NNDL: ICP5

Jyothi Kiran Boddeda
700769023

GITHUB LINK : https://github.com/jyothikiranboddeda/Neural-Network-Deep-Learning.git

VIDEO LINK :

https://drive.google.com/file/d/1s_pxOAKn0qqNZCUdmbp-3NHuZ3iB4zCj/view?usp=drive_link

1.Follow the instruction below and then report how the performance changed.(apply all at once)

• Convolutional input layer, 32 feature maps with a size of 3×3 and a rectifier activation function.

• Dropout layer at 20%.

• Convolutional layer, 32 feature maps with a size of 3×3 and a rectifier activation function.

• Max Pool layer with size 2×2.

• Convolutional layer, 64 feature maps with a size of 3×3 and a rectifier activation function.

• Dropout layer at 20%.

• Convolutional layer, 64 feature maps with a size of 3×3 and a rectifier activation function.

• Max Pool layer with size 2×2.

• Convolutional layer, 128 feature maps with a size of 3×3 and a rectifier activation function.

• Dropout layer at 20%.

• Convolutional layer,128 feature maps with a size of 3×3 and a rectifier activation function.

• Max Pool layer with size 2×2.

• Flatten layer.

• Dropout layer at 20%.

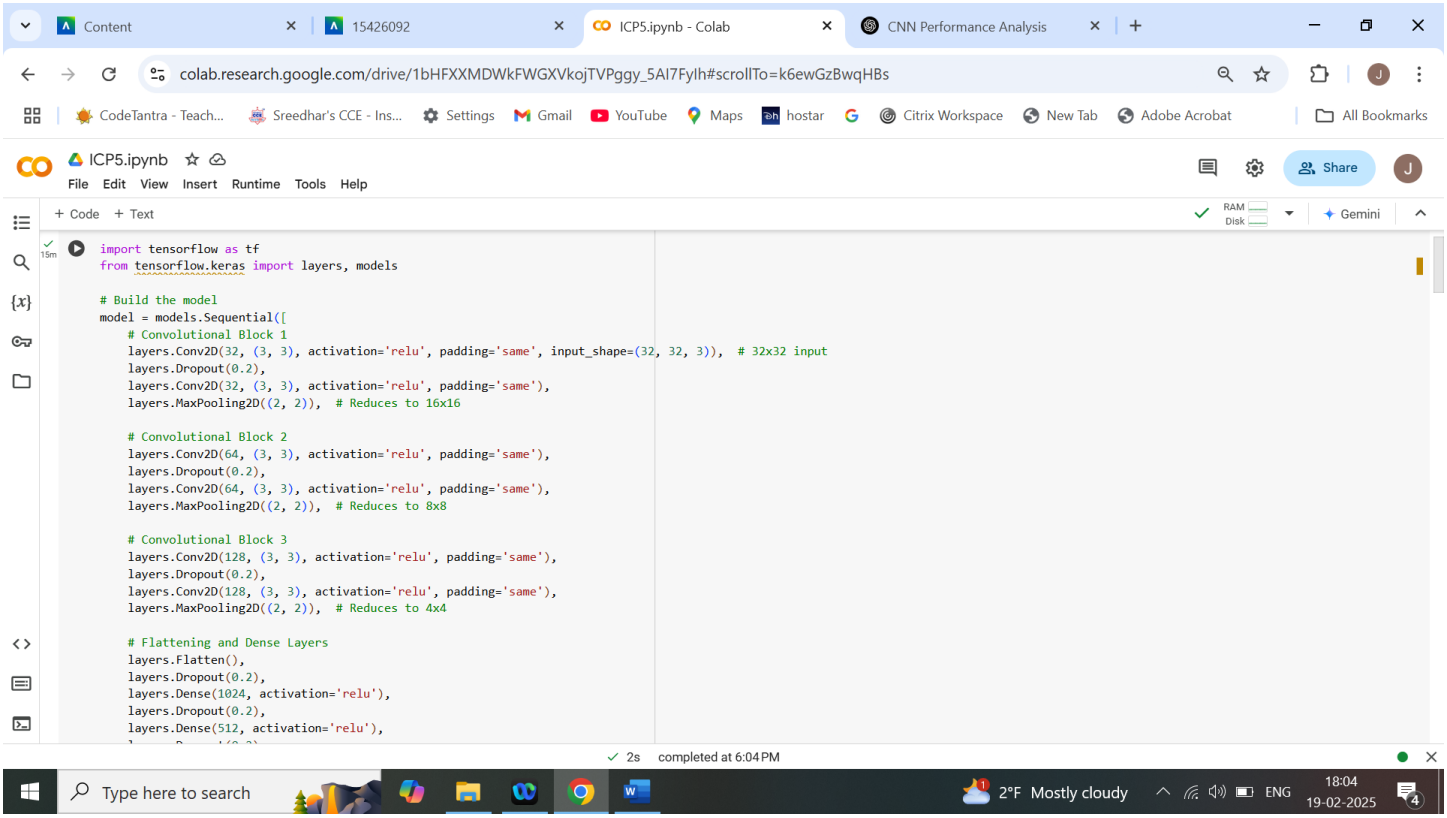
• Fully connected layer with 1024 units and a rectifier activation function.

