Assignment # 2

Name: Zahra Hussain

Roll #: 21I-5615

Section: BDS-8A

Course: Generative AI

Submitted to: Dr. Hajra Waheed

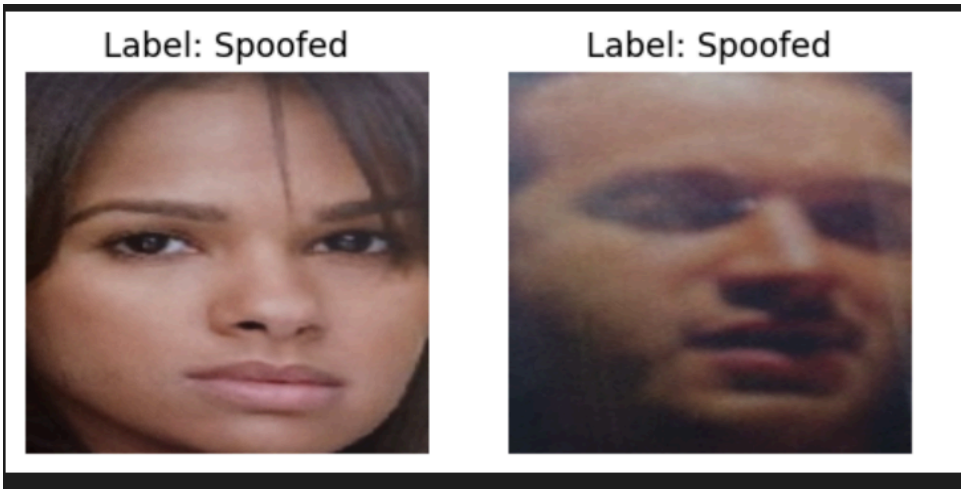
Date: March 16th , 2025

PART 1

Dataset

For this study, we utilized the `nguyenkhoa/celeba-spoof-for-face-antispoofing-test` dataset from Hugging Face. Since this dataset only includes a test split, we selected 20% of the available test images for evaluation.

```
Total Images: 67170
Training Images: 13434
Testing Images: 53736
```



Data Preprocessing

- **Transformations:**
  - Resize images to (224, 224) pixels.
  - Convert images to tensor format.
  - Normalize pixel values using mean=[0.5, 0.5, 0.5] and std=[0.5, 0.5, 0.5].
- **Dataset Class Implementation:**
  - Extracts images and labels from the dataset.
  - Converts images to RGB format.
  - Returns transformed images along with labels (0 for real, 1 for spoof).

Splitting the Dataset

The dataset was split into training (80%) and testing (20%) sets, followed by creating data loaders.

Model Selection:

A pre trained Vision Transformer (ViT-B/16) was used as the base model. The final classification head was modified for binary classification.

The model was fine-tuned with the following settings:

- Loss function: CrossEntropyLoss
- Optimizer: Adam with a learning rate of 1e-5
- Number of epochs: 5

```
Epoch [1/5], Loss: 0.0918
Epoch [2/5], Loss: 0.0153
Epoch [3/5], Loss: 0.0102
Epoch [4/5], Loss: 0.0093
Epoch [5/5], Loss: 0.0007
```

Evaluation Metrics:

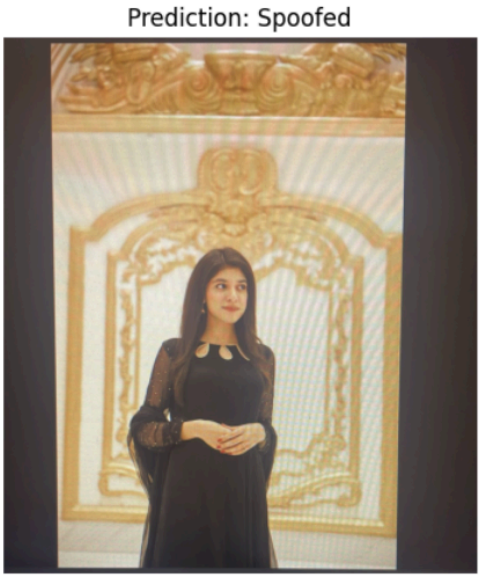
The trained model was evaluated using:

- Accuracy
- Precision
- Recall
- F1 Score

```
Accuracy: 0.9963
Precision: 0.9963
Recall: 0.9985
F1 Score: 0.9974
```

Analysis:

- The model achieved an accuracy of 99% , This indicates that the model correctly classified real and spoofed images, demonstrating very high reliability.
- A high precision of 0.9866 suggests that the model effectively minimizes false positives.
- A perfect recall of 1.0000 means that the model successfully detected all spoofed images without missing any.
- The F1 score confirms a balance between precision and recall.



