EEAIPROJECTREPORT

Project Name: Implementation of Testing with Concept Activation Vectors (TCAV)

SR UNIVERSITY

A Project Report in partial fulfillment of the degree

Bachelor of Technology in Computer Science & Artificial Intelligence By

Group Name: Ping	
Roll No	**Name of the Student**
2203A52085	**G.Krishna Priya**
2203A52130	**V.Divya**
2203A52115	**P. SiddharthRao**
2303A51LB6	**K. Deepak**

SchoolofComputerScience&ArtificialIntelligence
SR University, Ananthsagar, Hasanparthy (M),Warangal, Telangana 506371, India

2025-2026

Introduction to TCAV (Testing with Concept Activation Vectors)

TCAV (Testing with Concept Activation Vectors) is a technique developed to help interpret and explain machine learning models, particularly deep learning models. The core idea behind TCAV is to understand how a model's predictions are influenced by high-level human-interpretable concepts rather than just raw features in the input data. It focuses on explaining "why" a model makes a certain prediction by associating it with concepts that are meaningful to humans, rather than relying solely on the weights and activations inside the network.

Key Concepts of TCAV:

1.**Concepts**:Concepts are high-level features that humans can easily understand (e.g., "striped patterns," "color," "shapes," "objects," "textures"). These can be defined by the user and are often extracted from the model's activations, class labels, or even external knowledge.

2. **Activation Vectors**:TCAV works by defining a concept in terms of an "activation vector" (CAV), which represents the influence of a particular concept on the neural network's predictions. It uses this vector to test how much a model is paying attention to a certain concept during its decision-making process.

3. **Testing with CAVs**:TCAV compares the model’s activations in response to the input that represents a concept. It measures the relationship between a concept's activation vector and the model's output across a set of data points. The higher the correlation between a concept and the model's decision, the stronger the model's reliance on that concept.

colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=N127Ur5guNhw

Untitled9.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

Files

Analyse your files with code written by Gemini Upload

dataset
dtd
tcav_dataset...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...

Disk 77.78 GB available

```
[1] !pip install tcav tensorflow matplotlib scikit-learn --quiet
```

```
[2] !pip install protobuf==3.20.3
```

```
Requirement already satisfied: protobuf==3.20.3 in /usr/local/lib/python3.11/dist-packages (3.20.3)
```

```
[3] import os
    os.environ["PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION"] = "python"
```

```
import os
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.applications.inception_v3 import InceptionV3, preprocess_input, decode_predictions
from tensorflow.keras.preprocessing import image
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import tcav.model as model
import tcav.utils as utils
from tcav.tcav import TCAV
```

```
[5] base_dir = '/content/drive/MyDrive/tcav_dataset_XAI'  # <- CHANGE THIS to your path
    concept_dir = os.path.join(base_dir, 'chequered')
    target_dir = os.path.join(base_dir, 'stripped')
    bottleneck_name = 'mixed10'  # layer name in InceptionV3
    target_class = 'stripped'
```

```
def load_images_from_folder(folder, target_size=(299, 299)):
    images = []
```

0s completed at 23:49

colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=N127Ur5guNhw

Untitled9.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

Files

Analyse your files with code written by Gemini Upload

dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...