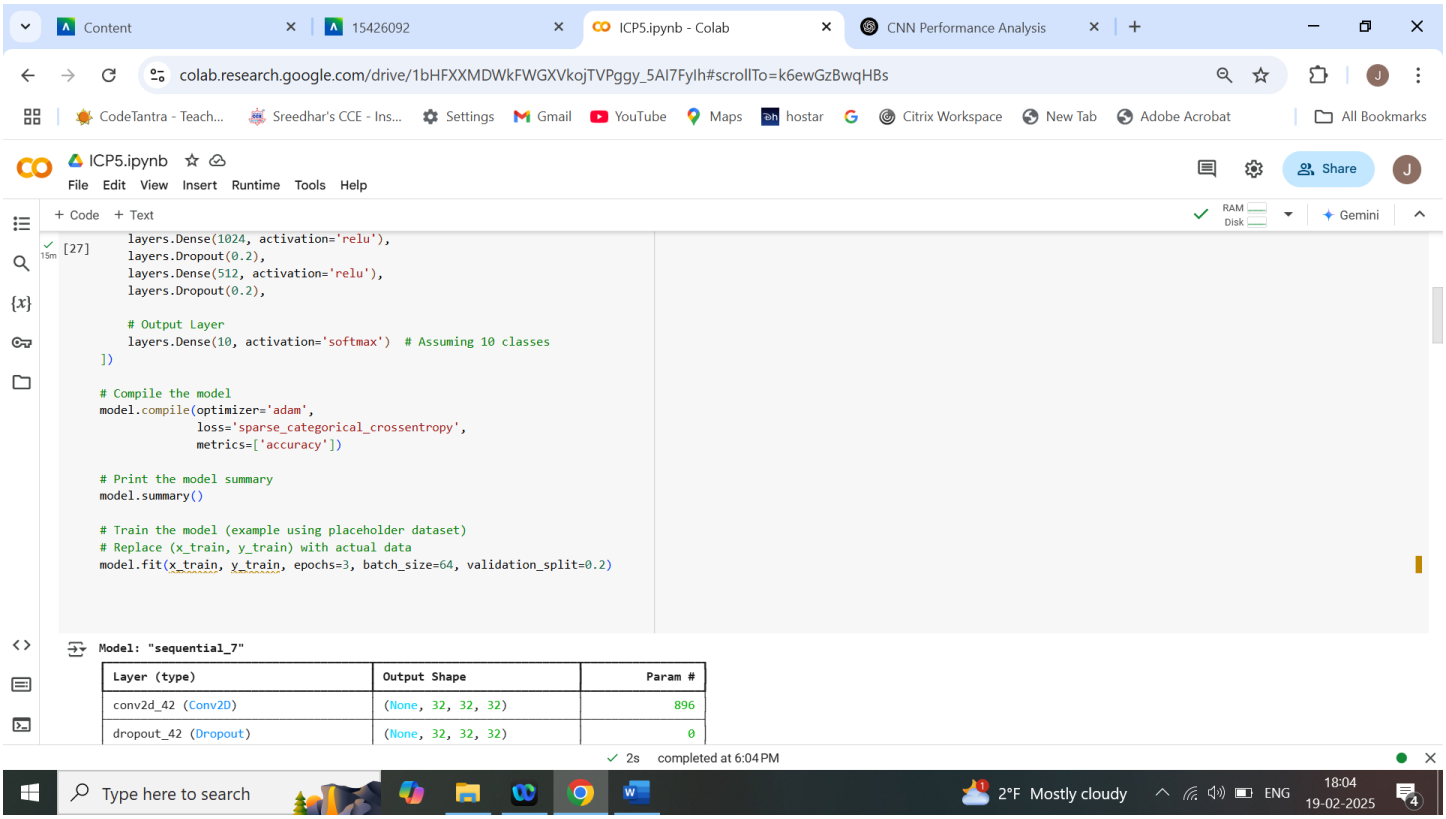
• Dropout layer at 20%.

