Department of Computer Science
Gujarat University
5 Year Integrated M.Sc.(Computer Science)
Semester - VIII

Text Encryption and Decryption using Gen AI

Presented By:
Trupal Lathiya (80007)
Ansh Yadav (80021)

Under the Guidance of :
Ms. Saloni Sah

Index

Project Profile

TITLE	DESCRIPTION
Project Title	Text Encryption and Decryption using Gen AI
Aim of the Project	The project aims to use Generative AI for secure, adaptive text encryption and decryption.
Project duration	6 month
Team size	2 Member
Tools used	Visual Studio code, Git hub
Technologies used	Cryptography, Tensorflow, Stepic , Matplotlib, Numpy, Panda, PyTorch , Next.js, Flask
Guide	Ms. Saloni Sah

OBJECTIVE

Develop a method for securely embedding and sharing hidden data within images while minimizing noise and distortion. Utilize Generative Adversarial Networks (GANs) and Generative AI (GenAI) to enhance data concealment, ensuring high image quality and security during transmission.

INTRODUCTION

- Securing sensitive information during transmission is a critical challenge. This project explores an advanced method for embedding and sharing hidden data within images while ensuring minimal noise and distortion. By leveraging **Least Significant Bit (LSB) steganography**, hidden data is subtly encoded within pixel grids without compromising image quality. **Generative Adversarial Networks (GANs)** further enhance data concealment by optimizing imperceptible modifications, making detection by unauthorized entities extremely difficult. The synergy of **cryptographic techniques** ensures robust encryption, adding an additional layer of security to prevent unauthorized access.

- To strengthen data embedding and retrieval, we integrate **pose estimation and Multi-View Stereo (MVS)** techniques. Pose estimation helps in extracting key features from human figures for intelligent concealment, while MVS reconstructs high-fidelity 3D models, ensuring depth-aware embedding across multiple perspectives. This combination of **AI-driven methods and cryptographic security** creates a powerful framework for secure image-based data transmission, offering a highly resilient solution against detection and interception threats.

LITERATURE REVIEW

LITERATURE REVIEW

Author	Title	Year	Description	Result
1. Nataniel Ruiz 2. Yuanzhen Li 3. Varun Jampani 4. Yael Pritch 5. Michael Rubinstein 6. Kfir Aberman	DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation	*25 Aug 2022*	This work introduces a novel approach to personalize text-to-image diffusion models by fine-tuning them with just a few images of a specific subject. The model learns to associate a unique identifier with the subject, allowing it to generate diverse and photorealistic images of that subject in different contexts, poses, views, and lighting conditions. By incorporating a new class-specific prior preservation loss, the method ensures the subject's key features are preserved. This technique enables tasks like subject recontextualization, text-guided view synthesis, and artistic rendering, and includes a new dataset and evaluation protocol for subject-driven generation.	CLIP-I : 0.812 CLIP-T : 0.306 DINO : 0.696
1. Jingxiang Sun 2. Bo Zhang 3. Ruizhi Shao 4. Lizhen Wang 5. Wen Liu 6. Zhenda Xie 7. Yebin Liu	DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior	*25 Oct 2023*	DreamCraft3D is a hierarchical 3D content generation method that creates high-fidelity, coherent 3D objects by leveraging a 2D reference image to guide geometry sculpting and texture enhancement. It addresses the consistency issues of existing methods by utilizing score distillation sampling through a view-dependent diffusion model to ensure geometry coherence, though at the cost of texture fidelity. To improve texture quality, DreamCraft3D introduces Bootstrapped Score Distillation, training a personalized diffusion model, Dreambooth, on augmented scene renderings, which infuses it with 3D scene knowledge. This 3D-aware diffusion prior provides view-consistent guidance, and through alternating optimization of the diffusion model and 3D scene representation, the system achieves mutual improvements, enhancing both geometry and texture. As a result, DreamCraft3D generates photorealistic, coherent 3D objects and advances the state-of-the-art in 3D content creation.	CLIP : 0.896 Contextual : 1.579 PSNR: 31.801 LPIPS : 0.005

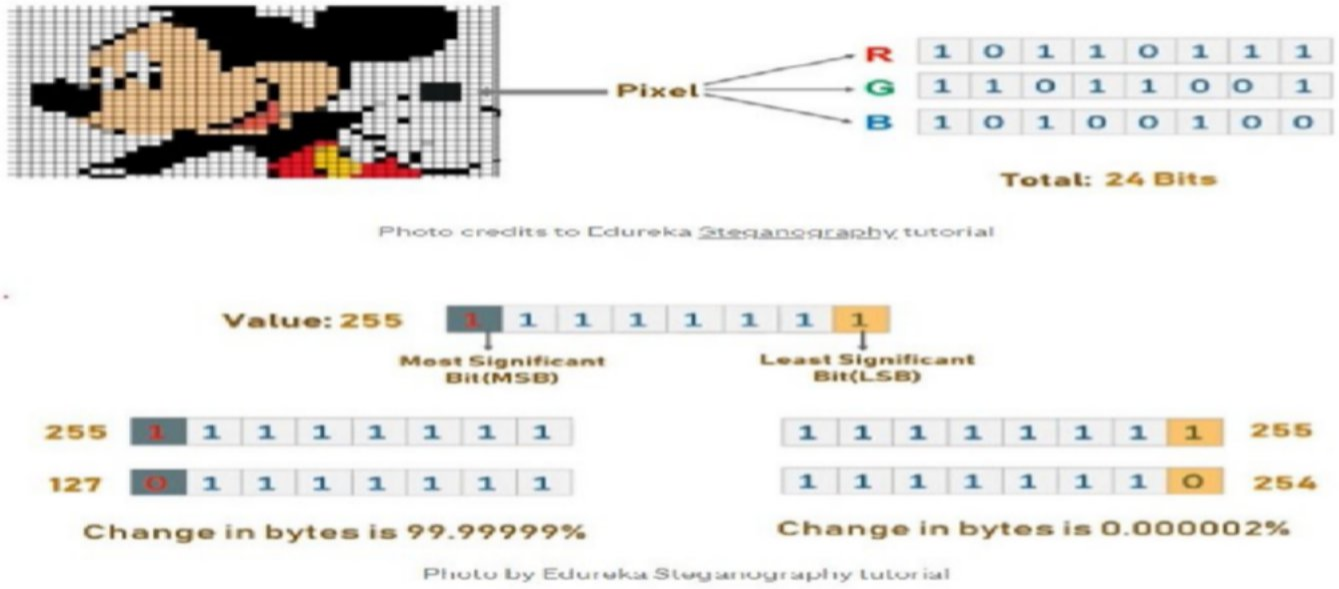
LITERATURE REVIEW

Author	Title	Year	Description	Result
1. Shume et Baluja	Hiding Images within Images	7 July 2020	This system uses deep neural networks to hide one full-color image inside another of the same size, with minimal quality loss to either image. Trained on random images from the ImageNet database, the system performs well on natural images. It explores how deep learning can be applied to image hiding and evaluates its effectiveness against various transformations, including simple manipulations and advanced machine learning-based attacks. The system includes two extensions to improve security and enable the hiding of multiple images, with potential applications in image authentication, digital watermarking, and detecting image manipulations.	-
1. Bishwa Karki 2. Chun-Hua Tsai 3. Pei-Chi Huang 4. Xin Zhong	Deep Learning-based Text-in-Image Watermarking	19 April 2024	We present a novel deep learning-based approach to text-in-image watermarking, which embeds and extracts textual information within images to enhance data security. By utilizing Transformer-based architectures for text processing and Vision Transformers for image feature extraction, our method sets new benchmarks in the field. It is the first deep learning application in this area, offering improved adaptivity to specific image characteristics and emerging threats. Testing shows our method outperforms traditional techniques, delivering superior robustness and imperceptibility, ensuring the watermark remains undetectable across different image contents.	-

METHODOLOGY

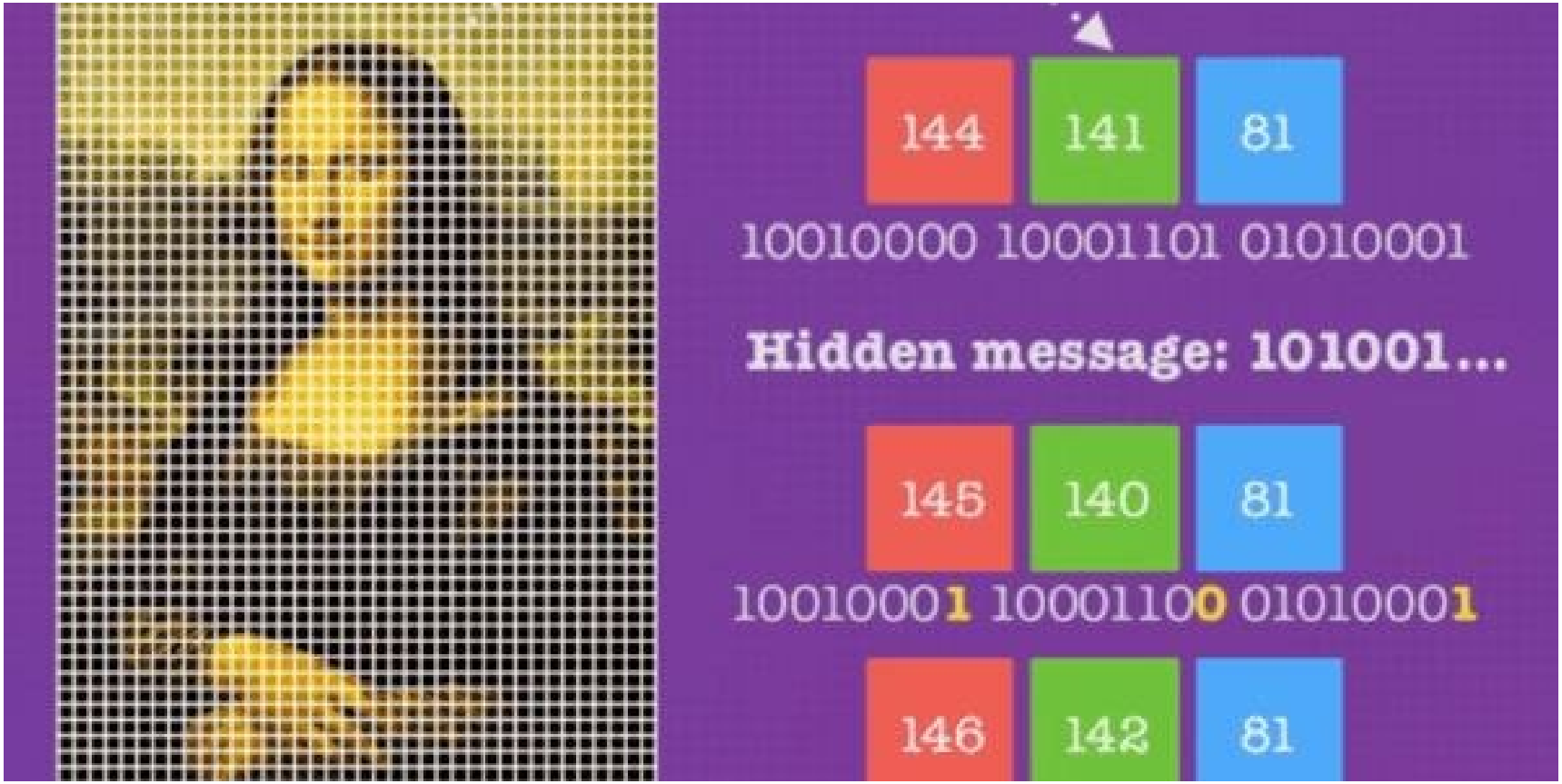
METHODOLOGY

- **LSB (Least Significant Bit) Steganography**: Visualized as digital pixel grids, with some bits subtly altered or hidden within a larger image, symbolizing data concealment.



