Release notes ✕ •••

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

```
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, roc_curve,
from tensorflow.keras.applications import VGG16, ResNet5
from tensorflow.keras.layers import Concatenate, Dense,
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDa
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import VGG16, ResNet5
```

```
import os
import random
import shutil

# Set the path to your main 'train' folder
main_folder = '/content/drive/MyDrive/Melanoma_Skin_Canc

# Create directories for training and testing data
train_folder = '/content/drive/MyDrive/Melanoma_Skin_Can
test_folder = '/content/drive/MyDrive/Melanoma_Skin_Canc
os.makedirs(train_folder, exist_ok=True)
os.makedirs(test_folder, exist_ok=True)

# Define the subfolders (benign and malignant)
subfolders = ['benign', 'malignant']

# Set the split ratio (70% training, 30% testing)
split_ratio = 0.7

# Iterate through each subfolder
for subfolder in subfolders:
    subfolder_path = os.path.join(main_folder, subfolder
    images = os.listdir(subfolder_path)
    num_images = len(images)
    num_train = int(num_images * split_ratio)

    # Shuffle the list of images
    random.shuffle(images)

    # Split into training and testing sets
    train_images = images[:num_train]
    test_images = images[num_train:]

    # Copy training images to the train_data_folder
    for image in train_images:
        src = os.path.join(subfolder_path, image)
        dst = os.path.join(train_folder, subfolder)
        os.makedirs(dst, exist_ok=True)
        shutil.copy(src, dst)

    # Copy testing images to the test_data_folder
    for image in test_images:
        src = os.path.join(subfolder_path, image)
        dst = os.path.join(test_folder, subfolder)
        os.makedirs(dst, exist_ok=True)
        shutil.copy(src, dst)

print("Splitting completed successfully.")
```

```
Splitting completed successfully.
```

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

2024-08-20

- TPU memory usage and utilization can now be checked with `!tpu-info`
- Gemini Chat responses are now grounded in relevant sources
- Added a new "Create Gemini API key" link in the user secrets panel
- Added a new "Gemini: Creating a prompt" snippet and touched up the existing "Gemini: Connecting to Gemini" snippet
- Added the ability to specify custom placeholder text for various interactive form params (see examples)
- Keyboard navigation a11y improvements to comments UI
- Various minor rendering improvements to interactive forms UI
- A11y improvements for the run button and header
- Updated tooltip styling
- A11y improvements for the file browser's disk usage bar
- On mobile, tooltips now trigger on long press
- On mobile, release notes updates will no longer display automatically
- Python package upgrades
  - astropy 5.3.4 -> 6.1.2
  - bigframes 1.11.1 -> 1.14.0
  - bokeh 3.3.4 -> 3.4.3
  - dask 2023.8.1 -> 2024.7.1
  - earthengine-api 0.1.412 -> 0.1.416
  - geopandas 0.13.2 -> 0.14.4
  - kagglehub 0.2.8 -> 0.2.9
  - keras 2.15.0 -> 3.4.1
  - lightgbm 4.1.0 -> 4.4.0
  - malloy 2023.1067 -> 2024.1067
  - numba 0.58.1 -> 0.60.0

```
input_shape = (224, 224, 3)
vgg_model = VGG16(weights='imagenet', include_top=False, i
resnet_model = ResNet50(weights='imagenet', include_top=Fa
```

```
Downloading data from https://storage.googleapis.com
58889256/58889256 [==============================] -
Downloading data from https://storage.googleapis.com
94765736/94765736 [==============================] -
```

```
for layer in vgg_model.layers:
    layer.trainable = False
for layer in resnet_model.layers:
    layer.trainable = False
input_tensor = Input(shape=input_shape)
vgg_output = vgg_model(input_tensor)
resnet_output = resnet_model(input_tensor)
merged = Concatenate()([GlobalAveragePooling2D()(vgg_out
x = Dense(512, activation='relu')(merged)
x = Dense(256, activation='relu')(x)
output = Dense(2, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output)
model.compile(optimizer=Adam(lr=0.001), loss='sparse_cat
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer,
```

```
model.summary()
```

```
Model: "model"
____________________________________________________
 Layer (type)                Output Shape
====================================================
 input_3 (InputLayer)        [(None, 224, 224, 3)]

 vgg16 (Functional)          (None, 7, 7, 512)


 resnet50 (Functional)       (None, 7, 7, 2048)


 global_average_pooling2d (  (None, 512)
 GlobalAveragePooling2D)

 global_average_pooling2d_1  (None, 2048)
  (GlobalAveragePooling2D)

 concatenate (Concatenate)   (None, 2560)
```

  - numpy 1.25.2 -> 1.26.4
  - opencv-python 4.8.0.76 -> 4.10.0.84
  - pandas 2.0.3 -> 2.1.4
  - pandas-gbq 0.19.2 -> 0.23.1
  - panel 1.3.8 -> 1.4.5
  - requests 2.31.0 -> 2.32.3
  - scikit-learn 1.2.2. -> 1.3.2
  - scipy 1.11.4 -> 1.13.1
  - tensorboard 2.15.2 -> 2.17.0
  - tensorflow 2.15.0 -> 2.17.0
  - tf-keras 2.15.1 -> 2.17.0
  - xarray 2023.7.0 -> 2024.6.0
  - xgboost 2.0.3 -> 2.1.1
- Python package inclusions
  - einops 0.8.0

2024-07-22

- You can now embed Google sheets directly into Colab to streamline interactions with data with InteractiveSheet.

  Example:

  ```
  from google.colab import shee
  sh = sheets.InteractiveSheet(
  df = sh.as_df()
  ```

- Fixed multiple rendering bugs in cell editors with wide text content (i.e. text is no longer hidden or clipped)
- Fixed multiple accessibility issues in Colab's comments feature (e.g. proper keyboard focus management, added accessibility landmarks, etc)
- Fixed bug where AI code generation would fail for extremely long broken code snippets
- Fixed multiple scrollbar bugs in the user secrets panel
- Added the ability for workspace admin to purchase Colab Pro and Pro+ Subscriptions for users
- Fixed bug where user secrets couldn’t be moved to a tab
- Fixed several focus management accessibility issues in tabs, the table of contents, the left toolbar, and the run button

```
 dense (Dense)               (None, 512)

 dense_1 (Dense)             (None, 256)

 dense_2 (Dense)             (None, 2)

=====================================================
Total params: 39745474 (151.62 MB)
Trainable params: 1443074 (5.50 MB)
Non-trainable params: 38302400 (146.11 MB)
_____________________________________________________
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_diseaseDa
    target_size=(224, 224),
    batch_size=32,
    class_mode='sparse'
)

validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Melanoma_Skin_Cancer_disea
        target_size=(224, 224),
        batch_size=32,
        class_mode='sparse')
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

```
history = model.fit_generator(
    train_generator,
    steps_per_epoch=2000 // 32,
    epochs=100,
    validation_data=validation_generator,
    validation_steps=800 // 32
)
```

```
<ipython-input-6-794030e3dcad>:1: UserWarning: `Mo
  history = model.fit_generator(
Epoch 1/100
62/62 [==============================] - 620s 10s/
Epoch 2/100
62/62 [==============================] - 252s 4s/s
Epoch 3/100
62/62 [==============================] - 181s 3s/s
Epoch 4/100
```

- Fixed bug where overflowing cells may be omitted when pasting from Google Sheets
- Fixed bug where the generate code button did not activate on touch
- Python package upgrades
  - bigframes 1.9.0 -> 1.11.1
  - cvxpy 1.3.4 -> 1.5.2
  - earthengine-api 0.1.408 -> 0.1.412
  - google-api-core 2.11.1 -> 2.19.1
  - google-api-python-client 2.84.0 -> 2.137.0
  - google-cloud-aiplatform 1.56.0 -> 1.59.0
  - google-cloud-bigquery 3.21.0 -> 3.25.0
  - google-cloud-core 2.3.3 -> 2.4.1
  - google-cloud-datastore 2.15.2 -> 2.19.0
  - google-cloud-firestore 2.11.1 -> 2.16.1
  - google-cloud-functions 1.13.3 -> 1.16.4
  - google-generativeai 0.5.4 -> 0.7.2
  - kagglehub 0.2.5 -> 0.2.8
  - pip 23.1.2 -> 24.1.2
  - setuptools 67.7.2 -> 71.0.4
  - sympy 1.12.1 -> 1.13.1
  - torch 2.3.0 -> 2.3.1
  - transformers 4.41.2 -> 4.42.4
- Python package inclusions
  - accelerate 0.32.1

2024-06-18

- Inline AI completions are now available to users on the free-of-charge tier
- Reduced latency for LSP and terminal connections
- Improved quality of inline completions
- Visual improvements to switch controls across Colab
- Various bug fixes, performance and a11y improvements to the user secrets panel

```
62/62 [==============================] - 138s 2s/s
Epoch 5/100
62/62 [==============================] - 120s 2s/s
Epoch 6/100
62/62 [==============================] - 93s 1s/st
Epoch 7/100
62/62 [==============================] - 75s 1s/st
Epoch 8/100
62/62 [==============================] - 65s 1s/st
Epoch 9/100
62/62 [==============================] - 59s 942ms
Epoch 10/100
62/62 [==============================] - 49s 799ms
Epoch 11/100
62/62 [==============================] - 39s 637ms
Epoch 12/100
62/62 [==============================] - 34s 555ms
Epoch 13/100
62/62 [==============================] - 34s 544ms
Epoch 14/100
62/62 [==============================] - 28s 453ms
Epoch 15/100
62/62 [==============================] - 22s 359ms
Epoch 16/100
62/62 [==============================] - 25s 410ms
Epoch 17/100
62/62 [==============================] - 26s 412ms
Epoch 18/100
62/62 [==============================] - 20s 326ms
Epoch 19/100
62/62 [==============================] - 25s 404ms
Epoch 20/100
62/62 [==============================] - 25s 398ms
Epoch 21/100
62/62 [==============================] - 25s 399ms
Epoch 22/100
62/62 [==============================] - 25s 404ms
Epoch 23/100
62/62 [==============================] - 20s 323ms
Epoch 24/100
62/62 [==============================] - 20s 324ms
Epoch 25/100
62/62 [==============================] - 20s 317ms
Epoch 26/100
62/62 [==============================] - 24s 393ms
Epoch 27/100
62/62 [==============================] - 20s 319ms
Epoch 28/100
```

```
model.save('/content/drive/MyDrive/melanoma_cancer_datas
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

- Improved tooltip UX behavior
- Improved behavior when copying data from Google Sheets and pasting in Colab
- Scroll to cell fixes for single tabbed view and jump to cell command
- Improved tab header behavior
- A11y improvements for notebook-focused cells
- Python package upgrades
  - torch 2.2.1 -> 2.3.0
  - torchaudio 2.2.1 -> 2.3.0
  - torchvision 0.17.1 -> 0.18.0
  - torchtext 0.17.1 -> 0.18.0
  - google-cloud-aiplatform 1.51.0 -> 1.56.0
  - bigframes 1.5.0 -> 1.8.0
  - regex 2023.12.25 -> 2024.5.15

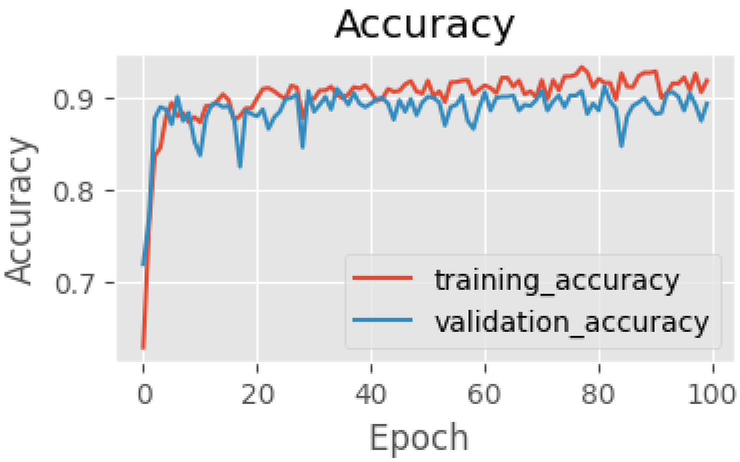
2024-05-13

- Code actions are now supported to automatically improve and refactor code. Code actions can be triggered by the keyboard shortcut "Ctrl/⌘ + ."
- Python package upgrades
  - bigframes 1.0.0 -> 1.5.0
  - google-cloud-aiplatform 1.47.0 -> 1.51.0
  - jax[tpu] 0.4.23 -> 0.4.26
- Python package inclusions
  - cudf 24.4.1