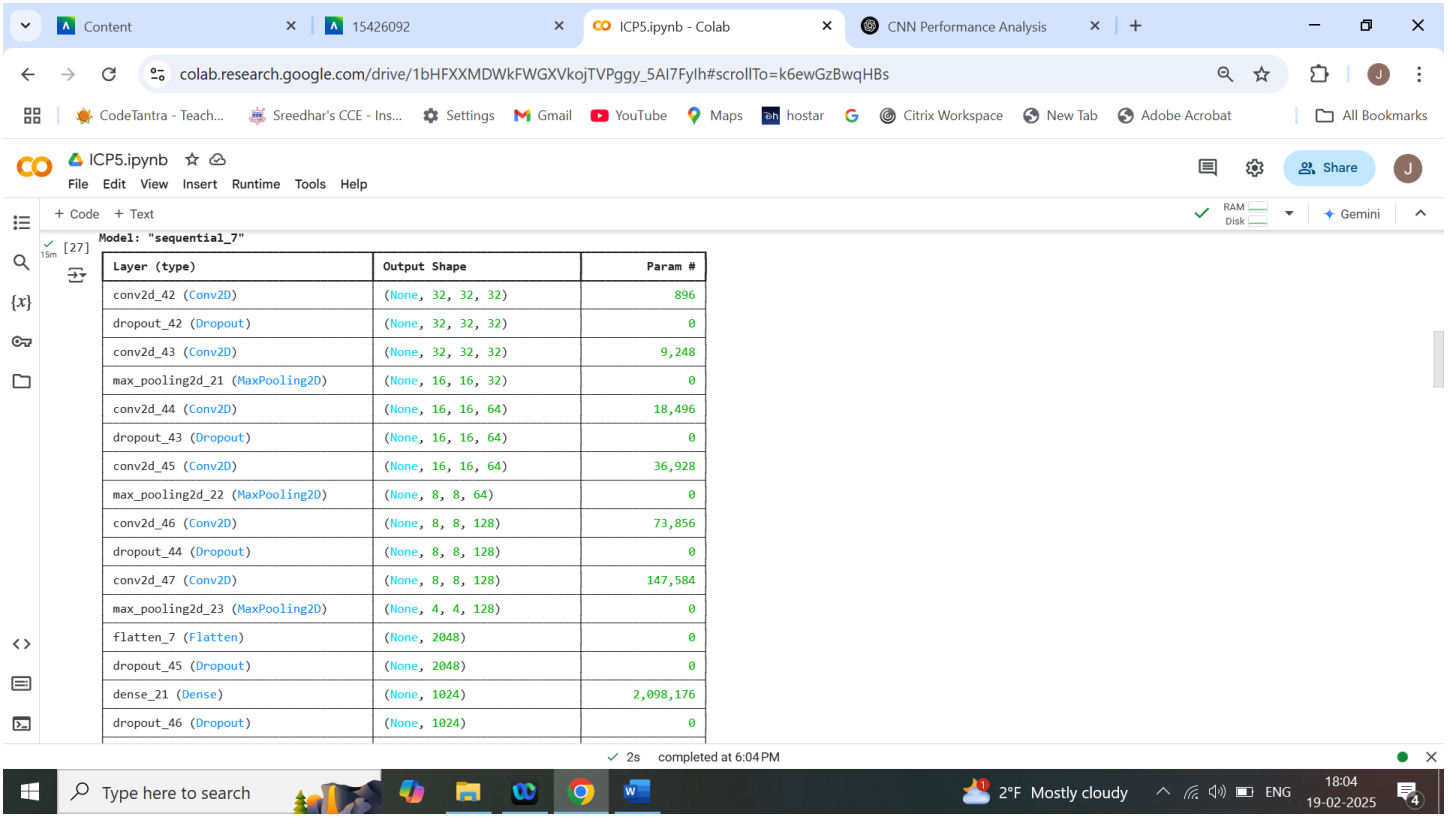
• Fully connected layer with 512 units and a rectifier activation function.

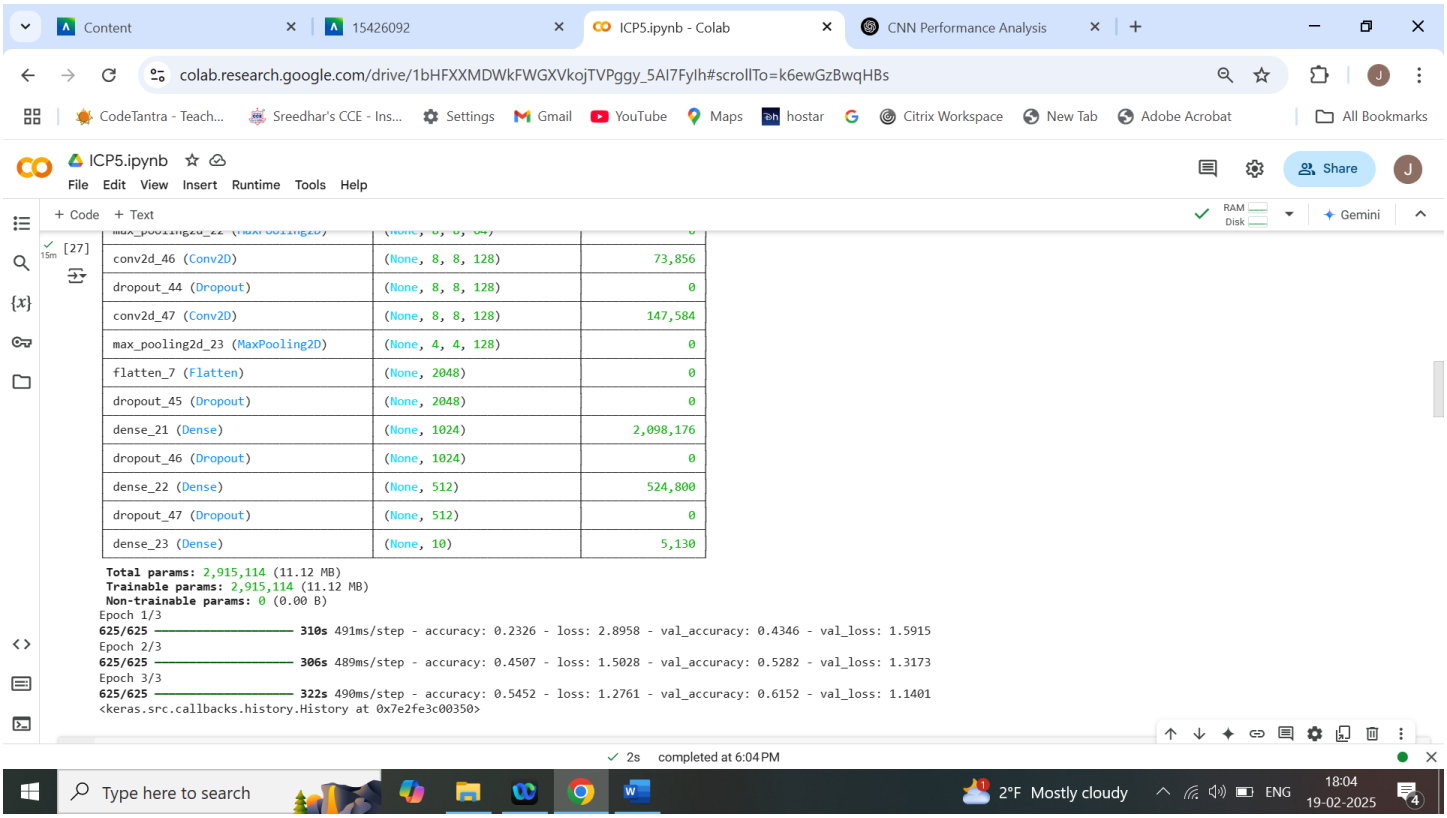
• Dropout layer at 20%.