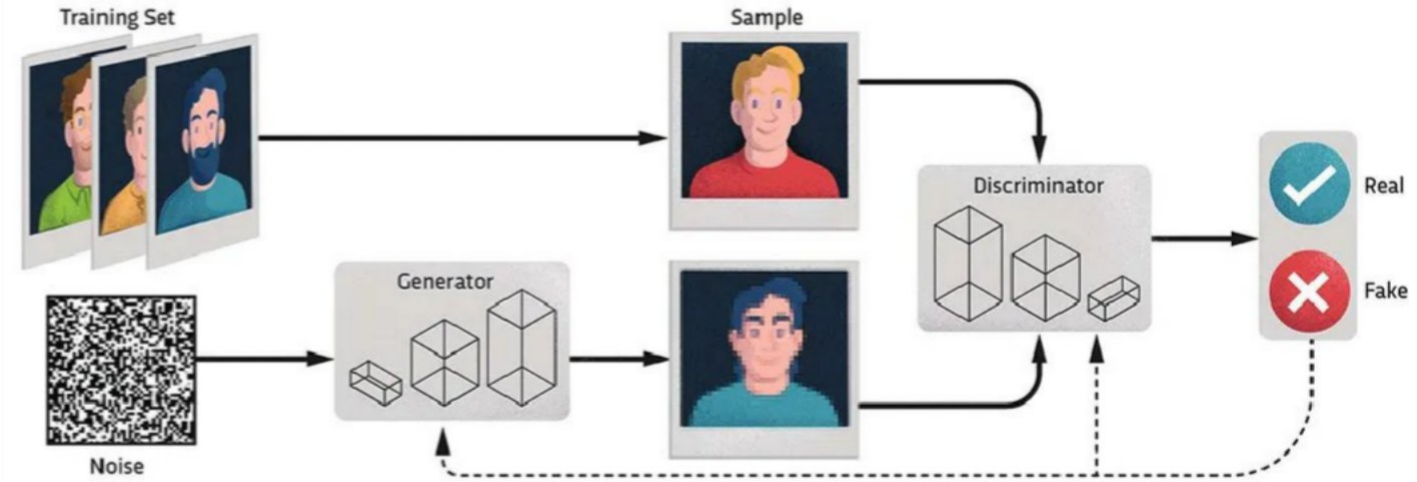
METHODOLOGY

- **Steganography**: Embed hidden messages inside an image through steganography



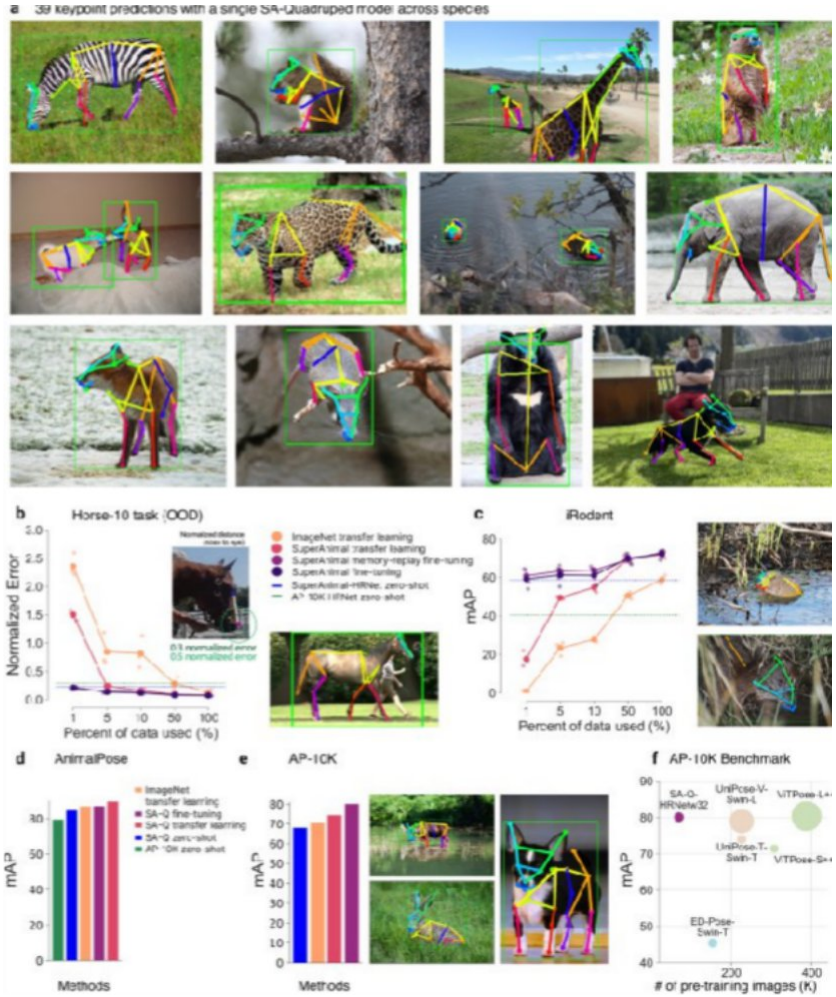
METHODOLOGY

- **GANs (Generative Adversarial Networks)**: Represented by two neural network structures, one generating and the other distinguishing, with flowing data between them, symbolizing AI learning.



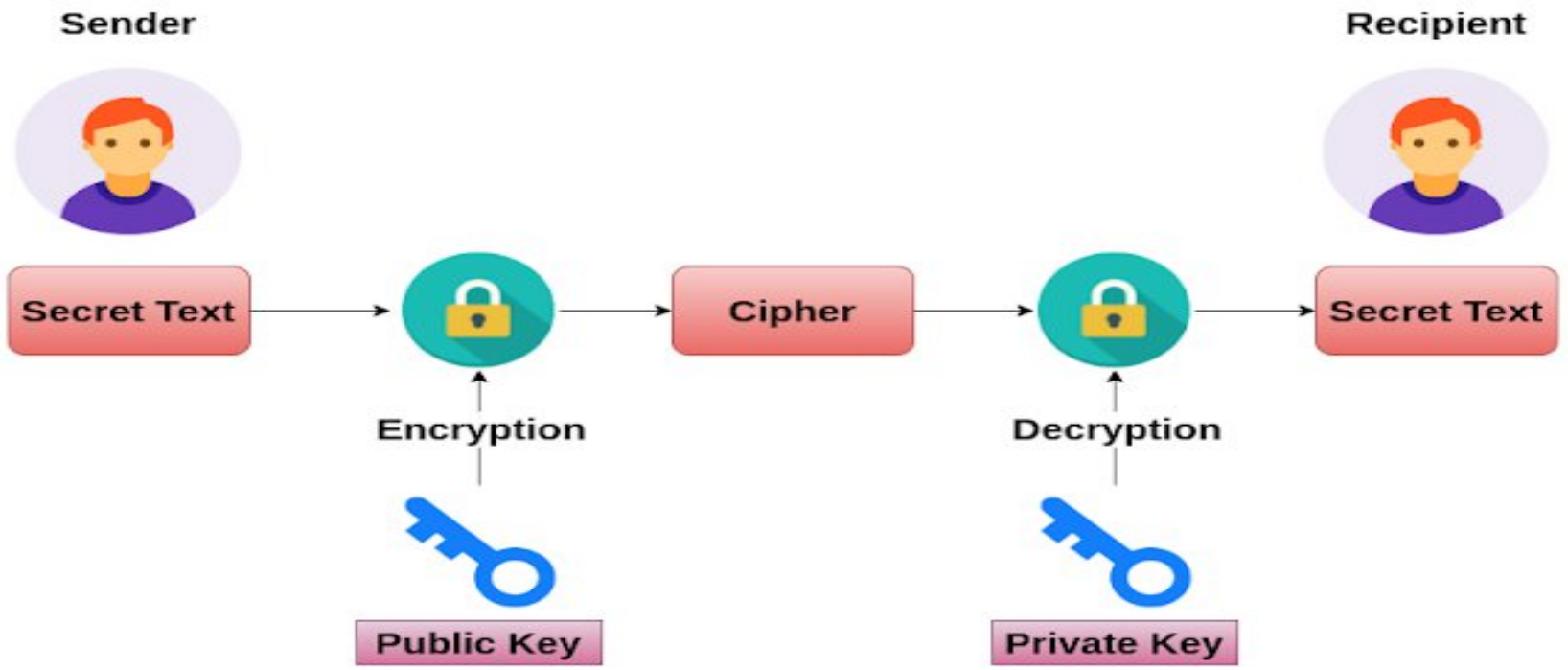
METHODOLOGY

- **Pose Estimation**: Illustrated by a human figure with joints and angles highlighted, representing key body points, with lines connecting the points to form a skeletal structure.