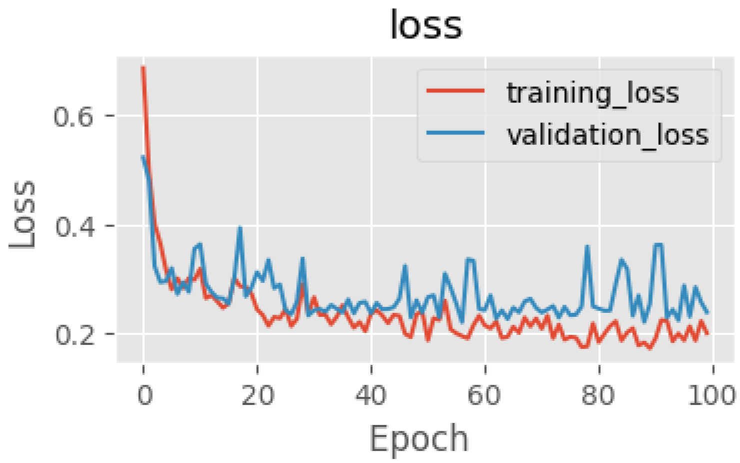
2024-04-15

- TPU v2 runtime is now available
- L4 runtime is now available for paid users
- New distributed fine-tuning Gemma tutorial on TPUs (GitHub)
- Symbol rename is now supported with keyboard shortcut F2
- Fixed bug causing inability to re-upload deleted files
- Fixed breaking bug in colabtools %upload_files_async
- Added syntax highlighting to %%writefile cells

```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["accuracy"], label="training
plt.plot(N, history.history["val_accuracy"], label="vali
plt.title("Accuracy")
plt.xlabel("Epoch ")
plt.ylabel("Accuracy")
plt.legend()
plt.savefig("accc.png")
```



```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["loss"], label="training_los
plt.plot(N, history.history["val_loss"], label="validati
plt.title("loss")
plt.xlabel("Epoch ")
plt.ylabel("Loss")
plt.legend()
plt.savefig("loss.png")
```



- Cuda dependencies that come with Torch are cached for faster downloads for packages that require Torch and its dependencies (GitHub issue)
- Python package upgrades
  - bigframes 0.24.0 -> 1.0.0
  - duckdb 0.9.2 -> 0.10.1
  - google-cloud-aiplatform 1.43.0 -> 1.47.0
  - jax 0.4.23 -> 0.4.26

2024-03-13

- Fixed bug that sometimes caused UserSecrets to move / disappear
- Improved messaging for mounting drive in an unsupported environment (GitHub issue)
- Python package upgrades
  - torch 2.1.0 -> 2.2.1
  - torchaudio 2.1.0 -> 2.2.1
  - torchvision 0.16.0 -> 0.17.1
  - torchtext 0.16.0 -> 0.17.1
  - PyMC 5.7.2 -> 5.10.4
  - BigFrames 0.21.0 -> 0.24.0
  - google-cloud-aiplatform 1.42.1 -> 1.43.0
  - tornado 6.3.2 -> 6.3.3

2024-02-21

- Try out Gemma on Colab!
- Allow unicode in form text inputs
- Display documentation and link to source when displaying functions
- Display image-like ndarrays as images
- Improved UX around quick charts and execution error suggestions
- Released Marketplace image for the month of February (GitHub issue)
- Python package upgrades
  - bigframes 0.19.2 -> 0.21.0
  - regex 2023.6.3 -> 2023.12.25
  - spacy 3.6.1 -> 3.7.4
  - beautifulsoup4 4.11.2 -> 4.12.3
  - tensorflow-probability 0.22.0 -> 0.23.0

```
test_data_dir='/content/drive/MyDrive/Melanoma_Skin_Canc
test_generator = train_datagen.flow_from_directory(test_
                                                   targe
                                                   batch
```

```
Found 3183 images belonging to 2 classes.
```

```
import cv2


def mode(my_list):
    ct = Counter(my_list)
    max_value = max(ct.values())
    return ([key for key, value in ct.items() if value =

true_value = []
model_one_pred = []
for folder in os.listdir(test_data_dir):

    test_image_ids = os.listdir(os.path.join(test_data_d

    for image_id in test_image_ids[:int(len(test_image_i

        path = os.path.join(test_data_dir,folder,image_i

        true_value.append(test_generator.class_indices[f
        img = cv2.resize(cv2.imread(path),(224,224))
        img_normalized = img/255

        model_one_prediction = np.argmax(model.predict(n
        model_one_pred.append(model_one_prediction)
```

```
1/1 [==============================] - 3s 3s/step
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 47ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 24ms/ste
```

  - google-cloud-language 2.9.1 -> 2.13.1
  - google-cloud-aiplatform 1.39.0 -> 1.42.1
  - transformers 4.35.2 -> 4..37.2
  - pyarrow 10.0.1 -> 14.0.2

2024-01-29

- New Kaggle Notebooks <> Colab updates! Now you can:
  - Import directly from Colab without having to download/re-upload
  - Upload via link, by pasting Google Drive or Colab URLs
  - Export & run Kaggle Notebooks on Colab with 1 click
- Try these notebooks that talk to Gemini:
  - Gemini and Stable Diffusion
  - Learning with Gemini and ChatGPT
  - Talk to Gemini with Google's Speech to Text API
  - Sell lemonade with Gemini and Sheets
  - Generate images with Gemini and Vertex
- Python package upgrades
  - google-cloud-aiplatform 1.38.1 -> 1.39.0
  - bigframes 0.18.0 -> 0.19.2
  - polars 0.17.3 -> 0.20.2
  - gdown 4.6.6 -> 4.7.3 (GitHub issue)
  - tensorflow-hub 0.15.0 -> 0.16.0
  - flax 0.7.5 -> 0.8.0
- Python package inclusions
  - sentencepiece 0.1.99

2024-01-08

- Avoid nested scrollbars for large outputs by using `google.colab.output.no_vertic` Example notebook
- Fix bug where downloading models from Hugging Face could freeze

```
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 29ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 26ms/ste
1/1 [==============================] - 0s 27ms/ste
1/1 [==============================] - 0s 27ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 28ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
```

- Python package upgrades
  - huggingface-hub 0.19.4 -> 0.20.2
  - bigframes 0.17.0 -> 0.18.0

2023-12-18

- Expanded access to AI coding has arrived in Colab across 175 locales for all tiers of Colab users
- Improvements to display of ML-based inline completions (for eligible Pro/Pro+ users)
- Started a series of notebooks highlighting Gemini API capabilities
- Enable ⌘/Ctrl+L to select the full line in an editor
- Fixed bug where we weren't correctly formatting output from multiple execution results
- Python package upgrades
  - CUDA 11.8 to CUDA 12.2
  - tensorflow 2.14.0 -> 2.15.0
  - tensorboard 2.14.0 -> 2.15.0
  - keras 2.14.0 -> 2.15.0
  - Nvidia drivers 525.105.17 -> 535.104.05
  - tensorflow-gcs-config 2.14.0 -> 2.15.0
  - bigframes 0.13.0 -> 0.17.0
  - geemap 0.28.2 -> 0.29.6
  - pyarrow 9.0.0 -> 10.0.1
  - google-generativeai 0.2.2 -> 0.3.1
  - jax 0.4.20 —> 0.4.23
  - jaxlib 0.4.20 —> 0.4.23
- Python package inclusions
  - kagglehub 0.1.4
  - google-cloud-aiplatform 1.38.1

2023-11-27

- Removed warning when calling `await` to make it render as code
- Added "Run selection" to the cell context menu
- Added highlighting for the `%%python` cell magic

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import itertools
#from mlxtend.plotting import plot_confusion_matrix
def clf_report(true_value, model_pred):

    classes = test_generator.class_indices.keys()
    TP_count = [true_value[i] == model_pred[i] for i in
    model_accuracy = np.sum(TP_count)/len(TP_count)
    print('Model Accuracy', model_accuracy)

    plt.figure(figsize=(3,3))
    cm = confusion_matrix(true_value,model_pred)
    plt.imshow(cm,interpolation='nearest',cmap=plt.cm.vi
    plt.title('Confusion Matrix')
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=15) #15 ref
    plt.yticks(tick_marks, classes)
    thresh = cm.max()*0.8
    for i,j in itertools.product(range(cm.shape[0]),rang
        plt.text(j,i,cm[i,j],
                horizontalalignment="center",
                color="black" if cm[i,j] > thresh else "
        pass

    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    pass

    print(classification_report(true_value, model_pred,




clf_report(true_value, model_one_pred)
```

- Launched AI coding features for Pro/Pro+ users in more locales
- Python package upgrades
  - bigframes 0.12.0 -> 0.13.0
- Python package inclusions
  - transformers 4.35.2
  - google-generativeai 0.2.2

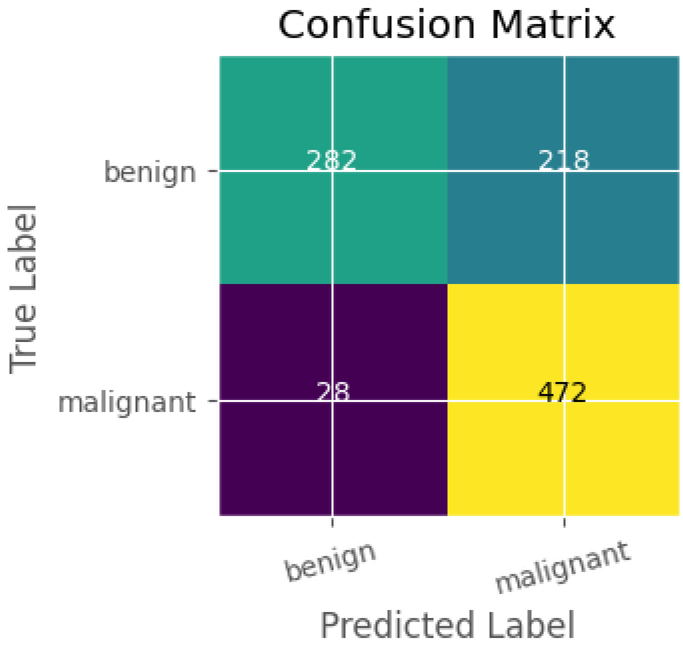
2023-11-08

- Launched Secrets, for safe storage of private keys on Colab (tweet)
- Fixed issue where TensorBoard would not load (#3990)
- Python package upgrades
  - lightgbm 4.0.0 -> 4.1.0
  - bigframes 0.10.0 -> 0.12.0
  - bokeh 3.2.2 -> 3.3.0
  - duckdb 0.8.1 -> 0.9.1
  - numba 0.56.4 -> 0.58.1
  - tweepy 4.13.0 -> 4.14.0
  - jax 0.4.16 -> 0.4.20
  - jaxlib 0.4.16 -> 0.4.20

2023-10-23

- Updated the **Open notebook** dialog for better usability and support for smaller screen sizes
- Added smart paste support for data from Google Sheets for R notebooks
- Enabled showing release notes in a tab
- Launched AI coding features for Pro/Pro+ users in Australia AU Canada CA India IN and Japan JP (tweet)
- Python package upgrades
  - earthengine-api 0.1.357 -> 0.1.375
  - flax 0.7.2 -> 0.7.4
  - geemap 0.27.4 -> 0.28.2
  - jax 0.4.14 -> 0.4.16
  - jaxlib 0.4.14 -> 0.4.16
  - keras 2.13.1 -> 2.14.0
  - tensorboard 2.13.0 -> 2.14.1
  - tensorflow 2.13.0 -> 2.14.0
  - tensorflow-gcs-config 2.13.0 -> 2.14.0

```
Model Accuracy 0.754
              precision    recall  f1-score   suppor

      benign       0.91      0.56      0.70       50
   malignant       0.68      0.94      0.79       50

    accuracy                           0.75      100
   macro avg       0.80      0.75      0.74      100
weighted avg       0.80      0.75      0.74      100
```



model using U-Net Segmentation

