Unveiling Depth Anything V2: A Breakthrough in Monocular Depth Estimation

Sunidhi Ashtekar
8 min read · Jun 16, 2024

Depth Anything V2 On RF100 Dataset

In the realm of computer vision, monocular depth estimation (MDE) has emerged as a pivotal task. The latest advancement in this field comes from the innovative “Depth Anything V2” model, which promises unprecedented accuracy and efficiency. Let’s dive into the key aspects of this groundbreaking work.

Introduction

Depth estimation is essential for various applications like 3D reconstruction, navigation, and even AI-generated content. Additionally, depth estimation is crucial for augmented reality (AR) and virtual reality (VR) experiences, autonomous driving, robotics, obstacle detection, path planning, and surveillance systems. Traditional methods struggled with certain challenges, including dealing with complex scenes and achieving fine-grained details. Depth Anything V2 aims to overcome these limitations, providing robust and detailed depth predictions.

The paper introduces models of various scales:

- **Small:** 25 million parameters
- **Base:** 335 million parameters
- **Large:** 891 million parameters
- **Giant:** 1.3 billion parameters

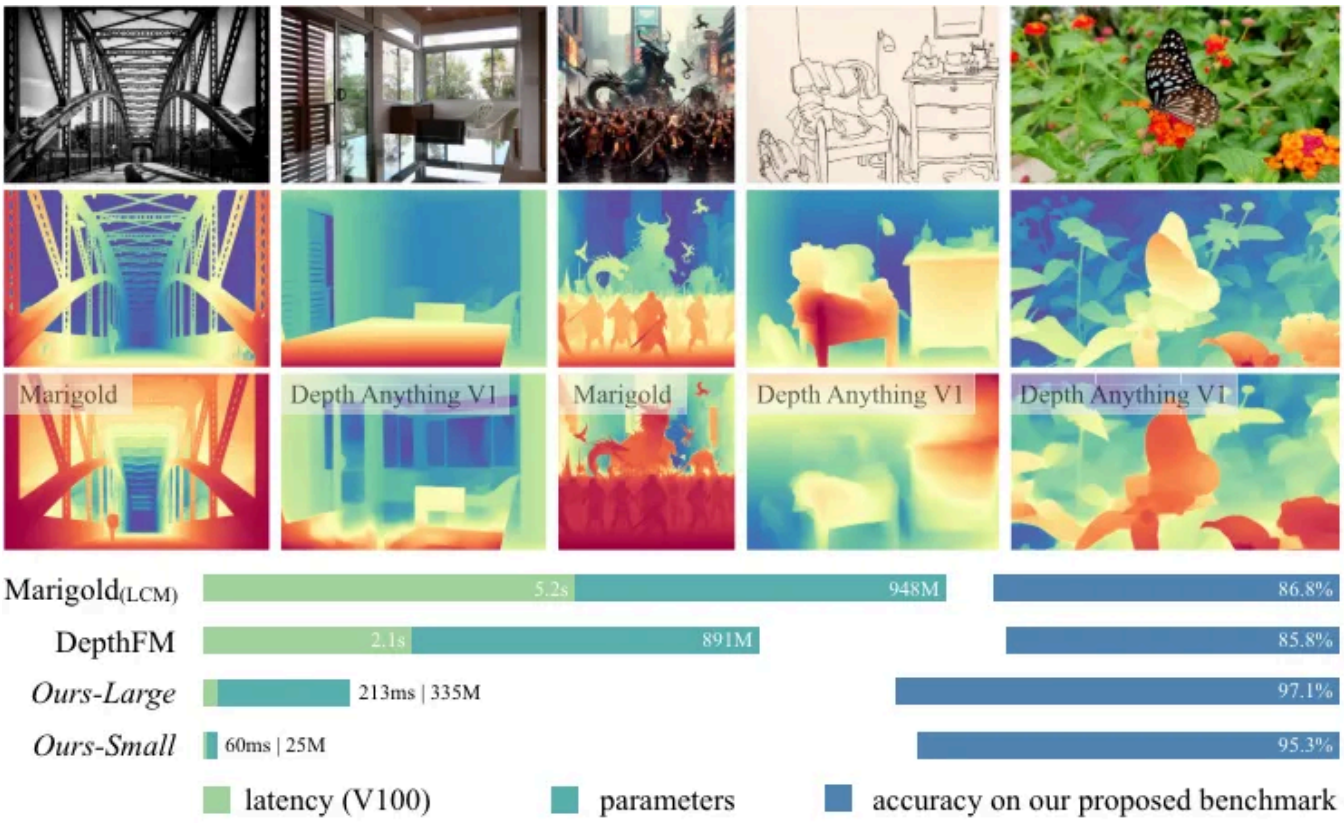


Image from Official Paper Depth Anything V2

What Makes Depth Anything V2 Special?

Depth Anything V2 stands out due to its innovative approach:

1. **Synthetic Data Utilization:** Replacing all labeled real images with synthetic ones to enhance label precision and detail.
2. **Scalable Model Architecture:** Offering models ranging from 25 million to 1.3 billion parameters to cater to different needs.
3. **Pseudo-Labeled Real Images:** Bridging the gap between synthetic and real data using large-scale pseudo-labeled real images.
4. **Versatile Evaluation Benchmark:** Introducing DA-2K, a new benchmark with diverse and precisely annotated scenes.

Improvements Over Depth Anything V1

Depth Anything V2 introduces several significant improvements over V1:

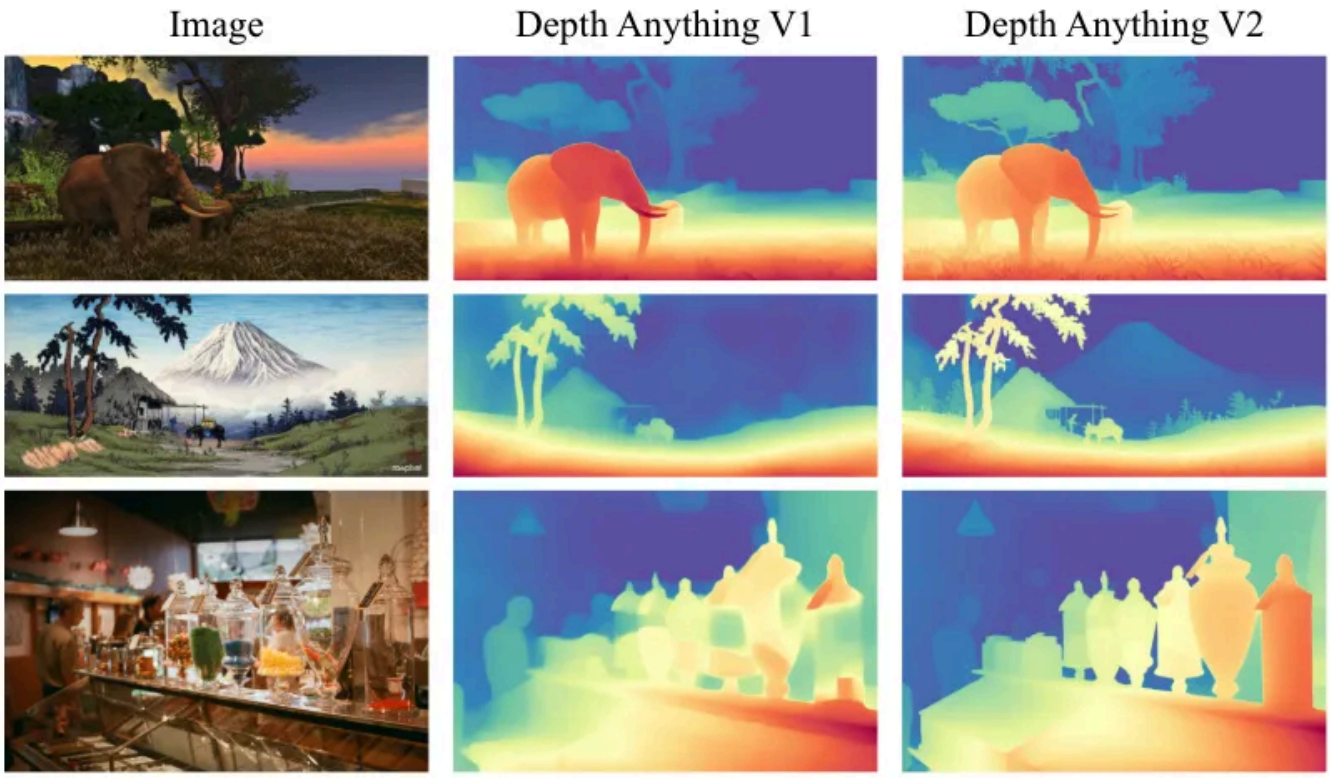


Image from Official Paper Depth Anything V2

1. **Data Quality:** Replaces noisy labeled real images with precise synthetic images.

2. **Efficiency**: Models are more efficient, with faster inference times and fewer parameters.
3. **Robustness**: Produces robust depth predictions for complex scenes, including transparent and reflective surfaces.
4. **Generalization**: Improved generalization capabilities due to training on large-scale pseudo-labeled real images.
5. **Model Scalability**: Offers models of varying scales to cater to different application needs, from lightweight (25M parameters) to large-scale (1.3B parameters).Depth Anything V1

- **Data:** Used a mix of labeled real images and synthetic images, which introduced noise and inconsistencies.
- **Model Efficiency:** Slower inference speeds and larger models compared to V2.