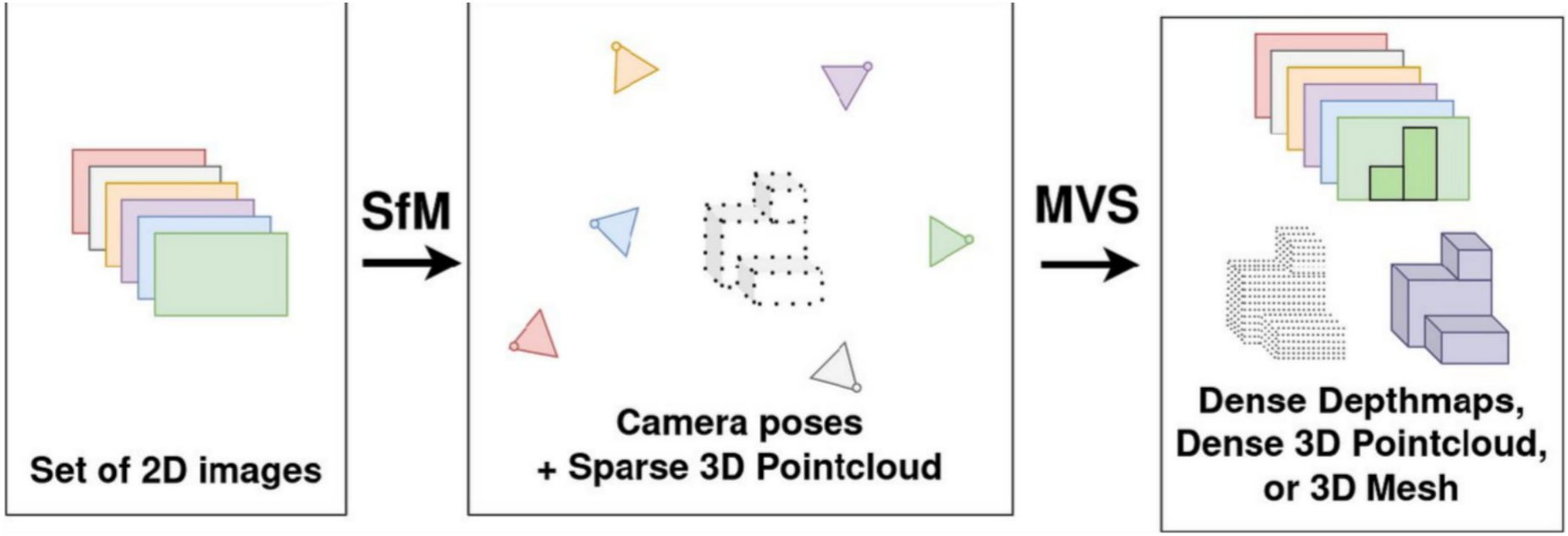
METHODOLOGY

- **Cryptography**: Depicted by a locked padlock, with encrypted binary code or cryptographic symbols in the background, symbolizing data protection.



METHODOLOGY

- **MVS (Multi-View Stereo)**: Shown through multiple overlapping 3D views of an object, representing the collection of different perspectives to reconstruct a detailed 3D model.

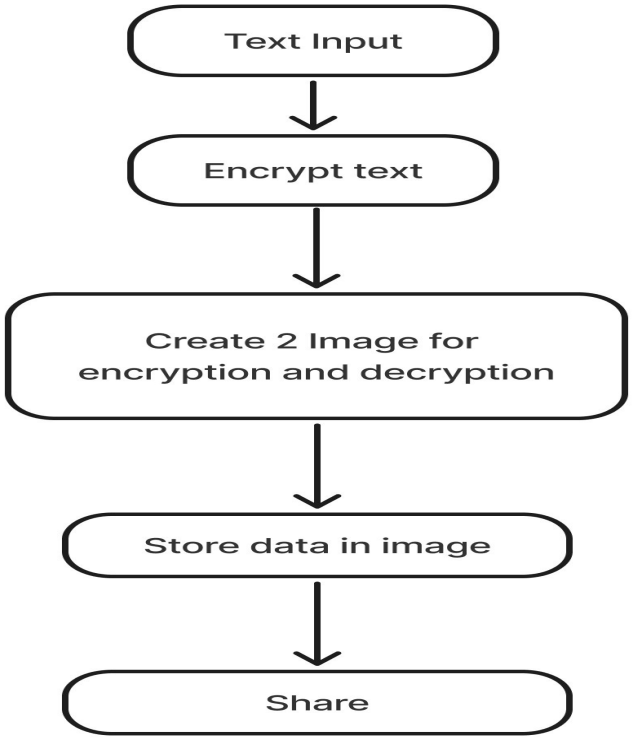


CHALLENGES

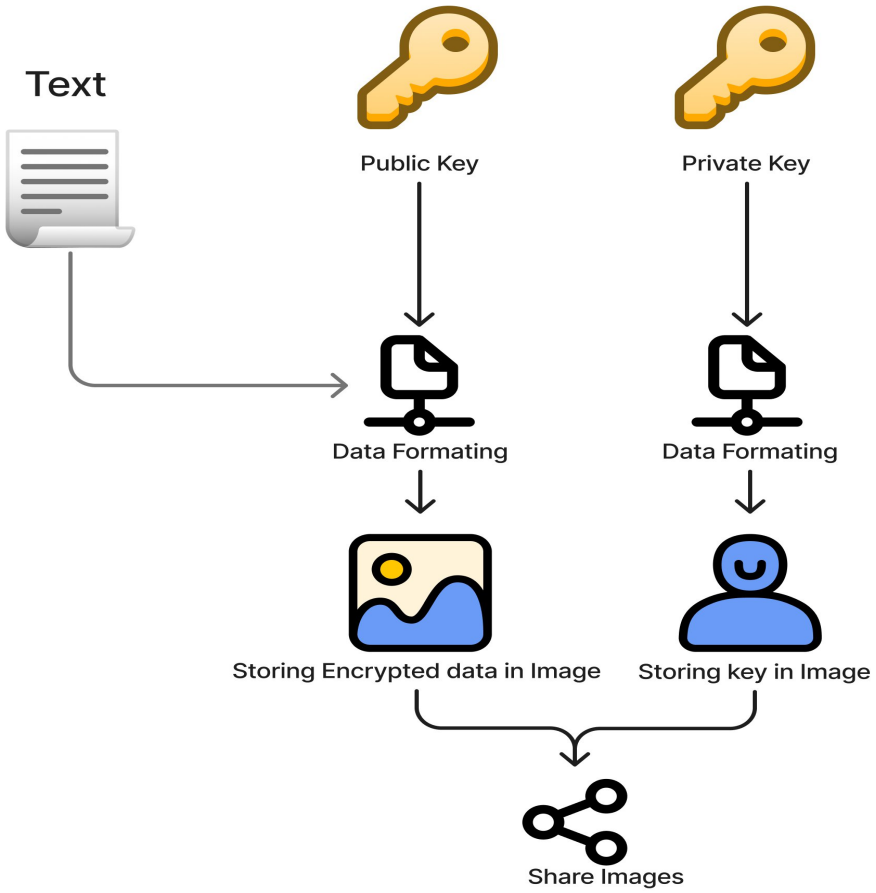
- **Noise and Distortion Control** – Ensuring minimal visual degradation while embedding hidden data.
- **Steganalysis Resistance** – Making hidden data undetectable against advanced forensic and AI-based attacks.
- **Cryptographic Integration** – Combining encryption with steganography without increasing computational overhead.
- **Pose Estimation Accuracy** – Ensuring precise feature extraction for secure and intelligent data embedding.
- **MVS Complexity** – Handling high computational costs and alignment challenges in multi-view 3D reconstruction.

PROCESS

PROCESS


Text Input
Encrypt text
Create 2 Image for encryption and decryption
Store data in image
Share

PROCESS

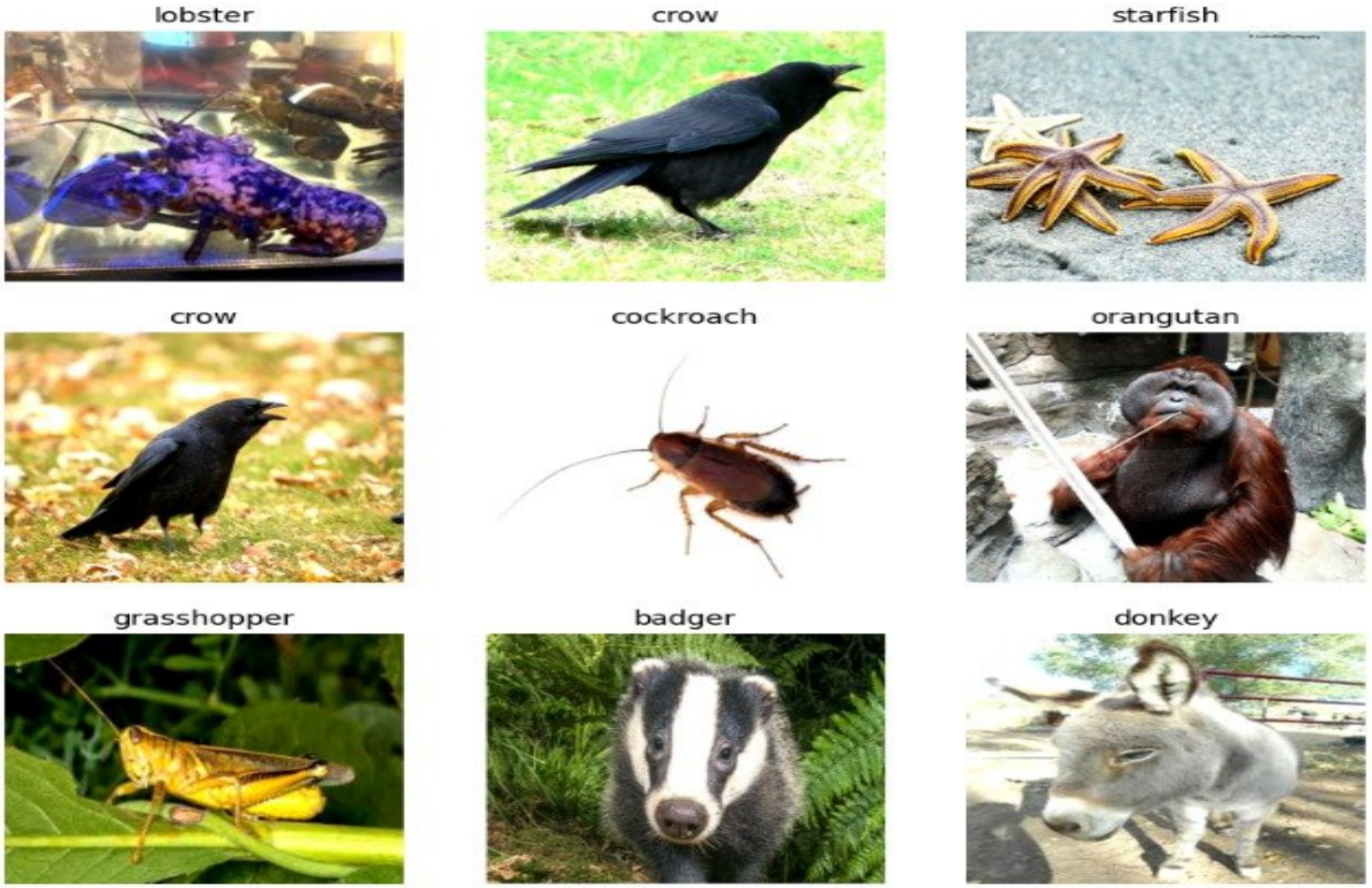

Text
Public Key
Private Key
Data Formating
Data Formating
Storing Encrypted data in Image
Storing key in Image
Share Images