```
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, roc_curve,
from tensorflow.keras.applications import VGG16, ResNet5
from tensorflow.keras.layers import Concatenate, Dense,
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDa
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import VGG16, ResNet5
```

  - tensorflow-hub 0.14.0 -> 0.15.0
  - tensorflow-probability 0.20.1 -> 0.22.0
  - torch 2.0.1 -> 2.1.0
  - torchaudio 2.0.2 -> 2.1.0
  - torchtext 0.15.2 -> 0.16.0
  - torchvision 0.15.2 -> 0.16.0
  - xgboost 1.7.6 -> 2.0.0
- Python package inclusions
  - bigframes 0.10.0
  - malloy 2023.1056

2023-09-22

- Added the ability to scope an AI generated suggestion to a specific Pandas dataframe (tweet)
- Added Colab link previews to Docs (tweet)
- Added smart paste support for data from Google Sheets
- Increased font size of dropdowns in interactive forms
- Improved rendering of the notebook when printing
- Python package upgrades
  - tensorflow 2.12.0 -> 2.13.0
  - tensorboard 2.12.3 -> 2.13.0
  - keras 2.12.0 -> 2.13.1
  - tensorflow-gcs-config 2.12.0 -> 2.13.
  - scipy 1.10.1-> 1.11.2
  - cython 0.29.6 -> 3.0.2
- Python package inclusions
  - geemap 0.26.0

2023-08-18

- Added "Change runtime type" to the menu in the connection button
- Improved auto-reconnection to an already running notebook (#3764)
- Increased the specs of our highmem machines for Pro users
- Fixed `add-apt-repository` command on Ubuntu 22.04 runtime (#3867)
- Python package upgrades
  - bokeh 2.4.3 -> 3.2.2
  - cmake 3.25.2 -> 3.27.2
  - cryptography 3.4.8 -> 41.0.3

```
input_shape = (224, 224, 3)
vgg_model = VGG16(weights='imagenet', include_top=False,
resnet_model = ResNet50(weights='imagenet', include_top=


for layer in vgg_model.layers:
    layer.trainable = False
for layer in resnet_model.layers:
    layer.trainable = False
input_tensor = Input(shape=input_shape)
vgg_output = vgg_model(input_tensor)
resnet_output = resnet_model(input_tensor)
merged = Concatenate()([GlobalAveragePooling2D()(vgg_out
x = Dense(1024, activation='relu')(merged)
#x = Dense(1024, activation='relu')(x)
#x = Dense(256, activation='relu')(x)
output = Dense(2, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output)
model.compile(optimizer=Adam(lr=0.001), loss='sparse_cat
```

```
Downloading data from https://storage.googleapis.com
58889256/58889256 [==============================] -
Downloading data from https://storage.googleapis.com
94765736/94765736 [==============================] -
WARNING:absl:`lr` is deprecated in Keras optimizer,
```

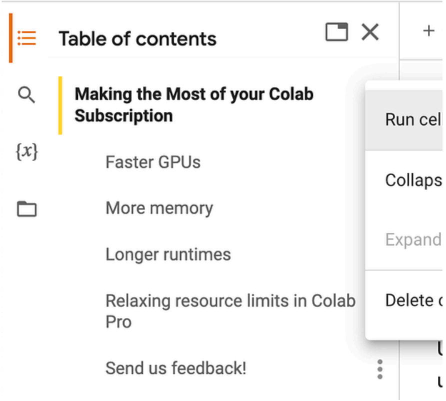
```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)


train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(224, 224),
    batch_size=32,
    class_mode='sparse'
)

validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Melanoma_Skin_Cancer_dis
        target_size=(224, 224),
        batch_size=32,
        class_mode='sparse')
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

  - dask 2022.12.1 -> 2023.8.0
  - distributed 2022.12.1 -> 2023.8.0
  - earthengine-api 0.1.358 -> 0.1.364
  - flax 0.7.0 -> 0.7.2
  - ipython-sql 0.4.0 -> 0.5.0
  - jax 0.4.13 -> 0.4.14
  - jaxlib 0.4.13 -> 0.4.14
  - lightgbm 3.3.5 -> 4.0.0
  - mkl 2019.0 -> 2023.2.0
  - notebook 6.4.8 -> 6.5.5
  - numpy 1.22.4 -> 1.23.5
  - opencv-python 4.7.0.72 -> 4.8.0.76
  - pillow 8.4.0 -> 9.4.0
  - plotly 5.13.1 -> 5.15.0
  - prettytable 0.7.2 -> 3.8.0
  - pytensor 2.10.1 -> 2.14.2
  - spacy 3.5.4 -> 3.6.1
  - statsmodels 0.13.5 -> 0.14.0
  - xarray 2022.12.0 -> 2023.7.0
- Python package inclusions
  - PyDrive2 1.6.3

2023-07-21

- Launched auto-plotting for dataframes, available using the chart button that shows up alongside datatables (post)
- Added a menu to the table of contents to support running a section or collapsing/expanding sections (post)



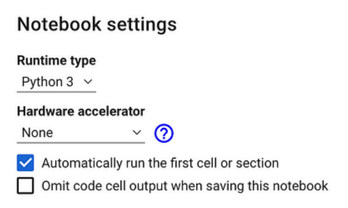
Making the Most of your Colab Subscription
Faster GPUs
More memory
Longer runtimes
Relaxing resource limits in Colab Pro
Send us feedback!

- Added an option to automatically run the first cell or section,

```
history = model.fit_generator(
    train_generator,
    steps_per_epoch=2000 // 32,
    epochs=100,
    validation_data=validation_generator,
    validation_steps=800 // 32
)
```

```
<ipython-input-10-794030e3dcad>:1: UserWarning: `M
  history = model.fit_generator(
Epoch 1/100
62/62 [==============================] - 73s 942ms
Epoch 2/100
62/62 [==============================] - 29s 474ms
Epoch 3/100
62/62 [==============================] - 26s 418ms
Epoch 4/100
62/62 [==============================] - 21s 345ms
Epoch 5/100
62/62 [==============================] - 25s 411ms
Epoch 6/100
62/62 [==============================] - 26s 412ms
Epoch 7/100
62/62 [==============================] - 26s 416ms
Epoch 8/100
62/62 [==============================] - 22s 348ms
Epoch 9/100
62/62 [==============================] - 26s 419ms
Epoch 10/100
62/62 [==============================] - 22s 348ms
Epoch 11/100
62/62 [==============================] - 23s 364ms
Epoch 12/100
62/62 [==============================] - 25s 405ms
Epoch 13/100
62/62 [==============================] - 23s 363ms
Epoch 14/100
62/62 [==============================] - 26s 422ms
Epoch 15/100
62/62 [==============================] - 36s 576ms
Epoch 16/100
62/62 [==============================] - 24s 392ms
Epoch 17/100
62/62 [==============================] - 28s 450ms
Epoch 18/100
62/62 [==============================] - 22s 359ms
Epoch 19/100
62/62 [==============================] - 21s 344ms
Epoch 20/100
62/62 [==============================] - 27s 428ms
Epoch 21/100
62/62 [==============================] - 25s 406ms
Epoch 22/100
62/62 [==============================] - 26s 426ms
Epoch 23/100
```

available under Edit -> Notebook settings (post)



Notebook settings

Runtime type

Python 3

Hardware accelerator

None

☑ Automatically run the first cell or section

□ Omit code cell output when saving this notebook

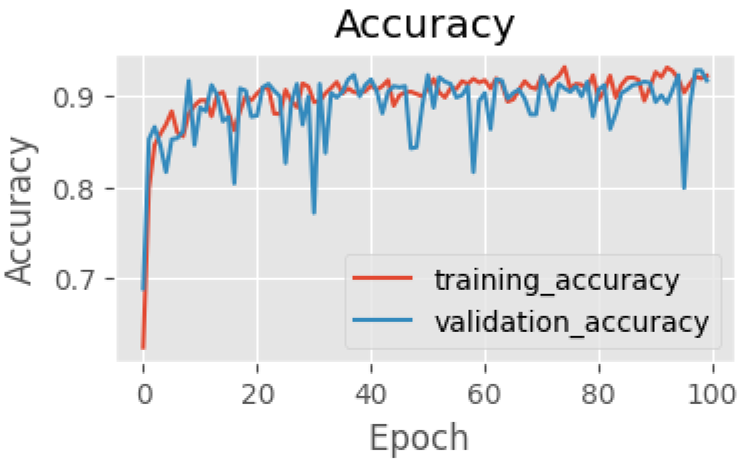
- Launched Pro/Pro+ to Algeria, Argentina, Chile, Ecuador, Egypt, Ghana, Kenya, Malaysia, Nepal, Nigeria, Peru, Rwanda, Saudi Arabia, South Africa, Sri Lanka, Tunisia, and Ukraine (tweet)
- Added a command, "Toggle tab moves focus" for toggling tab trapping in the editor (Tools -> Command palette, "Toggle tab moves focus")
- Fixed issue where `files.upload()` was sometimes returning an incorrect filename (#1550)
- Fixed f-string syntax highlighting bug (#3802)
- Disabled ambiguous characters highlighting for commonly used LaTeX characters (#3648)
- Upgraded Ubuntu from 20.04 LTS to 22.04 LTS
- Updated the Colab Marketplace VM image
- Python package upgrades:
    - autograd 1.6.1 -> 1.6.2
    - drivefs 76.0 -> 77.0
    - flax 0.6.11 -> 0.7.0
    - earthengine-api 0.1.357 -> 0.1.358
    - GDAL 3.3.2->3.4.3
    - google-cloud-bigquery-storage 2.20.0 -> 2.22.2
    - gspread-dataframe 3.0.8 -> 3.3.1
    - holidays 0.27.1 -> 0.29
    - jax 0.4.10 -> jax 0.4.13
    - jaxlib 0.4.10 -> jax 0.4.13
    - jupyterlab-widgets 3.0.7 -> 3.0.8
    - nbformat 5.9.0 -> 5.9.1

```
62/62 [==============================] - 21s 345ms
Epoch 24/100
62/62 [==============================] - 23s 367ms
Epoch 25/100
62/62 [==============================] - 25s 407ms
Epoch 26/100
62/62 [==============================] - 23s 366ms
Epoch 27/100
62/62 [==============================] - 21s 342ms
Epoch 28/100
```

```
model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer_
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals cl
plt.plot(N, history.history["accuracy"], label="training_a
plt.plot(N, history.history["val_accuracy"], label="valida
plt.title("Accuracy")
plt.xlabel("Epoch ")
plt.ylabel("Accuracy")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer_c
```

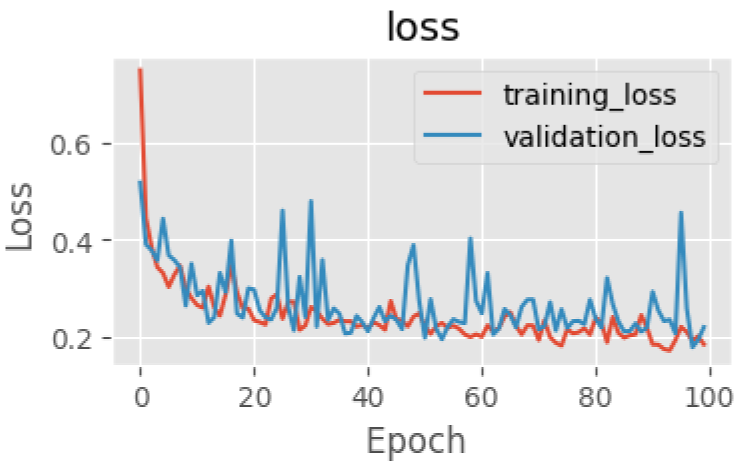


- opencv-python-headless 4.7.0.72 -> 4.8.0.74
- pygame 2.4.0 -> 2.5.0
- spacy 3.5.3 -> 3.5.4
- SQLAlchemy 2.0.16 -> 2.0.19
- tabulate 0.8.10 -> 0.9.0
- tensorflow-hub 0.13.0 -> 0.14.0

