AN APPROACH: COMPUTER VISION INDOOR SCENE UNDERSTANDING IN 2.5RGB-D AND 3D

ABSTRACT

The article provides a background on visual scene understanding of indoor environments and what method can be applied on each stage. The article covers at a high-level indoor scene understanding as well as subtask such as scene classification, object detection, pose estimation, semantic segmentation, 3D reconstruction, saliency detection, physics-based reasoning, and affordance prediction. The performance metrics used for evaluation in different tasks can be seen at last of section and a quantitative comparison among few techniques. Before proceeding it would be better to highlight that the article is not having any codes that will be required to achieve the results for indoor object detection but discusses on techniques and approach for the same. The code can be seen with each section and can be referred independently. For detailed code feel free to reach out on respective github. Also, it is assumed that reader is familiar with CNN, RNN, Encoder decoder architecture, MRF (Markov random field), Decision Forest (Not decision trees), and SVM as we will be comparing these techniques and using at various stages.

INTRODUCTION

An image to a machine can be represented as nothing but a grid of numbers. In order to develop a model which has a comprehensive understanding of visual content, it is necessary to uncover the underlying geometric and semantic clues between various scene elements present in its field of vision. it is also required to comprehend both the apparent and hidden relationships present between scene elements.

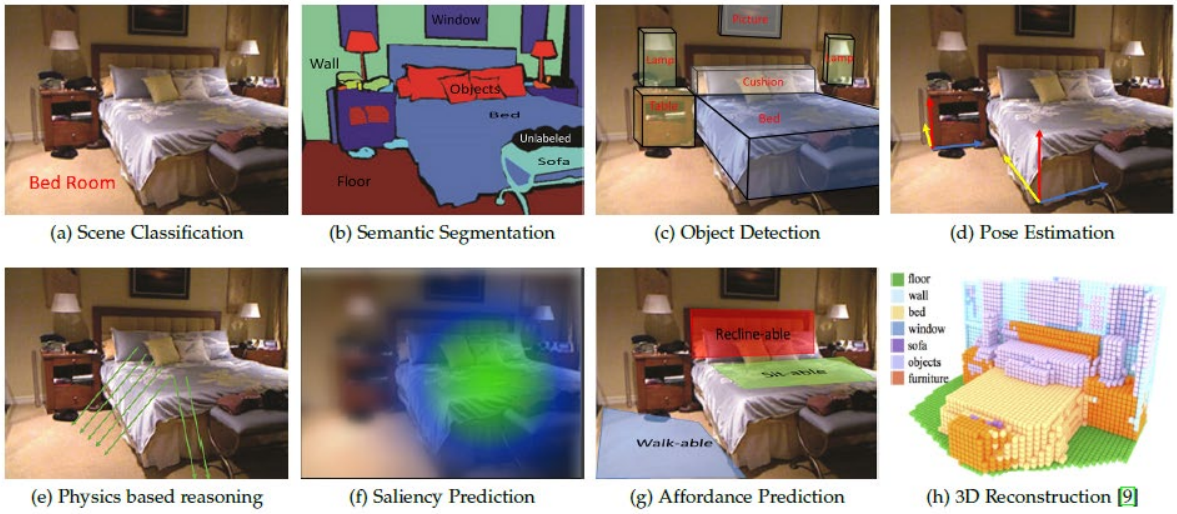
The first question that comes to our mind is what we mean by scene understanding:

It can be defined as “To analyze a scene by carrying out object detection, considering the geometric and semantic context of its contents, motion and action recognition and finally followed by defining intrinsic relationships between different elements”

BACKGROUND AND SCOPE FOR THIS PAPER

The Visual scene understanding can be broadly categorized into either static understanding (image based) or dynamic understanding (video based).

