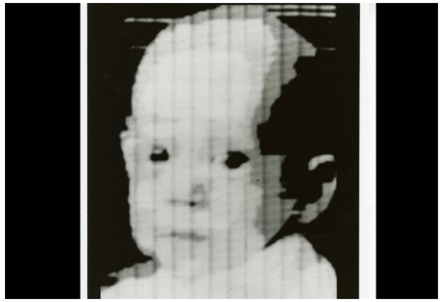
Vision AI Demystified

Ayesha Asif, Developer Marketing | GTC March 2024

Agenda

- The History of Computer Vision
- From CNNs to Vision Transformers
- Generative AI: GANs and Diffusion Models
- 3D Reconstruction with NeRFs and Gaussian Splatting
- Emerging Computer Vision Frontiers

A Brief History of Computer Vision



1957

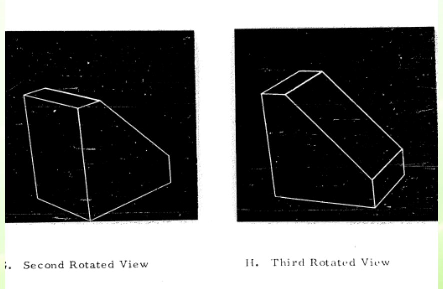
The first digitally scanned photos

Russell Kirsch

1963

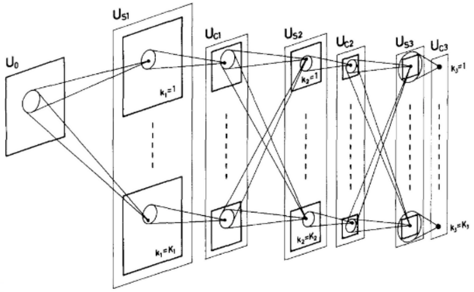
Machine Perception of Three-Dimensional Solids

Lawrence Roberts



. Second Rotated View

H. Third Rotated View



1970

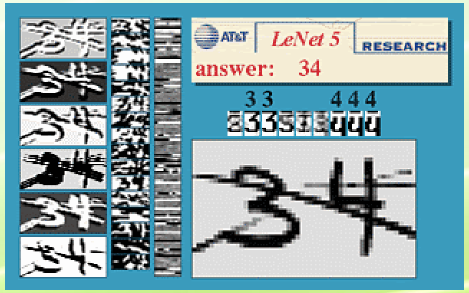
Neocognitron

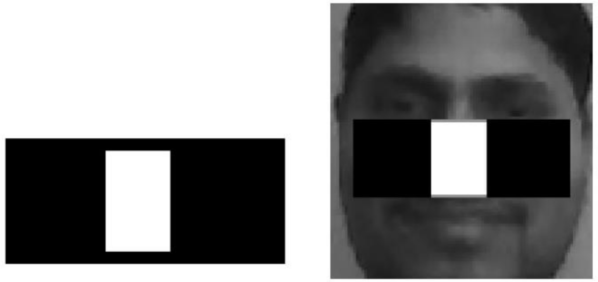
Kunihiko Fukushima

1989

LeNet-5

Yann LeCun





2001

The Viola-Jones algorithm

Paul Viola and Michael Jones

2009

ImageNet Dataset

Fei-Fei Li

2012-2024

Deep Learning

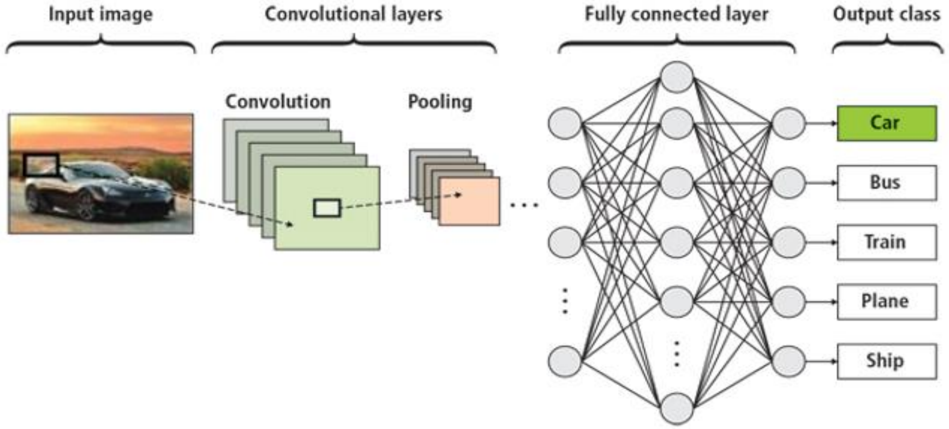
Vision Transformers

Diffusion Models

Generative AI

3D Reconstruction

Convolutional Neural Networks

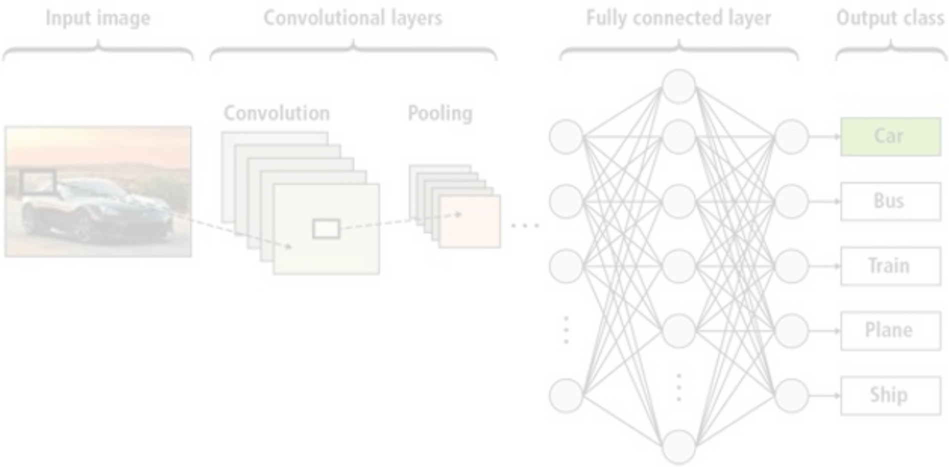


CNN Architecture

Hierarchical feature extraction and classification

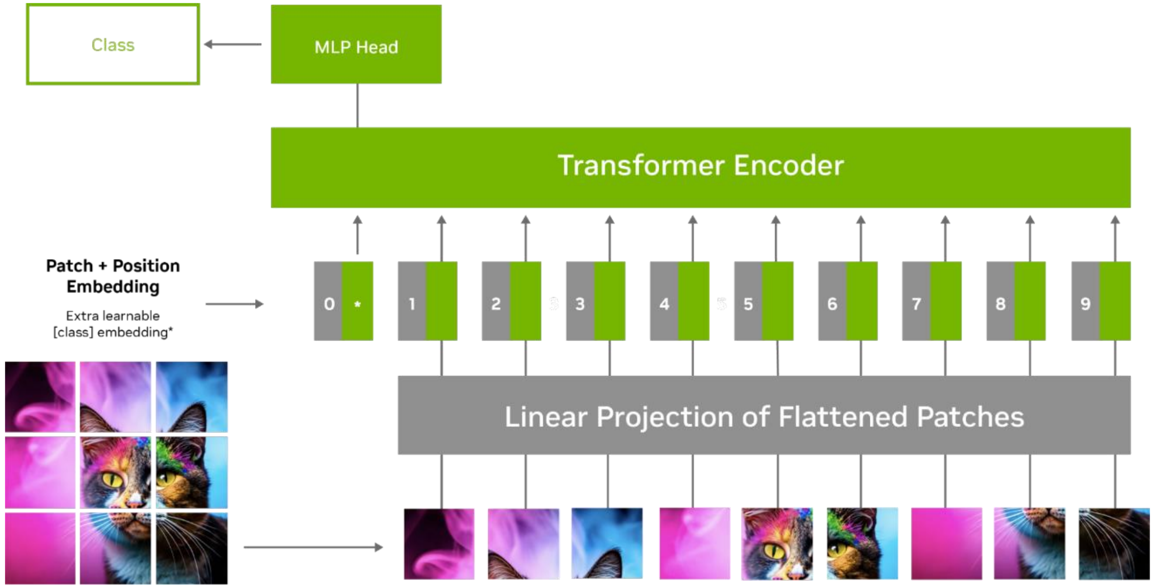
Convolutional Neural Networks and Transformers

What are Vision Transformers (ViTs)?