2023-06-23

- Launched AI coding features to subscribed users starting with Pro+ users in the US (tweet, post)
- Added the Kernel Selector in the Notebook Settings (tweet)
- Fixed double space trimming issue in markdown #3766
- Fixed run button indicator not always centered #3609
- Fixed inconsistences for automatic indentation on multi-line #3697
- Upgraded Python from 3.10.11 to 3.10.12
- Python package updates:
  - duckdb 0.7.1 -> 0.8.1
  - earthengine-api 0.1.350 -> 0.1.357
  - flax 0.6.9 -> 0.6.11
  - google-cloud-bigquery 3.9.0 -> 3.10.0
  - google-cloud-bigquery-storage 2.19.1 -> 2.20.0
  - grpcio 1.54.0 -> 1.56.0
  - holidays 0.25 -> 0.27.1
  - nbformat 5.8.0 -> 5.9.0
  - prophet 1.1.3 -> 1.1.4
  - pydata-google-auth 1.7.0 -> 1.8.0
  - spacy 3.5.2 -> 3.5.3
  - tensorboard 2.12.2 -> 2.12.3
  - xgboost 1.7.5 -> 1.7.6
- Python package inclusions:
  - gcsfs 2023.6.0
  - geopandas 0.13.2
  - google-cloud-bigquery-connection 1.12.0
  - google-cloud-functions 1.13.0
  - grpc-google-iam-v1 0.12.6
  - multidict 6.0.4

```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["loss"], label="training_los
plt.plot(N, history.history["val_loss"], label="validati
plt.title("loss")
plt.xlabel("Epoch ")
plt.ylabel("Loss")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
import cv2
```

```
test_data_dir='/content/drive/MyDrive/Melanoma_Skin_Canc
test_generator = train_datagen.flow_from_directory(test_
                                                   targe
                                                   batch
```

```
Found 3183 images belonging to 2 classes.
```

```
import tensorflow as tf
model = tf.keras.models.load_model('/content/drive/MyDri
```

  - tensorboard-data-server 0.7.1

2023-06-02

- Released the new site colab.google
- Published Colab's Docker runtime image to us-docker.pkg.dev/colab-images/public/runtime (tweet, instructions)
- Launched support for Google children accounts (tweet)
- Launched DagsHub integration (tweet, post)
- Upgraded to Monaco Editor Version 0.37.1
- Fixed various Vim keybinding bugs
- Fixed issue where the N and P letters sometimes couldn't be typed (#3664)
- Fixed rendering support for compositional inputs (#3660, #3679)
- Fixed lag in notebooks with lots of cells (#3676)
- Improved support for R by adding a Runtime type notebook setting (Edit -> Notebook settings)
- Improved documentation for connecting to a local runtime (Connect -> Connect to a local runtime)
- Python package updates:
  - holidays 0.23 -> 0.25
  - jax 0.4.8 -> 0.4.10
  - jaxlib 0.4.8 -> 0.4.10
  - pip 23.0.1 -> 23.1.2
  - tensorflow-probability 0.19.0 -> 0.20.1
  - torch 2.0.0 -> 2.0.1
  - torchaudio 2.0.1 -> 2.0.2
  - torchdata 0.6.0 -> 0.6.1
  - torchtext 0.15.1 -> 0.15.2
  - torchvision 0.15.1 -> 0.15.2
  - tornado 6.2 -> 6.3.1

2023-05-05

- Released GPU type selection for paid users, allowing them to choose a preferred NVidia GPU
- Upgraded R from 4.2.3 to 4.3.0

```
import cv2
def mode(my_list):
    ct = Counter(my_list)
    max_value = max(ct.values())
    return ([key for key, value in ct.items() if value =

true_value = []
model_one_pred = []
for folder in os.listdir(test_data_dir):

    test_image_ids = os.listdir(os.path.join(test_data_d

    for image_id in test_image_ids[:int(len(test_image_i

        path = os.path.join(test_data_dir,folder,image_i

        true_value.append(test_generator.class_indices[f
        img = cv2.resize(cv2.imread(path),(224,224))
        img_normalized = img/255

        model_one_prediction = np.argmax(model.predict(n
        model_one_pred.append(model_one_prediction)
```

```
1/1 [==============================] - 6s 6s/step
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 31ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 34ms/ste
1/1 [==============================] - 0s 33ms/ste
1/1 [==============================] - 0s 35ms/ste
1/1 [==============================] - 0s 33ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 28ms/ste
1/1 [==============================] - 0s 29ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 22ms/ste
```

- Upgraded Python from 3.9.16 to 3.10.11
- Python package updates:
    - attrs 22.2.0 -> attrs 23.1.0
    - earthengine-api 0.1.349 -> earthengine-api 0.1.350
    - flax 0.6.8 -> 0.6.9
    - grpcio 1.53.0 -> 1.54.0
    - nbclient 0.7.3 -> 0.7.4
    - tensorflow-datasets 4.8.3 -> 4.9.2
    - termcolor 2.2.0 -> 2.3.0
    - zict 2.2.0 -> 3.0.0

2023-04-14

- Python package updates:
    - google-api-python-client 2.70.0 -> 2.84.0
    - google-auth-oauthlib 0.4.6 -> 1.0.0
    - google-cloud-bigquery 3.4.2 -> 3.9.0
    - google-cloud-datastore 2.11.1 -> 2.15.1
    - google-cloud-firestore 2.7.3 -> 2.11.0
    - google-cloud-language 2.6.1 -> 2.9.1
    - google-cloud-storage 2.7.0 -> 2.8.0
    - google-cloud-translate 3.8.4 -> 3.11.1
    - networkx 3.0 -> 3.1
    - notebook 6.3.0 -> 6.4.8
    - jax 0.4.7 -> 0.4.8
    - pandas 1.4.4 -> 1.5.3
    - spacy 3.5.1 -> 3.5.2
    - SQLAlchemy 1.4.47 -> 2.0.9
    - xgboost 1.7.4 -> 1.7.5

2023-03-31

- Improve bash ! syntax highlighting (GitHub issue)
- Fix bug where VIM keybindings weren't working in the file editor
- Upgraded R from 4.2.2 to 4.2.3
- Python package updates:
    - arviz 0.12.1 --> 0.15.1
    - astropy 4.3.1 --> 5.2.2
    - dopamine-rl 1.0.5 --> 4.0.6
    - gensim 3.6.0 --> 4.3.1
    - ipykernel 5.3.4 -> 5.5.6
    - ipython 7.9.0 -> 7.34.0

```
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 31ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 38ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 21ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 35ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 21ms/ste
```

  - jax 0.4.4 -> 0.4.7
  - jaxlib 0.4.4 -> 0.4.7
  - jupyter_core 5.2.0 -> 5.3.0
  - keras 2.11.0 -> 2.12.0
  - lightgbm 2.2.3 -> 3.3.5
  - matplotlib 3.5.3 -> 3.7.1
  - nltk 3.7 -> 3.8.1
  - opencv-python 4.6.0.66 -> 4.7.0.72
  - plotly 5.5.0 -> 5.13.1
  - pymc 4.1.4 -> 5.1.2
  - seaborn 0.11.2 -> 0.12.2
  - spacy 3.4.4 -> 3.5.1
  - sympy 1.7.1 -> 1.11.1
  - tensorboard 2.11.2 -> 2.12.0
  - tensorflow 2.11.0 -> 2.12.0
  - tensorflow-estimator 2.11.0 -> 2.12.0
  - tensorflow-hub 0.12.0 -> 0.13.0
  - torch 1.13.1 -> 2.0.0
  - torchaudio 0.13.1 -> 2.0.1
  - torchtext 0.14.1 -> 0.15.1
  - torchvision 0.14.1 -> 0.15.1

2023-03-10

- Added the Colab editor shortcuts example notebook
- Fixed triggering of @-mention and email autocomplete for large comments (GitHub issue)
- Added View Resources to the Runtime menu
- Made file viewer images fit the view by default, resizing to original size on click
- When in VIM mode, enable copy as well as allowing propagation to monaco-vim to escape visual mode (GitHub issue)
- Upgraded CUDA 11.6.2 -> 11.8.0 and cuDNN 8.4.0.27 -> 8.7.0.84
- Upgraded Nvidia drivers 525.78.01 -> 530.30.02
- Upgraded Python 3.8.10 -> 3.9.16
- Python package updates:
  - beautifulsoup4 4.6.3 -> 4.9.3
  - bokeh 2.3.3 -> 2.4.3
  - debugpy 1.0.0 -> 1.6.6
  - Flask 1.1.4 -> 2.2.3
  - jax 0.3.25 -> 0.4.4
  - jaxlib 0.3.25 -> 0.4.4
  - Jinja2 2.11.3 -> 3.1.2

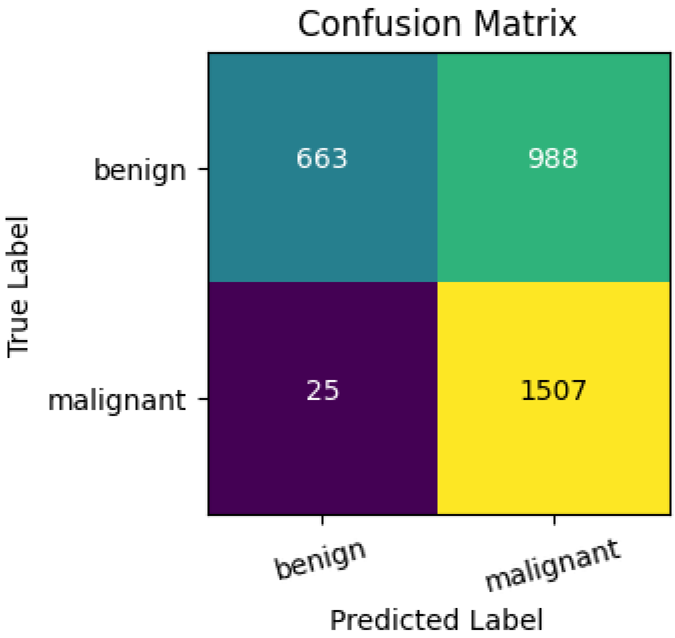
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import itertools
#from mlxtend.plotting import plot_confusion_matrix
def clf_report(true_value, model_pred):

    classes = test_generator.class_indices.keys()
    TP_count = [true_value[i] == model_pred[i] for i in
    model_accuracy = np.sum(TP_count)/len(TP_count)
    print('Model Accuracy', model_accuracy)

    plt.figure(figsize=(3,3))
    cm = confusion_matrix(true_value,model_pred)
    plt.imshow(cm,interpolation='nearest',cmap=plt.cm.vi
    plt.title('Confusion Matrix')
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=15) #15 ref
    plt.yticks(tick_marks, classes)
    thresh = cm.max()*0.8
    for i,j in itertools.product(range(cm.shape[0]),rang
        plt.text(j,i,cm[i,j],
                horizontalalignment="center",
                color="black" if cm[i,j] > thresh else "
        pass

    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    pass

    print(classification_report(true_value, model_pred,



clf_report(true_value, model_one_pred)
```

  - matplotlib 3.2.2 -> 3.5.3
  - nbconvert 5.6.1 -> 6.5.4
  - pandas 1.3.5 -> 1.4.4
  - pandas-datareader 0.9.0 -> 0.10.0
  - pandas-profiling 1.4.1 -> 3.2.0
  - Pillow 7.1.2 -> 8.4.0
  - plotnine 0.8.0 -> 0.10.1
  - scikit-image 0.18.3 -> 0.19.3
  - scikit-learn 1.0.2 -> 1.2.2
  - scipy 1.7.3 -> 1.10.1
  - setuptools 57.4.0 -> 63.4.3
  - sklearn-pandas 1.8.0 -> 2.2.0
  - statsmodels 0.12.2 -> 0.13.5
  - urllib3 1.24.3 -> 1.26.14
  - Werkzeug 1.0.1 -> 2.2.3
  - wrapt 1.14.1 -> 1.15.0
  - xgboost 0.90 -> 1.7.4
  - xlrd 1.2.0 -> 2.0.1

2023-02-17

- Show graphs of RAM and disk usage in notebook toolbar
- Copy cell links directly to the clipboard instead of showing a dialog when clicking on the link icon in the cell toolbar
- Updated the Colab Marketplace VM image
- Upgraded CUDA to 11.6.2 and cuDNN to 8.4.0.27
- Python package updates:
  - tensorflow 2.9.2 -> 2.11.0
  - tensorboard 2.9.1 -> 2.11.2
  - keras 2.9.0 -> 2.11.0
  - tensorflow-estimator 2.9.0 -> 2.11.0
  - tensorflow-probability 0.17.0 -> 0.19.0
  - tensorflow-gcs-config 2.9.0 -> 2.11.0
  - earthengine-api 0.1.339 -> 0.1.341
  - flatbuffers 1.12 -> 23.1.21
  - platformdirs 2.6.2 -> 3.0.0
  - pydata-google-auth 1.6.0 -> 1.7.0
  - python-utils 3.4.5 -> 3.5.2
  - tenacity 8.1.0 -> 8.2.1
  - tifffile 2023.1.23.1 -> 2023.2.3

