Comparative Analysis of Monocular Depth Estimation Models

Team Members:

1. Vorrapard Kumthongdee
2. Nikita Makarov

Mentor: Haoyang Pei

Abstract:

This project aims to conduct a comparative analysis of state-of-the-art models for monocular depth estimation. The models to be compared include traditional CNN-based architectures, hybrid CNN-transformer models, and purely transformer-based models. The primary goal is to evaluate each model's performance on key metrics such as accuracy (in terms of depth estimation), inference speed, and computational complexity, particularly focusing on their trade-offs between precision and efficiency. The dataset used for evaluation will be the NYU Depth v2, a widely recognized benchmark for depth estimation in indoor scenes. Through this comparison, we aim to provide insights into the relative strengths of each model, helping to determine the best choices for specific use cases such as real-time applications or high-accuracy depth prediction.

We will implement and test models such as Monodepth2, DepthAnythingv2, DenseDepth, Vision Transformer (ViT), and Swin Transformer (depending on time). Each model will be evaluated under the same conditions to ensure a fair comparison, and the results will be presented with visual depth maps and quantitative analysis.

(ref: https://paperswithcode.com/task/depth-estimation)

Project Schedule:

Week 6-7: Literature Review and Dataset Setup (Deadline: Oct 18)

- **Task**: Perform an in-depth literature review of monocular depth estimation models, especially focusing on models that leverage CNNs, transformers, and hybrid architectures.
    - **Approach**: Search for relevant papers and study the main points of each model from https://paperswithcode.com/task/depth-estimation. Identify key approaches used in the models we plan to implement, and prepare notes on the different methodologies.

    - **Responsibility**: Vorrapard & Nikita.
    - **Output**: A document summarizing the key findings from each paper, model descriptions, and potential evaluation metrics.
- **Task**: Download and preprocess the **NYU Depth v2 dataset**. This includes resizing images, normalizing pixel values, and dividing the dataset into training and testing sets.
    - **Approach**: Use standard data pre-processing techniques, relying on Python libraries such as OpenCV, PyTorch DataLoader, and NumPy.
    - **Responsibility**: Vorrapard & Nikita.
    - **Output**: Cleaned and ready-to-use dataset, with standardized image sizes and depth maps.

Week 8-10: Implementation of CNN-based Models (Deadline: Nov 8)

- **Task**: Implement or fine-tune CNN-based models such as Monodepth2 and MobileNet.
    - **Approach**: Utilize existing open-source implementations of Monodepth2 and MobileNet from GitHub repositories. Fine-tune the models on the NYU Depth v2 dataset using PyTorch.
    - **Responsibility**: One model for each
    - **Output**: Trained models ready for evaluation.
- **Task**: Begin training CNN-based models on the dataset and track performance metrics using TensorBoard.
    - **Approach**: Train the models on GPU resources (I don't know if we're given access to HPC), monitor training loss and depth accuracy, and perform early validation tests to ensure models are converging.
    - **Responsibility**: One model for each
    - **Output**: Trained CNN models with recorded metrics.

Week 11-13: Implementation of Transformer-based and Hybrid Models (Deadline: Nov 29)

- **Task**: Implement transformer-based models (such as Vision Transformer, Swin Transformer) and hybrid models (such as DenseDepth, DepthAnythingv2).
    - **Approach**: Adapt pre-trained versions of these models where possible and modify them for the depth estimation task. Use transformer-specific libraries like Hugging Face's transformers for the Vision Transformer.
    - **Responsibility**: One for ViT, one for hybrid
    - **Output**: Transformer-based models integrated into the depth estimation pipeline.
- **Task**: Train the transformer-based and hybrid models on the NYU Depth v2 dataset, with the same evaluation setup used for CNN-based models.
    - **Approach**: Train models using the same settings as the CNN-based models, track the learning curves, and visualize intermediate results.
    - **Responsibility**: One for ViT, one for hybrid
    - **Output**: Trained transformer-based models with initial evaluation metrics.

Week 14-15: Evaluation and Comparative Analysis (Deadline: Dec 11)

- **Task**: Conduct a comprehensive evaluation of all models based on key performance metrics: RMSE, MAE, delta accuracy, inference speed, and model complexity (parameters, FLOPs).
    - **Approach**: Run all trained models on the test set from NYU Depth v2. Use PyTorch scripts to calculate the performance metrics and generate depth map visualizations.
    - **Responsibility**: Vorrapard
    - **Output**: A detailed comparison table summarizing the metrics for each model.
- **Task**: Analyze and document the trade-offs between the models, such as accuracy vs. speed. Identify which models are more suitable for specific applications like real-time deployment or high-accuracy tasks.
    - **Approach**: Compare the models qualitatively (using visualizations) and quantitatively (using the comparison table), identifying where each model excels.
    - **Responsibility**: Nikita
    - **Output**: A report detailing the findings and recommendations for different use cases.

Week 16: Final Report and Presentation (Deadline: Dec 18)

- **Task**: Prepare the final project report, summarizing all the work done, findings from the comparative study, and lessons learned.
    - **Approach**: Create the report, including all relevant figures, tables, and citations.
    - **Responsibility**: Vorrapard & Nikita
    - **Output**: Completed final report.
- **Task**: Create a presentation that highlights the key findings, models compared, and results.
    - **Approach**: Use PowerPoint or Google Slides to develop a clear and concise presentation for the final class presentation.
    - **Responsibility**: Vorrapard & Nikita
    - **Output**: Final presentation slides.

List of References:

1. **A. Dosovitskiy et al.,** "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://arxiv.org/abs/2010.11929
2. **C. Godard et al.,** "Digging Into Self-Supervised Monocular Depth Estimation," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 3828-3838. doi: 10.1109/ICCV.2019.00393.
3. **R. Ranftl et al.,** "Vision Transformers for Dense Prediction," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. [Online]. Available: https://arxiv.org/abs/2103.13413

4. **Z. Liu et al.,** "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 10012-10022. doi: 10.1109/ICCV48922.2021.01000.
5. **F. B. Dijk and I. K. Reshadi,** "Monocular Depth Estimation Using Lightweight CNNs," *IEEE Transactions on Image Processing*, vol. 29, pp. 4789-4800, 2020. [Online]. Available: https://doi.org/10.1109/TIP.2020.2987129
6. **Yang, Lihe, et al.**, "Depth Anything V2." *arXiv preprint arXiv:2406.09414* (2024).
7. https://paperswithcode.com/task/depth-estimation