- **Accuracy:** Good accuracy but struggled with complex scenes and fine-grained details.
- **Robustness:** Less robust to transparent objects and reflective surfaces.

Architecture Overview

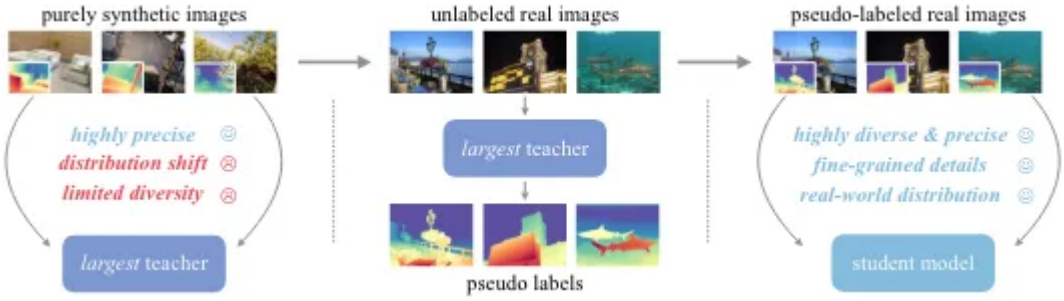


Image from Official Paper Depth Anything V2

Depth Anything V2 is built on the DINOv2 architecture and consists of three key steps:

1. **Teacher Model Training:** A large-scale teacher model is trained exclusively on high-quality synthetic images.

2. **Pseudo-Labeling Real Images:** The teacher model generates precise pseudo-depth labels for a large dataset of unlabeled real images.
3. **Student Model Training:** Student models are trained on these pseudo-labeled real images to ensure robust generalization.

Key Components:

- **Encoder:** Utilizes different scales of DINOv2 (small, base, large, giant) to process input images.
- **Depth Decoder:** Based on DPT, it converts features from the encoder into depth maps.