The scope for this publication is in Static scene understanding RGB-D image. Processing 2.5D or 3D visual data, also covering both the specific problem domains under the umbrella of scene understanding as well as solutions to various scene analysis problems. (refer figure below for better understanding on areas covered):



(a) Scene Classification (b) Semantic Segmentation (c) Object Detection (d) Pose Estimation

(e) Physics based reasoning (f) Saliency Prediction (g) Affordance Prediction (h) 3D Reconstruction [9]

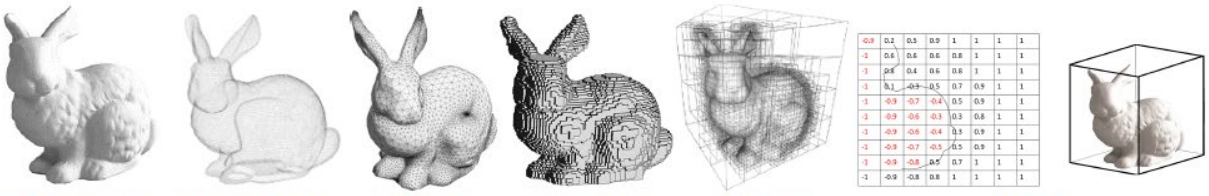
Fig. 1: Given a RGB-D image, visual scene understanding can involve the image and pixel level semantic labeling (a-b), 3D object detection and pose estimation (c-d), inferring physical relationships (e), identifying salient regions (f), predicting affordances (g), full 3D reconstruction (h) and holistic reasoning about multiple such tasks (sample image from the NYU-Depth dataset [10]).

An image or a video is a tensor with numeric values representing color (like r, g, b values) or location (like x, y, z space coordinates) information. Such information can be processed by computing local features representing color and texture characteristics. Several local feature descriptors have been designed over the years to faithfully encode visual information like: SIFT, HOG, SURF, Region Covariance, LBP etc.

To process the features, first thing required is to see the various ways in which the internal scene data can be represented. Below section has underlying data representations and datasets for RGB-D and 3D data.

DATA REPRESENTATIONS:

Below is the list of popular 2.5D and 3D data representations used for analyzing internal scenes.



(a) CAD Model (b) Point Cloud (c) Mesh (d) Voxelized (e) Octree (f) TSDF

Fig. 2: Visualization of different types of 3D data representations for Stanford bunny.

Point Cloud: A 'point cloud' is a collection of data points in 3D space. The combination of these points can be used to describe the geometry of the individual object or the complete scene. Every point in the point cloud is defined by x, y and z coordinates

Voxel Representation: A voxel (volumetric element) is the 3D counterpart of a pixel (picture element) in a 2D image. Voxelization is a process of converting a continuous geometric object into a set of discrete voxels that best approximate the object. Usually, a voxel value is mapped to either 0 or 1, where 0 indicates an empty voxel while 1 indicates the presence of range points inside the voxel.

3D Mesh: The mesh representation encodes a 3D object geometry in terms of a combination of edges, vertices, and faces. A commonly used mesh is a triangular mesh that is composed entirely of triangle shaped faces

Depth Channel and Encodings: A depth channel in a 2.5D representation shows the estimated distance of each pixel from the viewer. geocentric embedding encodes depth image using height above the ground, horizontal disparity, and angle with gravity for each pixel.

Octree Representations: An octree is a voxelized representation of a 3D shape that provides high compactness. The underlying data structure is a tree where each node has eight children

Stixels: The idea of stixels is to reduce the gap between pixel and object level information, thus reducing the number of pixels in a scene to few hundreds

Truncated Signed Distance Function: Truncated signed distance function (TSDF) is another volumetric representation of a 3D scene. Instead of mapping a voxel to 0 or 1, each voxel in the 3D grid is mapped to the signed distance to the nearest surface. The signed distance is negative if the voxel lies with in the shape and positive otherwise. Technologies like Kinect use this to obtain a complete 3D model.

Constructive Solid Geometry: Constructive solid geometry (CSG) is a building block technique in which simple objects such as cubes, spheres, cones, and cylinders are combined with a set of operations such as union, intersection, addition, and subtraction to model complex objects. CSG is represented as a binary tree with primitive shapes and the combination operations as its nodes. This representation is often used for CAD models in computer vision and graphics.

DATASETS

High quality datasets play important role in development of machine vision algorithms. Here, we review important datasets, below table, for scene understanding available for researching.

