Machine Learning Engineering Bootcamp
Capstone, Step 5: Data Wrangling
Student: Kenneth Fung
Date: 18 January 2025

1. INTRODUCTION:

In this step 5 of my Capstone project, I gathered a dataset of outdoor images to process using Photoshop into a dataset of images with embedded pictures and mirror reflections which are simulated portions of the original images. The objective is to create a dataset of images for self-supervised fine-tuning the Depth Anything v2 MDE model. The goal is to improve the pretrained Depth Anything model's performance in processing depth maps of images with flat pictures such as paintings, or mirror reflections.

2. Depth Anything v2 – Areas For Improvement:

The following are potential areas of improvement for the Depth Anything v2 MDE model:

A. Figure 1: This scene contains a painting of farm animals, as evidenced by the male and female figures behind the painting (hence it cannot be a window). The painting should have resulted in a constant depth assignment, and outlines of the animals have not have resulted.

B. Figure 2: There is a faint outline of the lady in the mirror. Since the mirror is flat, it should have resulted in a constant depth gradient instead of an outline of the lady. The image of the chess piece in the mirror fared better.

Figure 1. Depth Anything - Area for improvement: embedded pictures

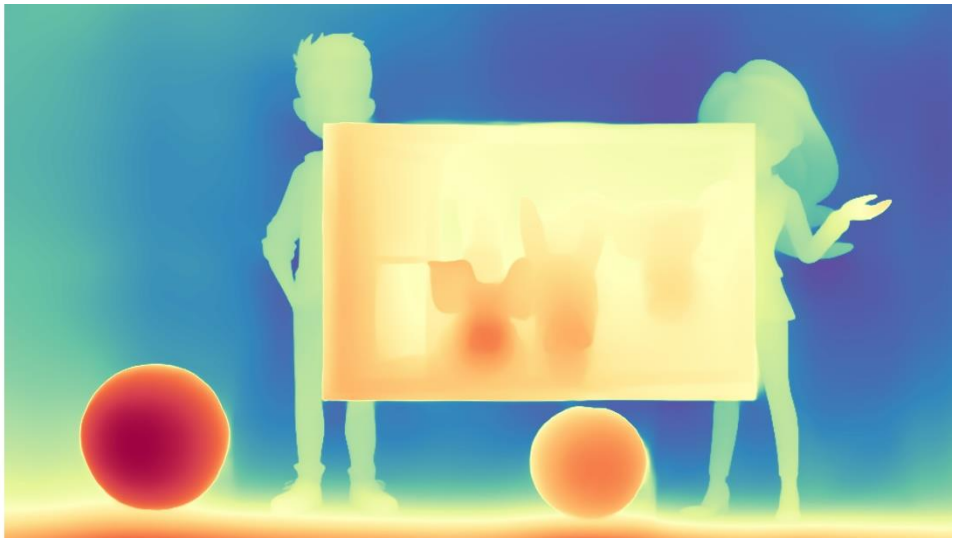

Figure 2. Depth Anything – Area for improvement: mirror reflections

3. ANALYSIS of Areas For Improvement (in-work)

…

4. PLAN FORWARD (in-work)

Overview…self-supervised fine-tuning vs incremental training…did not require data wrangling…

A. Gather a large (> 15,000 images) dataset of images of consistent sizes and resolution, consisting of either indoor or outdoor scenes.

B. Use Photoshop to batch process the dataset to produce two subsets of images – one subset with embedded pictures such as paintings, and one subset with embedded mirror reflections of portions of the original images.

Follow this process[1]:

1. Run Depth Anything on Your Dataset

- Use the pre-trained **Depth Anything** model to infer depth maps for all images in your dataset.
- Ensure your dataset is preprocessed correctly (e.g., resizing images to the required dimensions, normalizing values) to match the input requirements of the model.
- Save the generated depth maps as ground truth labels in a format compatible with the model's training pipeline.