• Fully connected output layer with 10 units and a Softmax activation function

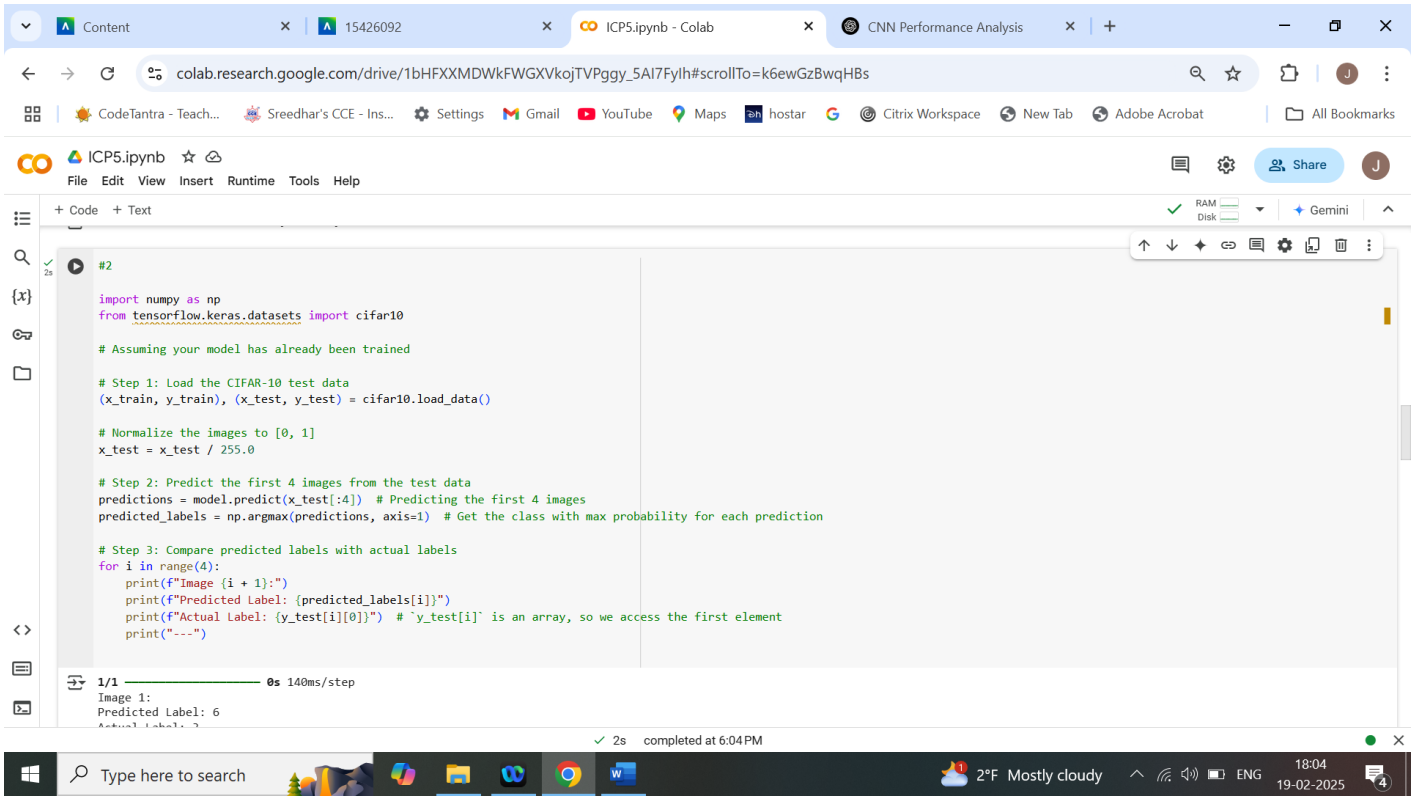

Content
15426092
ICP5.ipynb - Colab
CNN Performance Analysis
colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=k6ewGzBwqHBs
CodeTantra - Teach...
Sreedhar's CCE - Ins...
Settings
Gmail
YouTube
Maps
hostar
Citrix Workspace
New Tab
Adobe Acrobat
All Bookmarks
ICP5.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
import tensorflow as tf
from tensorflow.keras import layers, models
# Build the model
model = models.Sequential([
# Convolutional Block 1
layers.Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(32, 32, 3)), # 32x32 input
layers.Dropout(0.2),
layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
layers.MaxPooling2D((2, 2)), # Reduces to 16x16
Convolutional Block 2
layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
layers.Dropout(0.2),
layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
layers.MaxPooling2D((2, 2)), # Reduces to 8x8
Convolutional Block 3
layers.Conv2D(128, (3, 3), activation='relu', padding='same'),
layers.Dropout(0.2),
layers.Conv2D(128, (3, 3), activation='relu', padding='same'),
layers.MaxPooling2D((2, 2)), # Reduces to 4x4
Flattening and Dense Layers
layers.Flatten(),
layers.Dropout(0.2),
layers.Dense(1024, activation='relu'),
layers.Dropout(0.2),
layers.Dense(512, activation='relu'),
2s completed at 6:04 PM
Type here to search
2°F Mostly cloudy
ENG
18:04
19-02-2025