Datasets

Depth Anything V2 employs a mix of synthetic and real datasets:

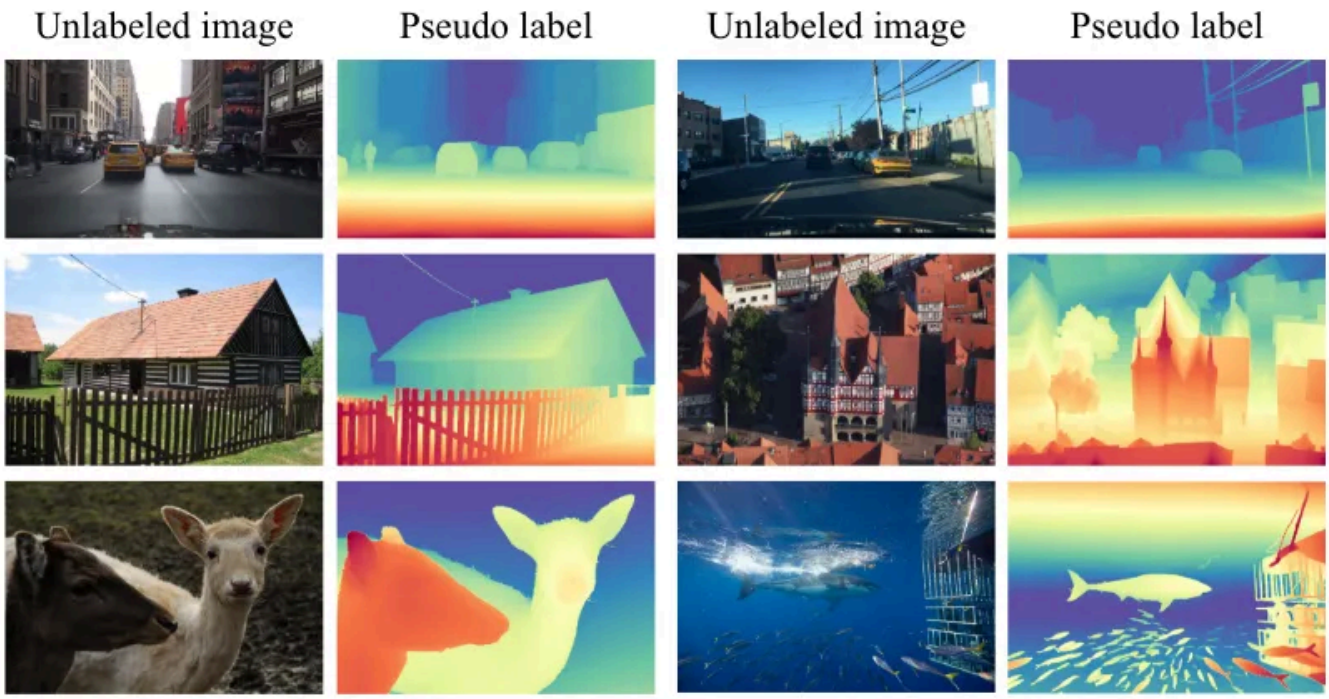


Image from Official Paper Depth Anything V2

1. **Synthetic Datasets:** These include BlendedMVS, Hypersim, IRS, TartanAir, and VKITTI 2, providing highly precise depth labels.

2. **Pseudo-Labeled Real Datasets**: Includes large-scale datasets like BDD100K, Google Landmarks, ImageNet-21K, LSUN, Objects365, Open Images V7, Places365, and SA-1B.

The combination of these datasets ensures a broad and diverse training set, enhancing the model's ability to generalize across various real-world scenarios.

Implementation Details

The training process involves two main stages:

1. **Training the Teacher Model**: The teacher model, a DINOv2-G (giant) encoder, is trained on synthetic datasets to capture detailed and accurate depth information.
2. **Training Student Models**: Smaller models (DINOv2 small, base, large) are trained on pseudo-labeled real images generated by the teacher model. This step involves the following specifics:

- **Resolution**: Training images are resized to 518x518 pixels.
- **Optimization**: Uses the Adam optimizer with specific learning rates for the encoder (5e-6) and decoder (5e-5).

Loss Functions

To ensure sharp and precise depth predictions, Depth Anything V2 combines two primary loss functions:

- **Scale and Shift-Invariant Loss (Lssi):** Ensures depth consistency regardless of scale and shift.
- **Gradient Matching Loss (Lgm):** Enhances sharpness and detail by matching gradients between predicted and ground truth depth maps.

