Pix 3D Object Dimension Estimator

MVSK Lalith Kumar - se24maid001
Mihir Varma Chintalapati – se24maid015

Access the Application:
https://tl-dimension-predictor-npfgj9ulyudtepasyvt5x6.streamlit.app/

1 Introduction

Estimating the real-world dimensions of objects from 2D images has become an essential task in various computer vision applications, including augmented reality, robotics, e-commerce visualization, and space planning. Traditionally, achieving this required complex 3D sensors like LiDAR, stereo cameras, or depth scanners. However, with the advancements in deep learning and transfer learning, it is now possible to infer approximate 3D dimensions from a single 2D image — using only RGB data.

This project, titled "3D Object Dimension Estimator", aims to build a deep learning-based system that predicts the Length, Breadth, and Height of an object in meters directly from a 2D image. The system is designed to be lightweight, user-friendly, and accurate for real-world object categories, particularly furniture items like chairs, beds, tables, and shelves.

The core of the model leverages Transfer Learning, where a ResNet-18 convolutional neural network (originally trained on the ImageNet dataset for object classification) is fine-tuned for a new task — regression. Specifically, the final layer of the network is replaced with a custom head that predicts three continuous values: the object's length, breadth, and height. To ensure stable training and better convergence across varied size scales, the model is trained on the logarithmic scale of dimensions, and the predictions are exponentiated at inference time to yield values in meters.

The model is trained using the Pix3D dataset, a richly annotated benchmark dataset for 3D vision tasks. It provides aligned 2D images and 3D CAD models of furniture items, along with accurate bounding boxes and ground-truth dimensions. This allows the model to learn realistic geometric priors of common household objects and perform well under various perspectives and lighting conditions.

Once trained, the model is deployed using Streamlit, a Python-based framework for creating interactive web applications. Users can upload their own images or select from a set of predefined sample images. The application displays the estimated dimensions along with the total volume of the object, making it useful for visualization and decision-making tasks like room arrangement or online product sizing.

This project not only demonstrates the power of deep learning and transfer learning in solving real-world spatial problems but also showcases how to build an end-to-end pipeline — from data preprocessing and model training to deployment and user interaction. The solution, while domain-specific to furniture, lays the foundation for extending to more complex categories such as appliances, tools, and consumer electronics with additional data and training.

2 Libraries and Tools Used

- `streamlit` – Front-end web application framework.

- `torch`, `torchvision` – Deep learning framework and model hub.
- `NumPy` – Efficient numerical computation.
- `Pillow` – Image handling.
- `gdown` – Google Drive model downloads.
- `matplotlib` – Visualization of prediction error.
- `scikit-learn` – Evaluation metrics like MAE, RMSE, and R^2.

3 Dataset: Pix3D – A Precise Dataset for Real-World 3D Vision

Dataset link: https://www.kaggle.com/datasets/ratneshkj/pix3d-dataset

The Pix3D dataset is one of the most reliable and richly annotated datasets for real-world 3D vision tasks such as shape reconstruction, pose estimation, and — in this project's context — dimension estimation. It contains over 10,000 real-world RGB images of furniture items, paired with accurate 3D CAD models, 2D-3D alignment, and ground truth object dimensions.

Pix3D stands out because it provides:

- Pixel-level 2D-3D alignment between images and 3D models
- Real-world, cluttered backgrounds, unlike synthetic datasets
- Accurate bounding boxes, key points, camera intrinsics, and 3D shape metadata
- Multiple images per object under different lighting, angles, and occlusion conditions

3.1 Object Categories in Pix3D:

- Chair, Bed, Table, Desk, Sofa, Bookcase, Wardrobe, Nightstand
- Mostly household furniture, making it ideal for interior design, space planning, or AR tasks

3.2 Why Pix3D Was Chosen for This Project

- **Real-world scenarios:** The images reflect real furniture in homes, not lab or rendered scenes
- **Ground-truth dimensions:** Essential for regression-based prediction of L × B × H
- **Furniture-focused:** Aligned with the scope of the project
- **Bounding boxes + camera data:** Help preprocess and crop inputs accurately

Pix3D hits the sweet spot between visual realism, 3D geometry, and label accuracy — making it a gold standard for this kind of dimension estimation project.