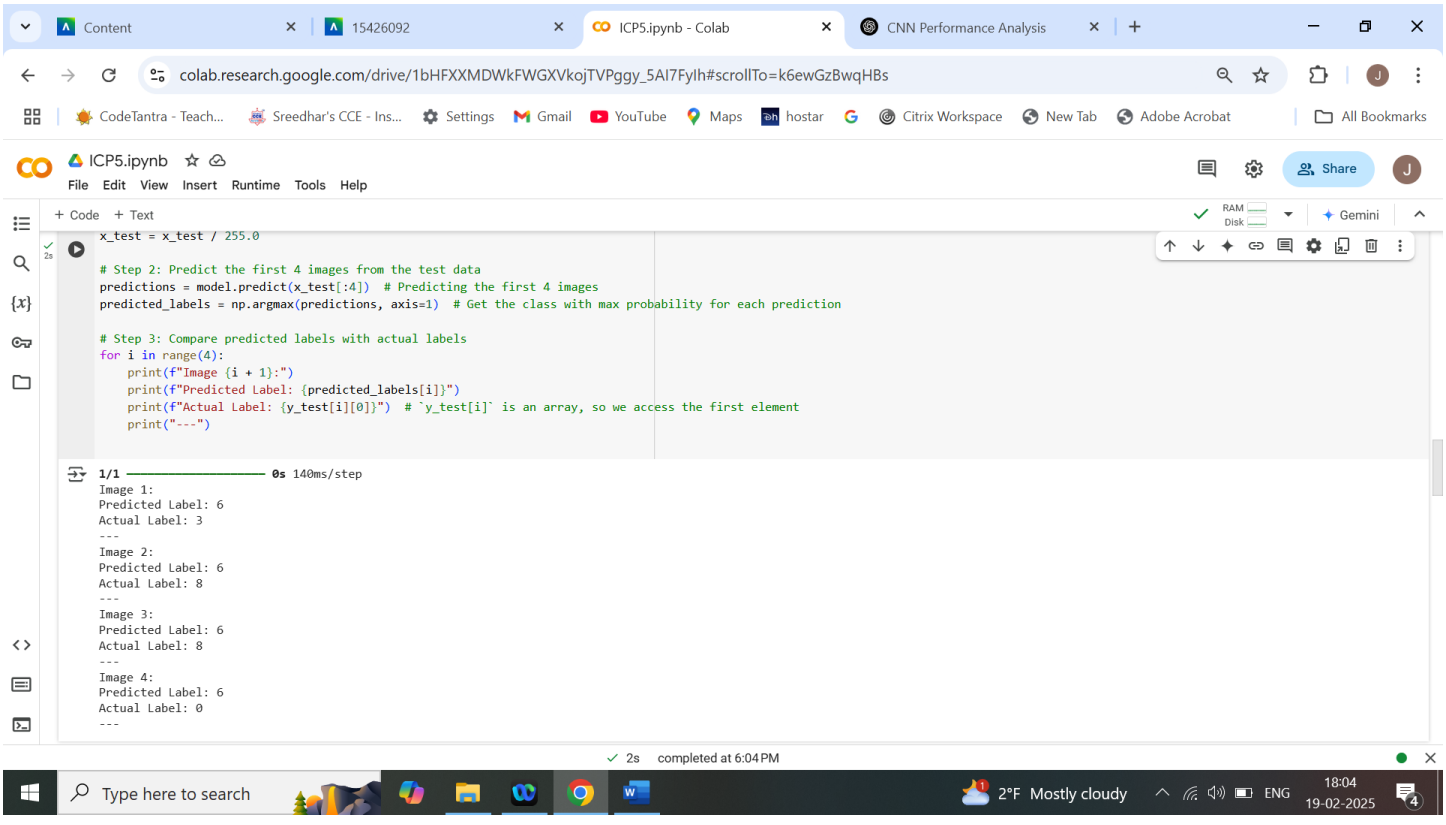

Content
15426092
ICP5.ipynb - Colab
CNN Performance Analysis
colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=k6ewGzBwqHBs
CodeTantra - Teach...
Sreedhar's CCE - Ins...
Settings
Gmail
YouTube
Maps
hostar
Citrix Workspace
New Tab
Adobe Acrobat
All Bookmarks
ICP5.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
[27]
layers.Dense(1024, activation='relu'),
layers.Dropout(0.2),
layers.Dense(512, activation='relu'),
layers.Dropout(0.2),
Output Layer
layers.Dense(10, activation='softmax') # Assuming 10 classes
])
Compile the model
model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
# Print the model summary
model.summary()
# Train the model (example using placeholder dataset)
# Replace (x_train, y_train) with actual data
model.fit(x_train, y_train, epochs=3, batch_size=64, validation_split=0.2)
Model: "sequential_7"
Layer (type)
Output Shape
Param #
conv2d_42 (Conv2D)
(None, 32, 32, 32)
896
dropout_42 (Dropout)
(None, 32, 32, 32)
0
2s completed at 6:04 PM
Type here to search
2°F Mostly cloudy
ENG
18:04
19-02-2025


