CSE463 Lab Assignment 3

Deadline: 6th December, 11:59PM

Submission instructions:

- Open a folder in Google Drive and name it as- ID_Name_Lab3
    a. For Q1, upload the **pizza image and the sandwich/cake image**.
    b. For Q2, upload **original cat image, transformed cat image, output image** with matched descriptors, and **dog image.**
    c. For Q3, **upload the original and final output image** after running the sequential model.
    d. Upload a **PDF** consisting of the codes & screenshots of the outputs of each code.
- Share the link of the folder in the submission form. (Make sure the folder is accessible(**Anyone with the link**)
- Submission Link: https://forms.gle/uwuLNkv4zrpATQnT7

QUESTION 1

Can Harris Find Corners on a Pizza? 🍕

Take an image of a **pizza** (with interesting toppings).

1. **Steps**:
    - Apply the Harris Corner Detection algorithm to the image.
    - Analyze the results:
        - Does it detect corners on the **pepperoni slices**?
        - Are the crust edges identified as corners?
        - How does the algorithm handle the "cheese texture"?
2. **Analysis**:
    - Explain why the algorithm might struggle with circular shapes or irregular toppings.

- Replace the pizza with a perfectly square sandwich or a triangle-shaped slice of cake. Compare the results—what kind of food makes Harris the happiest?

QUESTION 2

The Case of the Shape-Shifting Cat 🐾🐈‍⬛

Scenario:
A mischievous **shape-shifting cat** has escaped the virtual zoo and is causing chaos on the internet. The cat has been transforming its appearance to evade detection by the zookeepers! Your task is to help track it down using advanced image processing techniques.

Part 1: The Cat's Trickery
The shape-shifting cat loves to transform. To simulate its tricks, perform the following transformations on an image of the cat:

1. **The Shrinking Spell**: Scale the cat image down—now it's small enough to hide under a digital couch!
2. **The Teleportation Act**: Translate the cat image by shifting it horizontally and vertically.
3. **The Twisting Tail Move**: Rotate the image (e.g., by 45° or 90°)—it's trying to confuse its pursuers!
4. **The Sunbeam Glow**: Brighten the image dramatically—it's pretending to be a glowing angel.

Part 2: Deploying SIFT the Cat Tracker
SIFT (Scale-Invariant Feature Transform) is your **digital pet tracker**. Use it to:

1. Extract **keypoints and descriptors** from both the original and transformed images of the cat.
2. **Match the keypoints** between the images to prove they're the same shape-shifting MEOW🐈.
3. Visualize the matches by drawing lines between the corresponding points on the two images.
4. This clever cat doesn't stop at shape-shifting—it's been spotted disguised as a **dog**! Try using SIFT on two completely different images (e.g., a cat and a dog). Does it confuse the two? Write a short note explaining whether SIFT can handle such a devious disguise.

QUESTION 3

Build a CNN Model Using VGG-16 Architecture

Part 1: Use another image. Implement the **VGG-16** architecture(attached below) in Python using a deep learning library such as TensorFlow or PyTorch to build a sequential model.

Part 2: Discuss the role of each layer in the architecture and why VGG-16 uses small filters (3×3).

---

1. **For reference- VGG-16 Architecture**:

   VGG-16 consists of **13 convolutional layers** and **3 fully connected layers**, making it a total of 16 layers.

   Define the Input Layer:

   - Input size: 224 × 224 × 3 (Height × Width × Channels).

   Add Convolutional Layers:

- Use 3×3 filters for all convolutional layers.
- Apply ReLU activation after each convolutional operation.
- Group convolutional layers into blocks:
    - Block 1: 2 convolutional layers with 64 filters each.
    - Block 2: 2 convolutional layers with 128 filters each.
    - Block 3: 3 convolutional layers with 256 filters each.
    - Block 4: 3 convolutional layers with 512 filters each.
    - Block 5: 3 convolutional layers with 512 filters each.

Add Max Pooling Layers:

- After **each** block, add a MaxPooling layer with a 2×2 pool size and a stride of 2 to reduce spatial dimensions.

Flatten the Output:

- At the end of the convolutional blocks, flatten the 3D feature maps into a 1D vector.

Add Fully Connected Layers:

- Add two dense (fully connected) layers with 4096 neurons each and ReLU activation.
- Add a final dense layer with 1000 neurons (or the number of classes in your dataset) and softmax activation.