```
Model Accuracy 0.681746779767515
              precision    recall  f1-score   suppor

      benign       0.96      0.40      0.57       165
   malignant       0.60      0.98      0.75       153

    accuracy                           0.68       318
   macro avg       0.78      0.69      0.66       318
weighted avg       0.79      0.68      0.65       318
```



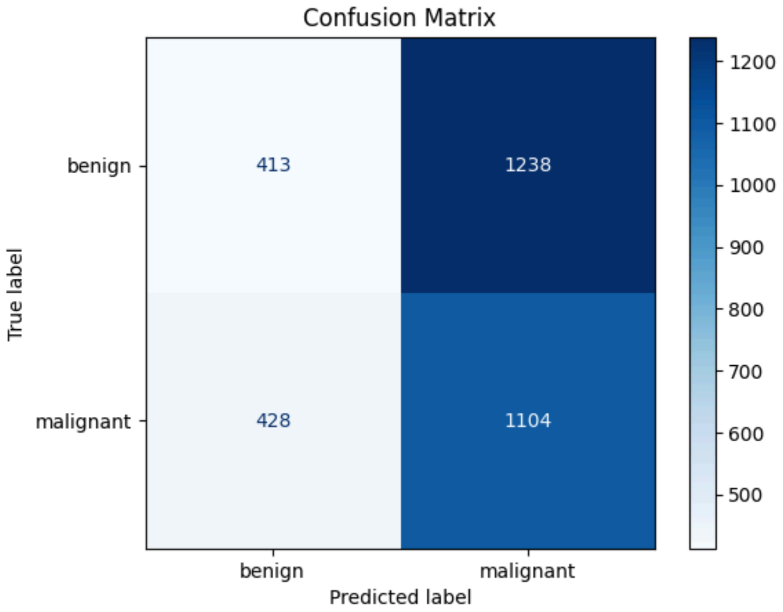
```
import numpy as np
from sklearn.metrics import confusion_matrix, ConfusionM
import matplotlib.pyplot as plt

# Step 1: Make Predictions
predictions = model.predict(test_generator, steps=len(te

# Step 2: Convert Predictions and Ground Truths to Label
predicted_classes = np.argmax(predictions, axis=1)  # As
true_classes = test_generator.classes
class_labels = list(test_generator.class_indices.keys())

# Step 3: Compute Confusion Matrix
cm = confusion_matrix(true_classes, predicted_classes)

# Step 4: Visualize the Confusion Matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, displ
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.show()
```

  - notebook 5.7.16 -> 6.3.0
  - tornado 6.0.4 -> 6.2
  - aiohttp 3.8.3 -> 3.8.4
  - charset-normalizer 2.1.1 -> 3.0.1
  - fastai 2.7.0 -> 2.7.1
  - soundfile 0.11.0 -> 0.12.1
  - typing-extensions 4.4.0 -> 4.5.0
  - widgetsnbextension 3.6.1 -> 3.6.2
  - pydantic 1.10.4 -> 1.10.5
  - zipp 3.12.0 -> 3.13.0
  - numpy 1.21.6 -> 1.22.4
  - drivefs 66.0 -> 69.0
  - gdal 3.0.4 -> 3.3.2 GitHub issue
- Added libudunits2-dev for smoother R package installs GitHub issue

2023-02-03

- Improved tooltips for pandas series to show common statistics about the series object
- Made the forms dropdown behave like an autocomplete box when it allows input
- Updated the nvidia driver from 460.32.03 to 510.47.03
- Python package updates:
  - absl-py 1.3.0 -> 1.4.0
  - bleach 5.0.1 -> 6.0.0
  - cachetools 5.2.1 -> 5.3.0
  - cmdstanpy 1.0.8 -> 1.1.0
  - dnspython 2.2.1 -> 2.3.0
  - fsspec 2022.11.0 -> 2023.1.0
  - google-cloud-bigquery-storage 2.17.0 -> 2.18.1
  - holidays 0.18 -> 0.19
  - jupyter-core 5.1.3 -> 5.2.0
  - packaging 21.3 -> 23.0
  - prometheus-client 0.15.0 -> 0.16.0
  - pyct 0.4.8 -> 0.5.0
  - pydata-google-auth 1.5.0 -> 1.6.0
  - python-slugify 7.0.0 -> 8.0.0
  - sqlalchemy 1.4.46 -> 2.0.0
  - tensorflow-io-gcs-filesystem 0.29.0 -> 0.30.0
  - tifffile 2022.10.10 -> 2023.1.23.1
  - zipp 3.11.0 -> 3.12.0

```
100/100 [==============================] - 38s 276ms
```


Confusion Matrix

  - Pinned sqlalchemy to version 1.4.46

2023-01-12

- Added support for @-mention and email autocomplete in comments
- Improved errors when GitHub notebooks can't be loaded
- Increased color contrast for colors used for syntax highlighting in the code editor
- Added terminal access for custom GCE VM runtimes
- Upgraded Ubuntu from 18.04 LTS to 20.04 LTS (GitHub issue)
- Python package updates:
  - GDAL 2.2.2 -> 2.2.3.
  - NumPy from 1.21.5 to 1.21.6.
  - attrs 22.1.0 -> 22.2.0
  - chardet 3.0.4 -> 4.0.0
  - cloudpickle 1.6.0 -> 2.2.0
  - filelock 3.8.2 -> 3.9.0
  - google-api-core 2.8.2 -> 2.11.0
  - google-api-python-client 1.12.11 -> 2.70.0
  - google-auth-httplib2 0.0.3 -> 0.1.0
  - google-cloud-bigquery 3.3.5 -> 3.4.1
  - google-cloud-datastore 2.9.0 -> 2.11.0
  - google-cloud-firestore 2.7.2 -> 2.7.3
  - google-cloud-storage 2.5.0 -> 2.7.0
  - holidays 0.17.2 -> holidays 0.18
  - importlib-metadata 5.2.0 -> 6.0.0
  - networkx 2.8.8 -> 3.0
  - opencv-python-headless 4.6.0.66 -> 4.7.0.68
  - pip 21.1.3 -> 22.04
  - pip-tools 6.2.0 -> 6.6.2
  - prettytable 3.5.0 -> 3.6.0
  - requests 2.23.0 -> 2.25.1
  - termcolor 2.1.1 -> 2.2.0
  - torch 1.13.0 -> 1.13.1
  - torchaudio 0.13.0 -> 0.13.1
  - torchtext 0.14.0-> 0.14.1
  - torchvision 0.14.0 -> 0.14.1

```
import numpy as np
from sklearn.metrics import accuracy_score, precision_sc
import matplotlib.pyplot as plt

# Step 1: Make Predictions
predictions = model.predict(test_generator, steps=len(te

# Step 2: Convert Predictions and Ground Truths to Label
predicted_classes = np.argmax(predictions, axis=1)  # Co
true_classes = test_generator.classes  # Ground truth la
class_labels = list(test_generator.class_indices.keys())

# Step 3: Compute Metrics
accuracy = accuracy_score(true_classes, predicted_classe
precision = precision_score(true_classes, predicted_clas
recall = recall_score(true_classes, predicted_classes, a
cm = confusion_matrix(true_classes, predicted_classes)

# Step 4: Display Metrics and Confusion Matrix
for i, label in enumerate(class_labels):
    print(f"Class: {label}")
    print(f"Precision: {precision[i]}")
    print(f"Recall: {recall[i]}")
    print("-------------------------")

disp = ConfusionMatrixDisplay(confusion_matrix=cm, displ
disp.plot(cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.show()

print(f"Overall Accuracy: {accuracy}")
```

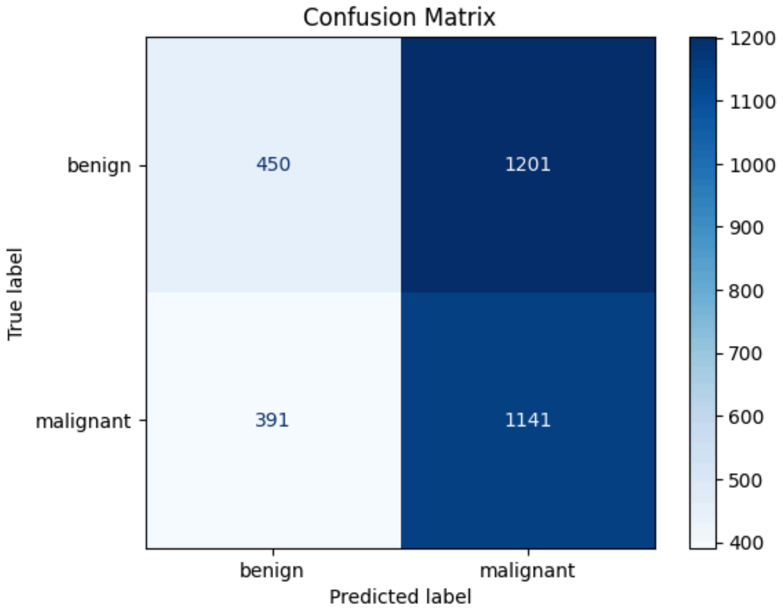
2022-12-06

- Made fallback runtime version available until mid-December (GitHub issue)
- Upgraded to Python 3.8 (GitHub issue)
- Python package updates:
  - jax from 0.3.23 to 0.3.25, jaxlib from 0.3.22 to 0.3.25
  - pyarrow from 6.0.1 to 9.0.0
  - torch from 1.12.1 to 1.13.0
  - torchaudio from 0.12.1 to 0.13.0
  - torchvision from 0.13.1 to 0.14.0
  - torchtext from 0.13.1 to 0.14.0
  - xlrd from 1.1.0 to 1.2.0
  - DriveFS from 62.0.1 to 66.0.3
- Made styling of markdown tables in outputs match markdown tables in text cells
- Improved formatting for empty interactive table rows
- Fixed syntax highlighting for variables with names that contain Python keywords (GitHub issue)

2022-11-11

- Added more dark editor themes for Monaco (when in dark mode, "Editor colorization" appears as an option in the Editor tab of the Tools → Settings dialog)
- Fixed bug where collapsed forms were deleted on mobile GitHub issue
- Python package updates:
  - rpy2 from 3.4.0 to 3.5.5 (GitHub issue)
  - notebook from 5.5.0 to 5.7.16
  - tornado from 5.1.1 to 6.0.4
  - tensorflow_probability from 0.16.0 to 0.17.0
  - pandas-gbq from 0.13.3 to 0.17.9
  - protobuf from 3.17.3 to 3.19.6
  - google-api-core[grpc] from 1.31.5 to 2.8.2

```
100/100 [==============================] - 22s 221ms
Class: benign
Precision: 0.535077288941736
Recall: 0.2725620835857056
-------------------------
Class: malignant
Precision: 0.48719043552519214
Recall: 0.7447780678851175
-------------------------
```



```
Overall Accuracy: 0.4998429154885328
```

Start coding or generate with AI.

**Changing learning rate and fully conected layers*

  - google-cloud-bigquery from 1.21.0 to 3.3.5
  - google-cloud-core from 1.0.1 to 2.3.2
  - google-cloud-datastore from 1.8.0 to 2.9.0
  - google-cloud-firestore from 1.7.0 to 2.7.2
  - google-cloud-language from 1.2.0 to 2.6.1
  - google-cloud-storage from 1.18.0 to 2.5.0
  - google-cloud-translate from 1.5.0 to 3.8.4

2022-10-21

- Launched a single-click way to get from BigQuery to Colab to further explore query results (announcement)
- Launched Pro, Pro+, and Pay As You Go to 19 additional countries: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, Greece, Hungary, Latvia, Lithuania, Norway, Portugal, Romania, Slovakia, Slovenia, and Sweden (tweet)
- Updated jax from 0.3.17 to 0.3.23, jaxlib from 0.3.15 to 0.3.22, TensorFlow from 2.8.2 to 2.9.2, CUDA from 11.1 to 11.2, and cuDNN from 8.0 to 8.1 (backend-info)
- Added a `readonly` option to `drive.mount`
- Fixed bug where Xarray was not working (GitHub issue)
- Modified Markdown parsing to ignore block quote symbol within MathJax (GitHub issue)

2022-09-30

- Launched Pay As You Go, allowing premium GPU access without requiring a subscription
- Added vim and tcllib to our runtime image
- Fixed bug where open files were closed on kernel disconnect (GitHub issue)
- Fixed bug where the play button/execution indicator was