Content
15426092
ICP5.ipynb - Colab
CNN Performance Analysis
colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=k6ewGzBwqHBs
CodeTantra - Teach...
Sreedhar's CCE - Ins...
Settings
Gmail
YouTube
Maps
hostar
Citrix Workspace
New Tab
Adobe Acrobat
All Bookmarks
ICP5.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
[27]
Model: "sequential_7"
Layer (type) Output Shape Param #
conv2d_42 (Conv2D) (None, 32, 32, 32) 896
dropout_42 (Dropout) (None, 32, 32, 32) 0
conv2d_43 (Conv2D) (None, 32, 32, 32) 9,248
max_pooling2d_21 (MaxPooling2D) (None, 16, 16, 32) 0
conv2d_44 (Conv2D) (None, 16, 16, 64) 18,496
dropout_43 (Dropout) (None, 16, 16, 64) 0
conv2d_45 (Conv2D) (None, 16, 16, 64) 36,928
max_pooling2d_22 (MaxPooling2D) (None, 8, 8, 64) 0
conv2d_46 (Conv2D) (None, 8, 8, 128) 73,856
dropout_44 (Dropout) (None, 8, 8, 128) 0
conv2d_47 (Conv2D) (None, 8, 8, 128) 147,584
max_pooling2d_23 (MaxPooling2D) (None, 4, 4, 128) 0
flatten_7 (Flatten) (None, 2048) 0
dropout_45 (Dropout) (None, 2048) 0
dense_21 (Dense) (None, 1024) 2,098,176
dropout_46 (Dropout) (None, 1024) 0
2s completed at 6:04 PM
Type here to search
2°F Mostly cloudy
ENG
18:04
19-02-2025