CNN Architecture

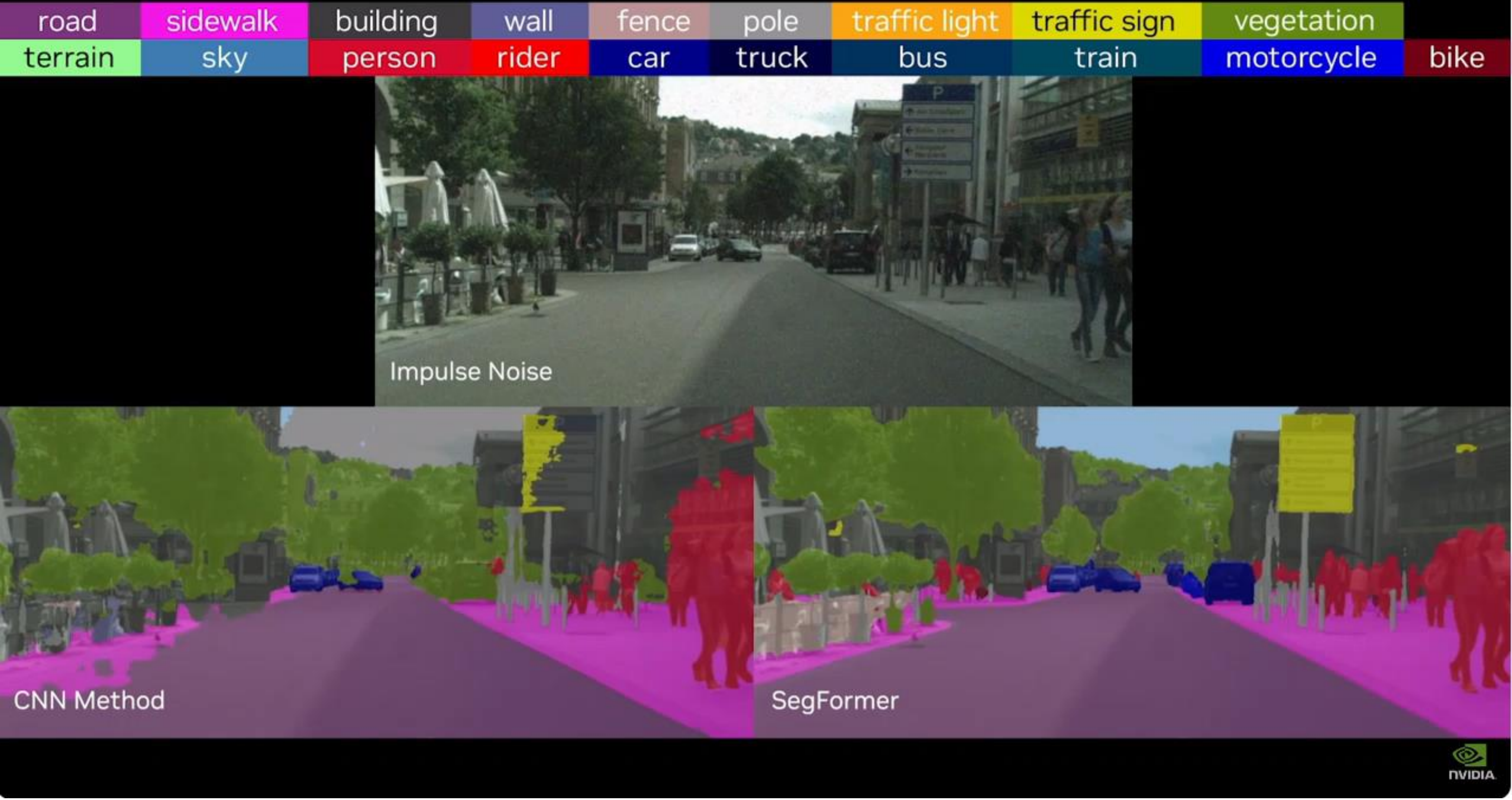
Hierarchical feature extraction and classification



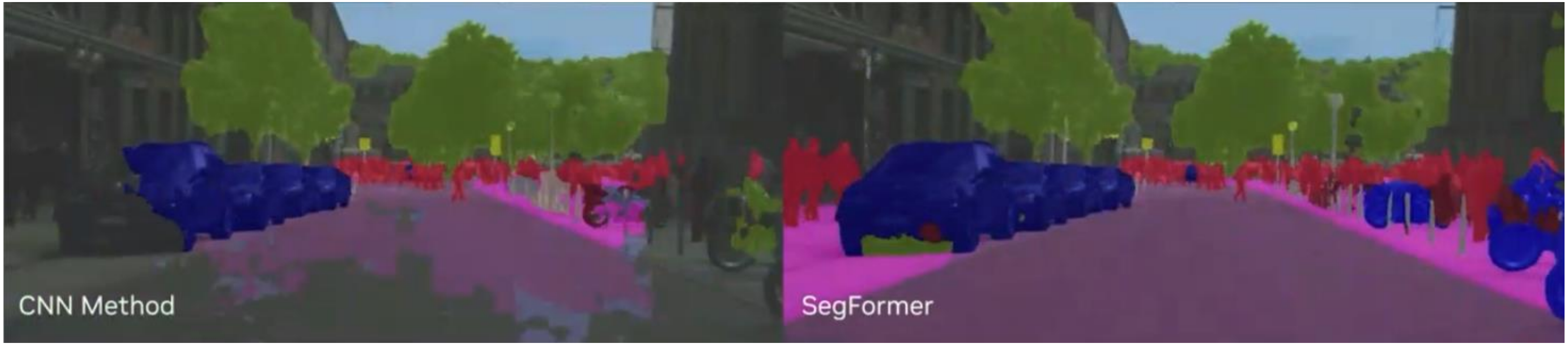
Class
MLP Head
Transformer Encoder
Patch + Position Embedding
Extra learnable [class] embedding*
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Vision Transformer Architecture

Attention-based patch embedding and classification

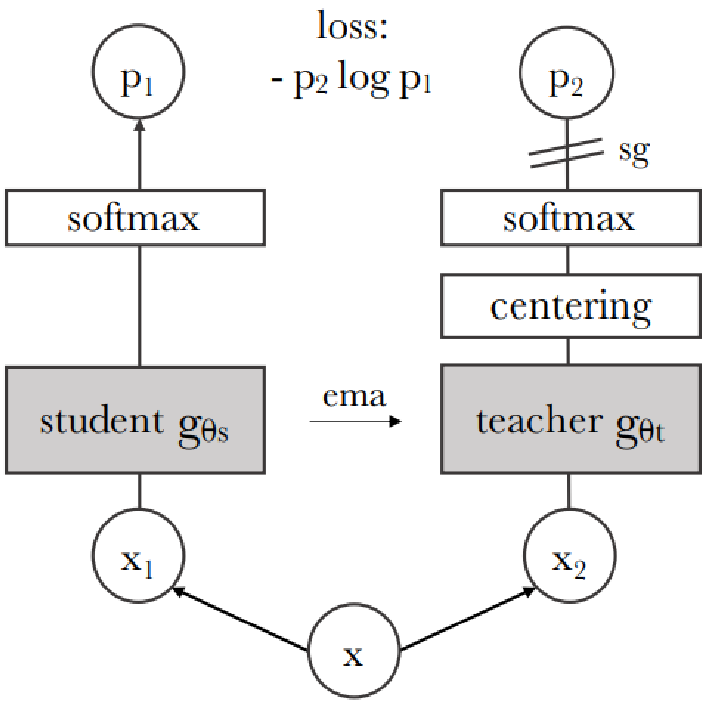

road
sidewalk
building
wall
fence
pole
traffic light
traffic sign
vegetation
terrain
sky
person
rider
car
truck
bus
train
motorcycle
bike
Impulse Noise
CNN Method
SegFormer
NVIDIA

Comparison of CNN and ViT Methods



Vision Transformers

self-**DI**stillation with **NO** labels (DINO)

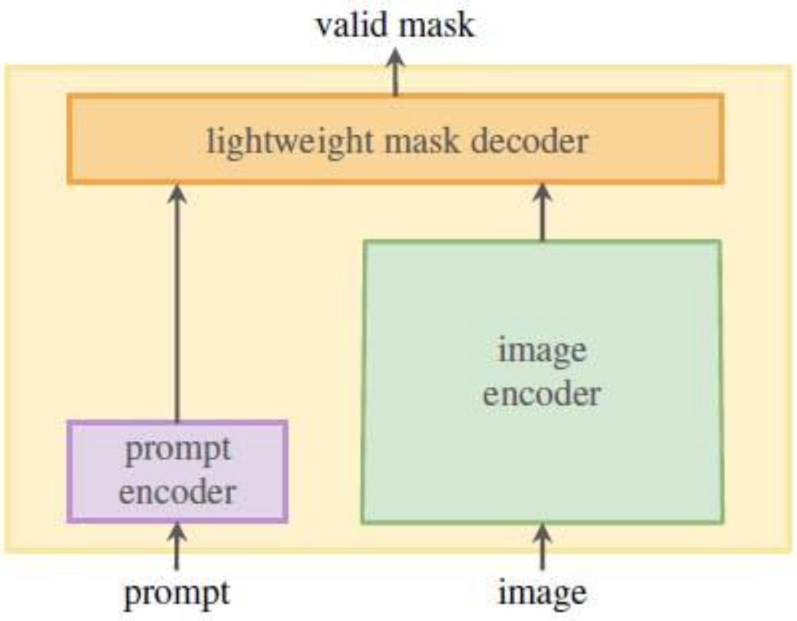


DINO

- Self-supervised learning method
- Same architecture for teacher and student networks
- Student network – matches the output distribution of a teacher network
- Teacher network – learns from the student parameters through exponential moving average (EMA)

Vision Transformers

Segment Anything Model (SAM)