4 Model Architecture and Training

The core of the project leverages Transfer Learning using a ResNet-18 convolutional neural network, which has been pretrained on the ImageNet dataset for large-scale image classification. The architecture is adapted to perform multivariate regression, where the output is a continuous 3D vector representing the estimated Length, Breadth, and Height of an object in meters.

4.1 Model Components

4.1.1 Base Network: ResNet-18 (Pretrained)

- A lightweight yet deep convolutional neural network with residual connections.
- Learns general-purpose visual features like edges, shapes, and textures.
- The initial layers are frozen or reused to reduce training time and data requirements.

4.1.2 Custom Regression Head

- The final classification layer (Linear(512 $\rightarrow$ 1000)) is removed.
- Replaced with a new fully connected layer: Linear(512 $\rightarrow$ 3).
- The output is a 3-element vector representing the log-transformed dimensions [log(L), log(B), log(H)].

4.1.3 Logarithmic Output Scaling

- The model is trained to predict dimensions in log-space to handle wide variation in furniture sizes (e.g., a lamp vs a bed).
- During inference, outputs are exponentiated to get real-world values: [log L, log B, log H] $\rightarrow$ [L, B, H] = e^{output}

4.2 Why ResNet-18?

ResNet-18 was selected because:

- It's compact and computationally efficient for real-time applications.
- Its skip connections preserve low-level spatial detail — important for geometric tasks.
- It provides robust general features when fine-tuned on a small dataset like Pix3D.

4.3 Training Process

Dataset: Pix3D (Furniture category)
Input: RGB images, cropped using bounding boxes
Output: Log-scaled ground truth dimensions (L, B, H)
Key Training Details:

- **Loss Function:** Mean Squared Error (MSE) on log-transformed targets
- **Optimizer:** Adam optimizer with a learning rate of 1e-4
- **Epochs:** $\sim$15–30 epochs depending on convergence
- **Batch Size:** 16–32
- **Image Size:** Resized to 224$\times$224 pixels for ResNet input
- **Cropping:** Objects are cropped using bounding box annotations before feeding into the network