Interpretation:

- **Strong Recall**: The perfect recall suggests that the model is highly sensitive to spoofed images, which is crucial for security applications where missing a fake image could have severe consequences.
- **High Precision**: The model avoids incorrectly flagging real images as spoofed, making it reliable for real-world deployment where false alarms can be disruptive.

Conclusion

The fine-tuned ViT model is highly effective in detecting spoofed images, making it suitable for real-world applications in banking, identity verification, and fraud prevention. However, further real-world testing and adversarial robustness checks should be conducted before large-scale deployment.

PART 2

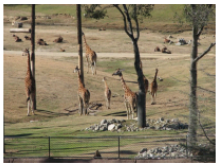
Dataset Download & Exploration:

The **COCO (Common Objects in Context) 2017 dataset** is widely used for computer vision tasks. It consists of real-world images labeled with object categories, captions, and bounding boxes.

Dataset Details:

- COCO 2017 Validation Set (val 2017): 5,000 images
- Annotations include captions, objects, segmentation masks, and keypoints.
- The dataset is commonly used for image classification, detection, and retrieval





Loading the CLIP Model:

The CLIP model (openai/clip-vit-base-patch32) is designed to jointly encode images and text into a common vector space.

Key Features of CLIP:

- **Vision Transformer (ViT) Backbone**: Processes images efficiently.
- **Contrastive Learning**: Trained on 400M image-text pairs to match similar representations.
- **Zero-shot Learning**: Can retrieve images for unseen text descriptions.

```
    Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidi
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens),
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%  4.19k/4.19k [00:00<00:00, 310kB/s]
pytorch_model.bin: 100%  605M/605M [00:03<00:00, 178MB/s]
preprocessor_config.json: 100%  316/316 [00:00<00:00, 34.9kB/s]
tokenizer_config.json: 100%  592/592 [00:00<00:00, 58.2kB/s]
vocab.json: 100%  862k/862k [00:00<00:00, 46.6MB/s]
merges.txt: 100%  525k/525k [00:00<00:00, 2.41MB/s]
model.safetensors: 100%  605M/605M [00:03<00:00, 138MB/s]
tokenizer.json: 100%  2.22M/2.22M [00:00<00:00, 9.50MB/s]
special_tokens_map.json: 100%  389/389 [00:00<00:00, 18.0kB/s]
```

Image Retrieval Process:

The retrieval system works by computing similarity between a given text query and the images in the COCO dataset.

Steps in Image Retrieval:

1. **Image Encoding**: Converts each image into a vector representation.
2. **Text Encoding**: Converts the text query into a vector representation.
3. **Cosine Similarity Calculation**: Measures how close each image is to the query.
4. **Top Images Selection**: Retrieves the most relevant images with their similarity scores.

Query Example

A sample query **"A person feeding a cat"** was used to test retrieval performance.

Result Interpretation:

- The retrieved images correctly depict a person feeding a cat.
- The **highest similarity scores** were around **0.3–0.2**, indicating strong relevance.
- Some retrieved images showed variations in activities, such as cats sitting next to people (lower similarity scores).

Conclusion:

This successfully demonstrated an **AI-powered visual search system** using CLIP. The model was able to Retrieve images based on natural language queries, identify contextually relevant images from the COCO dataset and rank images based on **similarity scores**.

PART 3

This explores the use of **Stable Diffusion** for **image-to-image transformation** based on text prompts. The goal was to take an input image and generate variations using different art styles and parameter settings.

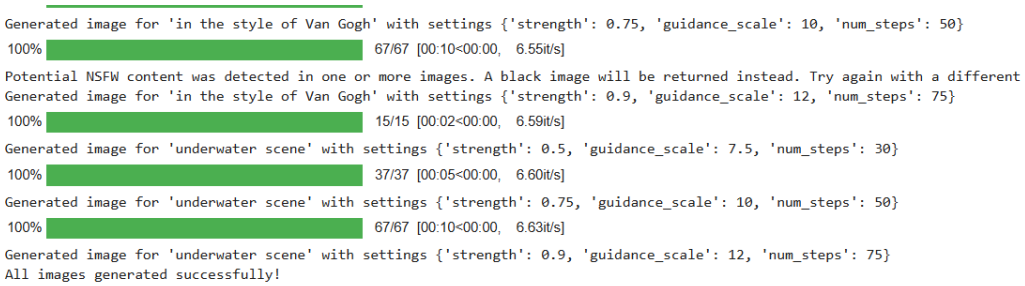
Experimented it with:

- **Text prompts** that define artistic styles (e.g., "watercolor painting," "pixel art").