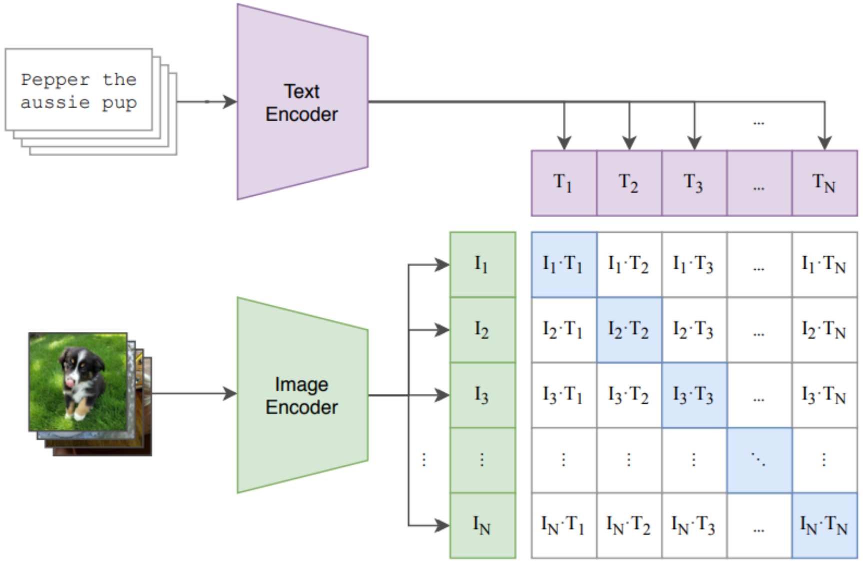


SAM

- SAM was trained on an unprecedented dataset of 11 million images and 1.1 billion segmentation masks
- Strong zero-shot performance
- Promptable segmentation

Vision Transformers

Contrastive Language-Image Pretraining (CLIP)



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

CLIP

- Trained on 400 million (image, text) pairs
- Jointly trains an image and text encoder to predict the correct pairings of a (image, text) examples
- Projected in a shared embedding space to understand the text/image relationships
- Applications – image search, image captioning

Getting Started With Vision Transformers

PEDESTRIAN
PEDESTRIAN
PEDESTRIAN
BIKE
BIKE
BIKE
CAR
CAR
CAR
CAR
CAR
CAR
CAR
CAR

DINO Notebook

#Installing the TAO launcher

```
!pip3 install nvidia-pyindex
!pip3 install nvidia-tao
```

#Pull pre-trained model from NGC

```
!ngc registry model download-version
nvidia/tao/pretrained_dino_nvimagenet:fan_small_hybri
d_nvimagenet --dest $LOCAL_PROJECT_DIR/dino/
```

#Run Inference of a pre-trained model

```
tao model dino inference -e /path/to/spec.yaml -r
/path/to/results/
inference.checkpoint=/path/to/model.pth
```

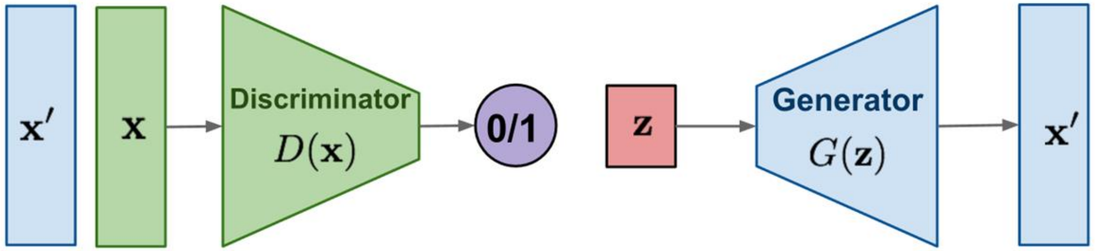
Finetuning on Custom Datasets:

- Tao model dino train
- Tao model dino evaluate

Generative Models

Generative Models

Generative Adversarial Networks (GANs)



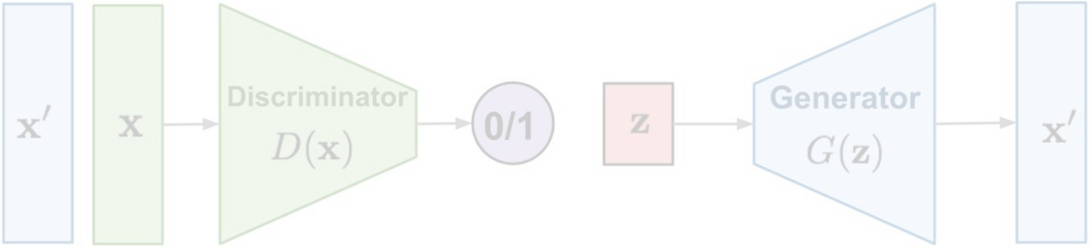
GAN Architecture

Adversarial training

- The generator creates synthetic data, and the discriminator tries to distinguish the generated data from real data
- Training instability and mode collapse issues

Generative Models

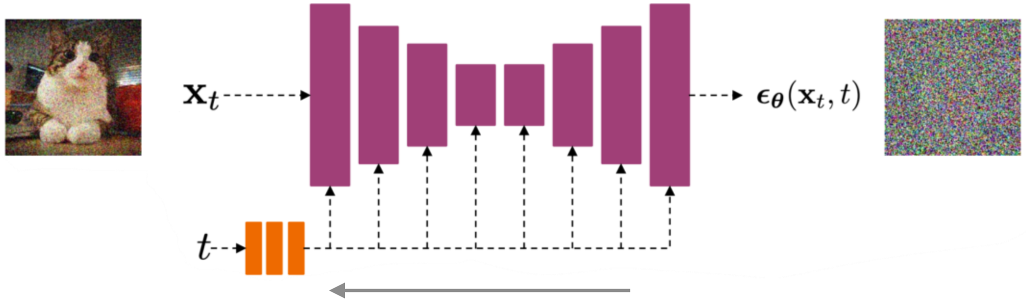
From GANs to Diffusion Models



GAN Architecture

Adversarial training

- The generator creates synthetic data, and the discriminator tries to distinguish the generated data from real data
- Training instability and mode collapse issues



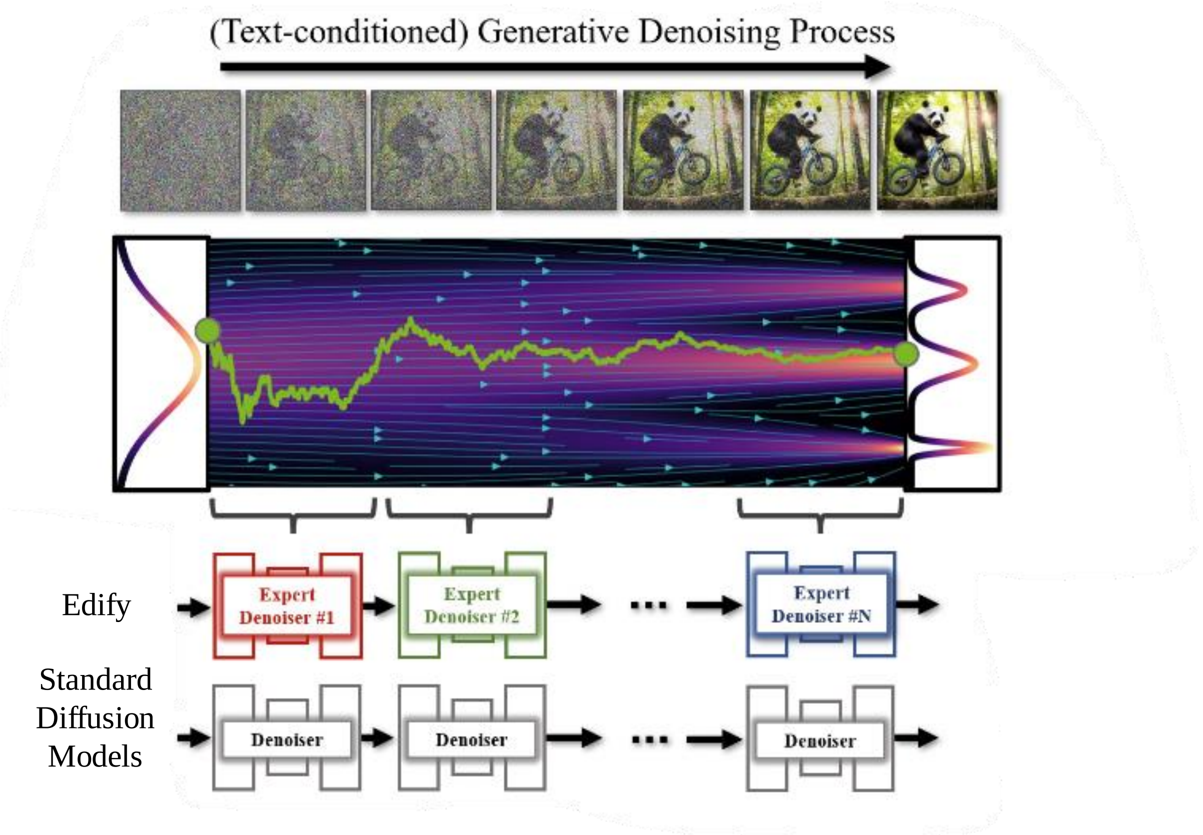
Diffusion Model Architecture

Gradual addition and removal of noise

- Learn to reverse a diffusion process by iteratively removing noise from data
- Can generate high-quality samples across various domains

Generative Models

NVIDIA Edify



- Instead of using a single denoiser, Edify trains an ensemble of expert denoising networks
- Edify also combines text embeddings from both T5 and CLIP models instead of using just one text encoder

