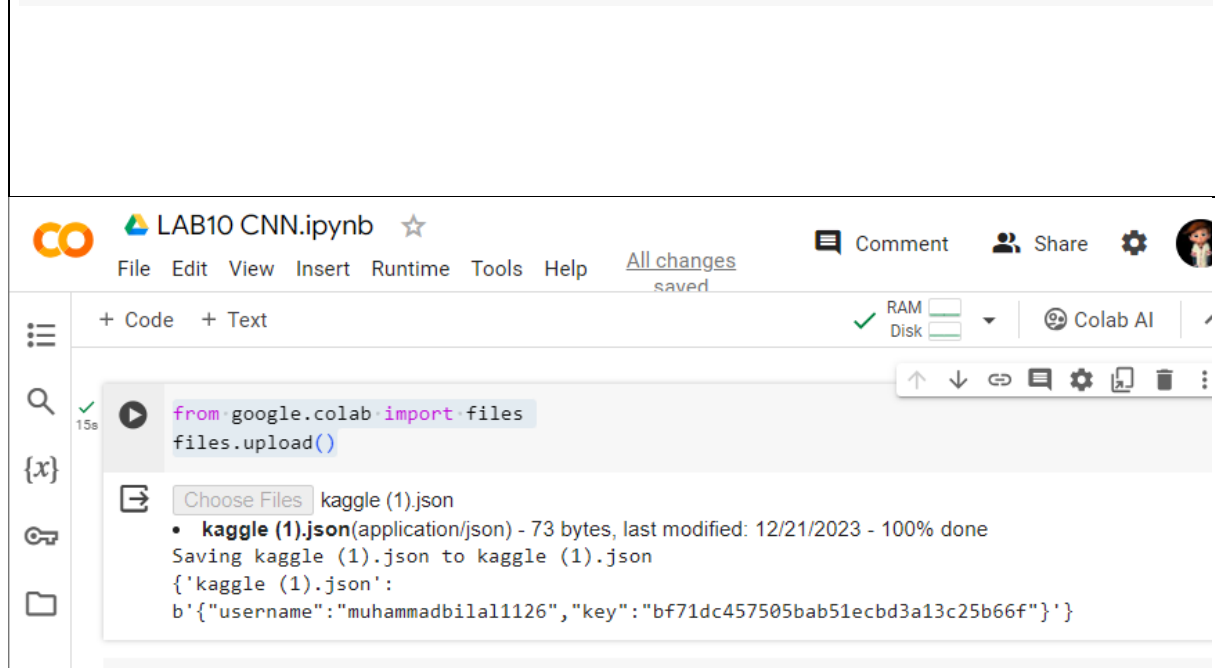


IN LAB

LAB TASK 1 :

```
from google.colab import files
files.upload()
```



LAB TASK 2:

```
!pip install -q kaggle
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets download -d moltean/fruits
```

```

b'{"username":"muhammadbilal1126","key":"15be4b1e4127168a12ab913db100d317"}'

!pip install -q kaggle
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets download -d moltean/fruits

mkdir: cannot create directory '/root/.kaggle': File exists
Downloading fruits.zip to /content
 99% 1.28G/1.28G [00:13<00:00, 53.2MB/s]
100% 1.28G/1.28G [00:13<00:00, 102MB/s]

```

LAB TASK 3:

```
!unzip fruits.zip
```

```

!unzip fruits.zip

inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_189_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_190_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_191_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_192_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_193_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_194_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_195_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_196_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_197_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_198_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_199_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_200_100.jpg
inflating: fruits-360_dataset/fruits-360/Training/Tomato Heart/r2_201_100.jpg

```

LAB TASK 4:

```

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

# Data Preparation

```

```

train_datagen = ImageDataGenerator(rescale=1./255,
validation_split=0.2) # Rescaling and splitting data

train_generator = train_datagen.flow_from_directory(
    directory='./fruits-360_dataset/fruits-360/Training',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training')

validation_generator = train_datagen.flow_from_directory(
    directory='./fruits-360_dataset/fruits-360/Test',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation')

```



```

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Data Preparation
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2) # Res

train_generator = train_datagen.flow_from_directory(
    directory='./fruits-360_dataset/fruits-360/Training',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training')

validation_generator = train_datagen.flow_from_directory(
    directory='./fruits-360_dataset/fruits-360/Test',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation')

```

```

Found 54190 images belonging to 131 classes.
Found 4481 images belonging to 131 classes.