DATASET

DATASET

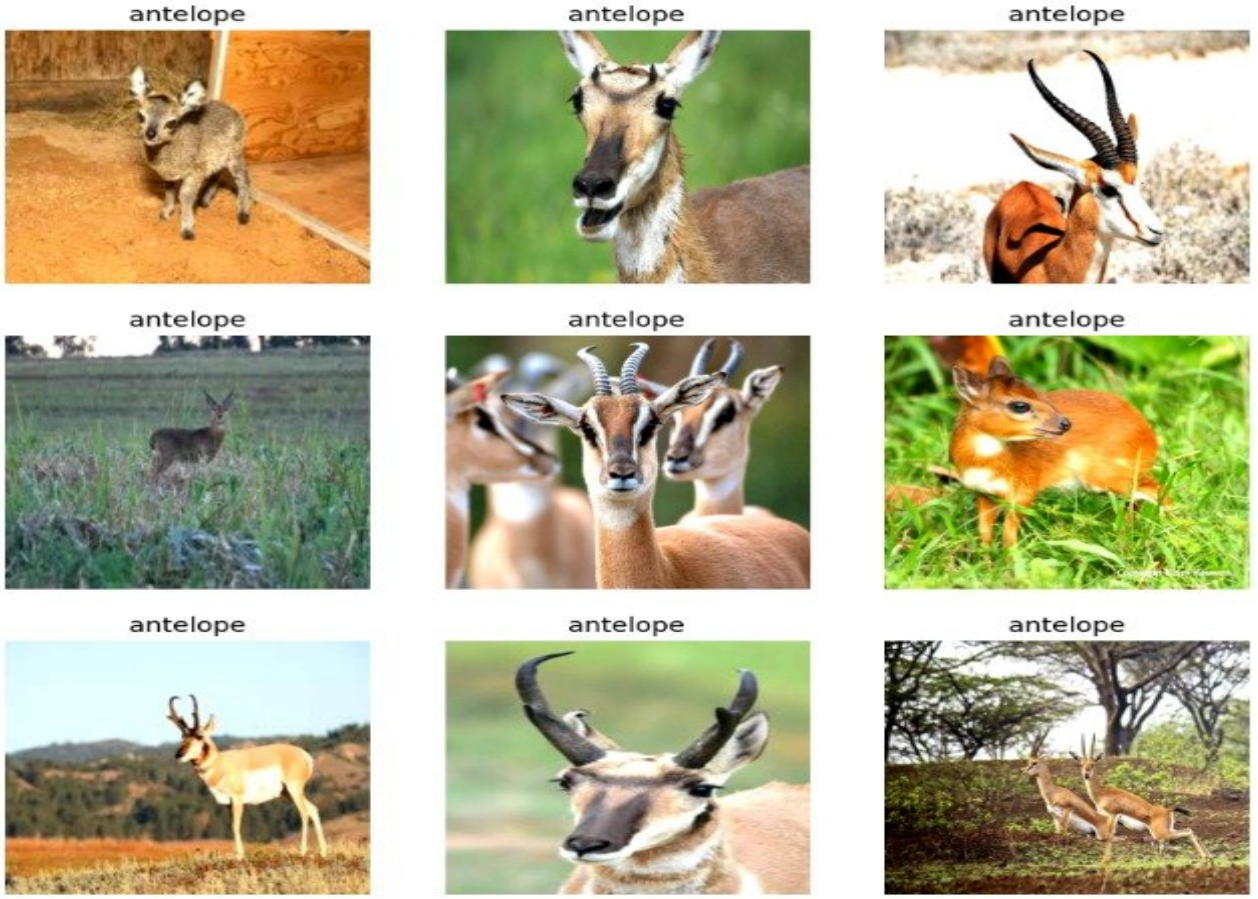
Found 5400 files belonging to 90 classes.
Using 1080 files for validation.
Classes : ['antelope', 'badger', 'bat', 'bear', 'bee', 'beetle', 'bison', 'boar', 'butterfly', 'cat', 'caterpillar', 'chimpanzee', 'cockroach', 'cow', 'coyote', 'crab', 'crow', 'deer', 'dog', 'dolphin', 'donkey', 'dragonfly', 'duck', 'eagle', 'elephant', 'flamingo', 'fly', 'fox', 'goat', 'goldfish', 'goose', 'gorilla', 'grasshopper', 'hamster', 'hare', 'hedgehog', 'hippopotamus', 'hornbill', 'horse', 'hummingbird', 'hyena', 'jellyfish', 'kangaroo', 'koala', 'ladybugs', 'leopard', 'lion', 'lizard', 'lobster', 'mosquito', 'moth', 'mouse', 'octopus', 'okapi', 'orangutan', 'otter', 'owl', 'ox', 'oyster', 'panda', 'parrot', 'pelecaniformes', 'penguin', 'pig', 'pigeon', 'porcupine', 'possum', 'raccoon', 'rat', 'reindeer', 'rhinoceros', 'sandpiper', 'seahorse', 'seal', 'shark', 'sheep', 'snake', 'sparrow', 'squid', 'squirrel', 'starfish', 'swan', 'tiger', 'turkey', 'turtle', 'whale', 'wolf', 'wombat', 'woodpecker', 'zebra']
Training size : 135
Validation size : 34

DATASET


lobster
crow
starfish
crow
cockroach
orangutan
grasshopper
badger
donkey

DATASET (antelope)

Number of batches with antelope images: 2



TOOL AND TECHNOLOGY

TOOLS AND TECHNOLOGY

1. Cryptography

- Ensures data security and confidentiality using encryption techniques.
- Used for securing sensitive information in the project.
- Common libraries: cryptography, PyCrypto, Fernet.

2. TensorFlow

- Open-source machine learning framework for deep learning models.
- Used for building and training neural networks in the project.
- Supports GPU acceleration for faster computation.

3. stepic

- Python-based library for image steganography using LSB technique.
- Used for hiding and retrieving secret messages within PNG images in the project.
- Ideal for concealing sensitive encrypted data without visibly altering the image.

TOOLS AND TECHNOLOGY

4. Matplotlib

- Powerful visualization library for creating static, animated, and interactive graphs.
- Used for plotting charts and data insights in the project.
- Works seamlessly with NumPy and Pandas..

5. NumPy

- Fundamental library for numerical computing in Python.
- Used for handling large datasets and mathematical operations.
- Provides efficient array operations and matrix computations.

6. Pandas

- Data analysis and manipulation tool for structured data.
- Used for handling datasets, preprocessing, and data cleaning.
- Supports operations like filtering, aggregation, and transformation.

7. PyTorch

- Deep learning framework for building and training neural networks.
- Used for developing machine learning models, especially in computer vision and NLP tasks.
- Supports operations like tensor computation, automatic differentiation, and GPU acceleration.

PRE-PROCESSING

PRE-PROCESSING

- **Resizing**: Adjusting the image dimensions to ensure uniformity and computational efficiency. It helps standardize input sizes for deep learning models and reduces processing time. Common interpolation techniques include bilinear, bicubic, and nearest-neighbor methods.

- **Augmentation**: Enhancing dataset diversity by applying transformations like rotation, flipping, scaling, brightness adjustment, and Gaussian blur. This improves model generalization and prevents overfitting.

- **Noise Removal**: Eliminating unwanted distortions (e.g., salt-and-pepper noise, Gaussian noise) using techniques such as median filtering, Gaussian blur, and bilateral filtering. This step enhances edge clarity and feature extraction.

IMPLEMENTATION

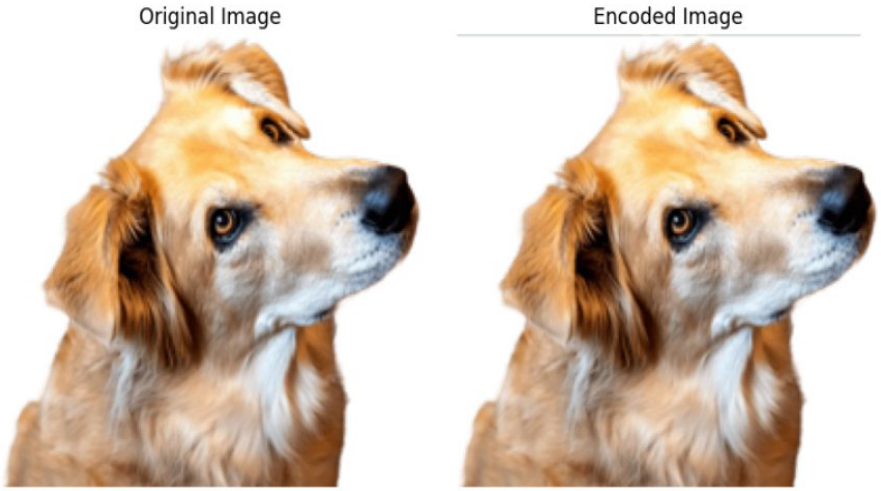
Text Encryption and Storing

Enter the message you want to encrypt and store in image: hello, This is a Secret Message

