpg   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10305.jpg   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10306.jpg   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10307.jpg   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10308.jpg   
 inflating: train\_data/train\_data/cbrtsh/\_D32\_10309.jpg   
 inflating: train\_data/train\_data/cmnmyn/100\_5763.JPG   
 inflating: train\_data/train\_data/cmnmyn/5866682091\_870ccc946c\_o.jpg   
 inflating: train\_data/train\_data/cmnmyn/6154956165\_64266b8b53\_o.jpg   
 inflating: train\_data/train\_data/cmnmyn/DSCN5784.jpg   
 inflating: train\_data/train\_data/cmnmyn/DSCN5787.jpg   
 inflating: train\_data/train\_data/cmnmyn/DSCN5790.jpg   
 inflating: train\_data/train\_data/cmnmyn/DSCN5791.jpg   
 inflating: train\_data/train\_data/gretit/100\_5042.JPG   
 inflating: train\_data/train\_data/gretit/100\_5043.JPG   
 inflating: train\_data/train\_data/gretit/100\_5044.JPG   
 inflating: train\_data/train\_data/gretit/100\_5045.JPG   
 inflating: train\_data/train\_data/gretit/100\_5046.JPG   
 inflating: train\_data/train\_data/gretit/100\_5047.JPG   
 inflating: train\_data/train\_data/hilpig/DSC\_6272.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6273.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6274.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6275.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6350.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6351.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6352.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6353.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6354.jpg   
 inflating: train\_data/train\_data/hilpig/DSC\_6358.jpg   
 inflating: train\_data/train\_data/himbul/100\_5028.JPG   
 inflating: train\_data/train\_data/himbul/100\_5029.JPG   
 inflating: train\_data/train\_data/himbul/100\_5096.JPG   
 inflating: train\_data/train\_data/himbul/100\_5099.JPG   
 inflating: train\_data/train\_data/himbul/100\_5100.JPG   
 inflating: train\_data/train\_data/himbul/100\_5102.JPG   
 inflating: train\_data/train\_data/himbul/100\_5110.JPG   
 inflating: train\_data/train\_data/himbul/100\_5112.JPG   
 inflating: train\_data/train\_data/himbul/100\_5113.JPG   
 inflating: train\_data/train\_data/himbul/100\_5159.JPG   
 inflating: train\_data/train\_data/himbul/100\_5160.JPG   
 inflating: train\_data/train\_data/himbul/100\_5161.JPG   
 inflating: train\_data/train\_data/himbul/100\_5162.JPG   
 inflating: train\_data/train\_data/himbul/10774570945\_9bc537c255\_o.jpg   
 inflating: train\_data/train\_data/himbul/11383838134\_96e891a316\_o.jpg   
 inflating: train\_data/train\_data/himbul/11383857084\_cdd7514746\_o.jpg   
 inflating: train\_data/train\_data/himbul/11387473723\_464965720a\_o.jpg   
 inflating: train\_data/train\_data/himbul/9007817678\_24db19fc63\_o.jpg   
 inflating: train\_data/train\_data/himgri/12029790563\_2fe54a68aa\_o.jpg   
 inflating: train\_data/train\_data/himgri/12029822653\_1185544e7b\_o.jpg   
 inflating: train\_data/train\_data/himgri/12029933015\_802428e277\_o.jpg   
 inflating: train\_data/train\_data/himgri/12029987304\_031b7a2d53\_o.jpg   
 inflating: train\_data/train\_data/himgri/12030165813\_f5341e8ed0\_o.jpg   
 inflating: train\_data/train\_data/himgri/12030221124\_1baaf905d1\_o.jpg   
 inflating: train\_data/train\_data/himgri/12030269234\_1c3f5a8e8c\_o.jpg   
 inflating: train\_data/train\_data/himgri/12152037683\_13e1556c41\_o.jpg   
 inflating: train\_data/train\_data/himgri/12152178844\_dc29e3126f\_o.jpg   
 inflating: train\_data/train\_data/himgri/12152510436\_23b62fed3b\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265055185\_138e3c7c70\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265498144\_d2e80faa3b\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265561764\_4814951031\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265582724\_61cb64e73a\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265738206\_2b9b25cc8a\_o.jpg   
 inflating: train\_data/train\_data/himgri/12265747116\_d4b4fa3741\_o.jpg   
 inflating: train\_data/train\_data/himgri/12266053036\_fcab92d8e8\_o.jpg   
 inflating: train\_data/train\_data/himgri/12266077576\_1d7143aaf7\_o.jpg   
 inflating: train\_data/train\_data/himgri/12266086526\_82cd337667\_o.jpg   
 inflating: train\_data/train\_data/himgri/IMG\_5463.JPG   
 inflating: train\_data/train\_data/hsparo/100\_4757.JPG   
 inflating: train\_data/train\_data/hsparo/100\_4758.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5039.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5040.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5041.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5048.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5049.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5050.JPG   
 inflating: train\_data/train\_data/hsparo/100\_5572.JPG   
 inflating: train\_data/train\_data/indvul/DSC\_0502.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0571e.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0572.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0576e.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0582.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0583e.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0584.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0616c.jpg   
 inflating: train\_data/train\_data/indvul/DSC\_0617.jpg   
 inflating: train\_data/train\_data/jglowl/12152151476\_7a1524aabb\_o.jpg   
 inflating: train\_data/train\_data/jglowl/DSC01335.jpg   
 inflating: train\_data/train\_data/jglowl/DSC01336.jpg   
 inflating: train\_data/train\_data/jglowl/\_D32\_10285.jpg   
 inflating: train\_data/train\_data/jglowl/\_D32\_10578.jpg   
 inflating: train\_data/train\_data/jglowl/\_D32\_10583.jpg   
 inflating: train\_data/train\_data/lbicrw/100\_4037.JPG   
 inflating: train\_data/train\_data/lbicrw/100\_4912.JPG   
 inflating: train\_data/train\_data/lbicrw/100\_4913.JPG   
 inflating: train\_data/train\_data/lbicrw/100\_4914.JPG   
 inflating: train\_data/train\_data/lbicrw/100\_4915.JPG   
 inflating: train\_data/train\_data/lbicrw/100\_4916.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5587.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5588.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5589.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5590.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5592.JPG   
 inflating: train\_data/train\_data/mgprob/100\_5762.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5744.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5745.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5746.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5748.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5749.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5750.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5751.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5752.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5754.JPG   
 inflating: train\_data/train\_data/rebimg/100\_5755.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4452.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4453.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4454.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4455.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4456.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4457.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4458.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4459.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4460.JPG   
 inflating: train\_data/train\_data/wcrsrt/100\_4461.JPG