Credit : Getty Images AI Generator

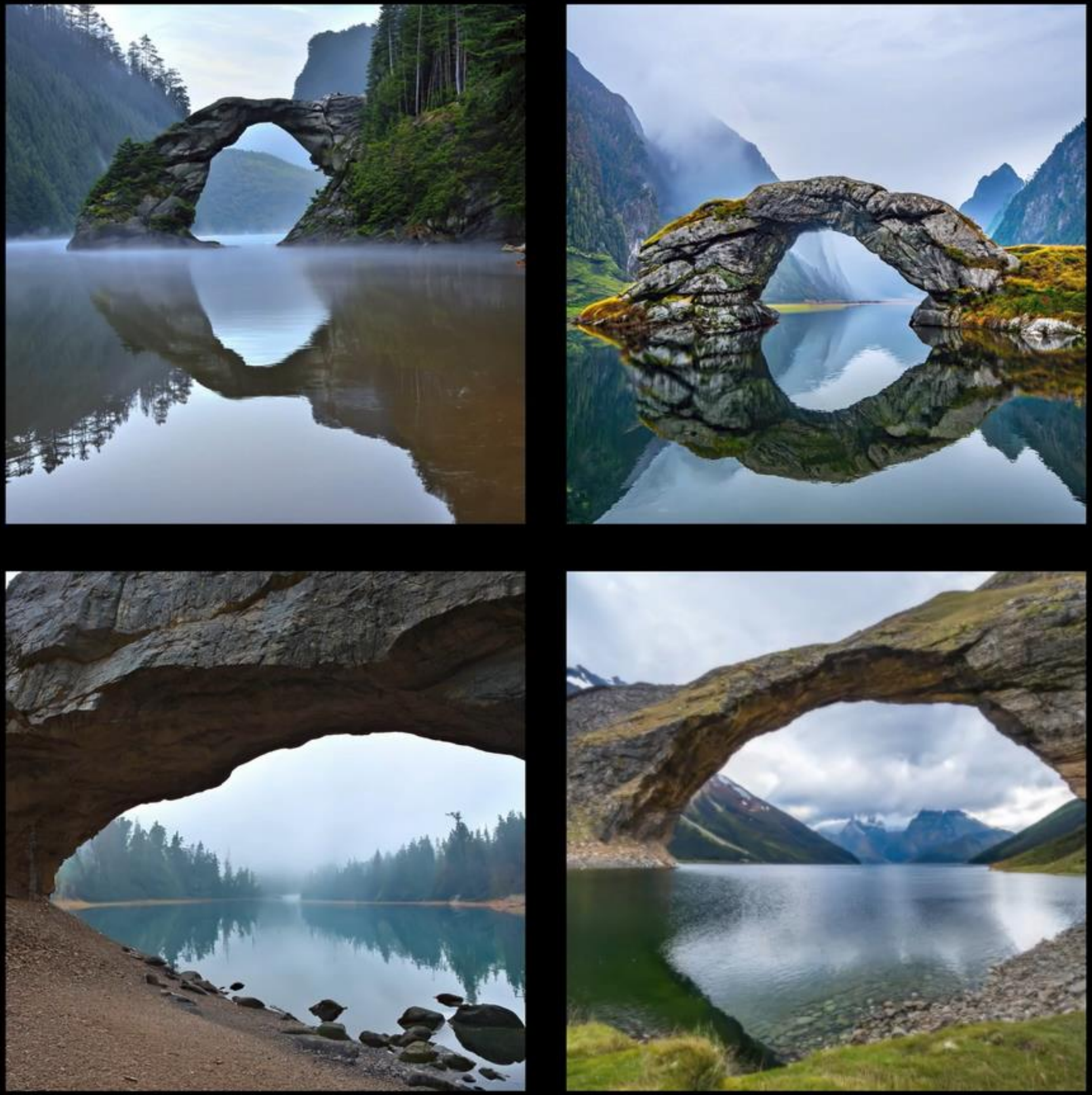
Wildlife photography of wolf, beautiful eyes, golden hour

Generative AI by gettyimages

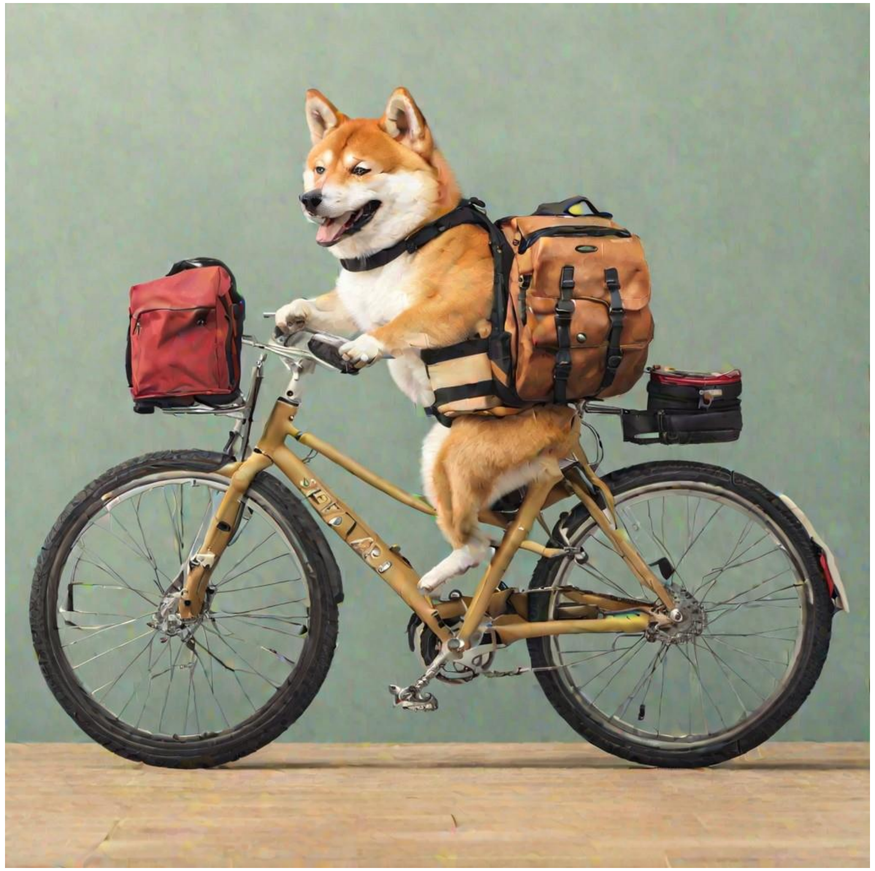
Built using NVIDIA Picasso

- State-of-the-art NVIDIA Edify model architecture for 4K generative photography
- Commercially safe—trained exclusively on Getty Images licensed data; uncapped indemnification
- Access via Web or iStock.com
- Advanced visual editing APIs for adding subjects, expanding images or replacing specific element

nature photography of a rock arch, a mountain lake with a forest and fog in the background, overcast, majestic



Getting Started With Diffusion Models

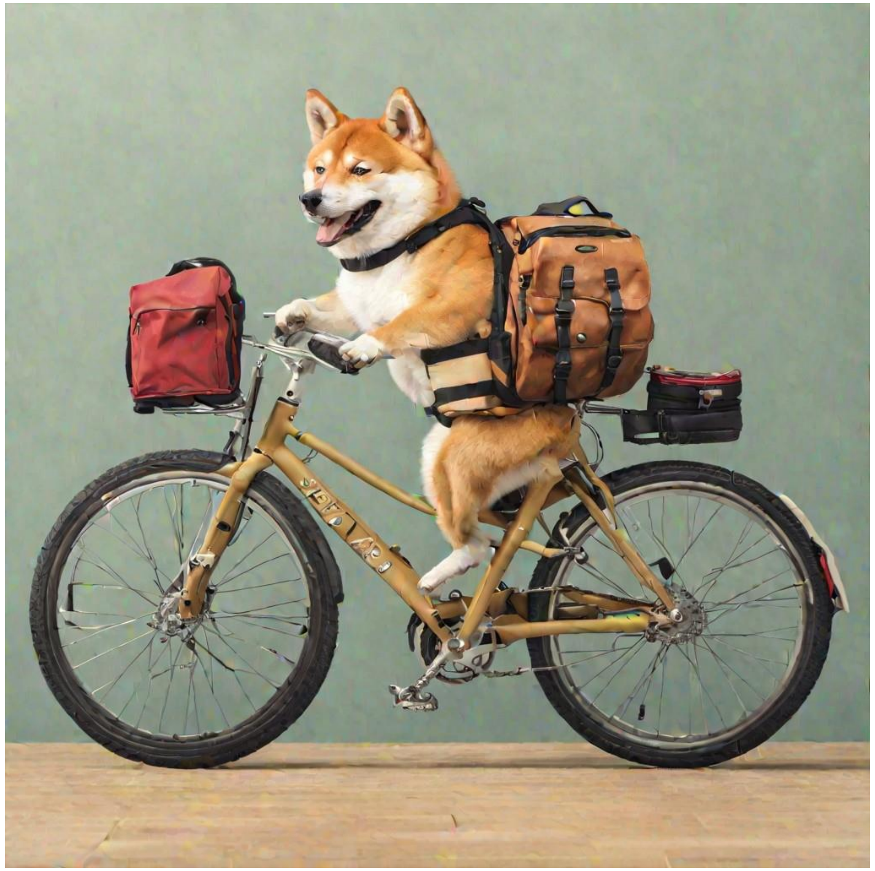


A photo of a Shiba Inu dog with a backpack riding a bike

```
import requests

# Call model from NVIDIA NGC
invoke_url = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/functions/89848fb8-549f-41bb-88cb-95d6597044a4"
fetch_url_format = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/status/"

# Provide API key here
headers = {
"Authorization": "Bearer $API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC",
"Accept": "application/json",
}

# Provide prompts and parameters
payload = {
"prompt": "A photo of a Shiba Inu dog with a backpack riding a bike",
"negative_prompt": "beach",
"inference_steps": 25
}

# re-use connections
session = requests.Session()

response = session.post(invoke_url, headers=headers, json=payload)

while response.status_code == 202:
        response = session.get(fetch_url_format + response.headers.get("NVCF-REQID”), headers=headers)

response.raise_for_status()
response_body = response.json()
print(response_body)
```