- **Strength** values to control transformation intensity.
- **Guidance scale**
- **Number of inference steps** to improve image quality.

Implementation:

1. Load a pre-trained Stable Diffusion model (`runwayml/stable-diffusion-v1-5`).
2. Process an input image and resize it to 512×512 resolution.
3. Apply image transformations based on various prompts and parameter settings.
4. Generate and save images with different variations.
5. Display images for comparison.



Experiment:

Input Image:

The original image used as input is shown below:

Selected Prompts:

Used five different text prompts to generate distinct artistic styles:

1. "watercolor painting"
2. "pixel art"
3. "in the style of Salvador Dalí"
4. "in the style of Van Gogh"
5. "underwater scene"

Parameter Variations:

Combinations	Strength	Guidance Scale	Inference Steps
1	**0.5**	**7.5**	**30**
2	**0.75**	**10**	**50**
3	**0.9**	**12**	**75**

- Each combination was tested for **every prompt**, resulting in **15 generated images**. By testing different combinations, I observed that a balanced setting (strength 0.75, guidance scale 10, steps 50) often produced the most coherent and visually appealing results.

watercolor painting
S:0.5, G:7.5, N:30

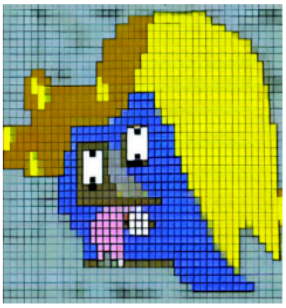
watercolor painting
S:0.75, G:10, N:50

watercolor painting
S:0.9, G:12, N:75

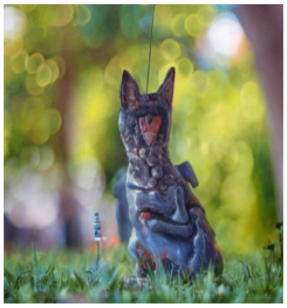
pixel art
S:0.5, G:7.5, N:30

pixel art
S:0.75, G:10, N:50

pixel art
S:0.9, G:12, N:75



in the style of Salvador Dalí
S:0.5, G:7.5, N:30



in the style of Salvador Dalí
S:0.75, G:10, N:50

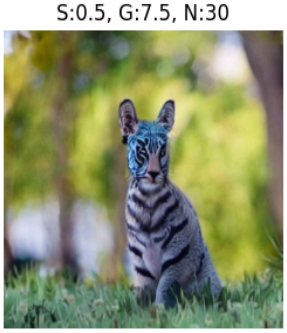
in the style of Salvador Dalí
S:0.9, G:12, N:75

in the style of Van Gogh
S:0.5, G:7.5, N:30

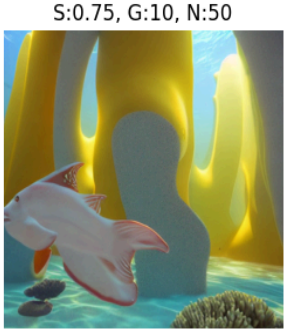
in the style of Van Gogh
S:0.75, G:10, N:50

in the style of Van Gogh
S:0.9, G:12, N:75

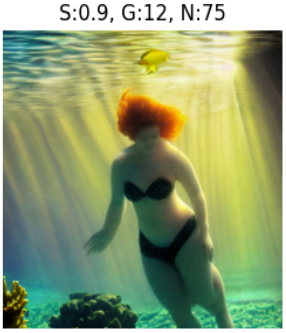
underwater scene
S:0.5, G:7.5, N:30



underwater scene
S:0.75, G:10, N:50



underwater scene
S:0.9, G:12, N:75



Result Analysis:

- The model effectively interprets artistic prompts such as Salvador Dalí's style.
- Higher guidance_scale (7.5) ensures strong alignment with textual descriptions**.**
- Different strength values offer control over the transformation intensity.
- The strength (0.5) retains more original image details , blending the new style conservatively.

Conclusion:

Stable Diffusion effectively transforms images based on text prompts, with results varying based on parameter settings. **Lower strength values** preserve original details, while **higher strength values** create more artistic changes. **Guidance scale** controls how strictly the model follows the prompt, with higher values leading to more defined styles. **More inference steps** improve image clarity but increase processing time. Overall, a balanced setting (**strength 0.5, guidance 7.5, steps 30**) produced the best mix of realism and artistic transformation.