#data augmentation  
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train\_gen=ImageDataGenerator(rescale=(1./255),horizontal\_flip=True,shear\_range=0.2)  
test\_gen=ImageDataGenerator(rescale=(1./255)) #rescale-->normalisation only

train=train\_gen.flow\_from\_directory('/content/train\_data/train\_data',  
 target\_size=(120, 120),  
 class\_mode='categorical',  
 batch\_size=8)  
test=test\_gen.flow\_from\_directory('/content/test\_data/test\_data',  
 target\_size=(120, 120),  
 class\_mode='categorical',  
 batch\_size=8)

Found 150 images belonging to 16 classes.  
Found 157 images belonging to 16 classes.

train.class\_indices,test.class\_indices

({'blasti': 0,  
 'bonegl': 1,  
 'brhkyt': 2,  
 'cbrtsh': 3,  
 'cmnmyn': 4,  
 'gretit': 5,  
 'hilpig': 6,  
 'himbul': 7,  
 'himgri': 8,  
 'hsparo': 9,  
 'indvul': 10,  
 'jglowl': 11,  
 'lbicrw': 12,  
 'mgprob': 13,  
 'rebimg': 14,  
 'wcrsrt': 15},  
 {'blasti': 0,  
 'bonegl': 1,  
 'brhkyt': 2,  
 'cbrtsh': 3,  
 'cmnmyn': 4,  
 'gretit': 5,  
 'hilpig': 6,  
 'himbul': 7,  
 'himgri': 8,  
 'hsparo': 9,  
 'indvul': 10,  
 'jglowl': 11,  
 'lbicrw': 12,  
 'mgprob': 13,  
 'rebimg': 14,  
 'wcrsrt': 15})

#build the cnn model  
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense  
from tensorflow.keras.models import Sequential

model=Sequential()  
#convolution layer  
model.add(Convolution2D(20,(3,3),activation='relu',input\_shape=(120,120,3))) #3-->color image.1-->grayscale  
#max\_pooling layer  
model.add(MaxPooling2D(pool\_size=(2,2)))  
#flattened layer  
model.add(Flatten())  
#fully connected(ANN)  
model.add(Dense(45,activation='relu')) #1st hidden layer  
model.add(Dense(16,activation='softmax')) #output layer 16--> 16classes only

#compile the model  
model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

#train the model  
model.fit(train,batch\_size=8,validation\_data=test,epochs=10)