Dataset	NYUv2 [10]	SUN3D [29]	SUN RGB-D [30]	Building Parser [31]	Matterport 3D [32]	ScanNet [33]	SUNCG [34]	RGBD Object [35]	SceneNN [36]	SceneNet RGB-D [37]	PiGraph [38]	TUM [39]	Pascal 3D+ [40]
Year	2012	2013	2015	2017	2017	2017	2016	2011	2016	2016	2016	2012	2014
Type	Real	Real	Real	Real	Real	Real	Synthetic	Real	Real	Synthetic	Synthetic	Real	Real
Total Scans	464	415	-	270	-	1513	45,622	900	100	57	63	39	-
Labels	1449 images	8 scans	10k images	70k images	194k images	1513 scans	130k images	900 scans	100 scans	5M images	21 scans*	39 scans	24k images
Objects/Scenes	Scene	Scene	Scene	Scene	Scene	Scene	Scene	Object	Scene	Scene	Scene	Scene	Object
Scene Classes	26	254	47	11	61	707	24	-	-	5	30	✗	-
Object Classes	894	-	800	13	40	50 - at least	84	51	50 - at least	255	5 subjects	✗	12
In/Outdoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	Indoor	In+Out
Available Data Types													
RGB	✓	-	✓	✓	✓	-	-	✓	-	✓	✗	-	✓
Depth	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✗	-	✗
Video	✓	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓	✗
Point cloud	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	-	✗	✗
Mesh/CAD	✗	✗	✗	✓	✓	✓	✓	✗	✓	✗	-	✗	✓
Available Annotation Types													
Scene Classes	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗
Semantic Label	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗
Object BB	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✓
Camera Poses	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Object Poses	✓	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
Trajectory	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗
Action	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

-: means information not available, ∗: Average reported; 4.9 actions annotated per scan and there are 298 actions with 8.4s length available.

CORE TECHNOQUES

Below is an overview of the core techniques employed for various scene understanding problems.

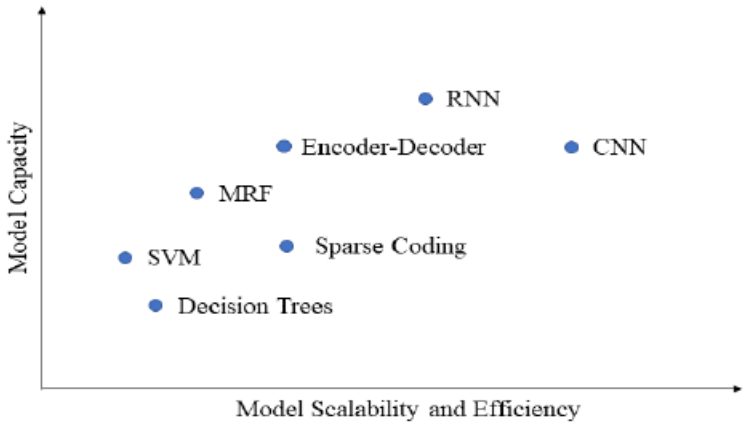


Fig. 3: Core-Techniques Comparison.

This was compared based on different tests on large and small datasets and average the results with overall effect on output.

APPROACH TO MODELLING

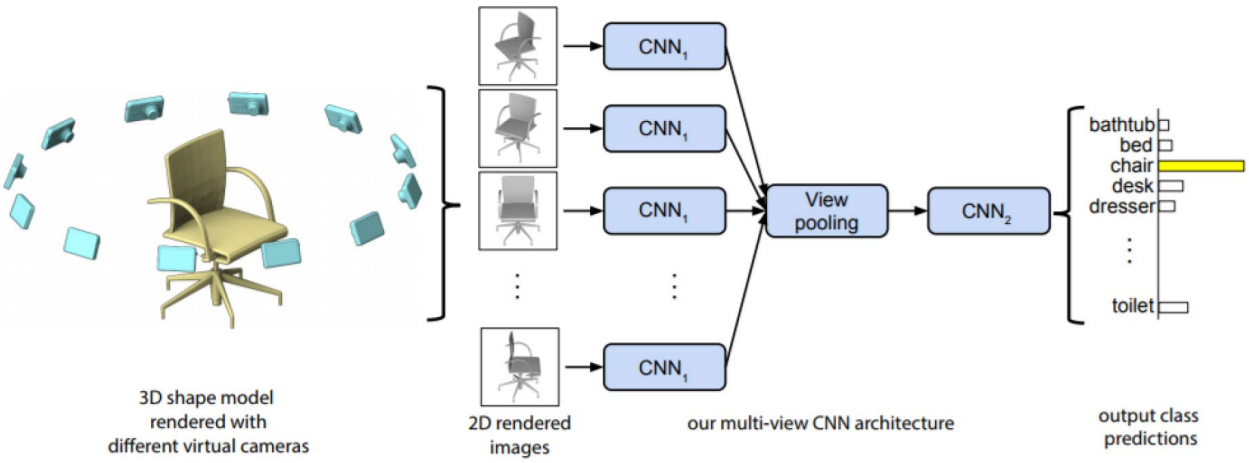
STEP-1: IMAGE RECOGNITION

The first task that we need to work on in visual scene understanding is image recognition. This will further lay down steps for Information about the scene or object category and can help in more sophisticated tasks such as scene segmentation and object detection. The key challenges in image recognition are:

- 2.5/3D data can be represented in multiple ways as discussed above. Challenge then is to choose the data representation that provides maximum information with minimum computational complexity.
- A key challenge is to distinguish between fine grained categories and appropriately model intraclass variations.
- Designing algorithms that can handle illuminations, background clutter and 3D deformations.
- Designing algorithm that can learn from limited data

METHOD OVERVIEW:

Using Multi View CNN architecture to recognize 3D shapes that can be trained on 2D rendered views. Using the Phong reflection method to render 2D views of 3D shapes. Afterwards, a pre-trained VGG-M network was fine-tuned on these rendered views. To aggregate the complementary information across different views, each rendered view was passed through the first part of the network (CNN1) separately, and the results across views were combined using element-wise maximum operation at the pooling layer before passing them through the rest of network (CNN2, see Figure below). MVCNN thus combines the multiple view information to better recognize 3D shapes. For full code access on how MVCNN works click here: https://github.com/suhangpro/mvcnn



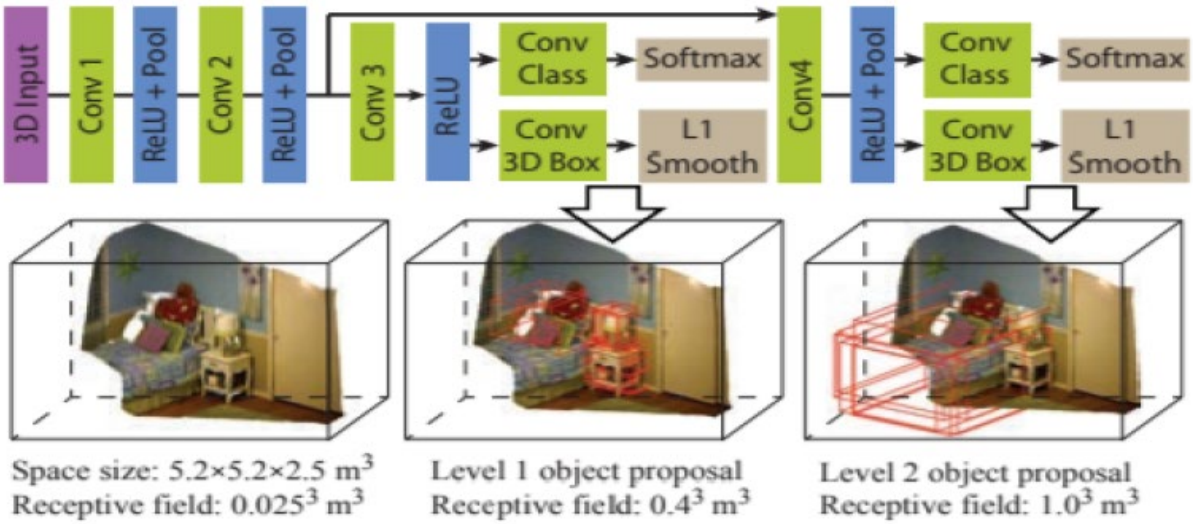
STEP-2: OBJECT DETECTION

Object detection deals with recognizing object instances and categories. Usually, an object detection algorithm outputs both the location (defined by a 2/3D bounding box around the visible parts of an object instance) and the class of an object, e.g., sofa, chair. Key challenges for object detection are:

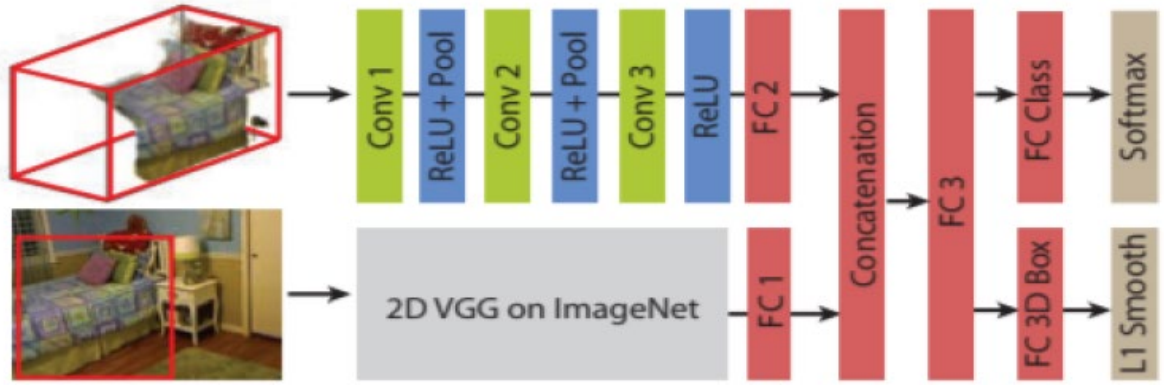
- Real world environments can be highly cluttered and object identification in such environments is very
- challenging.
- Detection algorithm should also be able to handle viewpoint and illuminations variations and deformations.
- In many scenarios, it is necessary to understand the scene context to successfully detect objects.
- Objects categories have a long-tail (imbalanced) distribution, which makes it challenging to model the infrequent classes.

METHOD OVERVIEW:

Typically, an object detection algorithm produces a bounding box on visible parts of the object on an image plane, but for practical reasons, it is desirable to capture the full extent of the object regardless of occlusion or truncation. Use three deep network architectures to produce object category labels along with 3D bounding boxes. First, a 3D network called Region Proposal Network (RPN) takes a 3D volume generated from depth map and produces 3D regional proposals for the whole object. Each region proposal is feed into another 3D convolutional net, and its 2D projection is fed to a 2D convolutional network to jointly learn color and depth features. The final output is the object category along with the 3D bounding box. A limitation of this work is that the object orientation is not explicitly considered. Check the figure below to see how the network works:



(a) 3D Region Proposals Network.



(b) Object Detection and 3D box regression Network.

For code click here: https://github.com/shurans/DeepSlidingShape

STEP-3: SEMANTIC SEGMENTATION

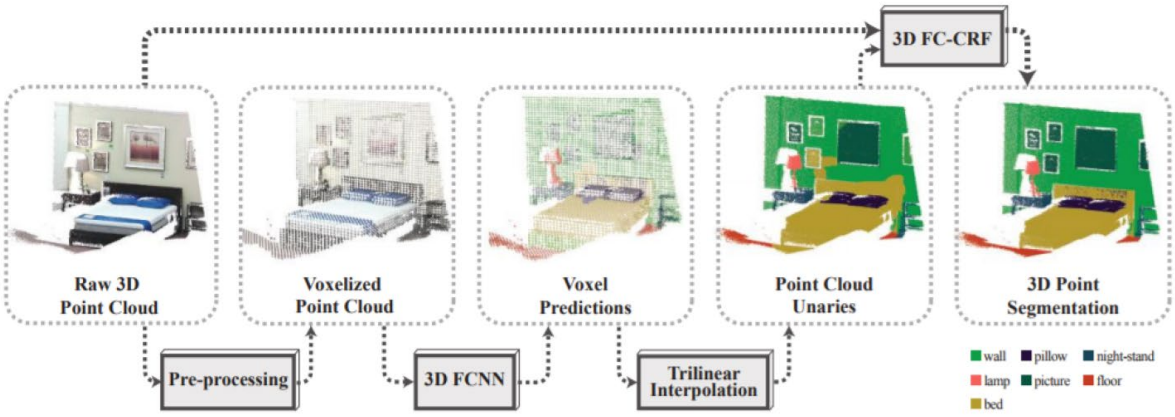
This task relates to the labeling of each pixel in an image with its corresponding semantically meaningful category. The key challenges for semantic segmentation is :

- Pixel level labeling requires both local and global information and challenge then is to design such algorithms that can incorporate the wide contextual information together.
- The difficulty level increases a lot for the case of instance segmentation, where the same class is segmented into different instances.
- Obtaining dense pixel level predictions, especially close to object boundaries, is challenging due to occlusions and confusing backgrounds.
- Segmentation is also affected by appearance, viewpoint, and scale changes

METHOD OVERVIEW:

Instead of solely using deep networks for context modeling, we can combine both CNN and CRFs for improved segmentations. As an example, a 3D point cloud segmentation combines the FCN and a fully connected CRF model which helps in better contextual modeling at each point in 3D.