catalog.ngc.nvidia.com/orgs/nvidia/teams/ai-foundation/models/sdxl

Getting Started With Diffusion Models

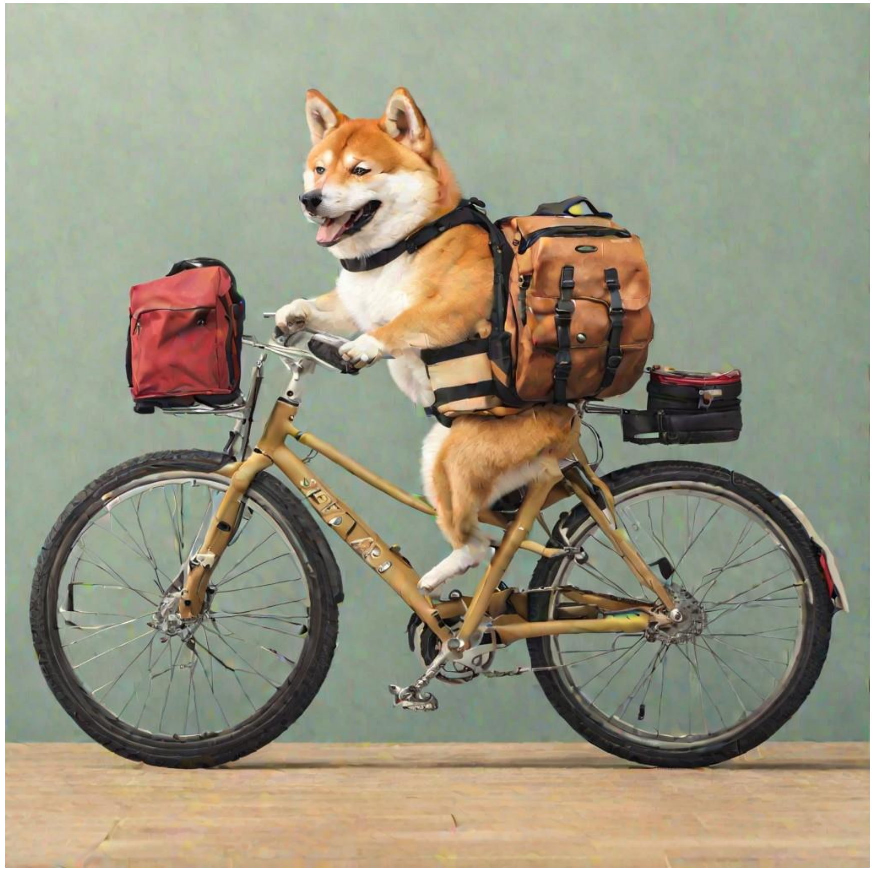


A photo of a Shiba Inu dog with a backpack riding a bike

```
import requests

# Call model from NVIDIA NGC
invoke_url = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/functions/89848fb8-549f-41bb-88cb-95d6597044a4"
fetch_url_format = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/status/"

# Provide API key here
headers = {
"Authorization": "Bearer $API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC",
"Accept": "application/json",
}

# Provide prompts and parameters
payload = {
"prompt": "A photo of a Shiba Inu dog with a backpack riding a bike",
"negative_prompt": "beach",
"inference_steps": 25
}

# re-use connections
session = requests.Session()

response = session.post(invoke_url, headers=headers, json=payload)

while response.status_code == 202:
        response = session.get(fetch_url_format + response.headers.get("NVCF-REQID”), headers=headers)

response.raise_for_status()
response_body = response.json()
print(response_body)
```

catalog.ngc.nvidia.com/orgs/nvidia/teams/ai-foundation/models/sdxl

Getting Started With Diffusion Models

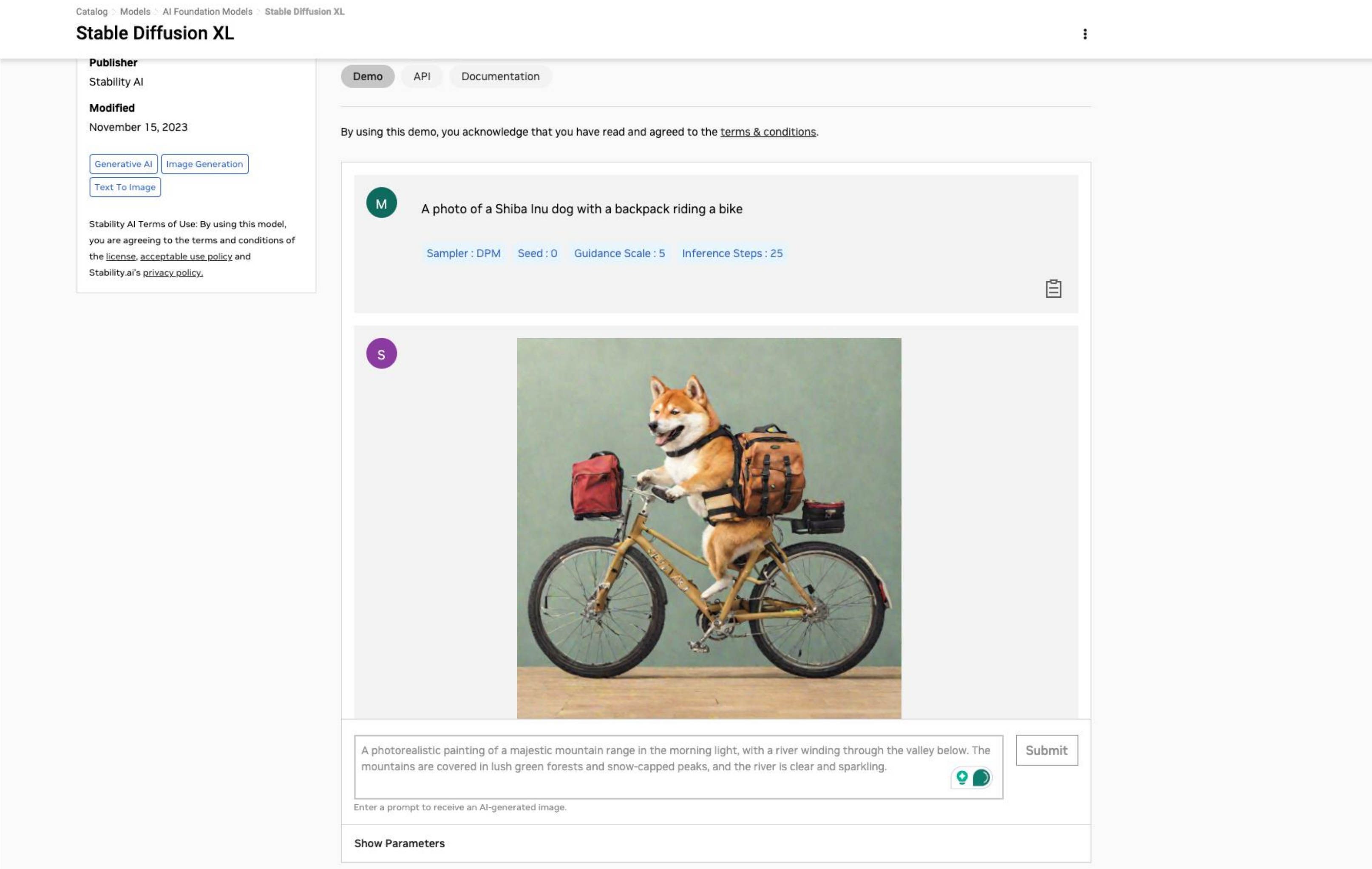


A photo of a Shiba Inu dog with
a backpack riding a bike

```
import requests

# Call model from NVIDIA NGC
invoke_url = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/functions/89848fb8-549f-41bb-88cb-95d6597044a4"
fetch_url_format = "https://api.nvcf.nvidia.com/v2/nvcf/pexec/status/"

# Provide API key here
headers = {
"Authorization": "Bearer $API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC",
"Accept": "application/json",
}

# Provide prompts and parameters
payload = {
"prompt": "A photo of a Shiba Inu dog with a backpack riding a bike",
"negative_prompt": "beach",
"inference_steps": 25
}

# re-use connections
session = requests.Session()

response = session.post(invoke_url, headers=headers, json=payload)

while response.status_code == 202:
        response = session.get(fetch_url_format + response.headers.get("NVCF-REQID”), headers=headers)

response.raise_for_status()
response_body = response.json()
print(response_body)
```

catalog.ngc.nvidia.com/orgs/nvidia/teams/ai-foundation/models/sdxl

Getting Started With Diffusion Models



Catalog > Models > AI Foundation Models > Stable Diffusion XL

Stable Diffusion XL

Publisher
Stability AI

Modified
November 15, 2023

Generative AI | Image Generation | Text To Image

Stability AI Terms of Use: By using this model, you are agreeing to the terms and conditions of the license, acceptable use policy and Stability.ai's privacy policy.