```
input_shape = (224, 224, 3)
vgg_model = VGG16(weights='imagenet', include_top=False,
resnet_model = ResNet50(weights='imagenet', include_top=


for layer in vgg_model.layers:
    layer.trainable = False
for layer in resnet_model.layers:
    layer.trainable = False
input_tensor = Input(shape=input_shape)
vgg_output = vgg_model(input_tensor)
resnet_output = resnet_model(input_tensor)
merged = Concatenate()([GlobalAveragePooling2D()(vgg_out
x = Dense(1024, activation='relu')(merged)
x = Dense(1024, activation='relu')(x)
#x = Dense(256, activation='relu')(x)
output = Dense(2, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output)
model.compile(optimizer=Adam(lr=0.001), loss='sparse_cat
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer,
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(224, 224),
    batch_size=32,
    class_mode='sparse'
)

validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Melanoma_Skin_Cancer_dis
        target_size=(224, 224),
        batch_size=32,
        class_mode='sparse')
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

```
history = model.fit_generator(
    train_generator,
    steps_per_epoch=2000 // 32,
    epochs=100,
    validation_data=validation_generator,
    validation_steps=800 // 32
)
```

not clickable when scrolled into the cell output (GitHub issue)
- Updated the styling for form titles so that they avoid obscuring the code editor
- Created a GitHub repo, backend-info, with the latest apt-list.txt and pip-freeze.txt files for the Colab runtime (GitHub issue)
- Added `files.upload_file(filename)` to upload a file from the browser to the runtime with a specified filename

2022-09-16

- Upgraded pymc from 3.11.0 to 4.1.4, jax from 0.3.14 to 0.3.17, jaxlib from 0.3.14 to 0.3.15, fsspec from 2022.8.1 to 2022.8.2
- Modified our save flow to avoid persisting Drive filenames as titles in notebook JSON
- Updated our Terms of Service
- Modified the `Jump to Cell` command to locate the cursor at the end of the command palette input (`Jump to cell` in Tools → Command palette in a notebook with section headings)
- Updated the styling of the Drive notebook comment UI
- Added support for terminating your runtime from code: `python from google.colab import runtime runtime.unassign()`
- Added regex filter support to the Recent notebooks dialog
- Inline google.colab.files.upload JS to fix `files.upload()` not working (GitHub issue)

2022-08-26

- Upgraded PyYAML from 3.13 to 6.0 (GitHub issue), drivefs from 61.0.3 to 62.0.1
- Upgraded TensorFlow from 2.8.2 to 2.9.1 and ipywidgets from 7.7.1 to 8.0.1 but rolled both back due to a number of user reports (GitHub issue, GitHub issue)
- Stop persisting inferred titles in notebook JSON (GitHub issue)

```
<ipython-input-5-794030e3dcad>:1: UserWarning: `Mo
  history = model.fit_generator(
Epoch 1/100
62/62 [==============================] - 778s 12s/
Epoch 2/100
62/62 [==============================] - 531s 9s/s
Epoch 3/100
62/62 [==============================] - 345s 6s/s
Epoch 4/100
62/62 [==============================] - 258s 4s/s
Epoch 5/100
62/62 [==============================] - 206s 3s/s
Epoch 6/100
62/62 [==============================] - 138s 2s/s
Epoch 7/100
62/62 [==============================] - 139s 2s/s
Epoch 8/100
62/62 [==============================] - 82s 1s/st
Epoch 9/100
62/62 [==============================] - 78s 1s/st
Epoch 10/100
62/62 [==============================] - 48s 773ms
Epoch 11/100
62/62 [==============================] - 39s 625ms
Epoch 12/100
62/62 [==============================] - 40s 642ms
Epoch 13/100
62/62 [==============================] - 37s 600ms
Epoch 14/100
62/62 [==============================] - 27s 430ms
Epoch 15/100
62/62 [==============================] - 36s 578ms
Epoch 16/100
62/62 [==============================] - 22s 351ms
Epoch 17/100
62/62 [==============================] - 22s 359ms
Epoch 18/100
62/62 [==============================] - 24s 390ms
Epoch 19/100
62/62 [==============================] - 25s 398ms
Epoch 20/100
62/62 [==============================] - 24s 390ms
Epoch 21/100
62/62 [==============================] - 25s 396ms
Epoch 22/100
62/62 [==============================] - 25s 396ms
Epoch 23/100
62/62 [==============================] - 20s 315ms
Epoch 24/100
62/62 [==============================] - 20s 319ms
Epoch 25/100
62/62 [==============================] - 20s 316ms
Epoch 26/100
62/62 [==============================] - 24s 392ms
Epoch 27/100
```

- Fix bug in background execution which affected some Pro+ users (GitHub issue)
- Fix bug where `Download as .py` incorrectly handled text cells ending in a double quote
- Fix bug for Pro and Pro+ users where we weren't honoring the preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Provide undo/redo for scratch cells
- When writing ipynb files, serialize empty multiline strings as `[]` for better consistency with JupyterLab

2022-08-11

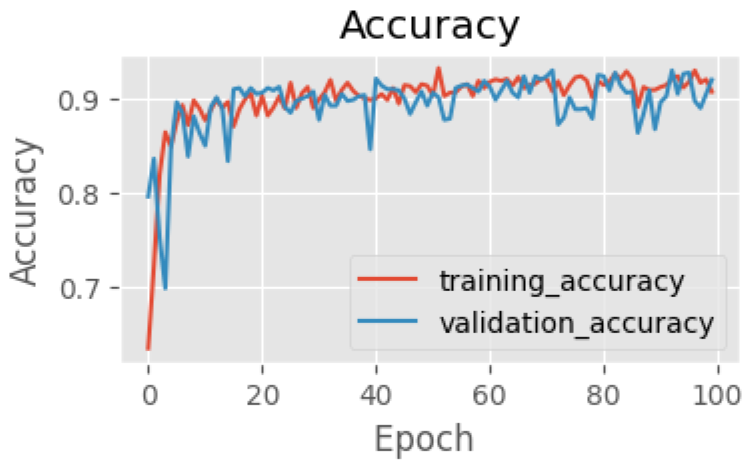
- Upgraded ipython from 5.5.0 to 7.9.0, fbprophet 0.7 to prophet 1.1, tensorflow-datasets from 4.0.1 to 4.6.0, drivefs from 60.0.2 to 61.0.3, pytorch from 1.12.0 to 1.12.1, numba from 0.51 to 0.56, and lxml from 4.2.0 to 4.9.1
- Loosened our `requests` version requirement (GitHub issue)
- Removed support for TensorFlow 1
- Added Help → Report Drive abuse for Drive notebooks
- Fixed indentation for Python lines ending in [
- Modified styling of tables in Markdown to left-align them rather than centering them
- Fixed special character replacement when copying interactive tables as Markdown
- Fixed ansi 8-bit color parsing (GitHub issue)
- Configured logging to preempt transitive imports and other loading from implicitly configuring the root logger
- Modified forms to use a value of `None` instead of causing a parse error when clearing raw and numeric-typed form fields

2022-07-22

- Update scipy from 1.4.1 to 1.7.3, drivefs from 59.0.3 to 60.0.2,

```
    62/62 [==============================] - 24s 388ms
    Epoch 28/100
```

```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["accuracy"], label="training
plt.plot(N, history.history["val_accuracy"], label="vali
plt.title("Accuracy")
plt.xlabel("Epoch ")
plt.ylabel("Accuracy")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals cl
plt.plot(N, history.history["loss"], label="training_loss'
plt.plot(N, history.history["val_loss"], label="validatior
plt.title("loss")
plt.xlabel("Epoch ")
plt.ylabel("Loss")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer_c
```

pytorch from 1.11 to 1.12, jax & jaxlib from 0.3.8 to 0.3.14, opencv-python from 4.1.2.30 to 4.6.0.66, spaCy from 3.3.1 to 3.4.0, and dlib from 19.18.0 to 19.24.0

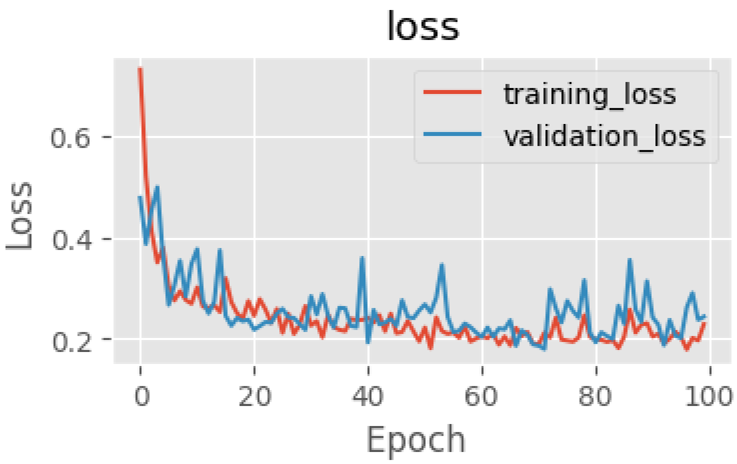
- Fix `Open in tab` doc link which was rendering incorrectly (GitHub issue)
- Add a preference for the default tab orientation to the Site section of the settings menu under Tools → Settings
- Show a warning for `USE_AUTH_EPHEM` usage when running authenticate_user on a TPU runtime (code)

2022-07-01

- Add a preference for code font to the settings menu under Tools → Settings
- Update drivefs from 58.0.3 to 59.0.3 and spacy from 2.2.4 to 3.3.1
- Allow display_data and execute_result text outputs to wrap, matching behavior of JupyterLab (does not affect stream outputs/print statements).
- Improve LSP handling of some magics, esp. %%writefile (GitHub issue).
- Add a FAQ entry about the mount Drive button behavior and include link buttons for each FAQ entry.
- Fix bug where the notebook was sometimes hidden behind other tabs on load when in single pane view.
- Fix issue with inconsistent scrolling when an editor is in multi-select mode.
- Fix bug where clicking on a link in a form would navigate away from the notebook
- Show a confirmation dialog before performing Replace all from the Find and replace pane.

2022-06-10

- Update drivefs from 57.0.5 to 58.0.3 and tensorflow from 2.8.0 to 2.8.2



```
model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer_
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

=================U net segmentation===========================

- Support more than 100 repos in the GitHub repo selector shown in the open dialog and the clone to GitHub dialog
- Show full notebook names on hover in the open dialog
- Improve the color contrast for links, buttons, and the `ipywidgets.Accordion` widget in dark mode

2022-05-20

- Support URL params for linking to some common pref settings: force_theme=dark, force_corgi_mode=1, force_font_size=14. Params forced by URL are not persisted unless saved using Tools → Settings.
- Add a class `markdown-google-sans` to allow Markdown to render in Google Sans
- Update monaco-vim from 0.1.19 to 0.3.4
- Update drivefs from 55.0.3 to 57.0.5, jax from 0.3.4 to 0.3.8, and jaxlib from 0.3.2 to 0.3.7

2022-04-29

- Added 🦀 mode (under Miscellaneous in Tools → Settings)
- Added "Disconnect and delete runtime" option to the menu next to the Connect button
- Improved rendering of filter options in an interactive table
- Added git-lfs to the base image
- Updated torch from 1.10.0 to 1.11.0, jupyter-core from 4.9.2 to 4.10.0, and cmake from 3.12.0 to 3.22.3
- Added more details to our FAQ about unsupported uses (using proxies, downloading torrents, etc.)
- Fixed issue with apt-get dependencies

2022-04-15

- Add an option in the file browser to show hidden files.