---

2. Evaluate the Generated Depth Maps

- Check the quality of the depth maps produced by **Depth Anything**.
    - Visualize a subset of the outputs.
    - Look for areas with significant noise or artifacts.
- If the quality is not consistent, consider:
    - Using additional post-processing techniques like edge-preserving filters or smoothing.
    - Combining depth maps with other sources (e.g., stereo depth, SfM outputs) for more reliable labels.

---

3. Augment the Dataset

- Perform **data augmentation** to improve the robustness of the retraining process:
    - Flip, rotate, or scale the images and corresponding depth maps.

[1] OpenAI. (2025). Explanation of self-supervised fine-tuning process. Retrieved from ChatGPT

    - Add synthetic reflections, noise, or blur to mimic challenging conditions.
- Augmentation helps avoid overfitting to the biases of the pre-trained model.

---

4. Fine-tune the Depth Anything Model

- Set up the training pipeline for **Depth Anything**:
    - Use the inferred depth maps as pseudo-labels for supervision.
    - Implement loss functions such as **L1 Loss**, **Huber Loss**, or **SSIM Loss** to compare predicted depth maps with pseudo-labels.
    - Use regularization techniques (e.g., smoothness losses) to enforce realistic depth continuity.
- Train with a lower learning rate initially to prevent catastrophic forgetting of the pre-trained knowledge.

---

5. Validate on a Separate Dataset

- Split your dataset into training, validation, and test sets.
- Ensure that the model generalizes well to unseen images by monitoring validation metrics like **RMSE**, **Abs Rel Error**, and **Delta Accuracy**.

---

6. Iterative Refinement

- Once the model is retrained, you can:
    - Use it to generate more accurate depth maps for your dataset.
    - Repeat the process iteratively for further refinement.
- Optionally, combine the predictions of the retrained model with the original pseudo-labels to improve robustness.

---

7. Potential Issues and Considerations

- **Bias Propagation**: Since the pseudo-labels are generated by the same model, any inherent biases in the pre-trained model might persist.
    - Mitigation: Introduce additional high-quality ground truth depth maps (if available) to fine-tune alongside pseudo-labels.
- **Domain Shift**: If your dataset differs significantly from the pre-trained model's training data, results might be suboptimal.

    - Mitigation: Include domain-specific augmentations or a few real depth labels from your target domain.

Tools and Resources for Implementation

- **Frameworks**: PyTorch, TensorFlow, or any library compatible with Depth Anything.
- **Pre-trained Model Repository**: Ensure you have access to the original **Depth Anything** weights and inference scripts.

Visualization Libraries: Use tools like Matplotlib or OpenCV for depth map visualization and evaluation.

5. Using Photoshop to process Mapillary dataset: (in-work)

…paintings and mirror reflections of portions of original images…adding borders to simulate picture and mirror frames

A. Mapillary dataset (mapillary-vistas-dataset_public_v1.2.zip):
https://www.mapillary.com/dataset/vistas

B. Photoshop script for adding embedded pictures (paintings) to original images:
GitHub: https://github.com/kentheman4AI/SB-Capstone-Project-Monocular-Depth-Estimation/blob/3edd44604f4ee6b7f730d72efdc2537226132bdd/Photoshop%20scripts/ResizeAndFrame.jsx

C. Photoshop script for adding mirror reflections to original images:
GitHub: https://github.com/kentheman4AI/SB-Capstone-Project-Monocular-Depth-Estimation/blob/3edd44604f4ee6b7f730d72efdc2537226132bdd/Photoshop%20scripts/MirrorImageProcessing.jsx

D. Sample Mapillary images processed with Photoshop:
GitHub: https://github.com/kentheman4AI/SB-Capstone-Project-Monocular-Depth-Estimation/tree/3edd44604f4ee6b7f730d72efdc2537226132bdd/data

Figure 3. Sample image of Mapillary image processed by Photoshop to add an embedded picture (painting)

Figure 4. Sample image of Mapillary image processed by Photoshop to add a mirror reflection