Original Message: hello, This is a Secret Message

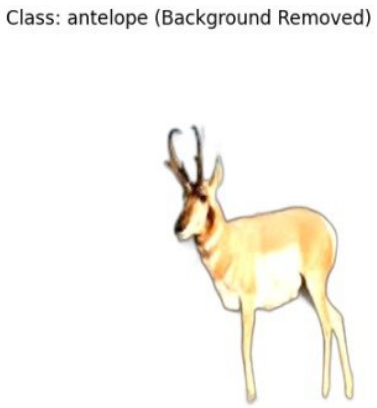
 Encrypted Message Bytes: b'\x1e\xc5e\xc7a\xad\xd8,\x8c\xdd\xea}$\x13\xa3\xcdl\xb2<\x98\xf9\xaf\x0c\x0cH\xcf\xcf\xc0i\x14C\xf5-\x90;"w\x06~_N\xadb\x88+Q\x1d8YKs\x93\x97S\xa7\xbe\x15{\xb1\x16\xdd\x052q\xb9\xecx\x83A\x00\xe5\xdb\x91\xb2\x9fql\xae\xff8l*U\xa4K\x11C\xe4\xd2\x87\x12\xcaM\x1e$\x93T\xe4\xa84\xae\xcd\xc6<n\x1c\xcfA\x08*\xd95\xe5\x93\xec\x8fsl\xef\x13q\xbf\xdd\x8c\xadt\xe6*\x9eH\'\x06\x88T\xd0I\x07\x8e\x89\x8cq\xa0OM\xca\xb9\xc4\xf7\x0c\\\x9c\reGW\xf9\xbd\x88:M\xe9\xb9\x0b\xd1\xa3\x9c\xf7,\x08\xae\x18\r\xfeV\xde\x9a\x0c(DuH\x0f!\xa2\x93\x1fh\x0c\xe0\xcf\x93b6ke&\xd8\n\x03\x92\x8d\x96=\x88\xb4\xb3Y0M\xad\xff\x06/7\xfa\xc1+\x8f\x9e$O,w\xce\x16"\xe3\x96\xaf\x0b\x8f\x14\x9f\xc1=7|\x95w\xa1\xa0*5Y\x9e{\xc5E3\xb2@\xcfE-('

 Encrypted message hidden and image saved as: /kaggle/working/encoded_image.png



 Decrypted Message: hello, This is a Secret Message

Remove Background



Remove Background

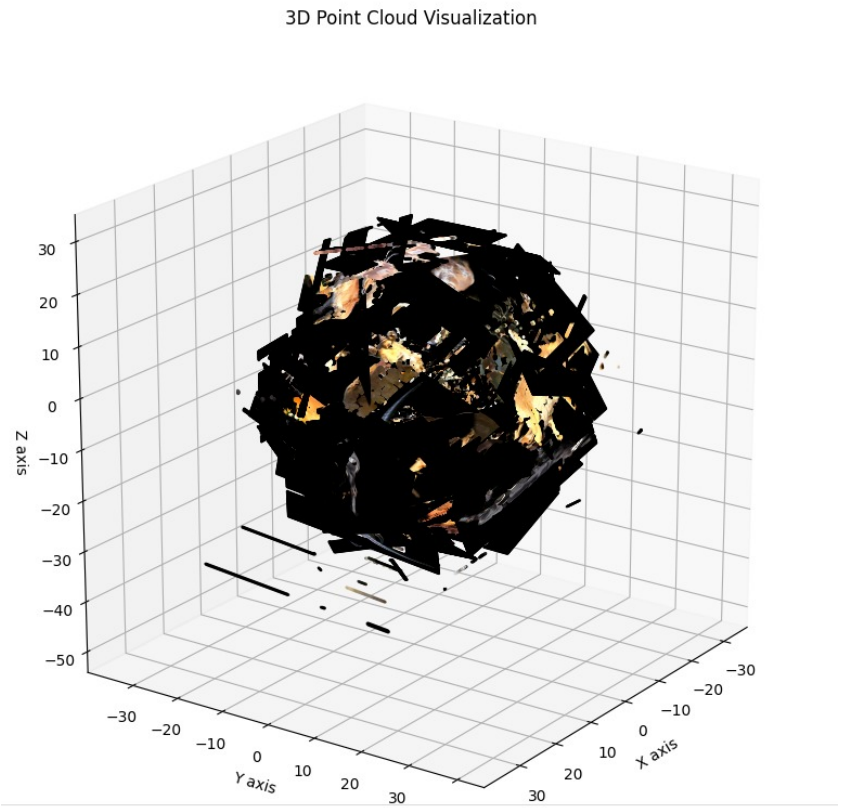
```
]:
```

```python
from rembg import remove
for images, labels in antelope_dataset.take(1):  # Take one batch
    for i in range(min(9, len(images))):
        single_image = images[i].numpy().astype("uint8")  # Extract the first image
        single_label = labels[i].numpy()  # Extract the corresponding label
        # break
        print("1")

        img = Image.fromarray(single_image).convert("RGB")
        img_no_bg = remove(img)
        img_bright = ImageEnhance.Brightness(img_no_bg).enhance(1.2)
        img_contrast = ImageEnhance.Contrast(img_bright).enhance(1.1)
        enhanced_image_no_bg = np.array(img_contrast)

        # Visualize the image with background removed
        plt.figure(figsize=(5, 5))
        plt.imshow(enhanced_image_no_bg)
        plt.title(f"Class: {class_names[single_label]} (Background Removed)")
        plt.axis("off")
        plt.show()
```

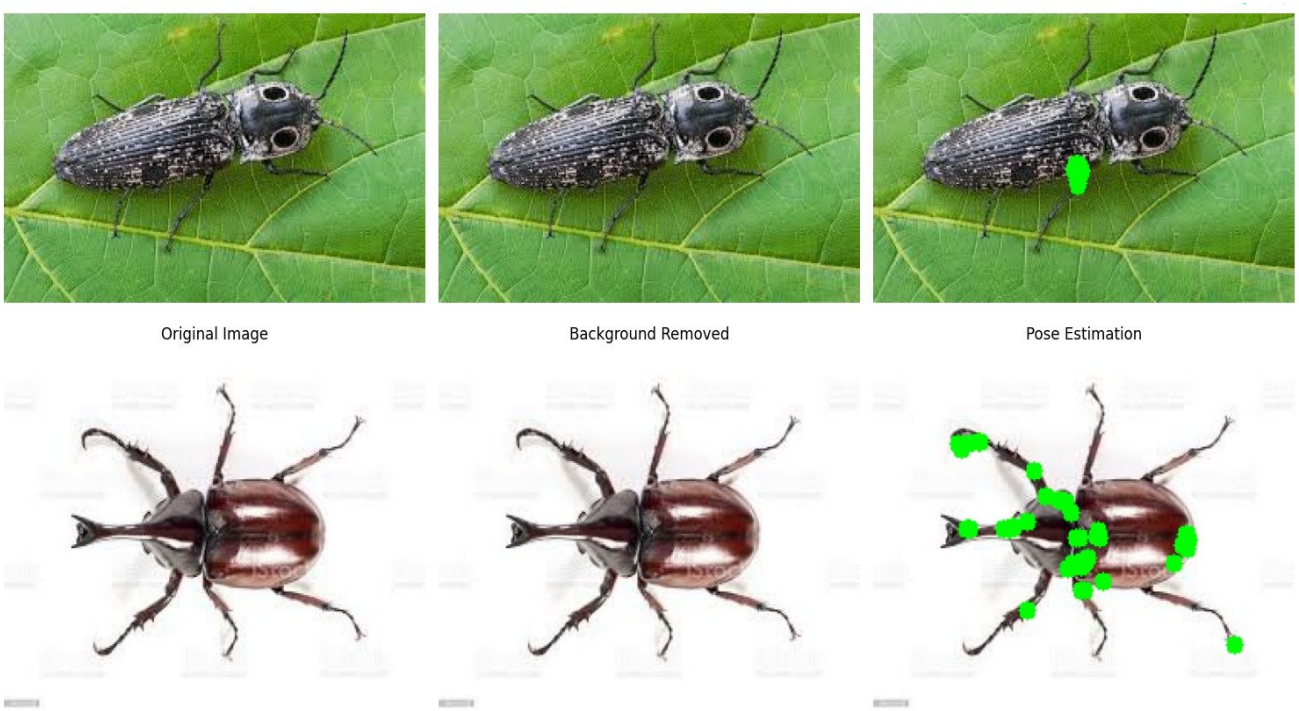
```
Downloading data from 'https://github.com/danielgatis/rembg/releases/download/v0.0.0/u2net.onnx' to file
'/root/.u2net/u2net.onnx'.
1
100%|████████████████████████████████████████| 176M/176M [00:00<00:00, 316GB/s]
```

3D Image Creation

```
Jupyter environment detected. Enabling Open3D WebVisualizer.
[Open3D INFO] WebRTC GUI backend enabled.
[Open3D INFO] WebRTCWindowSystem: HTTP handshake server disabled.
60
Preprocessing images...
Processing image 1/60
Processing image 2/60
Processing image 3/60
Processing image 4/60
Processing image 5/60
Processing image 6/60
Processing image 7/60
Processing image 8/60
Processing image 9/60
Processing image 10/60
Processing image 11/60
Processing image 12/60
Processing image 13/60
Processing image 14/60
Processing image 15/60
Processing image 16/60
Processing image 17/60
Processing image 18/60
Processing image 19/60
Processing image 20/60
Processing image 21/60
Processing image 22/60
Processing image 23/60
Processing image 24/60
Processing image 25/60
Processing image 26/60
Processing image 27/60
Processing image 28/60
Processing image 29/60
Processing image 30/60
Processing image 31/60
Processing image 32/60
Processing image 33/60
Processing image 34/60
Processing image 35/60
```



