2D to 3D

# Possible Approaches

**1. NeRF (Neural Radiance Fields)**

* **Best for:** Generating high-quality 3D objects from multiple 2D images.
* **How it works:** NeRF trains a neural network to model a 3D scene by predicting the color and density of points in space.
* **Requirements:** Multiple images from different angles.
* **Tools:** PyTorch/TensorFlow, Instant-NGP (NVIDIA), or Nerfstudio.

**2. DeepVoxels & Occupancy Networks**

* **Best for:** Single-image to 3D reconstruction.
* **How it works:** Uses voxel grids or signed distance functions (SDFs) to generate 3D structures.
* **Requirements:** Training on 2D-3D datasets.
* **Tools:** PyTorch, Open3D.

**3. GANs (Generative Adversarial Networks)**

* **Best for:** Generating 3D objects from images.
* **How it works:** GANs like Pix2Vox and 3D-GANs predict 3D voxel representations from 2D images.
* **Tools:** PyTorch, TensorFlow.

**4. Structure from Motion (SfM) + Neural Networks**

* **Best for:** If you want to use video input.
* **How it works:** SfM extracts 3D structure from multiple video frames, and deep learning refines it.
* **Tools:** COLMAP (for SfM) + Deep Learning.

# Steps

**Phase 1: Research & Setup**

1. **Decide on Input Type** – Will you use single images, multiple images or video?
2. **Choose a Framework** – PyTorch or TensorFlow?
3. **Set Up Development Tools** – Install OpenCV, Open3D and any necessary deep-learning libraries.

**Phase 2: Data Collection**

1. **Dataset Selection** – Get a dataset like ShapeNet, Pascal3D or Google Scanned Objects.
2. **Preprocessing** – Convert 2D images to a format suitable for training (resize, normalize).

**Phase 3: Model Selection & Training**

1. **Start with a Pre-trained Model** – Use an existing NeRF, GAN or Vox-based model.
2. **Train on Your Data** – Fine-tune the model on your own images.
3. **Test & Evaluate** – Use metrics like IoU (Intersection over Union) for accuracy.

**Phase 4: Visualization & Output**

1. **Generate 3D Models** – Convert to mesh (.obj, .stl) using Open3D or Meshlab.
2. **Visualize** – Render results using Blender or Three.js.

[Papers-in-100-Lines-of-Code/NeRF\_Representing\_Scenes\_as\_Neural\_Radiance\_Fields\_for\_View\_Synthesis at main · MaximeVandegar/Papers-in-100-Lines-of-Code](https://github.com/MaximeVandegar/Papers-in-100-Lines-of-Code/tree/main/NeRF_Representing_Scenes_as_Neural_Radiance_Fields_for_View_Synthesis)