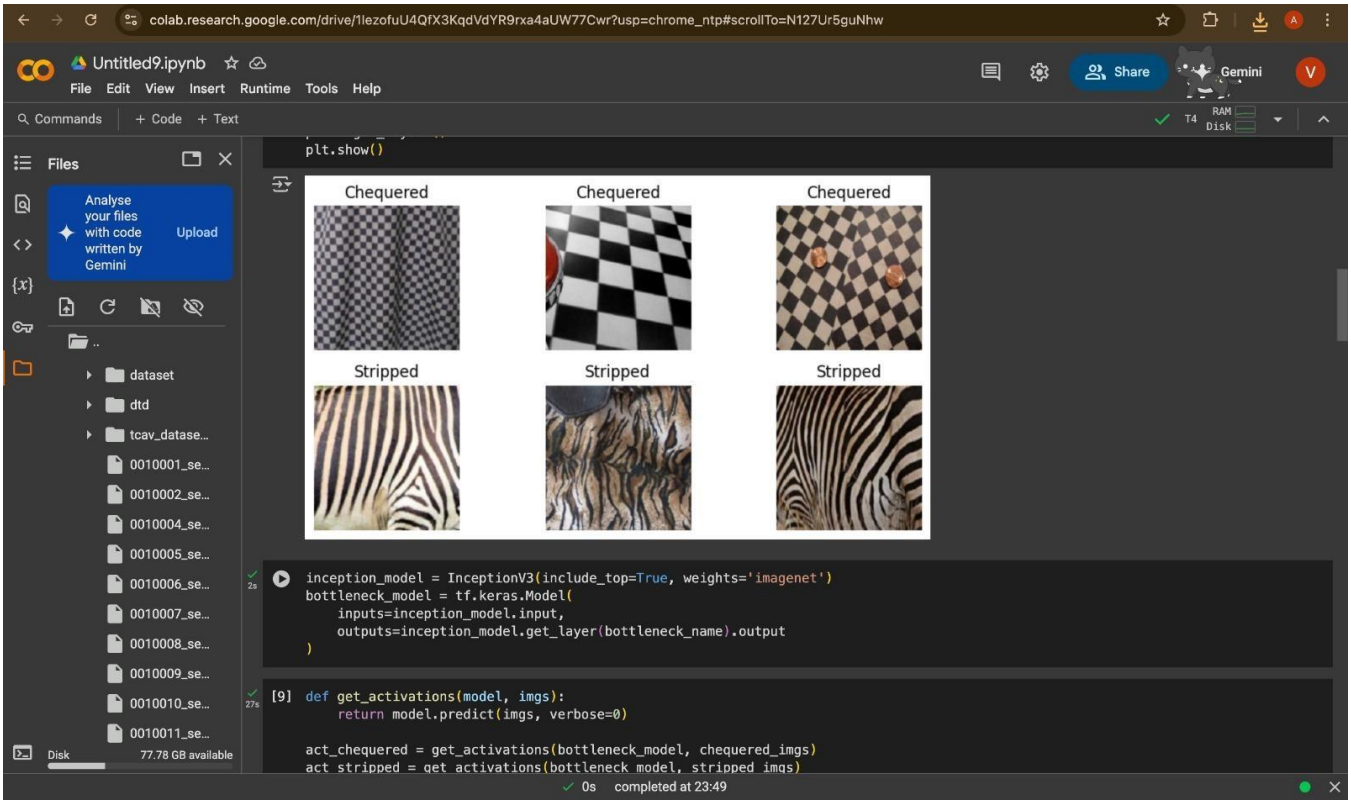
Disk 77.78 GB available

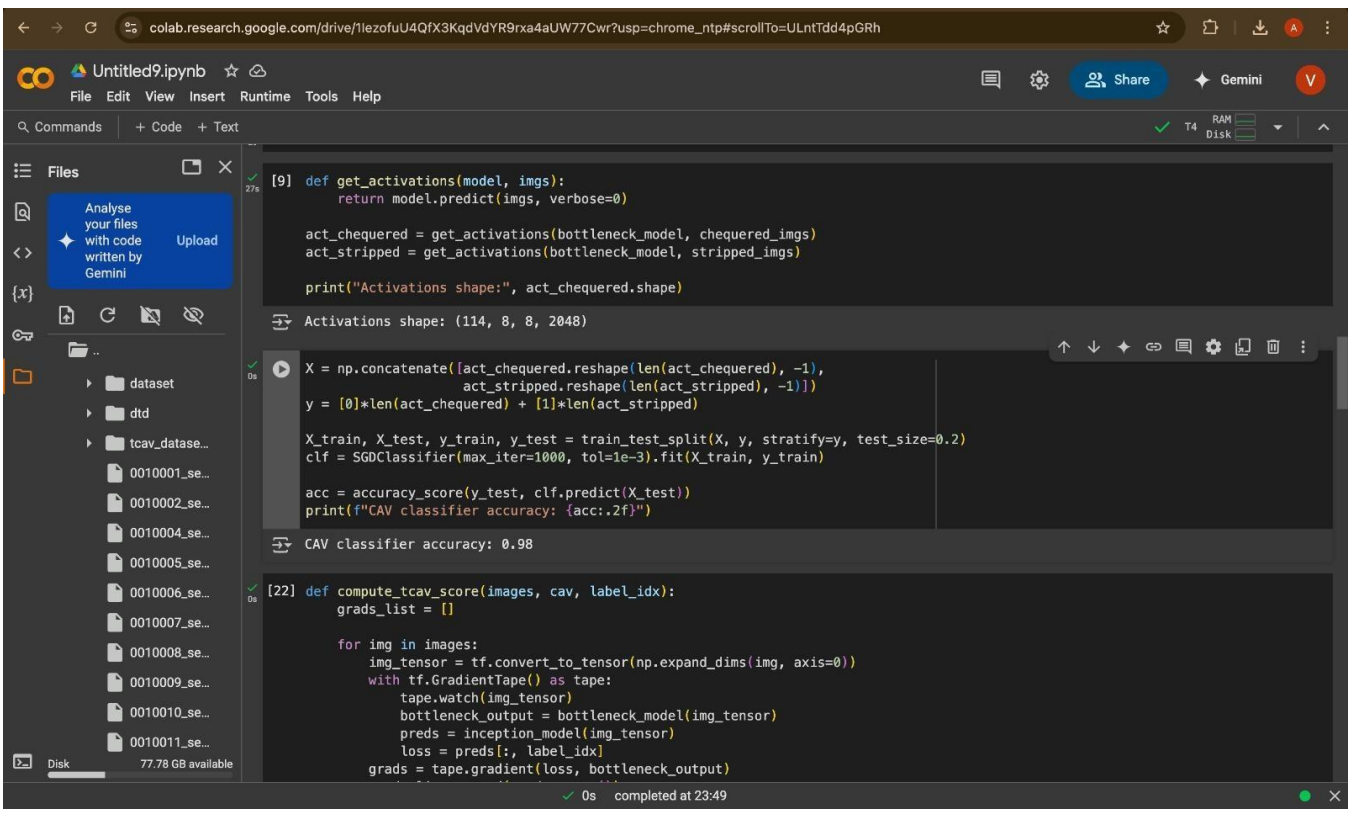
```
[6] def load_images_from_folder(folder, target_size=(299, 299)):
        images = []
        for filename in os.listdir(folder):
            img_path = os.path.join(folder, filename)
            img = image.load_img(img_path, target_size=target_size)
            img = image.img_to_array(img)
            images.append(preprocess_input(img))
        return np.array(images)

    chequered_imgs = load_images_from_folder(concept_dir)
    stripped_imgs = load_images_from_folder(target_dir)

    print(f'Loaded {len(chequered_imgs)} chequered images')
    print(f'Loaded {len(stripped_imgs)} stripped images')
```

```
Loaded 114 chequered images
Loaded 114 stripped images
```

```
# Show samples
plt.figure(figsize=(8, 4))
for i in range(3):
    plt.subplot(2, 3, i+1)
    plt.imshow(tf.cast((chequered_imgs[i] + 1) * 127.5, tf.uint8))
    plt.title("Chequered")
    plt.axis("off")

    plt.subplot(2, 3, i+4)
    plt.imshow(tf.cast((stripped_imgs[i] + 1) * 127.5, tf.uint8))
    plt.title("Stripped")
    plt.axis("off")
plt.tight_layout()
plt.show()
```

Chequered Chequered Chequered


colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=N127Ur5guNhw
Untitled9.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
Files
Analyse your files with code written by Gemini
Upload
dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...
Disk 77.78 GB available
plt.show()
Chequered
Chequered
Chequered
Stripped
Stripped
Stripped
inception_model = InceptionV3(include_top=True, weights='imagenet')
bottleneck_model = tf.keras.Model(
inputs=inception_model.input,
outputs=inception_model.get_layer(bottleneck_name).output
)
[9] def get_activations(model, imgs):
return model.predict(imgs, verbose=0)
act_chequered = get_activations(bottleneck_model, chequered_imgs)
act_stripped = get_activations(bottleneck_model, stripped_imgs)
0s completed at 23:49


colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=ULntTdd4pGRh
Untitled9.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
Files
Analyse your files with code written by Gemini
Upload
dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...
Disk 77.78 GB available
[9] def get_activations(model, imgs):
return model.predict(imgs, verbose=0)
act_chequered = get_activations(bottleneck_model, chequered_imgs)
act_stripped = get_activations(bottleneck_model, stripped_imgs)
print("Activations shape:", act_chequered.shape)
Activations shape: (114, 8, 8, 2048)
X = np.concatenate([act_chequered.reshape(len(act_chequered), -1),
act_stripped.reshape(len(act_stripped), -1)])
y = [0]*len(act_chequered) + [1]*len(act_stripped)
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2)
clf = SGDClassifier(max_iter=1000, tol=1e-3).fit(X_train, y_train)
acc = accuracy_score(y_test, clf.predict(X_test))
print(f"CAV classifier accuracy: {acc:.2f}")
CAV classifier accuracy: 0.98
[22] def compute_tcav_score(images, cav, label_idx):
grads_list = []
for img in images:
img_tensor = tf.convert_to_tensor(np.expand_dims(img, axis=0))
with tf.GradientTape() as tape:
tape.watch(img_tensor)
bottleneck_output = bottleneck_model(img_tensor)
preds = inception_model(img_tensor)
loss = preds[:, label_idx]
grads = tape.gradient(loss, bottleneck_output)
0s completed at 23:49

colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=ULntTdd4pGRh

Untitled9.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

Files

Analyse your files with code written by Gemini Upload

dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...

Disk 77.78 GB available

[23]
```
def directional_derivative(gradients, cav):
    flat_grads = gradients.reshape(len(gradients), -1)
    flat_cav = cav.reshape(-1)
    return np.dot(flat_grads, flat_cav)
```

[29]
```
def compute_tcav_score(images, cav, label_idx):
    grads_list = []

    for img in images:
        img_tensor = tf.convert_to_tensor(np.expand_dims(img, axis=0), dtype=tf.float32)

        with tf.GradientTape() as tape:
            # Watch the input tensor (img_tensor)
            tape.watch(img_tensor)

            # Get bottleneck output and prediction from full model
            bottleneck_output = bottleneck_model(img_tensor)
            preds = inception_model(img_tensor)
            loss = preds[:, label_idx]

        # Get gradients w.r.t. bottleneck layer output
        grads = tape.gradient(loss, bottleneck_output)

        # Raise ValueError if gradient is still None
        if grads is None:
            raise ValueError("Gradient is None. Check if img_tensor is being watched correctly. Ensure all operations between both

        grads_list.append(grads.numpy())

    grads_arr = np.array(grads_list)
    dd = directional_derivative(grads_arr, cav)
    return np.mean(dd < 0)
```

0s completed at 23:49

colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=ULntTdd4pGRh

Untitled9.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Share Gemini

Files

Analyse your files with code written by Gemini Upload

dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...