By using this demo, you acknowledge that you have read and agreed to the terms & conditions.

M: A photo of a Shiba Inu dog with a backpack riding a bike

Sampler : DPM | Seed : 0 | Guidance Scale : 5 | Inference Steps : 25

A photorealistic painting of a majestic mountain range in the morning light, with a river winding through the valley below. The mountains are covered in lush green forests and snow-capped peaks, and the river is clear and sparkling.

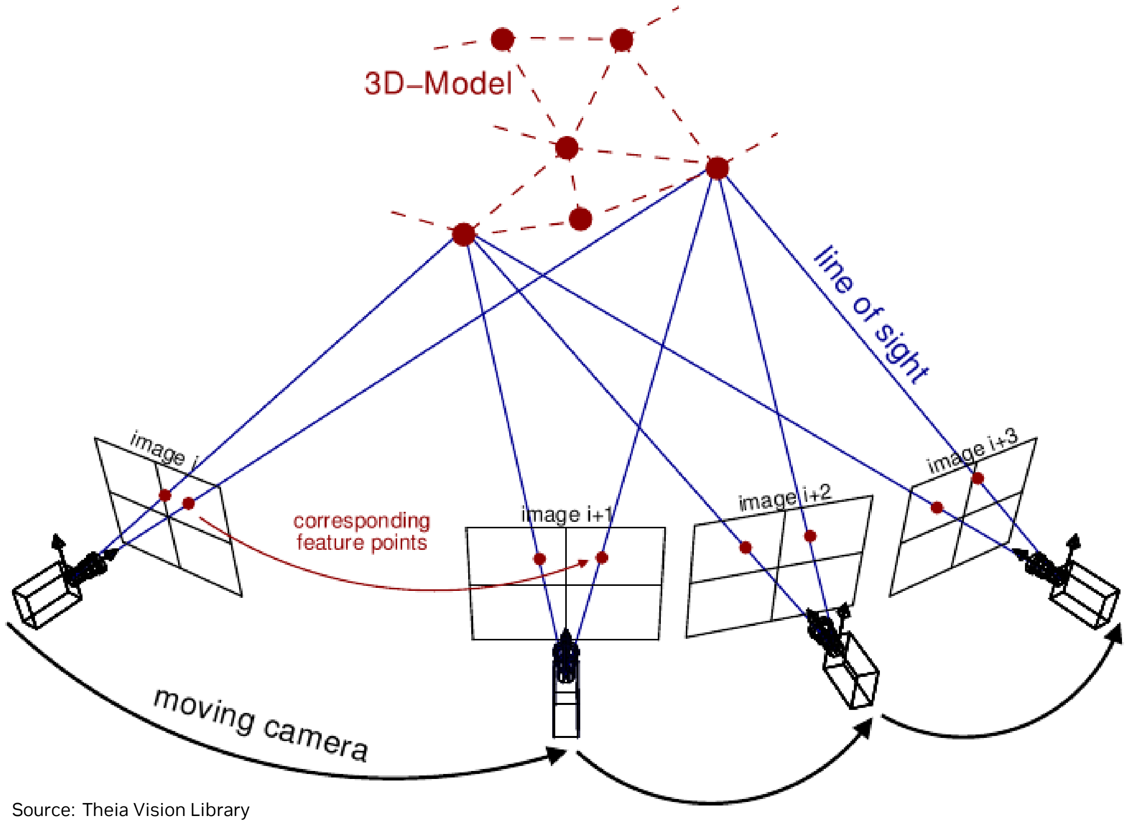
Enter a prompt to receive an AI-generated image.

catalog.ngc.nvidia.com/orgs/nvidia/teams/ai-foundation/models/sdxl

3D Reconstruction

3D Reconstruction

Structure from Motion (SfM)

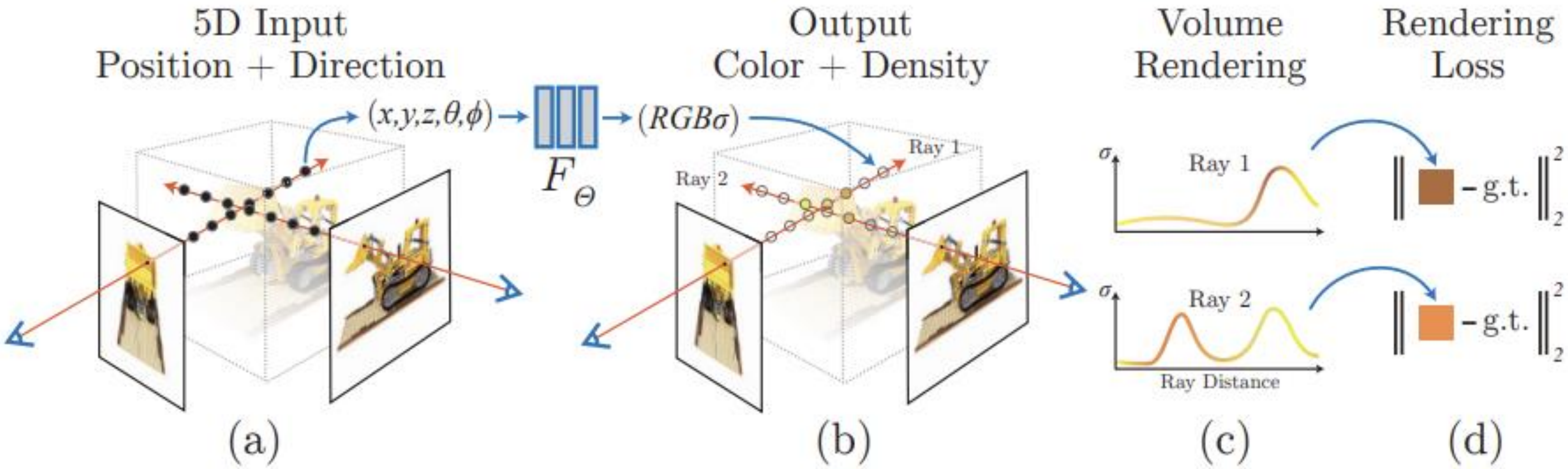


Source: Theia Vision Library

- An imaging technique for estimating 3D structures from 2D image sequences
- SfM tracks features (e.g. corner points, edges) across multiple images to find correspondences.
- This method struggles to capture fine details and complex geometry and struggles with certain materials

3D Reconstruction

Neural Radiance Fields (NeRFs)



NeRF architecture

- NeRFs can generate novel views of complex 3D scenes, based on a partial set of 2D images
- Several decoupled modules - Sampling, encoding, neural network, and rendering
- Diverse applications spanning virtual/augmented reality, medical imaging, robotics, and autonomous navigation

3D Reconstruction

Neuralangelo

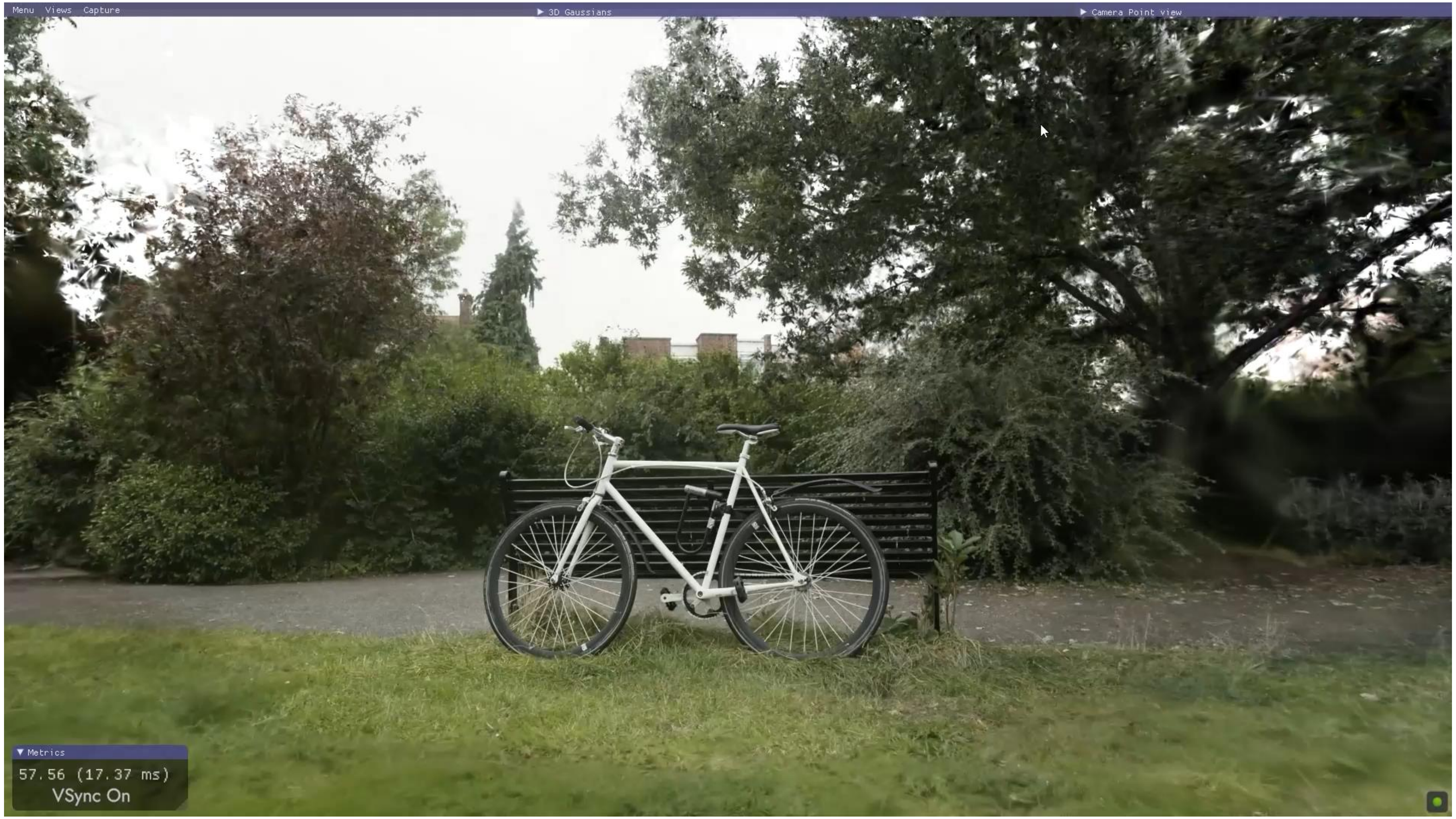
3D Online Shopping Experiences Powered by NeRFs