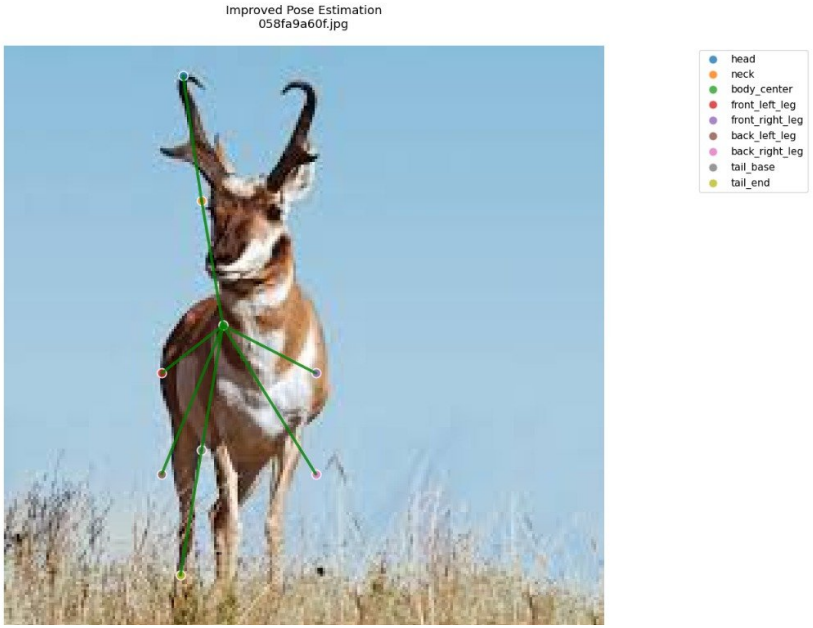
Pose Estimation 1



Pose Estimation 1

Pose Estimation 2

Pose Estimation 3



Fine tuning GAN3 model

```
!git clone https://github.com/NVlabs/stylegan3.git
%cd stylegan3
```

```
Cloning into 'stylegan3'...
remote: Enumerating objects: 212, done.
remote: Counting objects: 100% (163/163), done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 212 (delta 99), reused 90 (delta 90), pack-reused 49 (from 1)
Receiving objects: 100% (212/212), 4.16 MiB | 14.16 MiB/s, done.
Resolving deltas: 100% (108/108), done.
/kaggle/working/stylegan3
```

```
# For StyleGAN2-ADA / StyleGAN3 (latest version):
!python /kaggle/working/stylegan3/dataset_tool.py --source=/kaggle/working/animal_dataset --dest=/kaggle/working/animal_dataset-512x512.zip --resolution=512x512
```

```
100%|██████████| 5000/5000 [01:18<00:00, 63.39it/s]
```

```
!wget https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/stylegan3-r-afhqv2-512x512.pkl -O stylegan3-pretrained-512x512.pkl
!mv stylegan3-pretrained-512x512.pkl /kaggle/working/
!ls -lh /kaggle/working/stylegan3-pretrained-512x512.pkl
```

```
--2025-06-11 08:58:26--  https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/stylegan3-r-afhqv2-512x512.pkl
Resolving api.ngc.nvidia.com (api.ngc.nvidia.com)... 54.68.249.82, 52.25.204.74
Connecting to api.ngc.nvidia.com (api.ngc.nvidia.com)|54.68.249.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://xfiles.ngc.nvidia.com/org/nvidia/team/research/models/stylegan3/versions/1/files/stylegan3-r-afhqv2-512x512.pkl?ssec-algo=AES256&versionId=TsHJSzIRVUwRkXNjCC6J7cCR7nmREwJv&ssec-key=c89IwM5JotbIqGKN1zk00l4m4o%2Fm%2Fp3b05bqa%2FhR59SUXGEwESweunnjTBe7gv9mAo40c%2B0b72i2YheM9YrLxqXtmYxh1g%2By1gWstnYvX5ltE2OJXOnnHgJ2Bsk0Fyi6E6x%2BlG1aRvQQwaQM1ZsXmmFJz8uoCB4k4qYWsapSgS10wuvcOgQfvJNQvgdz1mjyVRm3LIpW7Q1%2BDmCTF4ihvR4Gxjr%2BqZaTyortRRNWrS%2FigA2PfR%2FMarwJxBI7IoTl7MZxCvtjU%2FlqOESqSkaasfN82Y6xxld9Pnfr2gh9I7%2B5ROqGp0dn6Y%2BwQPRzxHBo%2BguInq9IH4tc6Ke%2BC3WQgh2U0S6ojiEvQNXl5B3Db3fTp90T9UQPvX9CHcovTRvBmXWB4lLll%2FM1fVj7GR6Bn5EwwEdCZnF%2FdS7dCUU2RG4cu9pbKME2D%2BU224Zv0ueZRkQppokaTi8e6oXldeHXg%2BuVCPABLiNrC5wj6q0CR9n9MtfF4VSPETjIxDSMkYn6&Signature=WnysOGb1ZZyjQEps~EsS~I~yanJNVlPa7y67EnHmRxKo7bPx0lwL~o97OCqMDtugeWNXSrwaj67Bkd6Ek7gHSkyJa3rT9mXnzaIa3k2GZ5cDXRDGFlo6VF18Qn2jjEy4f7GO4e6X4aQP1Fen7CZK8xCEkn4fRJfXqDhPK1vuyIY92eq~XhHiI2h5C1xsALICF~tl8Jo80YbQSCG18PXV-PJzZw6e8rI9mEuLmVI7D~133xi-S1IxJfViRbae~9fkFEQsviu6~NpC-78DEauBBg4Al5CDS38Pxxz2wG4u69wj-P769DIXMtisFWBnb16NUu~35GOCZTINg4x87P7q6w__&kid=bXJrLWU3OGM1M2FhZjE4YzRiNmJiNjlkYmRhZjcxNjA3YWEw&Expires=1749718706&Key-Pair-Id=KCX06E8E9L60W&ssec-enabled=true [following]
--2025-06-11 08:58:26--  https://xfiles.ngc.nvidia.com/org/nvidia/team/research/models/stylegan3/versions/1/files/stylegan3-r-afhqv2-512x512.pkl?ssec-algo=AES256&versionId=TsHJSzIRVUwRkXNjCC6J7cCR7nmREwJv&ssec-key=c89IwM5JotbIqGKN1zk00l4m4o%2Fm%2Fp3b05bqa%2FhR59SUXGEwESweunnjTBe7gv9mAo40c%2B0b72i2YheM9YrLxqXtmYxh1g%2By1gWstnYvX5ltE2OJXOnnHgJ2Bsk0Fyi6E6x%2BlG1aRvQQwaQM1ZsXmmFJz8uoCB4k4qYWsapSgS10wuvcOgQfvJNQvgdz1mjyVRm3LIpW7Q1%2BDmCTF4ihvR4Gxjr%2BqZaTyortRRNWrS%2FigA2PfR%2FMarwJxBI7IoTl7MZxCvtjU%2FlqOESqSkaasfN82Y6xxld9Pnfr2gh9I7%2B5ROqGp0dn6Y%2BwQPRzxHBo%2BguInq9IH4tc6Ke%2BC3WQgh2U0S6ojiEvQNX15B3Db3fTp90T9UQPvX9CHcovTRvBmXWB4lLll%2FM1fVj7GR6Bn5EwwEdCZnF%2FdS7dCUU2RG4cu9pbKME2D%2BU224Zv0ueZRkQppokaTi8e6oXldeHXg%2BuVCPABLiNrC5wj6q0CR9n9MtfF4VSPETjIxDSMkYn6&Signature=WnysOGb1ZZyjQEps~EsS~I~yanJNVlPa7y67EnHmRxKo7bPx0lwL~o97OCqMDtugeWNXSrwaj67Bkd6Ek7gHSkyJa3rT9mXnzaIa3k2GZ5cDXRDGFlo6VF18Qn2jjEy4f7GO4e6X4aQP1Fen7CZK8xCEkn4fRJfXqDhPK1vuyIY92eq~XhHiI2h5C1xsALICF~tl8Jo80YbQSCG18PXV-PJzZw6e8rI9mEuLmVI7D~133xi-S1IxJfViRbae~9fkFEQsviu6~NpC-78DEauBBg4Al5CDS38Pxxz2wG4u69wj-P769DIXMtisFWBnb16NUu~35GOCZTINg4x87P7q6w__&kid=bXJrLWU3OGM1M2FhZjE4YzRiNmJiNjlkYmRhZjcxNjA3YWEw&Expires=1749718706&Key-Pair-Id=KCX06E8E9L60W&ssec-enabled=true
Resolving xfiles.ngc.nvidia.com (xfiles.ngc.nvidia.com)... 13.35.166.57, 13.35.166.124, 13.35.166.27, ...
Connecting to xfiles.ngc.nvidia.com (xfiles.ngc.nvidia.com)|13.35.166.57|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 249525556 (238M) [application/x-zerosize]
Saving to: ‘stylegan3-pretrained-512x512.pkl’
```

Fine tuning GAN3 model

```
!python /kaggle/working/stylegan3/train.py \
  --outdir=/kaggle/working/animal-stylegan-results-512x512 \
  --cfg=stylegan3-r \
  --data=/kaggle/working/animal_dataset-512x512.zip \
  --gpus=1 \
  --batch=8 \
  --mirror=1 \
  --gamma=8 \
  --snap=1 \
  --resume=/kaggle/working/stylegan3-pretrained-512x512.pkl \
  --kimg=5000 \
  --metrics=none \
  --glr=0.002 \
  --dlr=0.002 \
  --aug=ada \
  --freezed=0 \
  --workers=4 \
  --target=0.6
```

Fine tuning GAN3 model (Generator)

Generator	Parameters	Buffers	Output shape	Datatype
mapping.fc0	262656	-	[8, 512]	float32
mapping.fc1	262656	-	[8, 512]	float32
mapping	-	512	[8, 16, 512]	float32
synthesis.input.affine	2052	-	[8, 4]	float32
synthesis.input	1048576	3081	[8, 1024, 36, 36]	float32
synthesis.L0_36_1024.affine	525312	-	[8, 1024]	float32
synthesis.L0_36_1024	1049600	157	[8, 1024, 36, 36]	float32
synthesis.L1_36_1024.affine	525312	-	[8, 1024]	float32
synthesis.L1_36_1024	1049600	157	[8, 1024, 36, 36]	float32
synthesis.L2_52_1024.affine	525312	-	[8, 1024]	float32
synthesis.L2_52_1024	1049600	169	[8, 1024, 52, 52]	float32
synthesis.L3_52_1024.affine	525312	-	[8, 1024]	float32
synthesis.L3_52_1024	1049600	157	[8, 1024, 52, 52]	float32
synthesis.L4_84_1024.affine	525312	-	[8, 1024]	float32
synthesis.L4_84_1024	1049600	169	[8, 1024, 84, 84]	float16
synthesis.L5_84_1024.affine	525312	-	[8, 1024]	float32
synthesis.L5_84_1024	1049600	157	[8, 1024, 84, 84]	float16
synthesis.L6_148_1024.affine	525312	-	[8, 1024]	float32
synthesis.L6_148_1024	1049600	169	[8, 1024, 148, 148]	float16
synthesis.L7_148_967.affine	525312	-	[8, 1024]	float32
synthesis.L7_148_967	991175	157	[8, 967, 148, 148]	float16
synthesis.L8_276_645.affine	496071	-	[8, 967]	float32
synthesis.L8_276_645	624360	169	[8, 645, 276, 276]	float16
synthesis.L9_276_431.affine	330885	-	[8, 645]	float32
synthesis.L9_276_431	278426	157	[8, 431, 276, 276]	float16
synthesis.L10_532_287.affine	221103	-	[8, 431]	float32
synthesis.L10_532_287	123984	169	[8, 287, 532, 532]	float16
synthesis.L11_532_192.affine	147231	-	[8, 287]	float32
synthesis.L11_532_192	55296	157	[8, 192, 532, 532]	float16
synthesis.L12_532_128.affine	98496	-	[8, 192]	float32
synthesis.L12_532_128	24704	25	[8, 128, 532, 532]	float16
synthesis.L13_512_128.affine	65664	-	[8, 128]	float32
synthesis.L13_512_128	16512	25	[8, 128, 512, 512]	float16
synthesis.L14_512_3.affine	65664	-	[8, 128]	float32
synthesis.L14_512_3	387	1	[8, 3, 512, 512]	float16
synthesis	-	-	[8, 3, 512, 512]	float32
---	---	---	---	---
Total	16665594	5588	-	-