Disk 77.78 GB available

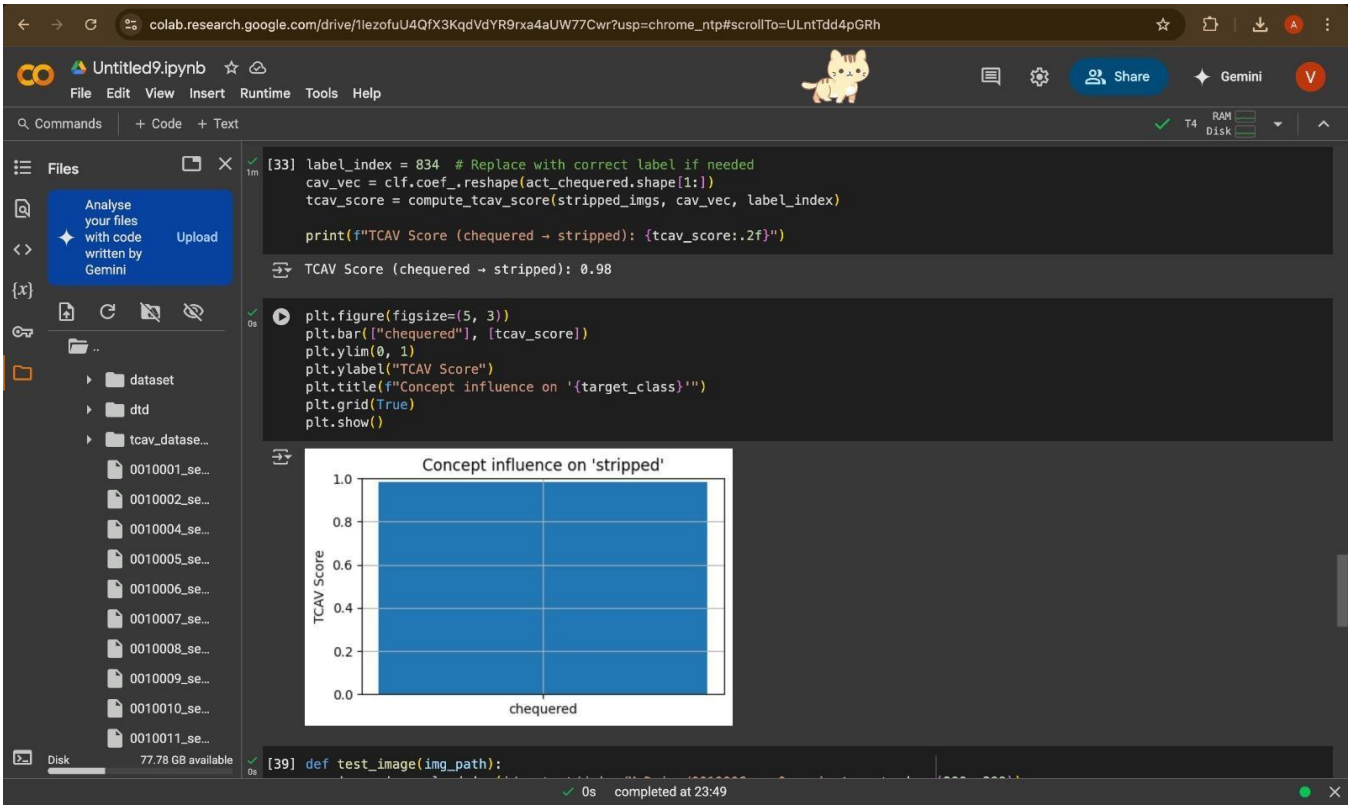
[31]
```
# Get layer after 'mixed10' (flatten → dense → softmax)
logit_model = tf.keras.Model(
    inputs=bottleneck_model.output,
    outputs=inception_model.output
)
```

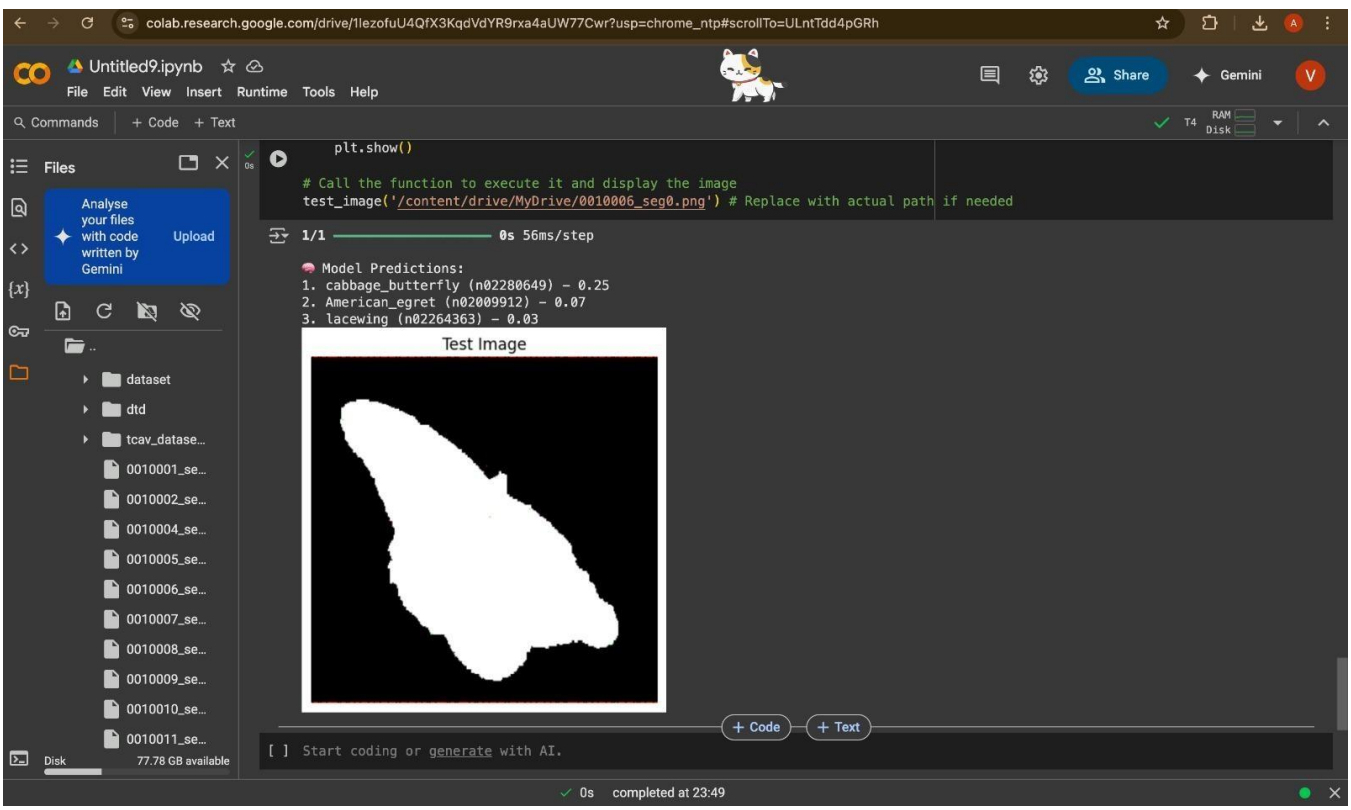
Generated code may be subject to a licence | dcabezas98/AA | stackoverflow.com/questions/74817507/training-multiple-models-using-multiprocessing

```
def compute_tcav_score(images, cav, label_idx):
    grads_list = []

    for img in images:
        img_tensor = tf.convert_to_tensor(np.expand_dims(img, axis=0), dtype=tf.float32)

        with tf.GradientTape() as tape:
            bottleneck_output = bottleneck_model(img_tensor)
            tape.watch(bottleneck_output)

            logits = logit_model(bottleneck_output)
            loss = logits[:, label_idx]

        grads = tape.gradient(loss, bottleneck_output)

        if grads is None:
            raise ValueError("Gradient is None. Ensure tape is watching bottleneck_output and logit_model is used.")

        grads_list.append(grads.numpy())

    grads_arr = np.array(grads_list)
    dd = directional_derivative(grads_arr, cav)
    return np.mean(dd < 0)
```