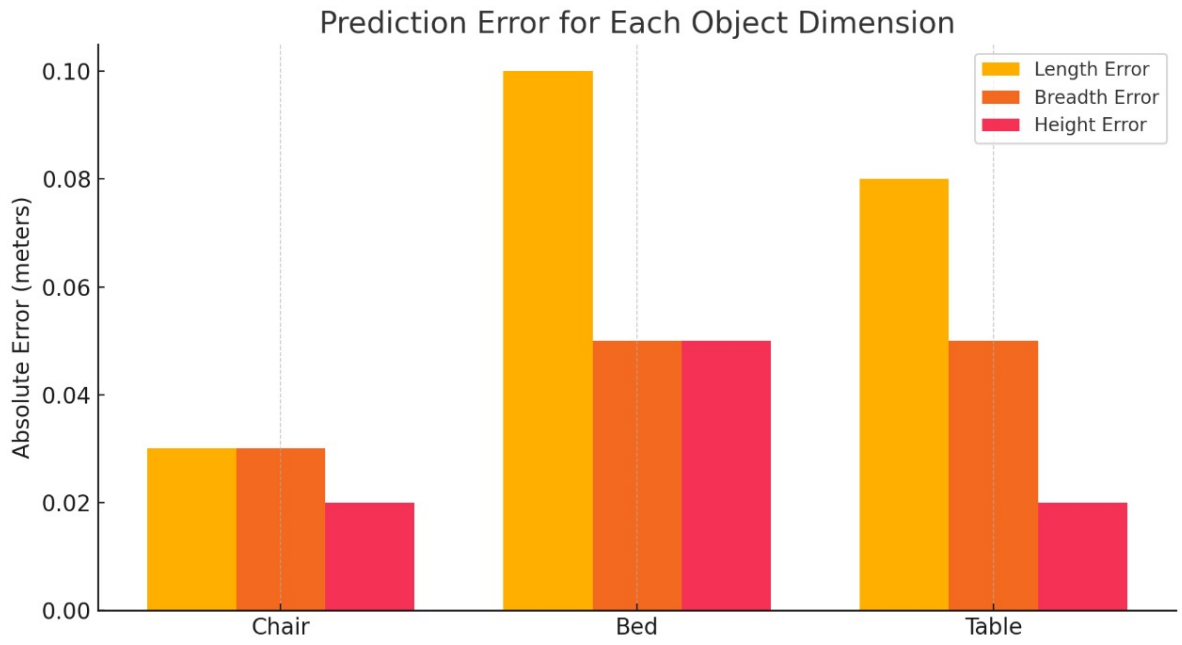

Figure 1: Prediction Error For Each Object Dimension

4.4 Inference Pipeline

1. User uploads or selects an image.
2. If available, the object is cropped using the bounding box.
3. The image is resized and normalized using ResNet preprocessing.
4. The model predicts [log L, log B, log H].
5. Output is exponentiated → final predicted dimensions.
6. Volume is calculated as: Length × Breadth × Height.

5 Sample Prediction Error Visualization

- Compares Length, Breadth, and Height prediction errors for 3 furniture objects: Chair, Bed, and Table.
- Each bar represents the absolute error (in meters) for that dimension.
- Makes it easy to see where the model performs well vs. where it struggles (e.g., breadth of the table shows higher error).

6 How to Run the Application

6.1 Step 1: Install Requirements

```
pip install -r requirements.txt
```

6.2 Step 2: Run the App

```
streamlit run app.py
```

6.3 Step 3: Upload or Select Image

Once the app opens in your browser:

- Choose "Upload your own" to test a new image
- Or select a sample from the "pix3d-samples" folder

The app will:

- Preprocess the image (crop, resize)
- Pass it through the pretrained ResNet-18 model
- Predict Length, Breadth, Height
- Calculate Volume
- Display all results in meters

6.4 How the Model Works (Behind the Scenes)

1. Loads a fine-tuned ResNet-18 model from Google Drive (trained_model.pth)
2. Replaces the final classification layer with a regression head
3. Accepts a single image input and preprocesses it (224×224, normalized)
4. Outputs log-scaled [L, B, H], then exponentiates them to get meters
5. Displays results via Streamlit UI

7 Future Improvements

The current system shows promising results on furniture images, but several enhancements can elevate its accuracy and applicability. Incorporating monocular depth estimation (e.g., MiDaS) alongside RGB inputs can significantly improve spatial awareness. Using object category labels as auxiliary inputs would help the model learn size priors specific to each furniture type. Enhancing the loss function with aspect ratio or volume consistency terms can address issues in shape proportionality. Training on augmented or synthetic multi-view data can improve generalization to different environments. Expanding the dataset beyond furniture to include household and consumer objects (e.g., bottles, phones, tools) would make the model more versatile. Adding uncertainty estimation can make predictions more reliable, while optimization techniques like quantization could enable real-time deployment on mobile or edge devices.

8 Conclusion

This project successfully demonstrates the use of deep learning and transfer learning to estimate real-world object dimensions (length, breadth, and height) from a single 2D image. By fine-tuning a pretrained ResNet-18 model on the Pix3D dataset, we built a regression system capable of predicting physical measurements for various furniture items with reasonable accuracy. The log-scale training, bounding box preprocessing, and Streamlit-based deployment contribute to both model robustness and user accessibility. While the system currently focuses on furniture, its modular design allows for easy extension to other object types with additional data. The project showcases the practicality of combining computer vision, regression modeling, and web-based deployment for solving spatial inference problems in real-world scenarios.

THANK YOU