Fine tuning GAN3 model (Discriminator)

Discriminator	Parameters	Buffers	Output shape	Datatype
b512.fromrgb	256	16	[8, 64, 512, 512]	float16
b512.skip	8192	16	[8, 128, 256, 256]	float16
b512.conv0	36928	16	[8, 64, 512, 512]	float16
b512.conv1	73856	16	[8, 128, 256, 256]	float16
b512	-	16	[8, 128, 256, 256]	float16
b256.skip	32768	16	[8, 256, 128, 128]	float16
b256.conv0	147584	16	[8, 128, 256, 256]	float16
b256.conv1	295168	16	[8, 256, 128, 128]	float16
b256	-	16	[8, 256, 128, 128]	float16
b128.skip	131072	16	[8, 512, 64, 64]	float16
b128.conv0	590080	16	[8, 256, 128, 128]	float16
b128.conv1	1180160	16	[8, 512, 64, 64]	float16
b128	-	16	[8, 512, 64, 64]	float16
b64.skip	262144	16	[8, 512, 32, 32]	float16
b64.conv0	2359808	16	[8, 512, 64, 64]	float16
b64.conv1	2359808	16	[8, 512, 32, 32]	float16
b64	-	16	[8, 512, 32, 32]	float16
b32.skip	262144	16	[8, 512, 16, 16]	float32
b32.conv0	2359808	16	[8, 512, 32, 32]	float32
b32.conv1	2359808	16	[8, 512, 16, 16]	float32
b32	-	16	[8, 512, 16, 16]	float32
b16.skip	262144	16	[8, 512, 8, 8]	float32
b16.conv0	2359808	16	[8, 512, 16, 16]	float32
b16.conv1	2359808	16	[8, 512, 8, 8]	float32
b16	-	16	[8, 512, 8, 8]	float32
b8.skip	262144	16	[8, 512, 4, 4]	float32
b8.conv0	2359808	16	[8, 512, 8, 8]	float32
b8.conv1	2359808	16	[8, 512, 4, 4]	float32
b8	-	16	[8, 512, 4, 4]	float32
b4.mbstd	-	-	[8, 513, 4, 4]	float32
b4.conv	2364416	16	[8, 512, 4, 4]	float32
b4.fc	4194816	-	[8, 512]	float32
b4.out	513	-	[8, 1]	float32
---	---	---	---	---
Total	28982849	480	-	-

RESULT

Image creation (without pose estimation)

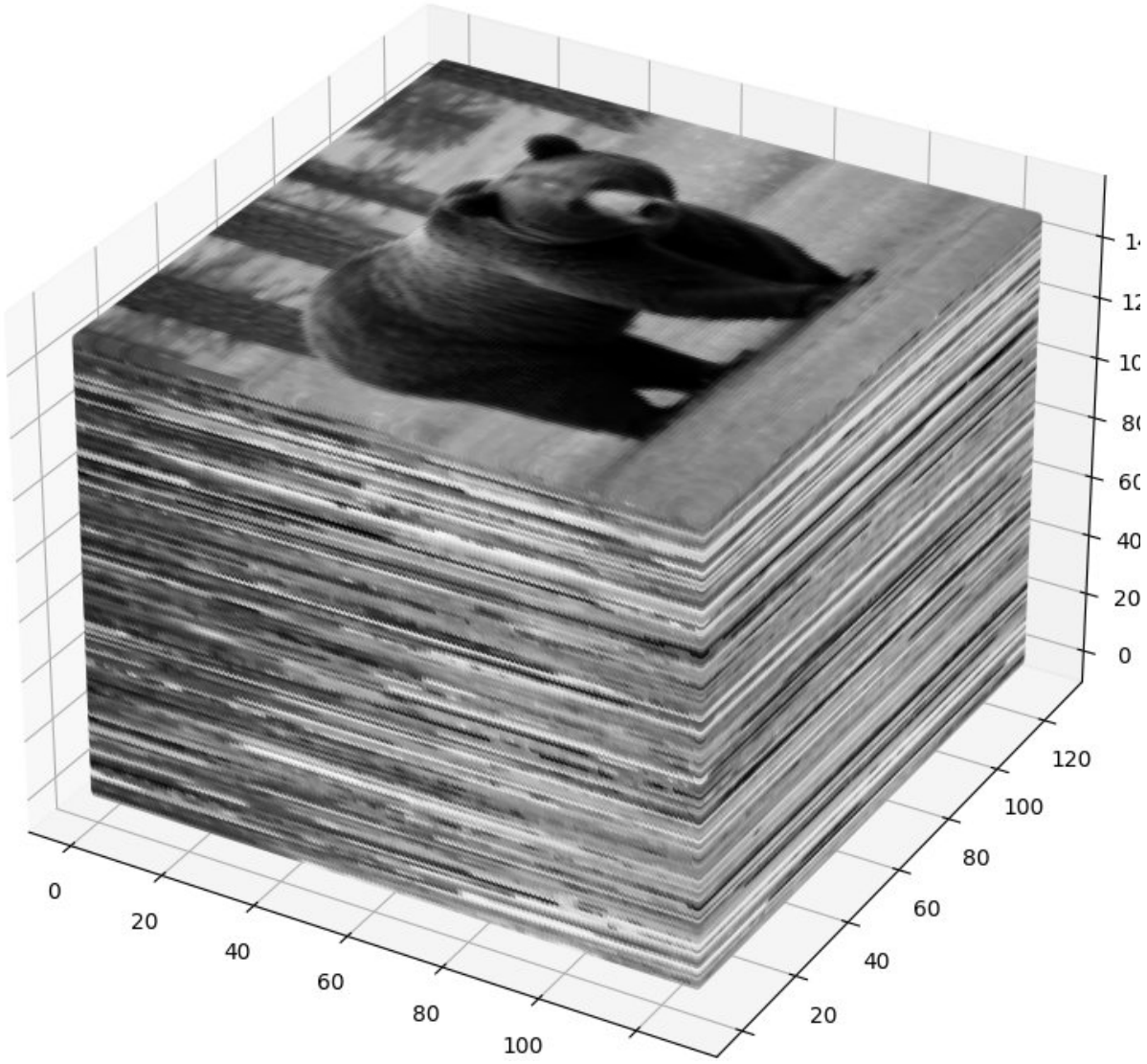
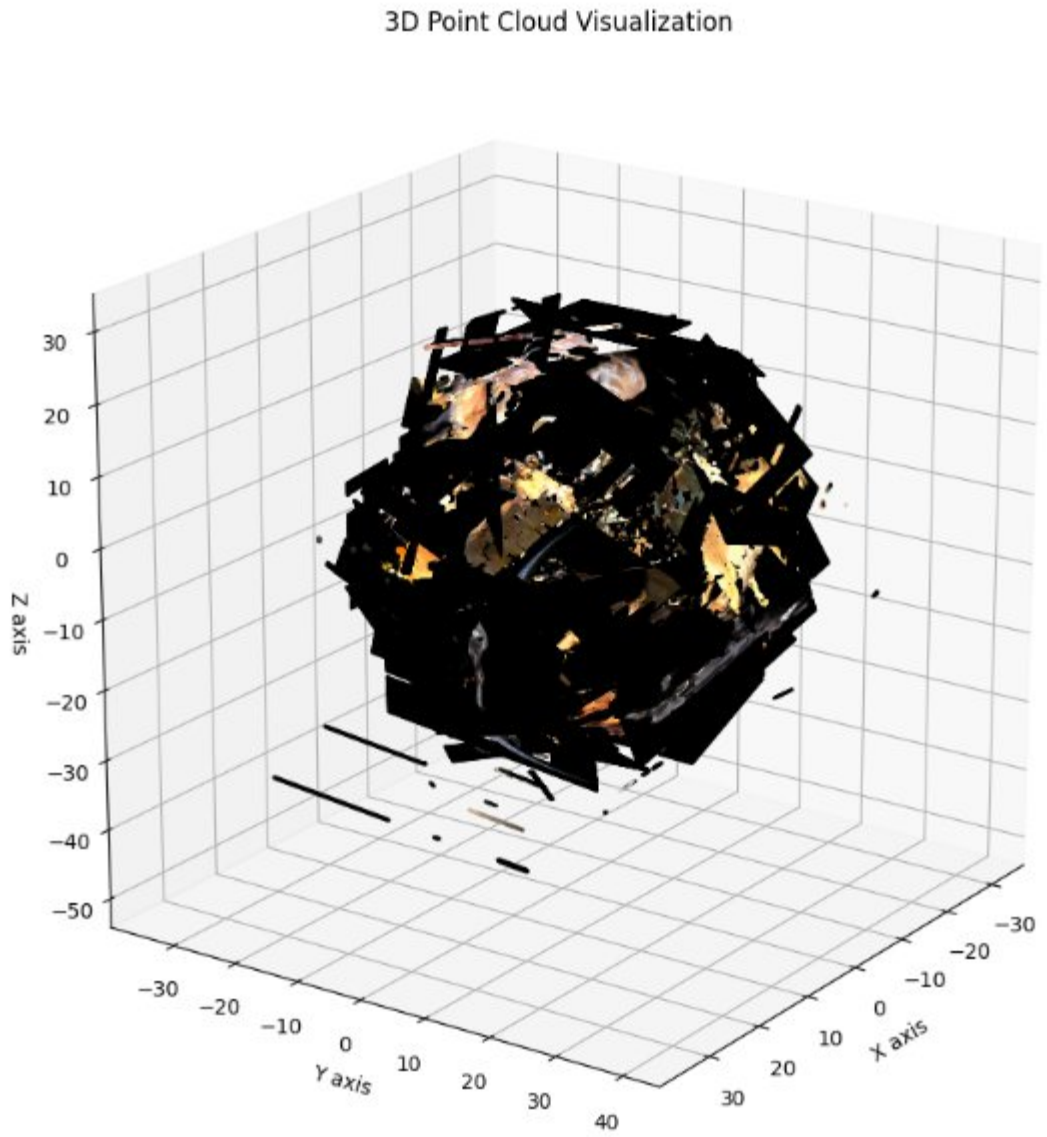
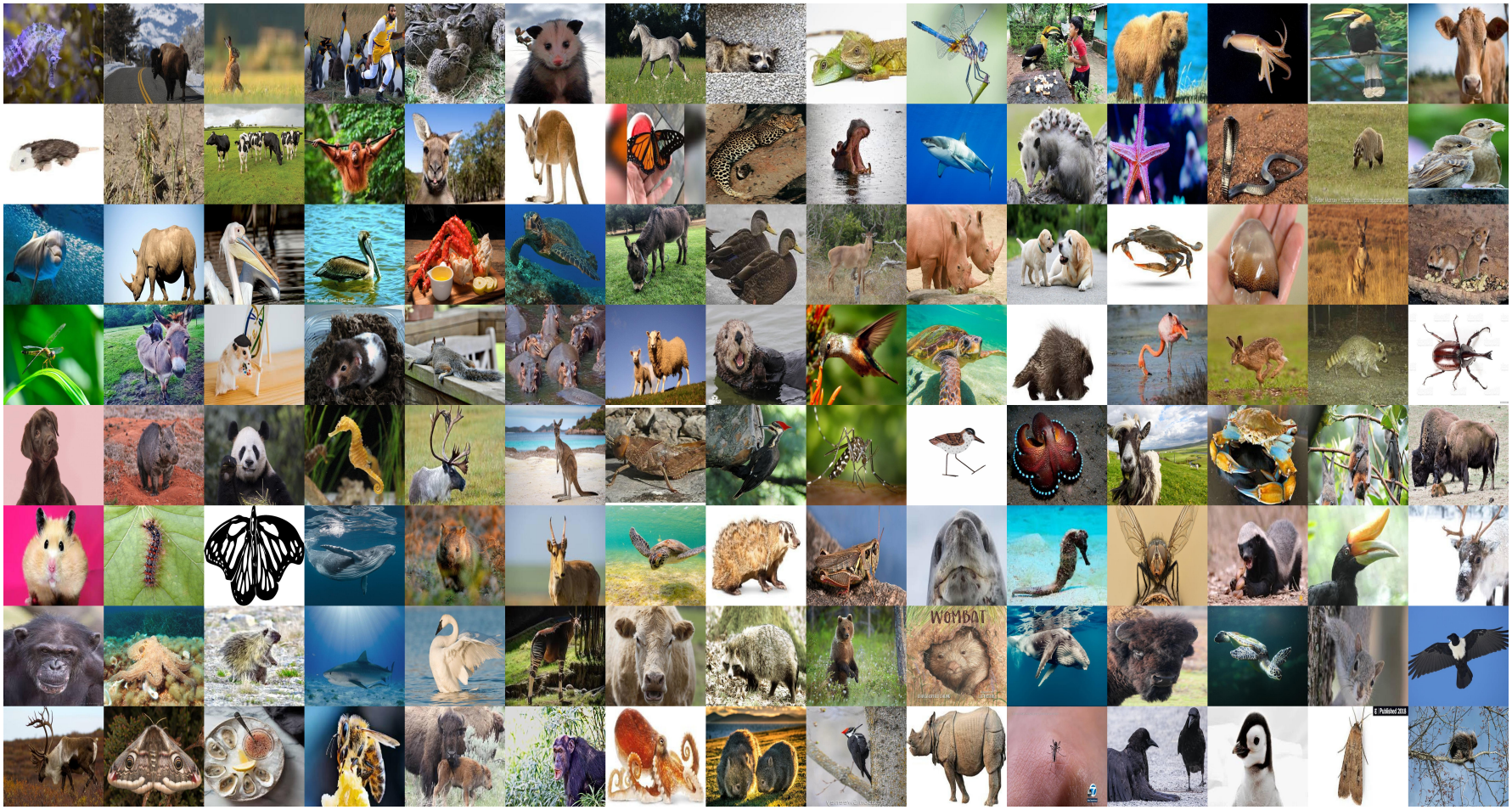