[33]
```
label_index = 834  # Replace with correct label if needed
cav_vec = clf.coef_.reshape(act_chequered.shape[1:])
```

0s completed at 23:49


colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=ULntTdd4pGRh
Untitled9.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
Files
Analyse your files with code written by Gemini
Upload
dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...
Disk 77.78 GB available
[33] label_index = 834 # Replace with correct label if needed
cav_vec = clf.coef_.reshape(act_chequered.shape[1:])
tcav_score = compute_tcav_score(stripped_imgs, cav_vec, label_index)
print(f"TCAV Score (chequered → stripped): {tcav_score:.2f}")
TCAV Score (chequered → stripped): 0.98
plt.figure(figsize=(5, 3))
plt.bar(["chequered"], [tcav_score])
plt.ylim(0, 1)
plt.ylabel("TCAV Score")
plt.title(f"Concept influence on '{target_class}'")
plt.grid(True)
plt.show()
Concept influence on 'stripped'
TCAV Score
1.0
0.8
0.6
0.4
0.2
0.0
chequered
[39] def test_image(img_path):
0s completed at 23:49


colab.research.google.com/drive/1lezofuU4QfX3KqdVdYR9rxa4aUW77Cwr?usp=chrome_ntp#scrollTo=ULntTdd4pGRh
Untitled9.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
Files
Analyse your files with code written by Gemini
Upload
dataset
dtd
tcav_datase...
0010001_se...
0010002_se...
0010004_se...
0010005_se...
0010006_se...
0010007_se...
0010008_se...
0010009_se...
0010010_se...
0010011_se...
Disk 77.78 GB available
plt.show()
# Call the function to execute it and display the image
test_image('/content/drive/MyDrive/0010006_seg0.png') # Replace with actual path if needed
1/1 0s 56ms/step
Model Predictions:
1. cabbage_butterfly (n02280649) - 0.25
2. American_egret (n02009912) - 0.07
3. lacewing (n02264363) - 0.03
Test Image
+ Code + Text
Start coding or generate with AI.
0s completed at 23:49


Ready
00:11