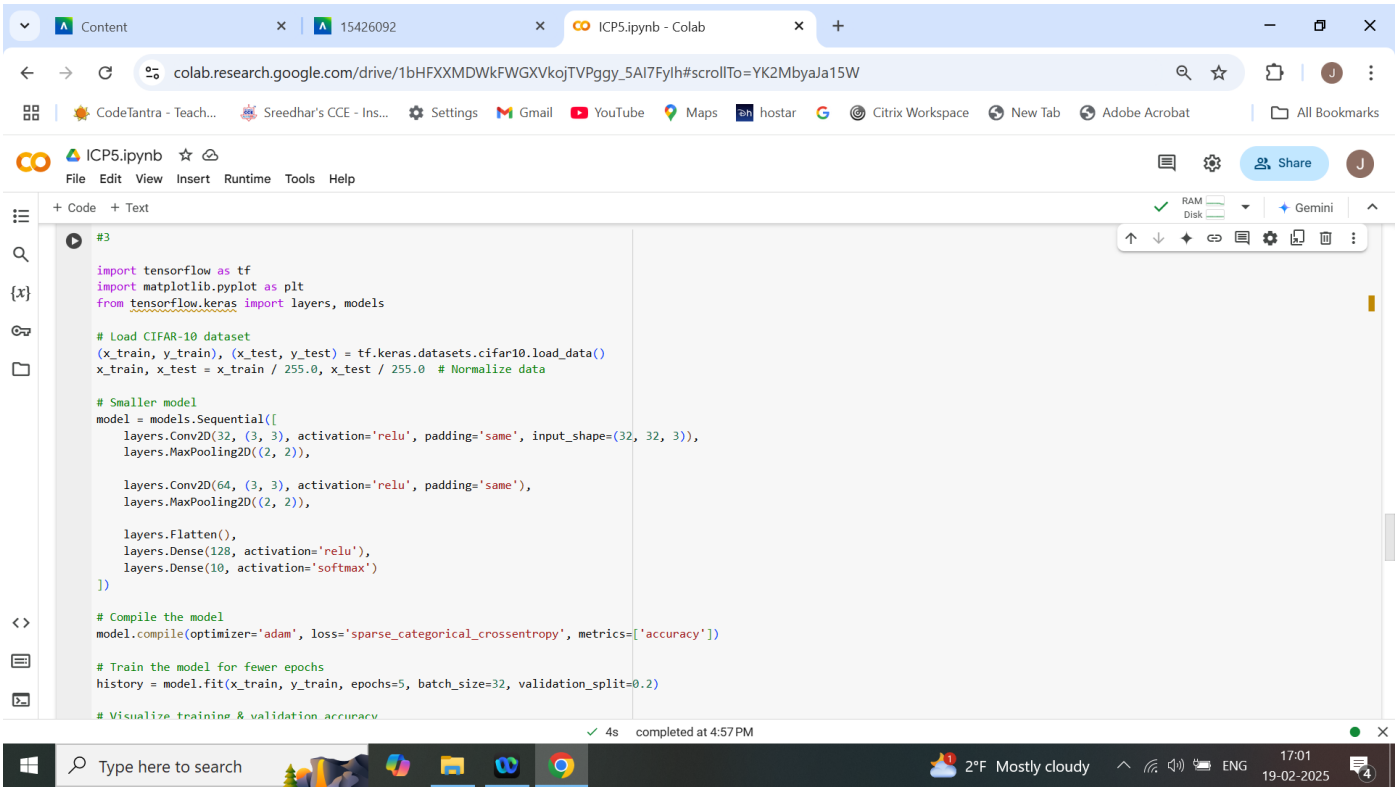

Content
15426092
ICP5.ipynb - Colab
CNN Performance Analysis
colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=k6ewGzBwqHBs
CodeTantra - Teach...
Sreedhar's CCE - Ins...
Settings
Gmail
YouTube
Maps
hostar
Citrix Workspace
New Tab
Adobe Acrobat
All Bookmarks
ICP5.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
RAM
Disk
Gemini
[27]
conv2d_46 (Conv2D) (None, 8, 8, 128) 73,856
dropout_44 (Dropout) (None, 8, 8, 128) 0
conv2d_47 (Conv2D) (None, 8, 8, 128) 147,584
max_pooling2d_23 (MaxPooling2D) (None, 4, 4, 128) 0
flatten_7 (Flatten) (None, 2048) 0
dropout_45 (Dropout) (None, 2048) 0
dense_21 (Dense) (None, 1024) 2,098,176
dropout_46 (Dropout) (None, 1024) 0
dense_22 (Dense) (None, 512) 524,800
dropout_47 (Dropout) (None, 512) 0
dense_23 (Dense) (None, 10) 5,130
Total params: 2,915,114 (11.12 MB)
Trainable params: 2,915,114 (11.12 MB)
Non-trainable params: 0 (0.00 B)
Epoch 1/3
625/625 310s 491ms/step - accuracy: 0.2326 - loss: 2.8958 - val_accuracy: 0.4346 - val_loss: 1.5915
Epoch 2/3
625/625 306s 489ms/step - accuracy: 0.4507 - loss: 1.5028 - val_accuracy: 0.5282 - val_loss: 1.3173
Epoch 3/3
625/625 322s 490ms/step - accuracy: 0.5452 - loss: 1.2761 - val_accuracy: 0.6152 - val_loss: 1.1401
<keras.src.callbacks.history.History at 0x7e2fe3c00350>
2s completed at 6:04 PM
Type here to search
2°F Mostly cloudy
ENG
18:04
19-02-2025

2. Predict the first 4 images of the test data using the above model. Then, compare with the actual label for those 4 images to check whether or not the model has predicted correctly.