To enable a fully learnable system, the CRF is implemented as a differentiable recurrent network. Local context is incorporate in the proposed scheme by obtaining a voxelized representation at a coarse scale, and the predictions over voxels are used as the unary potentials in the CRF model (see Figure below).



Refer to this link for detailed approach: http://segcloud.stanford.edu/

STEP-4 PHYSICS BASED REASONING

A scene is a static picture of the visual world. However, when humans look at the static image, they can infer hidden dynamics in a scene. As an example, from a still picture of a football field with players and a ball, we can understand the pre-existing motion patterns and guess the future events which are likely to happen in a scene. As a result, we can plan our moves and take well-informed decisions. In line with this human cognitive ability, efforts have been made in computer vision to develop an insight into the underlying physical properties of a scene. The key challenges that has been faced for physics-based reasoning is:

- This task requires starting with extremely limited information (e.g., a still image) and performing extrapolation to predict rich information about scene dynamics.
- A desirable characteristic is to adequately model prior information about the physical world.
- Physics based reasoning requires algorithms to reason about the contextual information

METHOD OVERVIEW:

The method that can be incorporated into a code can be broken down into multiple areas:

1. Dynamics prediction
2. Support relationships
3. Stability Analysis
4. Hazard detection
5. Occlusion reasoning

Each area has its own specialization of methods and approaches which cannot be generalized and will depend on the application at hand whether the code to be designed is to be applied on a game engine, AR-VR application, house help auto agents etc. Like for example for dynamics prediction we can use something like N-cube (Newtonian Neural Network) to find the scenario that best describes object motion in an image etc.

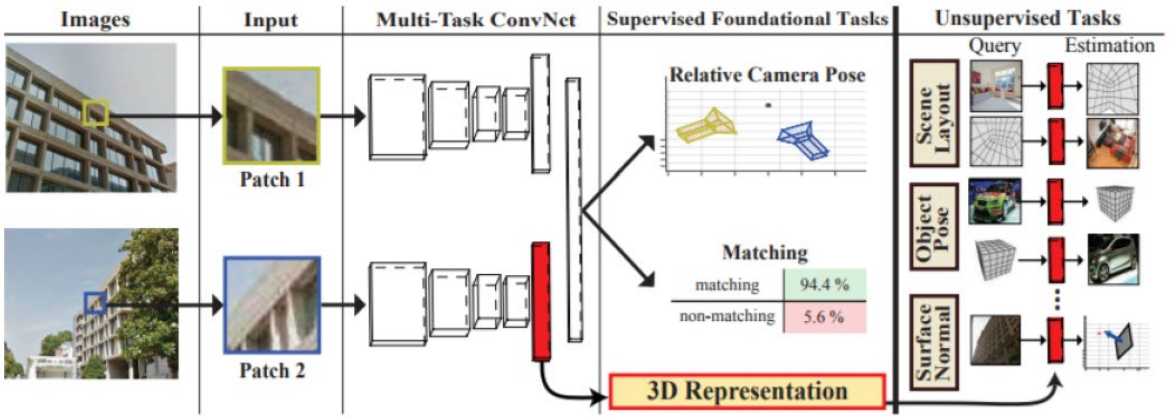
STEP-5: OBJECT POSE ESTIMATION

The pose estimation task deals with finding object's position and orientation with respect to a specific coordinate system. Information about an object's pose is crucial for object manipulation by robotic platforms and scene reconstruction. The key challenges for pose estimation algorithms encounter are:

- The requirement of detecting objects and estimating their orientation at the same makes this task particularly challenging.
- Object's pose can vary significantly from one scene to another; therefore algorithm should be invariant to these changes.
- Occlusions and deformations make the pose estimation task difficult especially when multiple objects are simultaneously present

METHOD OVERVIEW:

CNN models have an extraordinary ability to learn generic representations that are transferable across tasks. We can train a CNN to learn 3D generic representations to simultaneously address multiple tasks. In this regard, we can train a multi-task CNN to jointly learn camera pose estimation and key point matching across extreme poses and showed with extensive experimentation that internal representation of such a trained CNN can be used for other predictions tasks such as object pose, scene layout and surface normal estimation (see Figure below for reference)