RECON Labs Inc.

Take a video of the object

3D Reconstruction

3D Gaussian Splatting

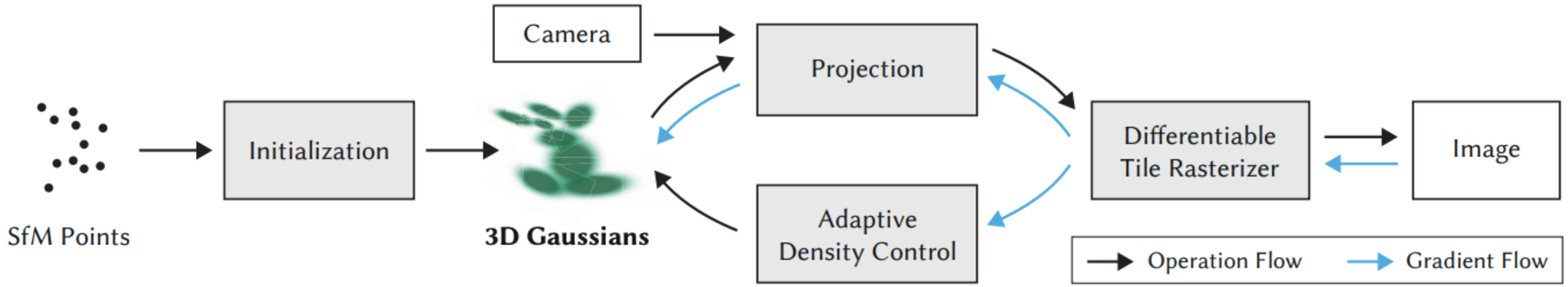


57.56 (17.37 ms)
VSync On

3D Gaussian Splatting for Real-Time Radiance Field Rendering

3D Reconstruction

3D Gaussian Splatting



Point Cloud Generation

- 3D sparse point are generated using 3D points

Gaussian Placement

- Each point transformed into a 3D Gaussian function defined by position, spread, color and opacity

Training & Optimization

- Optimizes parameter of each gaussian points (position, spread, color, opacity)

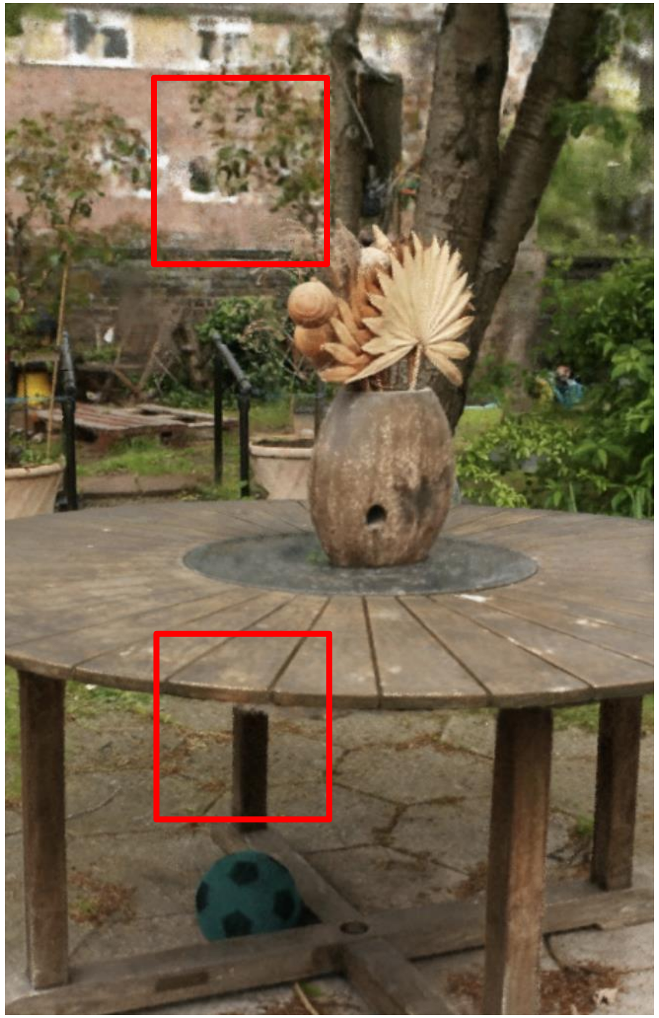
Rendering

- Optimized points are projected on the screen based on camera's viewpoint

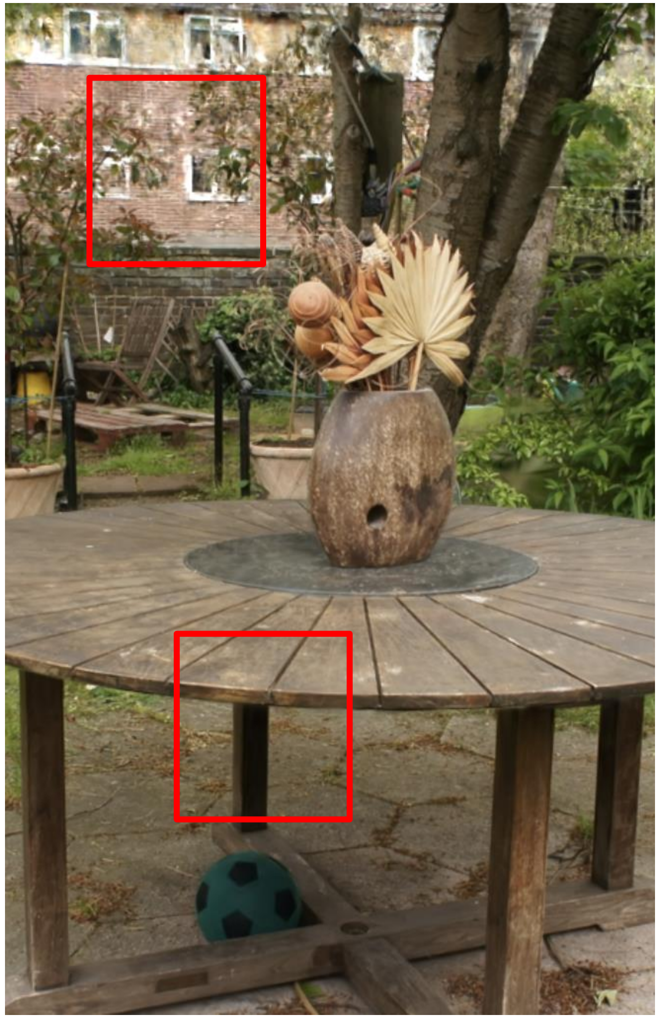
3D Gaussian Splatting for Real-Time Radiance Field Rendering