Depth Estimation Process

Depth estimation in Depth Anything V2 involves converting a 2D image into a depth map, representing the distance of each pixel from the camera. The process includes:

1. **Image Encoding:** The input image is passed through the DINOv2 encoder, extracting multi-scale features.
2. **Depth Decoding:** The depth decoder (DPT) processes these features to generate a high-resolution depth map.
3. **Inference**: During inference, the model predicts the depth map, which can be visualized or used in downstream applications like 3D reconstruction or navigation.

Detailed Overview of Depth Estimation

Here's a detailed overview of the process in Depth Anything V2:

1. **Input and Preprocessing:** The input image is preprocessed to standardize its size and format, including resizing and normalizing pixel values. This ensures consistent processing and enhances model robustness through data augmentation.
2. **Feature Extraction:** The preprocessed image is passed through a feature extractor using the DINOv2 architecture. This feature extractor, a Vision Transformer (ViT) or Convolutional Neural Network (CNN), captures essential details such as edges, textures, and objects.
3. **Depth Prediction:** Extracted features are fed into a Depth Decoder based on the Dense Prediction Transformer (DPT) architecture. The decoder predicts depth values for each pixel, creating a depth map where each pixel value represents the estimated distance from the camera.
4. **Loss Functions:** During training, the predicted depth map is compared against ground truth using two primary loss functions.

5. **Post-Processing:** Raw depth predictions may undergo post-processing, such as bilateral filtering, to refine the depth map, smoothing values while preserving edges.

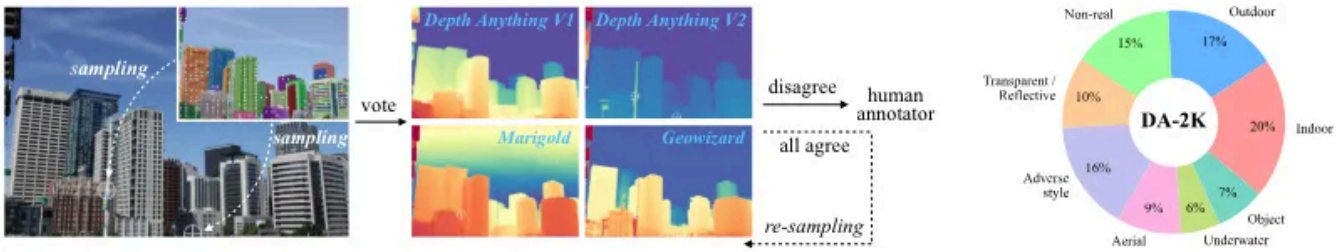
6. **Inference:** The model processes new images in real-time to generate depth maps efficiently. Depth Anything V2's architecture ensures fast and accurate depth prediction suitable for various applications.

7. **Robustness and Generalization:** Depth Anything V2 uses pseudo-labeled real images for robustness across scenarios. The teacher model, trained on synthetic images, generates pseudo-depth labels for unlabeled real images. Student models trained on these images generalize well to complex scenes, transparent objects, and reflective surfaces.

Depth Anything V2 combines precision, efficiency, and robustness to handle a wide range of real-world scenarios, achieving state-of-the-art performance in monocular depth estimation.

Annotation Process

Figure 8: Visualization of widely adopted but indeed noisy test benchmark [70]. As highlighted, the depth of the mirror and thin structures are incorrect (black pixels are ignored). In comparison, our model predictions are accurate. The noise will cause better models instead achieve lower scores.



(a) Annotation pipeline (b) Encompass 8 scenarios

Figure 9: Our proposed evaluation benchmark DA-2K. (a) The annotation pipeline for relative depth between two points. Points are sampled based on SAM [33] mask predictions. Disagreed pairs among four depth models will be popped out for annotators to label. (b) Detail of our scenario coverage.