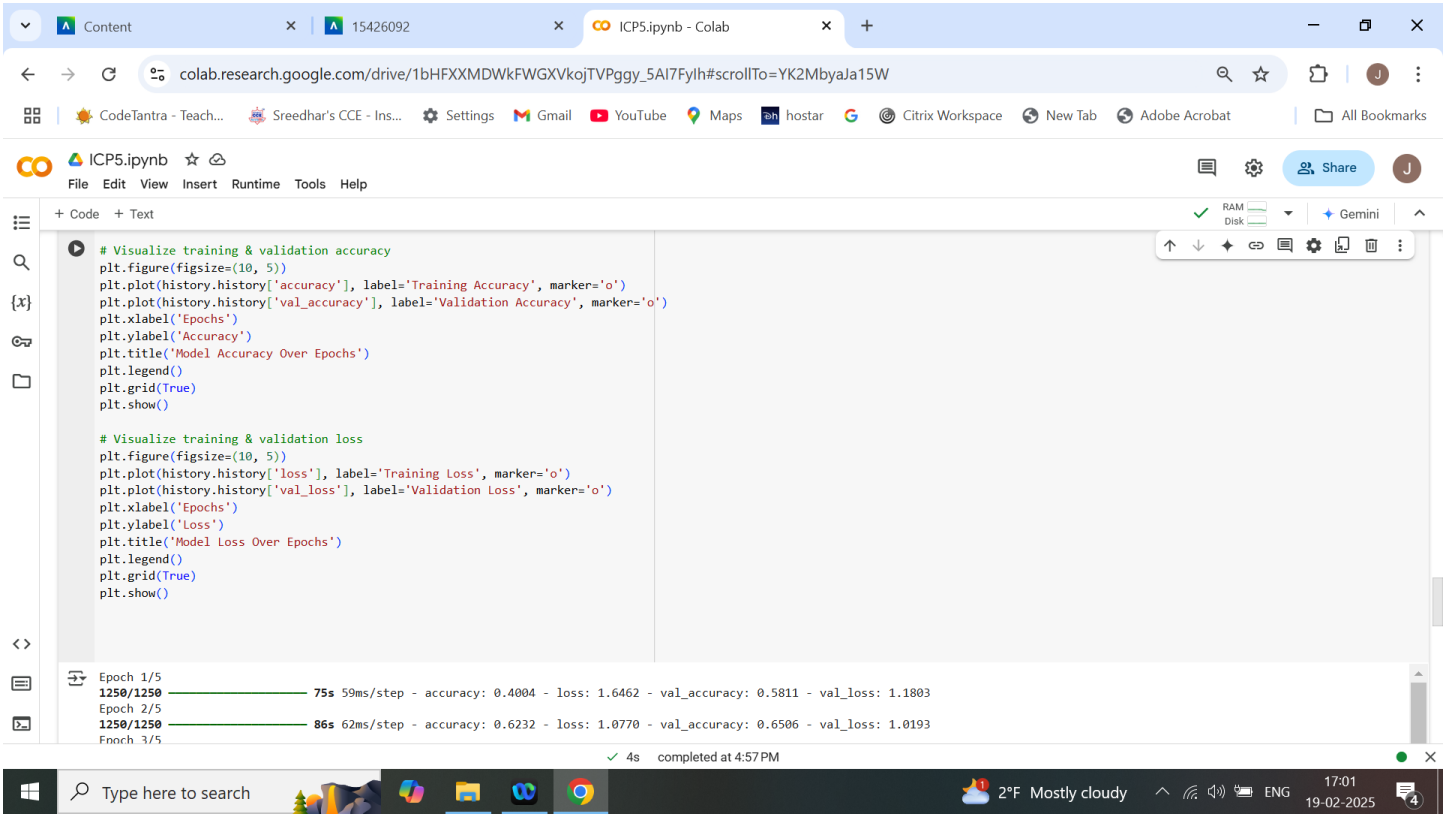




3. Visualize Loss and Accuracy using the history object\



```
#3

import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models

# Load CIFAR-10 dataset
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0  # Normalize data

# Smaller model
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
    layers.MaxPooling2D((2, 2)),

    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])

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

# Train the model for fewer epochs
history = model.fit(x_train, y_train, epochs=5, batch_size=32, validation_split=0.2)

# Visualize training & validation accuracy
```



```
# Visualize training & validation accuracy
plt.figure(figsize=(10, 5))
plt.plot(history.history['accuracy'], label='Training Accuracy', marker='o')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy', marker='o')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Model Accuracy Over Epochs')
plt.legend()
plt.grid(True)
plt.show()

# Visualize training & validation loss
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss', marker='o')
plt.plot(history.history['val_loss'], label='Validation Loss', marker='o')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Model Loss Over Epochs')
plt.legend()
plt.grid(True)
plt.show()
```

```
Epoch 1/5
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 75s 59ms/step - accuracy: 0.4004 - loss: 1.6462 - val_accuracy: 0.5811 - val_loss: 1.1803
Epoch 2/5
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 86s 62ms/step - accuracy: 0.6232 - loss: 1.0770 - val_accuracy: 0.6506 - val_loss: 1.0193
Epoch 3/5
```

Content | 15426092 | ICP5.ipynb - Colab

colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=YK2MbyaJa15W

CodeTantra - Teach... | Sreedhar's CCE - Ins... | Settings | Gmail | YouTube | Maps | hostar | Citrix Workspace | New Tab | Adobe Acrobat | All Bookmarks

ICP5.ipynb

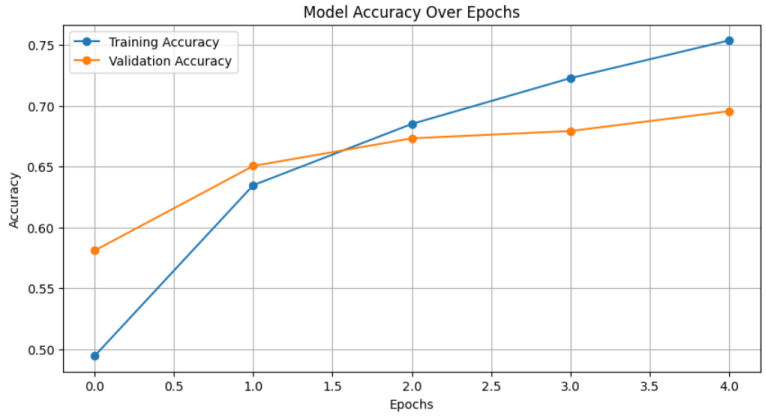
File Edit View Insert Runtime Tools Help

Share

+ Code + Text

RAM Disk | Gemini

```
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 81s 58ms/step - accuracy: 0.7254 - loss: 0.7862 - val_accuracy: 0.6793 - val_loss: 0.9364
Epoch 5/5
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 84s 60ms/step - accuracy: 0.7583 - loss: 0.6966 - val_accuracy: 0.6956 - val_loss: 0.9001
```



4s completed at 4:57 PM

Type here to search | 2°F Mostly cloudy | ENG | 17:02 19-02-2025

Content | 15426092 | ICP5.ipynb - Colab

colab.research.google.com/drive/1bHFXXMDWkFWGXVkojTVPggy_5AI7FyIh#scrollTo=YK2MbyaJa15W

CodeTantra - Teach... | Sreedhar's CCE - Ins... | Settings | Gmail | YouTube | Maps | hostar | Citrix Workspace | New Tab | Adobe Acrobat | All Bookmarks

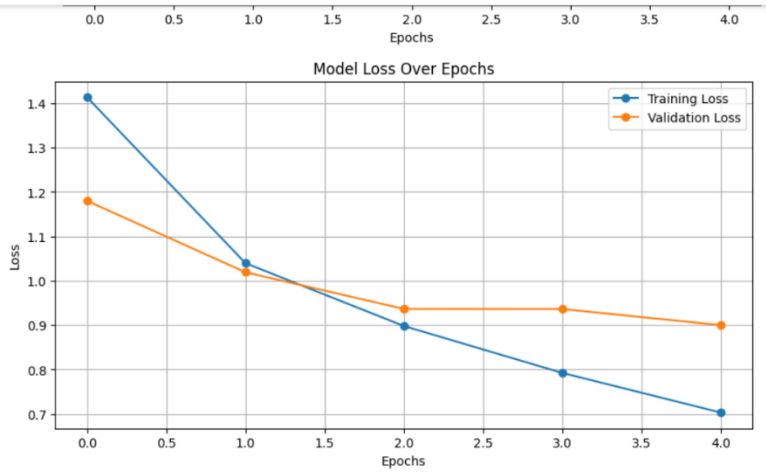
ICP5.ipynb

File Edit View Insert Runtime Tools Help

Share

+ Code + Text

RAM Disk | Gemini



4s completed at 4:57 PM

Type here to search | 2°F Mostly cloudy | ENG | 17:02 19-02-2025