3D Reconstruction

3D Gaussian Splatting



Instant-NGP

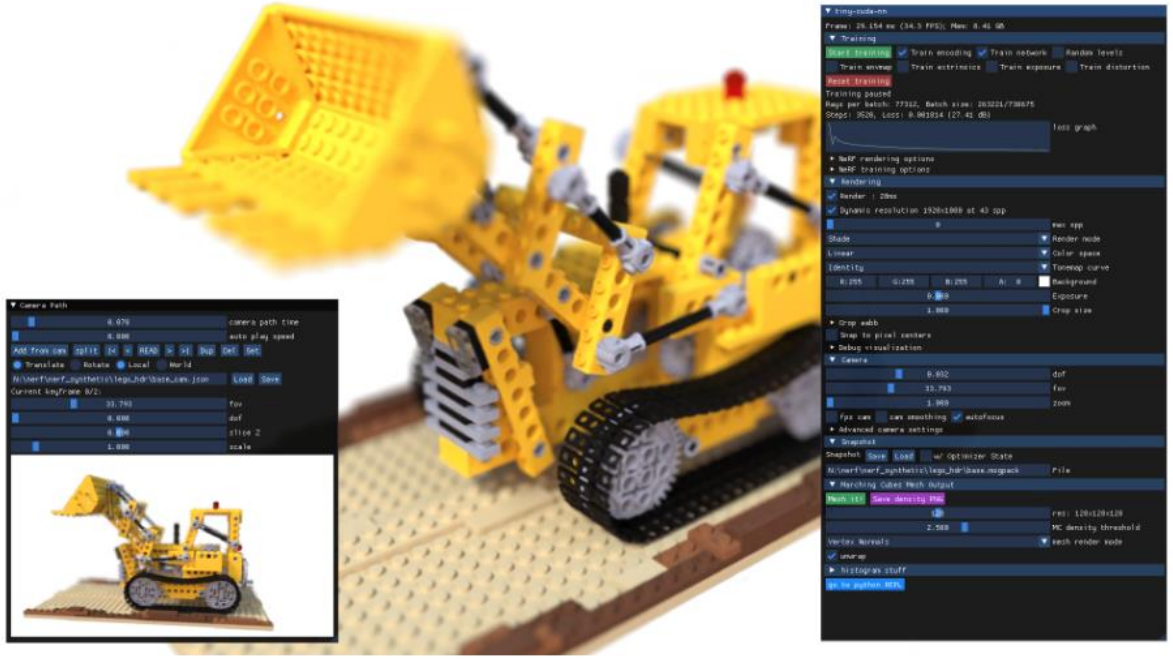


Gaussian Splatter

3D Gaussian Splatting for Real-Time Radiance Field Rendering

Created by @Yannick_Cerrutti using Luma AI Capture

Getting Started With Instant-NGP & 3D Gaussian Splatting

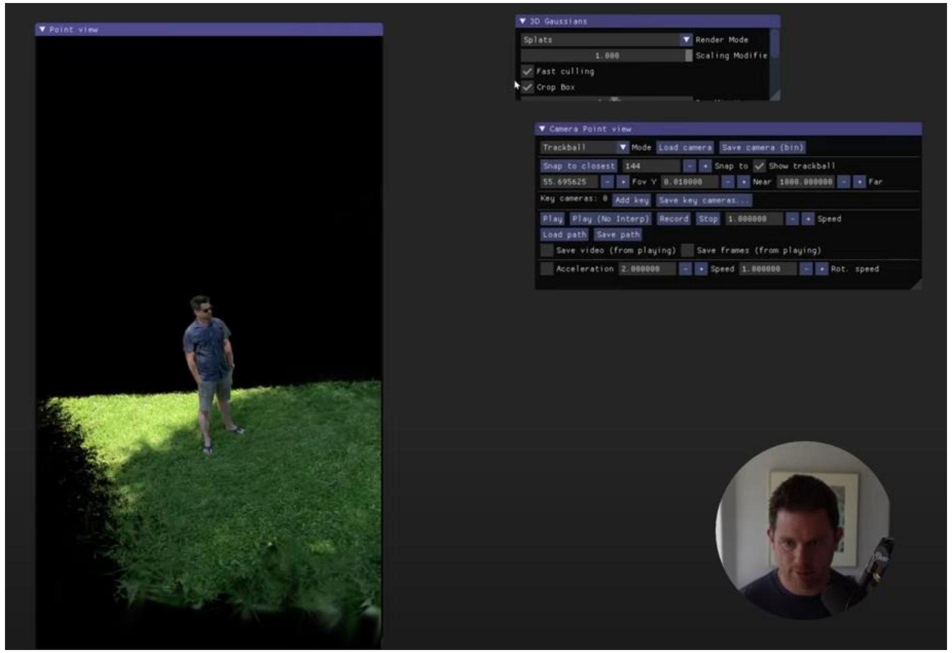


```
▼ tiny-cuda-nn
Frame: 29.154 ms (34.3 FPS); Mem: 8.46 GB
▼ Training
Start training  ☑ Train encoding  ☑ Train network  □ Random levels
□ Train envmap  □ Train extrinsics  □ Train exposure  □ Train distortion
Reset training
Training paused
Rays per batch: 77312, Batch size: 263221/738675
Steps: 3528, Loss: 0.001814 (27.41 dB)
                                   loss graph
▸ NeRF rendering options
▸ NeRF training options
▼ Rendering
☑ Render : 28ms
☑ Dynamic resolution 1920x1080 at 43 spp
                0                  max spp
Shade                              Render mode
Linear                             Color space
Identity                           Tonemap curve
R:255  G:255  B:255  A: 0          Background
              0.000                Exposure
              1.000                Crop size
▸ Crop aabb
□ Snap to pixel centers
▸ Debug visualization
▼ Camera
              0.032                dof
              33.793               fov
              1.000                zoom
□ fps cam  □ cam smoothing  ☑ autofocus
▸ Advanced camera settings
▼ Snapshot
Snapshot  Save  Load  □ w/ Optimizer State
N:\nerf\nerf_synthetic\lego_hdr\base.msgpack   File
▼ Marching Cubes Mesh Output
Mesh it!  Save density PNG
              128                  res: 128x128x128
              2.500                MC density threshold
Vertex Normals                     mesh render mode
☑ unwrap
▸ histogram stuff
go to python REPL
```

```
▼ Camera Path
              0.079                camera path time
              0.000                auto play speed
Add from cam  split  |<  <  READ  >  >|  Dup  Del  Set
● Translate  ○ Rotate  ● Local  ○ World
N:\nerf\nerf_synthetic\lego_hdr\base_cam.json  Load  Save
Current keyframe 0/2:
              33.793               fov
              0.000                dof
              0.000                slice Z
              1.000                scale
```

[Video Tutorial](#) by Eric Hanes, NVIDIA

[Step by Step Instructions in GitHub](#)



```
▼ 3D Gaussians
Splats                          Render Mode
        1.000                   Scaling Modifie
☑ Fast culling
☑ Crop Box

▼ Camera Point view
Trackball  Mode  Load camera  Save camera (bin)
Snap to closest  144  -  +  Snap to  ☑ Show trackball
55.695625  -  +  Fov Y  0.010000  -  +  Near  1000.000000  -  +  Far
Key cameras: 0  Add key  Save key cameras...
Play  Play (No Interp)  Record  Stop  1.000000  -  +  Speed
Load path  Save path
□ Save video (from playing)  □ Save frames (from playing)
□ Acceleration  2.000000  -  +  Speed  1.000000  -  +  Rot. speed
```

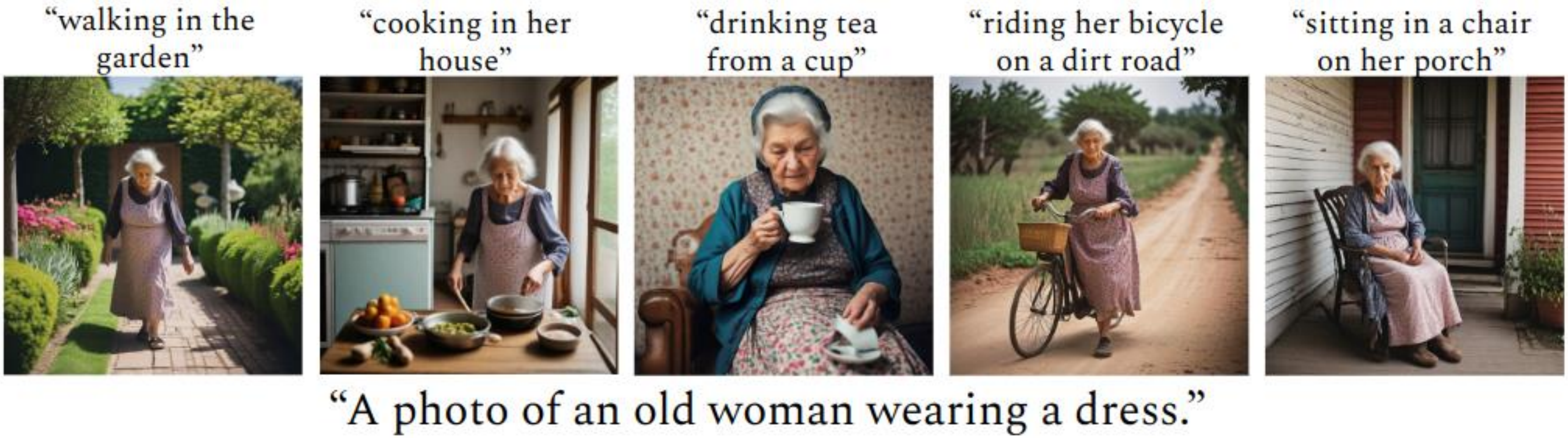
[Video Tutorial](#) by The NeRF Guru

[Step by Step Instructions in GitHub](#)

Emerging Computer Vision Frontiers

Emerging Computer Vision Frontiers

Text-2-Image with Character Consistency


"walking in the garden"
"cooking in her house"
"drinking tea from a cup"
"riding her bicycle on a dirt road"
"sitting in a chair on her porch"
"A photo of an old woman wearing a dress."

Emerging Computer Vision Frontiers

Text-to-4D (3D in motion)

https://research.nvidia.com/labs/toronto-ai/AlignYourGaussians/samples.html

Other Relevant GTC Sessions

- [S62724] Revolutionizing Vision AI: From 2D to 3D Worlds
- [S62818] The Visionaries: A Cross-Industry Exploration of Computer Vision
- [CWE63456] Build Accelerated Computer Vision Microservices With NVIDIA Libraries and SDKs
- [S62624] The Vision-AI Revolution powered by DeepStream

NVIDIA®