```
def unet_model(input_shape):
    inputs = Input(input_shape)

    # Encoder
    conv1 = Conv2D(32, (3, 3), activation='relu', paddin
    conv1 = Conv2D(32, (3, 3), activation='relu', paddin
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(64, (3, 3), activation='relu', paddin
    conv2 = Conv2D(64, (3, 3), activation='relu', paddin
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

    # Bottleneck
    conv3 = Conv2D(128, (3, 3), activation='relu', paddi
    conv3 = Conv2D(128, (3, 3), activation='relu', paddi

    # Decoder
    up4 = Conv2DTranspose(64, (2, 2), strides=(2, 2), pa
    up4 = concatenate([up4, conv2], axis=3)
    conv4 = Conv2D(64, (3, 3), activation='relu', paddin
    conv4 = Conv2D(64, (3, 3), activation='relu', paddin

    up5 = Conv2DTranspose(32, (2, 2), strides=(2, 2), pa
    up5 = concatenate([up5, conv1], axis=3)
    conv5 = Conv2D(32, (3, 3), activation='relu', paddin
    conv5 = Conv2D(32, (3, 3), activation='relu', paddin

    outputs = Conv2D(1, (1, 1), activation='sigmoid')(co

    model = Model(inputs=[inputs], outputs=[outputs])
    model.compile(optimizer=tf.keras.optimizers.legacy.A

    return model
```

- Upgrade gdown from 4.2.0 to 4.4.0, google-api-core[grpc] from 1.26.0 to 1.31.5, and pytz from 2018.4 to 2022.1

2022-03-25

- Launched Pro/Pro+ to 12 additional countries: Australia, Bangladesh, Colombia, Hong Kong, Indonesia, Mexico, New Zealand, Pakistan, Philippines, Singapore, Taiwan, and Vietnam
- Added `google.colab.auth.authenticat` to support using Service Account keys
- Update jax from 0.3.1 to 0.3.4 & jaxlib from 0.3.0 to 0.3.2
- Fixed an issue with Twitter previews of notebooks shared as GitHub Gists

2022-03-10

- Launched Pro/Pro+ to 10 new countries: Ireland, Israel, Italy, Morocco, the Netherlands, Poland, Spain, Switzerland, Turkey, and the United Arab Emirates
- Launched support for scheduling notebooks for Pro+ users
- Fixed bug in interactive datatables where filtering by number did not work
- Finished removing the python2 kernelspec

2022-02-25

- Made various accessibility improvements to the header
- Fix bug with forms run:auto where a form field change would trigger multiple runs
- Minor updates to the bigquery example notebook and snippet
- Include background execution setting in the sessions dialog for Pro+ users
- Update tensorflow-probability from 0.15 to 0.16
- Update jax from 0.2.25 to 0.3.1 & jaxlib from 0.1.71 to 0.3.0

2022-02-11

```
# Image data generator with augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    fill_mode='nearest'
)

test_datagen = ImageDataGenerator(rescale=1./255)

# Flow from directory with target size and class mode
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(256, 256),
    batch_size=16,
    class_mode='input',
    classes=['benign', 'malignant'],
    shuffle=True
)

validation_generator = test_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(256, 256),
    batch_size=16,
    class_mode='input',
    classes=['benign', 'malignant'],
    shuffle=False
)
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

- Improve keyboard navigation for the open dialog
- Fix issue where nvidia-smi stopped reporting resource utilization for some users who were modifying the version of nvidia used
- Update tensorflow from 2.7 to 2.8, keras from 2.7 to 2.8, numpy from 1.19.5 to 1.21.5, tables from 3.4.4 to 3.7.0

2022-02-04

- Improve UX for opening content alongside your notebook, such as files opened from the file browser. This includes a multi-pane view and drag-drop support
- Better Twitter previews when sharing example Colab notebooks and notebooks opened from GitHub Gists
- Update pandas from 1.1.5 to 1.3.5
- Update openpyxl from 2.5.9 to 3.0.0 and pyarrow from 3.0.0 to 6.0.0
- Link to the release notes from the Help menu

2022-01-28

- Add a copy button to data tables
- Python LSP support for better completions and code diagnostics. This can be configured in the Editor Settings (Tools → Settings)
- Update gspread examples in our documentation
- Update gdown from 3.6 to 4.2

2022-01-21

- New documentation for the `google.colab` package
- Show GPU RAM in the resource usage tab
- Improved security for mounting Google Drive which disallows mounting Drive from accounts other than the one currently executing the notebook

2022-01-14

```
import tensorflow as tf

input_shape = (256, 256, 3)  # Assuming input images are
model = unet_model(input_shape)

# Train the model
model.fit(
    train_generator,
    steps_per_epoch=len(train_generator),
    epochs=10,
    validation_data=validation_generator,
    validation_steps=len(validation_generator)
)
```

```
Epoch 1/10
464/464 [==============================] - 1902s 4s/
Epoch 2/10
464/464 [==============================] - 202s 435m
Epoch 3/10
464/464 [==============================] - 197s 425m
Epoch 4/10
464/464 [==============================] - 198s 427m
Epoch 5/10
464/464 [==============================] - 194s 419m
Epoch 6/10
464/464 [==============================] - 200s 431m
Epoch 7/10
464/464 [==============================] - 197s 424m
Epoch 8/10
464/464 [==============================] - 200s 431m
Epoch 9/10
464/464 [==============================] - 195s 420m
Epoch 10/10
464/464 [==============================] - 199s 429m
<keras.src.callbacks.History at 0x7f3458347400>
```

```
model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer_
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

```
import tensorflow as tf

# Load the model using tf.keras.models.load_model if it'
model = tf.keras.models.load_model('/content/drive/MyDri

# Now 'model' is ready to use for predictions or other o
```

- Add a preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Fix bug where / and : weren't working in VIM mode
- Update gspread from 3.0 to 3.4
- Update the Colab Marketplace VM image

Start coding or generate with AI.

=========Model after U net Segmenntation ====================

```
input_shape = (224, 224, 3)
vgg_model = VGG16(weights='imagenet', include_top=False,
resnet_model = ResNet50(weights='imagenet', include_top=


for layer in vgg_model.layers:
    layer.trainable = False
for layer in resnet_model.layers:
    layer.trainable = False
input_tensor = Input(shape=input_shape)
vgg_output = vgg_model(input_tensor)
resnet_output = resnet_model(input_tensor)
merged = Concatenate()([GlobalAveragePooling2D()(vgg_out
x = Dense(1024, activation='relu')(merged)
x = Dense(1024, activation='relu')(x)
#x = Dense(256, activation='relu')(x)
output = Dense(2, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output)
model.compile(optimizer=Adam(lr=0.001), loss='sparse_cat
```

```
Downloading data from https://storage.googleapis.com
58889256/58889256 [==============================] -
Downloading data from https://storage.googleapis.com
94765736/94765736 [==============================] -
WARNING:absl:`lr` is deprecated in Keras optimizer,
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(224, 224),
    batch_size=32,
    class_mode='sparse'
)

validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Melanoma_Skin_Cancer_dis
        target_size=(224, 224),
        batch_size=32,
        class_mode='sparse')
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

```
history = model.fit_generator(
    train_generator,
    steps_per_epoch=2000 // 32,
    epochs=100,
    validation_data=validation_generator,
    validation_steps=800 // 32
)
```

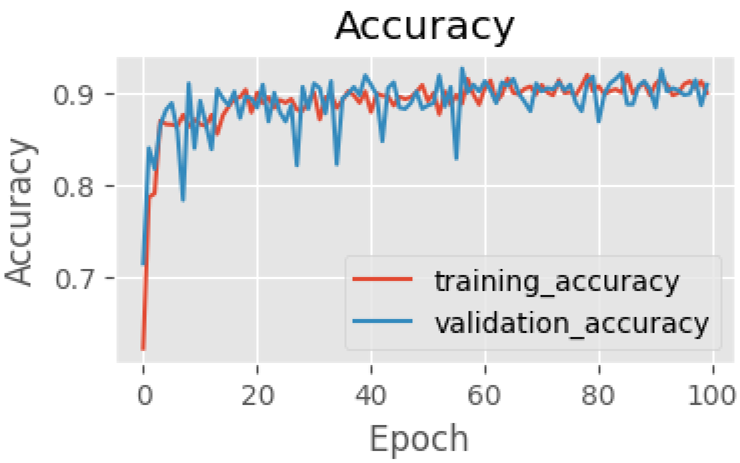
```
<ipython-input-4-794030e3dcad>:1: UserWarning: `Mo
  history = model.fit_generator(
Epoch 1/100
62/62 [==============================] - 1026s 16s
Epoch 2/100
62/62 [==============================] - 696s 11s/
Epoch 3/100
62/62 [==============================] - 494s 8s/s
Epoch 4/100
62/62 [==============================] - 371s 6s/s
Epoch 5/100
62/62 [==============================] - 279s 5s/s
Epoch 6/100
62/62 [==============================] - 215s 3s/s
Epoch 7/100
62/62 [==============================] - 161s 3s/s
Epoch 8/100
62/62 [==============================] - 102s 2s/s
Epoch 9/100
62/62 [==============================] - 87s 1s/st
Epoch 10/100
62/62 [==============================] - 60s 965ms
Epoch 11/100
62/62 [==============================] - 52s 839ms
Epoch 12/100
62/62 [==============================] - 44s 720ms
Epoch 13/100
62/62 [==============================] - 42s 683ms
Epoch 14/100
62/62 [==============================] - 29s 472ms
Epoch 15/100
62/62 [==============================] - 29s 467ms
Epoch 16/100
62/62 [==============================] - 35s 570ms
Epoch 17/100
62/62 [==============================] - 23s 372ms
Epoch 18/100
62/62 [==============================] - 25s 402ms
Epoch 19/100
62/62 [==============================] - 21s 334ms
Epoch 20/100
62/62 [==============================] - 24s 393ms
Epoch 21/100
```

```
62/62 [==============================] - 20s 327ms
Epoch 22/100
62/62 [==============================] - 20s 325ms
Epoch 23/100
62/62 [==============================] - 20s 319ms
Epoch 24/100
62/62 [==============================] - 24s 388ms
Epoch 25/100
62/62 [==============================] - 20s 314ms
Epoch 26/100
62/62 [==============================] - 25s 395ms
Epoch 27/100
62/62 [==============================] - 24s 389ms
Epoch 28/100
```

