Spatial Object Learning & Integrated Dimensions Multi-Modal Object Classification with RGB, Depth, and Geometry — Project Check-In

Moses Adewolu, Terrence An Malayvong, Zihan Chen, Felipe Bergerman

July 4th, 2025

1 Introduction

In this project, we address the problem of **multi-modal object classification** using RGB images, depth maps, and 3D geometry data.

Problem Definition: Given an input comprising:

- **RGB image** capturing visual texture and color.
- **Depth map** encoding spatial distance information.
- **Point cloud geometry** representing the object's 3D shape.

The task is to predict the correct object class label from a fixed set of categories.

Importance: Multi-modal classification enhances robustness and generalization in robotic perception, autonomous driving, and AR/VR systems. Single-modality systems often fail under poor lighting, occlusions, or texture-less surfaces, making integrated approaches critical.

Related Work:

1. **DGCNN (Wang et al., 2019)**: Uses dynamic graph convolution to capture local geometric relationships in point clouds, achieving state-of-the-art performance on 3D shape classification and segmentation.
2. **CLIP (Radford et al., 2021)**: Vision-language pretraining model providing strong semantic representations for RGB images through contrastive learning on image-text pairs.
3. **Swin Transformer (Liu et al., 2021)**: Hierarchical vision transformer architecture effectively encoding RGB and depth modalities using shifted windows for efficient and scalable feature extraction.

These works focus on single-modality or bi-modal encoding without an end-to-end transformer-based cross-modal fusion pipeline combining RGB, depth, and geometry systematically. Our project leverages these foundations to implement an integrated SOLID approach.

2 Technical Approach

We propose the following pipeline:

- **RGB:** Encoded using CLIP ViT-B/32 pretrained on image-text pairs.
- **Depth:** Encoded using Swin Transformer Tiny pretrained on ImageNet.

- **Geometry:** Encoded using Dynamic Graph CNN (DGCNN) pretrained on ModelNet40.
- **Fusion:** Cross-modal transformer encoder to integrate features from all modalities into a joint embedding space.
- **Classification Head:** Dense layers with softmax output for final object class prediction.

The model will be implemented in PyTorch with multimodal data loaders to synchronize RGB, depth, and point cloud inputs.

Baselines: We will establish single-modality baselines for comparison:

- **Geometry Only:** DGCNN trained solely on point cloud data.
- **RGB Only:** ResNet-18 trained solely on RGB images.

These baselines will help evaluate the effectiveness of multimodal fusion over single-modality approaches.

3 Evaluation

Metrics:

- **Overall Accuracy** – correct predictions / total samples.
- **Per-Class Accuracy** – class-wise performance.
- **Precision, Recall, F1-Score** – to measure performance under class imbalance.
- **Confusion Matrix** – for visual error analysis.
- **Feature Embedding Visualization** – t-SNE plots to validate modality fusion effectiveness.

Evaluation will be conducted on the test split of SUN RGB-D, with ModelNet40 used primarily for geometry pretraining.

4 Intermediate/Preliminary Results

Dataset: SUN RGB-D (main evaluation dataset), with ModelNet40 used for optional geometry encoder pretraining.

Current Progress:

- Completed detailed design of the multimodal pipeline integrating CLIP ViT-B/32 (RGB), Swin Transformer Tiny (Depth), and DGCNN (Geometry) encoders.
- Developed data loading and preprocessing plans to synchronize RGB images, depth maps, and point cloud geometry from SUN RGB-D.
- Finalized model architecture including cross-modal transformer fusion and classification head.
- Preparing to begin implementation and integration of individual modality encoders and fusion module.
- Established baseline models (PointNet++ for geometry, ResNet-18 for RGB) for future comparative evaluation.

5 Next Steps

By the end of the semester, we plan to:

Next Steps:

- Implement and integrate pretrained encoders for each modality.
- Develop the fusion transformer and classification layers.
- Conduct initial training runs and baseline comparisons.
- Perform debugging, optimization, and data augmentation.

Milestones:

1. **Week 1-2:** Begin implementation and integration of pretrained encoders and the fusion module.
2. **Week 3:** Complete initial training runs and debug the multimodal fusion pipeline.
3. **Week 4:** Conduct baseline comparisons, perform ablation studies, and prepare the progress report.

Potential Challenges:

- Managing GPU memory usage during training of the multimodal transformer – mitigated through mixed precision training and batch size tuning.
- Ensuring proper alignment and synchronization of RGB, depth, and point cloud inputs – addressed through careful preprocessing and data augmentation strategies.
- Integration complexity of multiple pretrained models – mitigated by modular code design and incremental testing.

Final Success Evaluation: The project will be considered successful if the multimodal SOLID pipeline achieves higher accuracy and better generalization than single-modality baselines, as demonstrated by quantitative metrics and qualitative embedding visualizations.