Epoch 1/10  
19/19 [==============================] - 125s 7s/step - loss: 2.5413 - accuracy: 0.2467 - val\_loss: 2.7623 - val\_accuracy: 0.1274  
Epoch 2/10  
19/19 [==============================] - 104s 6s/step - loss: 2.4987 - accuracy: 0.2333 - val\_loss: 2.7115 - val\_accuracy: 0.2166  
Epoch 3/10  
19/19 [==============================] - 125s 7s/step - loss: 2.4075 - accuracy: 0.2467 - val\_loss: 2.6416 - val\_accuracy: 0.1847  
Epoch 4/10  
19/19 [==============================] - 125s 7s/step - loss: 2.3059 - accuracy: 0.2467 - val\_loss: 2.7181 - val\_accuracy: 0.1975  
Epoch 5/10  
19/19 [==============================] - 124s 7s/step - loss: 2.2737 - accuracy: 0.2600 - val\_loss: 2.7076 - val\_accuracy: 0.2102  
Epoch 6/10  
19/19 [==============================] - 125s 7s/step - loss: 2.1683 - accuracy: 0.2467 - val\_loss: 2.7905 - val\_accuracy: 0.2102  
Epoch 7/10  
19/19 [==============================] - 124s 7s/step - loss: 2.1755 - accuracy: 0.2667 - val\_loss: 2.9647 - val\_accuracy: 0.1592  
Epoch 8/10  
19/19 [==============================] - 125s 7s/step - loss: 2.1434 - accuracy: 0.2600 - val\_loss: 2.7592 - val\_accuracy: 0.2102  
Epoch 9/10  
19/19 [==============================] - 125s 7s/step - loss: 2.0758 - accuracy: 0.2533 - val\_loss: 2.8624 - val\_accuracy: 0.1529  
Epoch 10/10  
19/19 [==============================] - 124s 7s/step - loss: 2.0396 - accuracy: 0.2467 - val\_loss: 2.6783 - val\_accuracy: 0.2229

<keras.callbacks.History at 0x7fa685bbc430>

model.save('birdspecies') #it will save the model in the local system

WARNING:absl:Found untraced functions such as \_jit\_compiled\_convolution\_op, \_update\_step\_xla while saving (showing 2 of 2). These functions will not be directly callable after loading.

#predict the model with real time image rat  
import numpy as np  
from tensorflow.keras.preprocessing import image

img=image.load\_img('/content/AdobeStock\_561780689\_Preview.jpeg',target\_size=(120,120))

img



#convert the image to array  
img=image.img\_to\_array(img)  
img

array([[[20., 19., 53.],  
 [20., 21., 52.],  
 [20., 21., 51.],  
 ...,  
 [59., 28., 46.],  
 [53., 24., 46.],  
 [48., 20., 43.]],  
  
 [[19., 19., 55.],  
 [20., 21., 52.],  
 [22., 23., 53.],  
 ...,  
 [58., 27., 45.],  
 [53., 24., 46.],  
 [49., 21., 44.]],  
  
 [[20., 20., 56.],  
 [21., 22., 53.],  
 [23., 24., 55.],  
 ...,  
 [58., 26., 47.],  
 [51., 23., 45.],  
 [49., 21., 44.]],  
  
 ...,  
  
 [[13., 16., 49.],  
 [13., 16., 47.],  
 [15., 19., 48.],  
 ...,  
 [50., 20., 44.],  
 [42., 16., 45.],  
 [34., 14., 42.]],  
  
 [[13., 16., 49.],  
 [12., 15., 46.],  
 [13., 16., 47.],  
 ...,  
 [47., 17., 43.],  
 [38., 15., 43.],  
 [33., 14., 42.]],  
  
 [[13., 18., 50.],  
 [12., 17., 47.],  
 [12., 17., 46.],  
 ...,  
 [46., 18., 43.],  
 [35., 16., 44.],  
 [28., 15., 43.]]], dtype=float32)

#expand the dimension from 2 to 3 because we use 3 channel rgb for tha model  
img=np.expand\_dims(img,axis=0)  
img

array([[[[20., 19., 53.],  
 [20., 21., 52.],  
 [20., 21., 51.],  
 ...,  
 [59., 28., 46.],  
 [53., 24., 46.],  
 [48., 20., 43.]],  
  
 [[19., 19., 55.],  
 [20., 21., 52.],  
 [22., 23., 53.],  
 ...,  
 [58., 27., 45.],  
 [53., 24., 46.],  
 [49., 21., 44.]],  
  
 [[20., 20., 56.],  
 [21., 22., 53.],  
 [23., 24., 55.],  
 ...,  
 [58., 26., 47.],  
 [51., 23., 45.],  
 [49., 21., 44.]],  
  
 ...,  
  
 [[13., 16., 49.],  
 [13., 16., 47.],  
 [15., 19., 48.],  
 ...,  
 [50., 20., 44.],  
 [42., 16., 45.],  
 [34., 14., 42.]],  
  
 [[13., 16., 49.],  
 [12., 15., 46.],  
 [13., 16., 47.],  
 ...,  
 [47., 17., 43.],  
 [38., 15., 43.],  
 [33., 14., 42.]],  
  
 [[13., 18., 50.],  
 [12., 17., 47.],  
 [12., 17., 46.],  
 ...,  
 [46., 18., 43.],  
 [35., 16., 44.],  
 [28., 15., 43.]]]], dtype=float32)

model.predict(img) #predict the model

1/1 [==============================] - 0s 211ms/step

array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]],  
 dtype=float32)

pred=np.argmax(model.predict(img))  
out=['blasti','bonegl','brhkyt','cbrtsh','cmnmyn','gretit','hilpig','himbul','himgri','hsparo','indvul','jglowl','lbicrw','mgprob','rebimg','wcrsrt']  
out[pred]

1/1 [==============================] - 0s 26ms/step

{"type":"string"}