Image creation using MVS



StyleGAN3 Fine tuning Fake Images

StyleGAN3 Fine tuning Real Images



StyleGAN3 Fine tuning Generated Images

Image 1

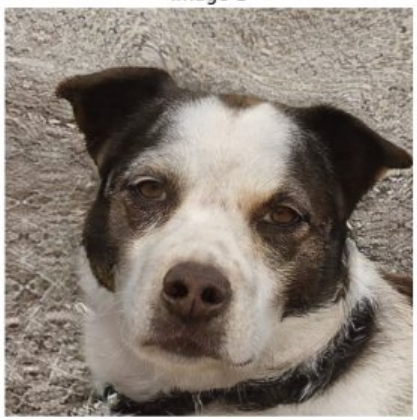


Image 2

Image 3

Image 4

Image 5

Image 6

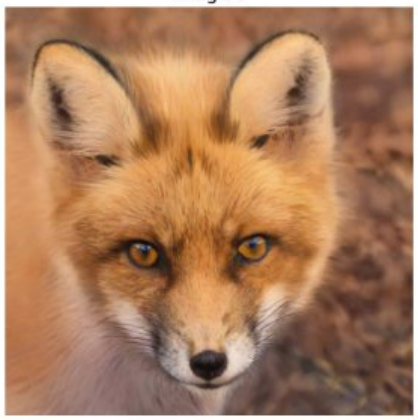


Image 7

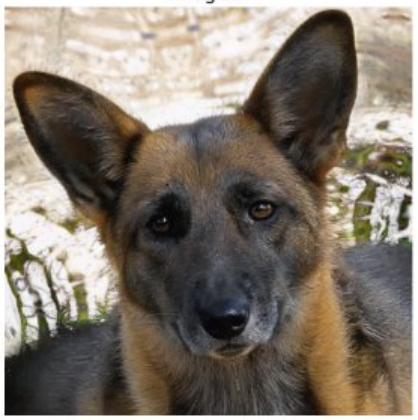
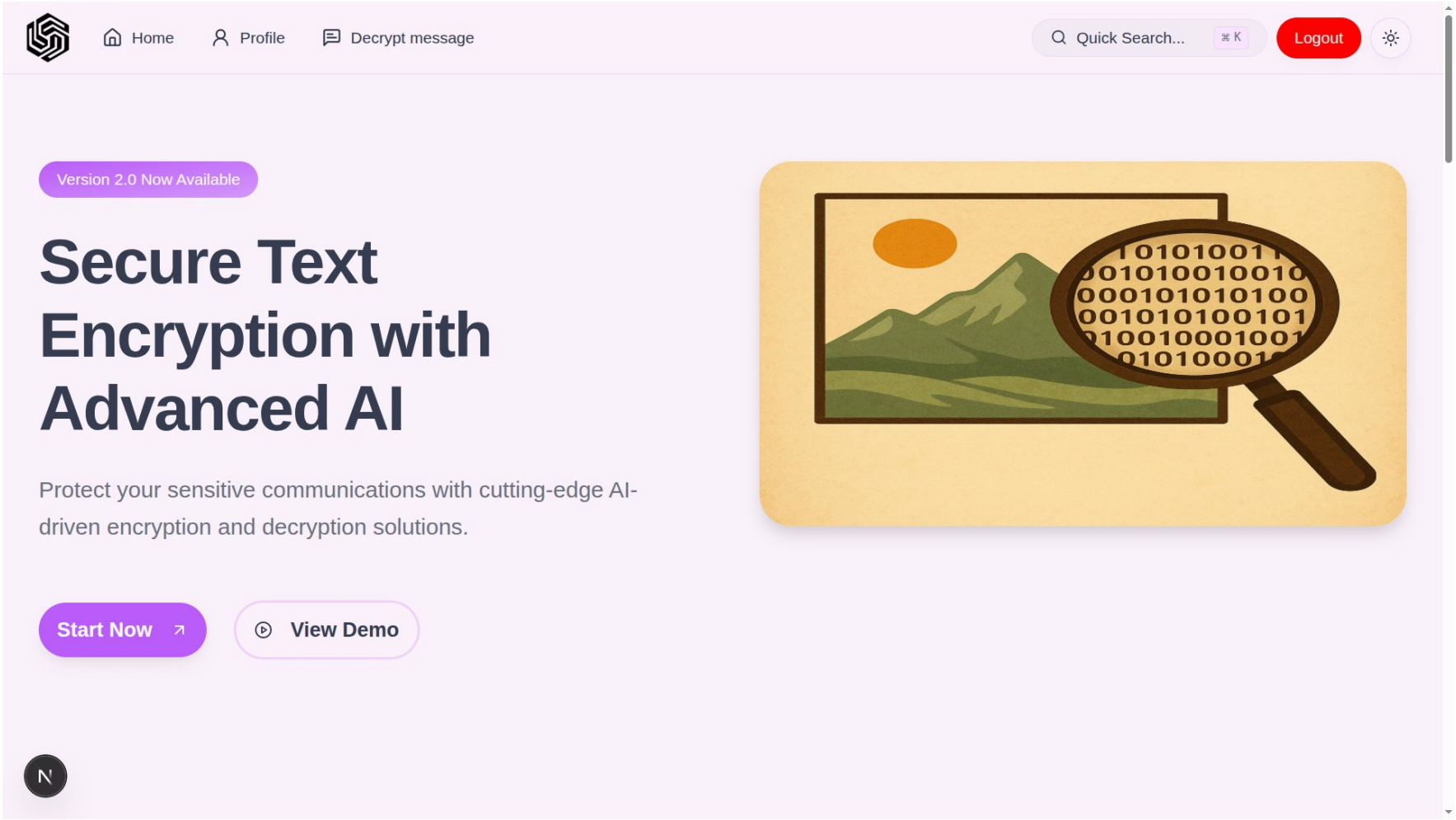


Image 8

Website



[Website link]

Future Work

- Web Scraping
- Improve accuracy
- Model Deployment

Bibliography

- DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation (https://arxiv.org/abs/2208.12242)
- DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior (https://arxiv.org/abs/2310.16818)
- Hiding Images within Images (https://ieeexplore.ieee.org/document/8654686)
- Deep Learning-based Text-in-Image Watermarking (https://arxiv.org/abs/2404.13134)

Bibliography

- Tenorflow (https://www.tensorflow.org/)
- Numpy (https://numpy.org/)
- Pandas (https://pandas.pydata.org/)
- Matplotlib (https://matplotlib.org/)
- Scikit-learn (https://scikit-learn.org/stable/)
- Cryptography (https://pypi.org/project/cryptography/)
- Streamlit (https://streamlit.io/)
- Keras (https://keras.io/)

THANK YOU ☺