For code link click here: https://github.com/yuxng/PoseCNN

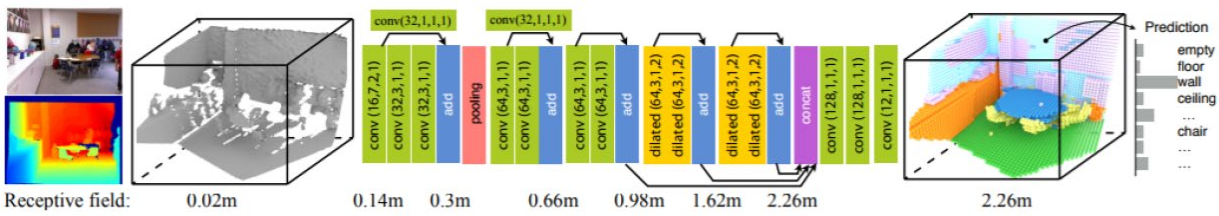
STEP-6: 3D RECONSTRUCTION FROM RGB-D

Humans visualize and interpret surrounding environments in 3D. The 3D reasoning about an object or a scene allows a deeper understanding of the mechanics, shape and 3D texture characteristics. For this purpose, it is often desirable to recover the full 3D shape from a single or multiple RGB-D images. The key challenges for 3D reconstruction are:

- Complete 3D reconstruction from incomplete information is an ill-posed problem with no unique solution.
- This problem poses a significant challenge due to sensor noise, low depth resolution, missing data and quantization errors.
- It requires appropriately incorporating external information about the scene or object geometry for a successful reconstruction

METHOD OVERVIEW:

A 3D CNN can be used to jointly perform semantic voxel labeling and scene completion from a single RGB-D image. The CNN architecture makes use of successful ideas in deep learning such as skip connections and dilated convolutions to aggregate scene context and the use of a large-scale dataset (like SUNCG). Refer below image to see how 3D reconstruction happens using CNN:



For full code refer here: https://github.com/shurans/sscnet

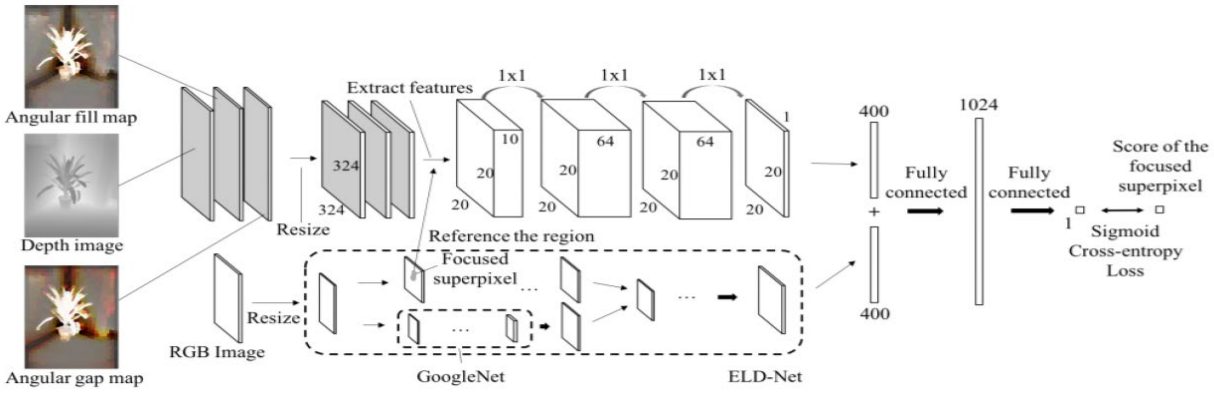
STEP-7: SALIENCY PREDICTION

The human visual system selectively attends to salient parts of a scene and performs a detailed understanding for the most salient regions. The detection of salient regions corresponds to important objects and events in a scene and their mutual relationships. The key challenges for saliency prediction task are:

- Saliency is a complex function of different factors including appearance, texture, background properties, location, depth etc. It is a challenge to model these intricate relationships.
- It requires both top-down and bottom-up cues to accurately model objects saliency.
- A key requisite is to adequately encode the local and global context.