Image from Official Paper Depth Anything V2

- **Automatic Selection:** Uses SAM to predict object masks and select key points.
- **Manual Selection:** Ensures challenging pairs are included for precision.
- **Triple-Checking:** Multiple annotators verify annotations to maintain accuracy.

Getting Started

To do inference on custom dataset these are the steps to follow,

- Environment Setup
- Download Pre-trained Model Weights for Depth Anything V2
- Install Dependencies
- Download datasets from Roboflow
- Visualization and Analysis

Setting Up Hugging Face Hub

To use the Hugging Face Model Hub, first install the `huggingface_hub` library, then log into your Hugging Face account. This setup enables you to download, manage, and upload models and datasets seamlessly.

```
!python -m pip install huggingface_hub
!huggingface-cli login
```

Cloning the Depth Anything V2 Repository

```
!git clone https://huggingface.co/spaces/depth-anything/Depth-Anything-V2
```

Mounting Google Drive

Next, mount the drive to use the downloaded model. I have used images from Roboflow RF-100 Construction Safety 2 dataset for inference.

```
from google.colab import drive
drive.mount('/content/drive')
```

Installing Dependencies

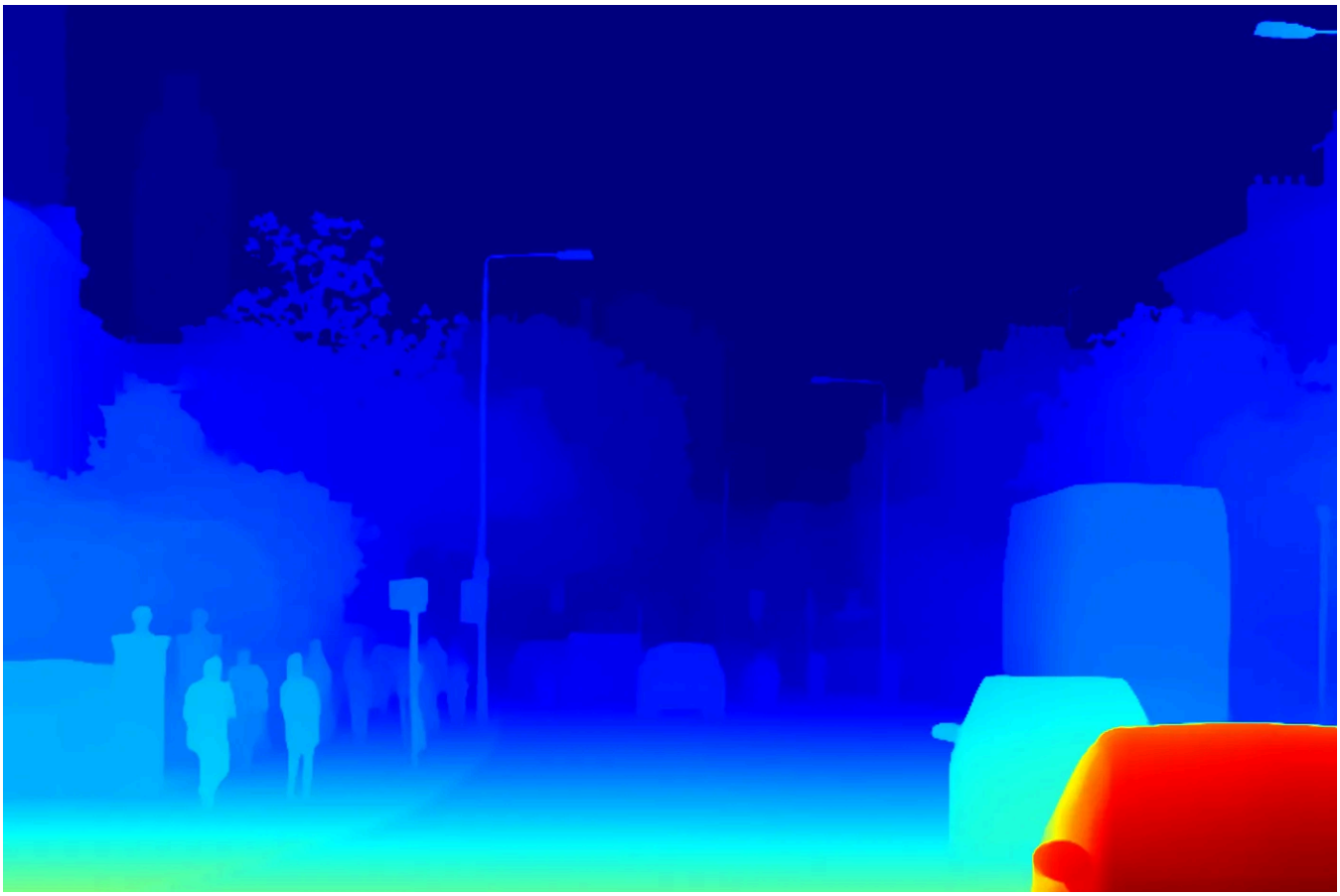
Navigate to the cloned Depth Anything V2 repository and install all necessary dependencies. Install the required Python packages listed in the `requirements.txt` file. Download the pre-trained model weights and save under checkpoints folder, if the folder doesn't exist create one!

```
!cd Depth-Anything-V2
!pip install -r /content/Depth-Anything-V2/requirements.txt

!mkdir -p checkpoints
!mv /content/drive/MyDrive/checkpoints/depth_anything_v2_vitl.pth checkpoints/d
```