```

LAB TASK 5:

```

# Model Definition
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu',
input_shape=(224, 224, 3)),

```

```

tf.keras.layers.MaxPooling2D(2, 2),
# Add more layers as needed
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(512, activation='relu'),
tf.keras.layers.Dense(131, activation='softmax') # Adjusted to
131 for the correct number of classes
])

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

```

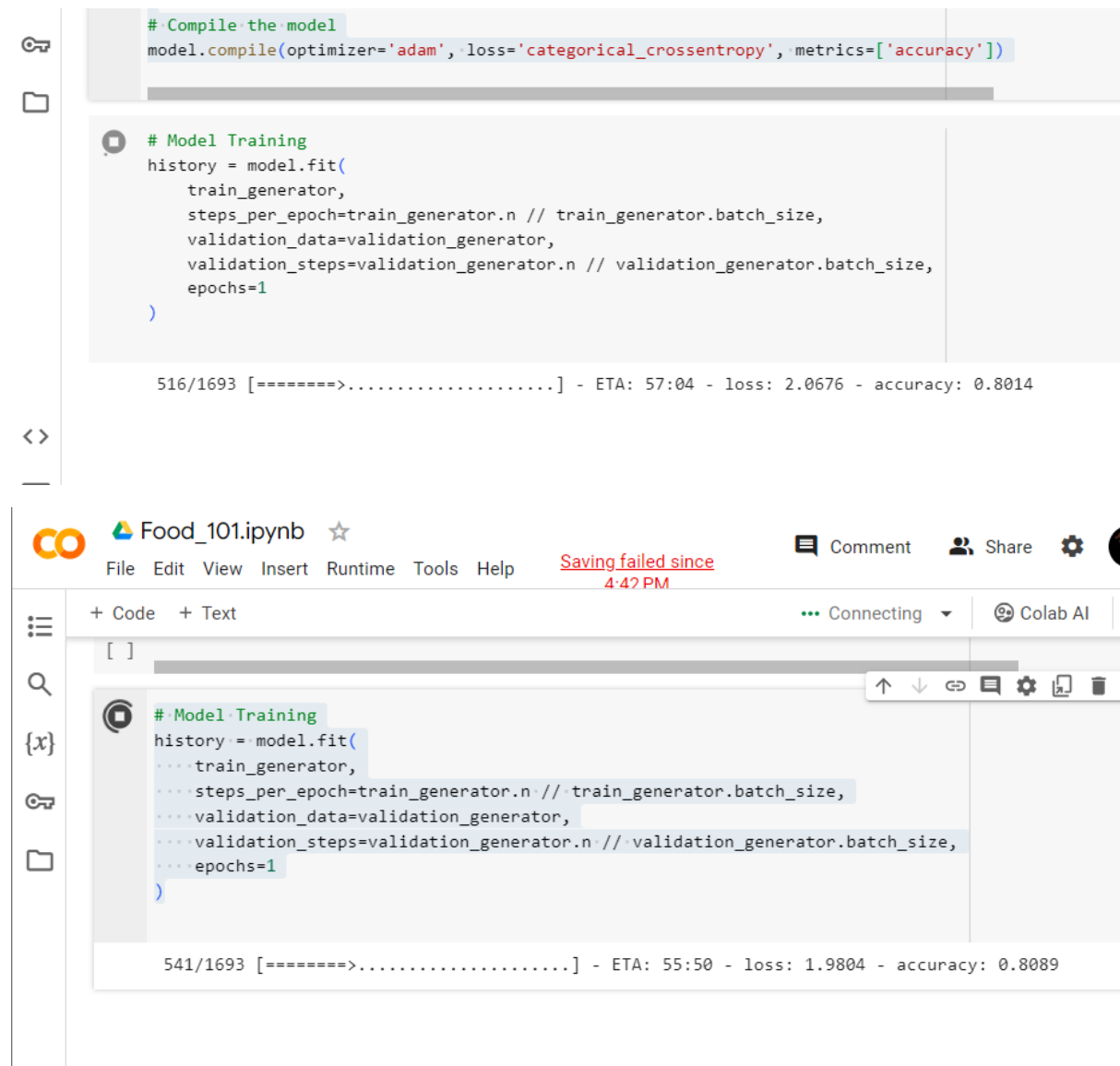


LAB TASK 6

```

# Model Training
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.n //
validation_generator.batch_size,
    epochs=1
)

```



The screenshot displays a Jupyter Notebook interface with a sidebar on the left containing icons for a key, a folder, and a double arrow. The main area shows a code cell with the following Python code:

```
# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Model Training
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.n // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.n // validation_generator.batch_size,
    epochs=1
)
```

Below the code, the output shows the progress of the training process:

```
516/1693 [=====>.....] - ETA: 57:04 - loss: 2.0676 - accuracy: 0.8014
```

The interface includes a top bar with the Colab logo, the filename "Food_101.ipynb", and a star icon. Below this is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A red status message "Saving failed since 4:42 PM" is visible. On the right side of the top bar are icons for "Comment", "Share", and "Colab AI". The main editor area has a toolbar with icons for undo, redo, copy, paste, and other editing functions. The code cell is currently selected, and the output is displayed below it.

541/1693 [=====>.....] - ETA: 55:50 - loss: 1.9804 - accuracy: 0.8089