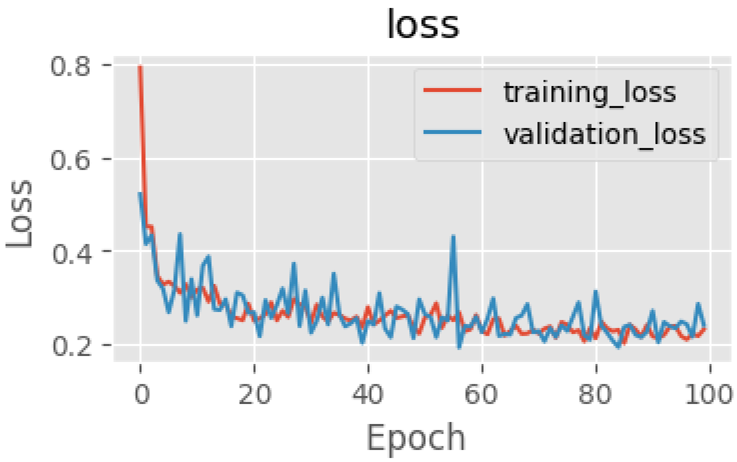
```
model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer_
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["accuracy"], label="training
plt.plot(N, history.history["val_accuracy"], label="vali
plt.title("Accuracy")
plt.xlabel("Epoch ")
plt.ylabel("Accuracy")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
N = np.arange(0, 100) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["loss"], label="training_los
plt.plot(N, history.history["val_loss"], label="validati
plt.title("loss")
plt.xlabel("Epoch ")
plt.ylabel("Loss")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
import cv2
test_data_dir='/content/drive/MyDrive/Melanoma_Skin_Canc
test_generator = train_datagen.flow_from_directory(test_
                                                   targe
                                                   batch
```

```
Found 3183 images belonging to 2 classes.
```

```
import cv2
def mode(my_list):
    ct = Counter(my_list)
    max_value = max(ct.values())
    return ([key for key, value in ct.items() if value =

true_value = []
model_one_pred = []
for folder in os.listdir(test_data_dir):

    test_image_ids = os.listdir(os.path.join(test_data_d

    for image_id in test_image_ids[:int(len(test_image_i

        path = os.path.join(test_data_dir,folder,image_i

        true_value.append(test_generator.class_indices[f
        img = cv2.resize(cv2.imread(path),(224,224))
        img_normalized = img/255

        model_one_prediction = np.argmax(model.predict(n
        model_one_pred.append(model_one_prediction)
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import itertools
#from mlxtend.plotting import plot_confusion_matrix
def clf_report(true_value, model_pred):

    classes = test_generator.class_indices.keys()
    TP_count = [true_value[i] == model_pred[i] for i in
    model_accuracy = np.sum(TP_count)/len(TP_count)
    print('Model Accuracy', model_accuracy)

    plt.figure(figsize=(3,3))
    cm = confusion_matrix(true_value,model_pred)
    plt.imshow(cm,interpolation='nearest',cmap=plt.cm.vi
    plt.title('Confusion Matrix')
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=15) #15 ref
    plt.yticks(tick_marks, classes)
    thresh = cm.max()*0.8
    for i,j in itertools.product(range(cm.shape[0]),rang
        plt.text(j,i,cm[i,j],
                horizontalalignment="center",
                color="black" if cm[i,j] > thresh else "
        pass

    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    pass

    print(classification_report(true_value, model_pred,


clf_report(true_value, model_one_pred)


#********************************************************


import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, roc_curve,
from tensorflow.keras.applications import VGG16, ResNet5
from tensorflow.keras.layers import Concatenate, Dense,
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDa
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import VGG16, ResNet5
```

```
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Melanoma_Skin_Cancer_disease
    target_size=(224, 224),
    batch_size=32,
    class_mode='sparse'
)

validation_generator = test_datagen.flow_from_directory(
        '/content/drive/MyDrive/Melanoma_Skin_Cancer_dis
        target_size=(224, 224),
        batch_size=32,
        class_mode='sparse')
```

```
Found 7423 images belonging to 2 classes.
Found 3183 images belonging to 2 classes.
```

```
input_shape = (224, 224, 3)
vgg_model = VGG16(weights='imagenet', include_top=False,
resnet_model = ResNet50(weights='imagenet', include_top=


for layer in vgg_model.layers:
    layer.trainable = False
for layer in resnet_model.layers:
    layer.trainable = False
input_tensor = Input(shape=input_shape)
vgg_output = vgg_model(input_tensor)
resnet_output = resnet_model(input_tensor)
merged = Concatenate()([GlobalAveragePooling2D()(vgg_out
x = Dense(256, activation='relu')(merged)
#x = Dense(1024, activation='relu')(x)
#x = Dense(1024, activation='relu')(x)
output = Dense(2, activation='softmax')(x)
model = Model(inputs=input_tensor, outputs=output)
model.compile(optimizer=Adam(lr=0.001), loss='sparse_cat
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer,
```

```
history = model.fit_generator(
    train_generator,
    steps_per_epoch=2000 // 32,
    epochs=50,
    validation_data=validation_generator,
    validation_steps=800 // 32
)
```

```
<ipython-input-10-3cb626d73511>:1: UserWarning: `M
  history = model.fit_generator(
Epoch 1/50
62/62 [==============================] - 26s 339ms
Epoch 2/50
62/62 [==============================] - 24s 390ms
Epoch 3/50
62/62 [==============================] - 25s 395ms
Epoch 4/50
62/62 [==============================] - 25s 399ms
Epoch 5/50
62/62 [==============================] - 20s 322ms
Epoch 6/50
62/62 [==============================] - 20s 326ms
Epoch 7/50
62/62 [==============================] - 27s 431ms
Epoch 8/50
62/62 [==============================] - 24s 393ms
Epoch 9/50
62/62 [==============================] - 20s 328ms
Epoch 10/50
62/62 [==============================] - 25s 398ms
Epoch 11/50
62/62 [==============================] - 20s 324ms
Epoch 12/50
62/62 [==============================] - 20s 323ms
Epoch 13/50
62/62 [==============================] - 20s 322ms
Epoch 14/50
62/62 [==============================] - 20s 320ms
Epoch 15/50
62/62 [==============================] - 24s 395ms
Epoch 16/50
62/62 [==============================] - 20s 318ms
Epoch 17/50
62/62 [==============================] - 20s 328ms
Epoch 18/50
62/62 [==============================] - 25s 398ms
Epoch 19/50
62/62 [==============================] - 20s 321ms
Epoch 20/50
62/62 [==============================] - 25s 397ms
Epoch 21/50
62/62 [==============================] - 20s 322ms
Epoch 22/50
62/62 [==============================] - 25s 396ms
Epoch 23/50
62/62 [==============================] - 20s 320ms
Epoch 24/50
62/62 [==============================] - 20s 323ms
Epoch 25/50
62/62 [==============================] - 25s 396ms
Epoch 26/50
62/62 [==============================] - 24s 391ms
Epoch 27/50
62/62 [==============================] - 25s 398ms
```

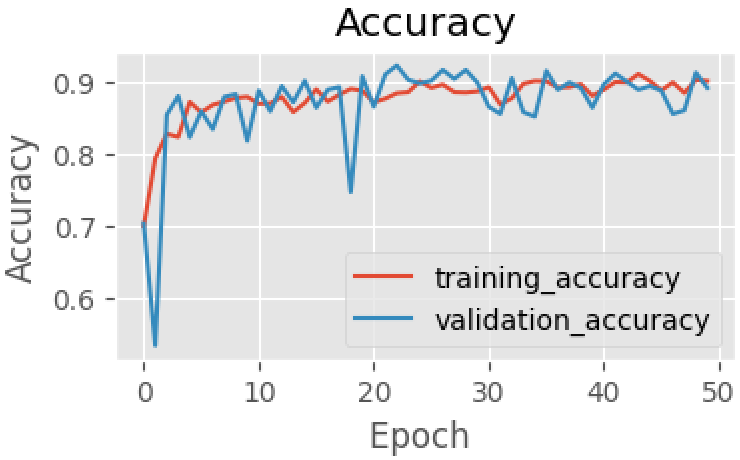
```
    Epoch 28/50
```

```
#model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer
```

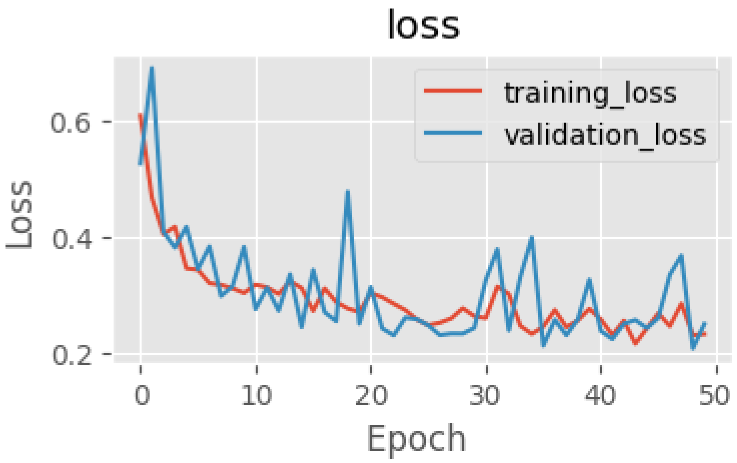
```
model.save('/content/drive/MyDrive/Melanoma_Skin_Cancer_
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/en
  saving_api.save_model(
```

```
N = np.arange(0, 50) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["accuracy"], label="training
plt.plot(N, history.history["val_accuracy"], label="vali
plt.title("Accuracy")
plt.xlabel("Epoch ")
plt.ylabel("Accuracy")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
N = np.arange(0, 50) #10 = epoch
plt.style.use("ggplot")
plt.figure(figsize=(4, 2)) #10=eqals width and 2 equals
plt.plot(N, history.history["loss"], label="training_los
plt.plot(N, history.history["val_loss"], label="validati
plt.title("loss")
plt.xlabel("Epoch ")
plt.ylabel("Loss")
plt.legend()
plt.savefig("/content/drive/MyDrive/Melanoma_Skin_Cancer
```



```
import cv2
test_data_dir='/content/drive/MyDrive/Melanoma_Skin_Canc
test_generator = train_datagen.flow_from_directory(test_
                                                   targe
                                                   batch
```

```
Found 3183 images belonging to 2 classes.
```

```
import cv2
def mode(my_list):
    ct = Counter(my_list)
    max_value = max(ct.values())
    return ([key for key, value in ct.items() if value =

true_value = []
model_one_pred = []
for folder in os.listdir(test_data_dir):

    test_image_ids = os.listdir(os.path.join(test_data_d

    for image_id in test_image_ids[:int(len(test_image_i

        path = os.path.join(test_data_dir,folder,image_i

        true_value.append(test_generator.class_indices[f
        img = cv2.resize(cv2.imread(path),(224,224))
        img_normalized = img/255

        model_one_prediction = np.argmax(model.predict(n
        model_one_pred.append(model_one_prediction)
```

```
1/1 [==============================] - 118s 118s/s
1/1 [==============================] - 0s 33ms/ste
1/1 [==============================] - 0s 25ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 39ms/ste
```

```
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 34ms/ste
1/1 [==============================] - 0s 38ms/ste
1/1 [==============================] - 0s 35ms/ste
1/1 [==============================] - 0s 47ms/ste
1/1 [==============================] - 0s 35ms/ste
1/1 [==============================] - 0s 48ms/ste
1/1 [==============================] - 0s 64ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 27ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 36ms/ste
1/1 [==============================] - 0s 42ms/ste
1/1 [==============================] - 0s 43ms/ste
1/1 [==============================] - 0s 33ms/ste
1/1 [==============================] - 0s 39ms/ste
1/1 [==============================] - 0s 32ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 36ms/ste
1/1 [==============================] - 0s 34ms/ste
1/1 [==============================] - 0s 34ms/ste
1/1 [==============================] - 0s 35ms/ste
1/1 [==============================] - 0s 33ms/ste
1/1 [==============================] - 0s 32ms/ste
1/1 [==============================] - 0s 40ms/ste
1/1 [==============================] - 0s 36ms/ste
1/1 [==============================] - 0s 38ms/ste
1/1 [==============================] - 0s 43ms/ste
1/1 [==============================] - 0s 37ms/ste
1/1 [==============================] - 0s 51ms/ste
1/1 [==============================] - 0s 43ms/ste
1/1 [==============================] - 0s 36ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 24ms/ste
1/1 [==============================] - 0s 23ms/ste
1/1 [==============================] - 0s 22ms/ste
1/1 [==============================] - 0s 23ms/ste
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import itertools
#from mlxtend.plotting import plot_confusion_matrix
def clf_report(true_value, model_pred):

    classes = test_generator.class_indices.keys()
    TP_count = [true_value[i] == model_pred[i] for i in
    model_accuracy = np.sum(TP_count)/len(TP_count)
    print('Model Accuracy', model_accuracy)

    plt.figure(figsize=(3,3))
    cm = confusion_matrix(true_value,model_pred)
    plt.imshow(cm,interpolation='nearest',cmap=plt.cm.vi
    plt.title('Confusion Matrix')
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=15) #15 ref
    plt.yticks(tick_marks, classes)
    thresh = cm.max()*0.8
    for i,j in itertools.product(range(cm.shape[0]),rang
        plt.text(j,i,cm[i,j],
                horizontalalignment="center",
                color="black" if cm[i,j] > thresh else "
        pass

    plt.ylabel('True Label')
    plt.xlabel('Predicted Label')
    pass

    print(classification_report(true_value, model_pred,


clf_report(true_value, model_one_pred)
```

```
Model Accuracy 0.8595664467483506
              precision    recall  f1-score   suppor

      benign       0.95      0.77      0.85      165
   malignant       0.79      0.96      0.87      153

    accuracy                           0.86      318
   macro avg       0.87      0.86      0.86      318
weighted avg       0.88      0.86      0.86      318
```

```
model.summary()
```

```
Model: "model_6"
____________________________________________________
 Layer (type)                Output Shape
====================================================
 input_7 (InputLayer)        [(None, 256, 256, 3)]

 conv2d_66 (Conv2D)          (None, 256, 256, 32)

 conv2d_67 (Conv2D)          (None, 256, 256, 32)

 max_pooling2d_12 (MaxPooli  (None, 128, 128, 32)
 ng2D)

 conv2d_68 (Conv2D)          (None, 128, 128, 64)

 conv2d_69 (Conv2D)          (None, 128, 128, 64)

 max_pooling2d_13 (MaxPooli  (None, 64, 64, 64)
 ng2D)

 conv2d_70 (Conv2D)          (None, 64, 64, 128)

 conv2d_71 (Conv2D)          (None, 64, 64, 128)

 conv2d_transpose_12 (Conv2  (None, 128, 128, 64)
 DTranspose)

 concatenate_12 (Concatenat  (None, 128, 128, 128)
```