Performing Depth Estimation with Depth Anything V2

This code will perform depth estimation on the specified image and display the resulting depth map within the Colab notebook. Adjust the paths as needed to match your file locations

```
import cv2
import sys
import os
import torch
from google.colab.patches import cv2_imshow

sys.path.append(os.path.abspath('/content/Depth-Anything-V2'))

from depth_anything_v2.dpt import DepthAnythingV2

# Initialize the model (using 'vitl' encoder as an example)
model = DepthAnythingV2(encoder='vitl', features=256, out_channels=[256, 512, 1
model.load_state_dict(torch.load('checkpoints/depth_anything_v2_vitl.pth', map_
model.eval()

# Read the image
raw_img = cv2.imread('/content/drive/MyDrive/input_imgs/demo01.jpg')

# Perform inference
depth = model.infer_image(raw_img)  # HxW raw depth map

# Optionally, visualize the depth map
depth_normalized = cv2.normalize(depth, None, 0, 255, cv2.NORM_MINMAX)
depth_colormap = cv2.applyColorMap(depth_normalized.astype('uint8'), cv2.COLORM
cv2_imshow(depth_colormap)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Running Depth Estimation on Images

To run the Depth Anything V2 model on a directory of images use the provided script. The model will process all images in the specified input directory and save the depth maps to the designated output directory

```
!python /content/Depth-Anything-V2/run.py --encoder vitl --img-path /content/dr
```

Running Depth Estimation on Videos

```
!python /content/Depth-Anything-V2/run_video.py --encoder vitl --video-path /cc
```

Conclusion

I hope this guide has provided a comprehensive overview of the newly launched Depth Anything V2 model, including its innovative features and the detailed process of monocular depth estimation. Depth Anything V2 leverages both synthetic and pseudo-labeled real datasets, and achieves robust, efficient, and accurate depth predictions. Additionally, we provided a step-by-step guide on how to perform inference on your images using this model.

If you have any questions, recommendations, or critiques, please don't hesitate to reach out on LinkedIn. I'm open to discussions and eager to hear your feedback or assist with any challenges you might encounter.

References

To further explore the concepts, I recommend visiting the following resources:

- **Depth Anything V2 Official Repository Github**
- **Depth Anything V2 Official Paper**
- **Hugging Face Depth Anything V2 Repository**: Explore models and datasets related to Depth Anything V2 on Hugging Face.
- **Depth Anything V2 Online Demo**
- For custom dataset visit **Roboflow 100**