METHOD OVERVIEW:

Based on the insight that salient objects are likely to appear at different depths, we can use a multistage model where local, global and background contrast-based cues were used to predict a rough estimate of saliency. This initial saliency estimate can be used to calculate a foreground probability map which was combined with an object prior to generate final saliency predictions. It augments the high-level feature description from a pre-trained CNN with a few low-level feature descriptions such as the depth contrast, angular disparity, and background enclosure



For code refer here: https://github.com/sshige/rgbd-saliency

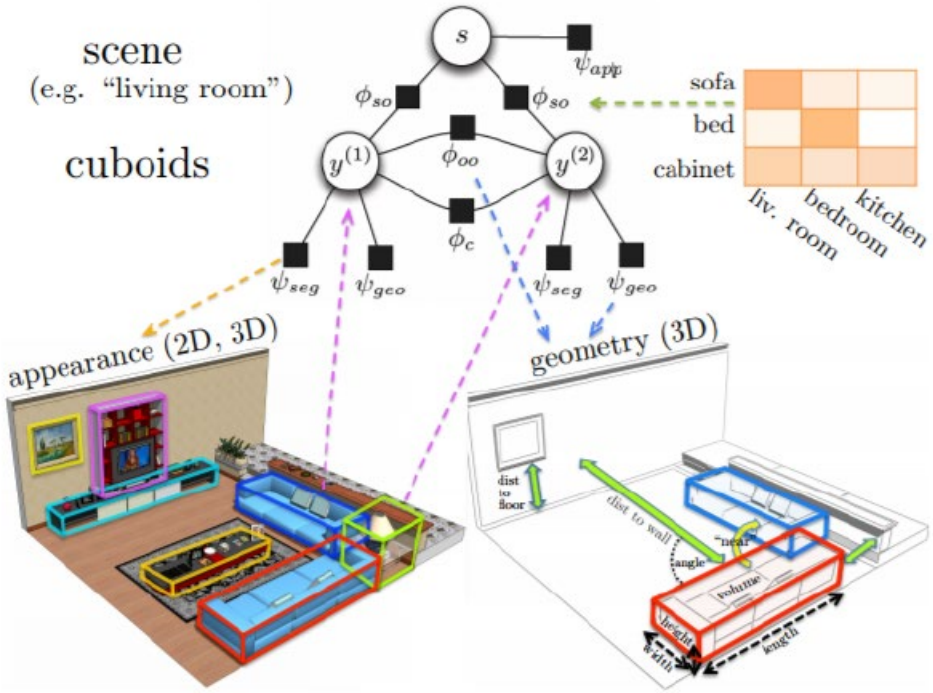
STEP-8: HOLISTIC OR HYBRID APPROACHES:

In holistic scene understanding, a model aims to simultaneously reason about multiple complimentary aspects of a scene to provide a detailed scene understanding. Such an integration of individual tasks can lead to practical systems which require joint reasoning. The Key challenges for Holistic or Hybrid approaches are:

- Accurately modeling relationships between objects and background is a hard task in real-world environments due to the complexity of inter-object interactions.
- Efficient training and inference is difficult due to the requirement of reasoning at multiple levels of scene decomposition.
- Integration of multiple individual tasks and complementing one source of information with another is a key challenge.

METHOD OVERVIEW:

From step-1 to step-7 we have seen individual approach for each task for holistic approach we can combine multiple models to create a holistic graphical model. We can use something like a CRF to integrate scene geometry, relations between objects, interaction of objects with scene environment for 3D object recognition (see Figure below):



Refer to the link for detailed deep vision codes and articles: https://github.com/kjw0612/awesome-deep-vision

METRICS

STEP-1 CLASSIFICATION:

Classifier performance can be measured by classification accuracy as follows

$$\text{Accuracy} = \frac{\text{Number of samples correctly classified}}{\text{Total number of samples}}$$

STEP-2 OBJECT DETECTION:

Object detection is the task of recognizing each object instance and its category. Average precision (AP) is a commonly used metric to measure an object detector's performance: (TP is true positive and FP is false positive)

$$\text{AP} = \frac{1}{\text{classes}} \sum_{i \in \text{classes}} \frac{TP(i)}{TP(i) + FP(i)},$$

STEP-3 POSE ESTIMATION

Objects pose estimation task deals with finding object's position and orientation with respect to a certain coordinate system. The percentage of correctly predicted poses is the efficiency measure of a pose estimator. A pose estimation is considered correct if average distance between the estimated pose and ground truth is less than a specific threshold

STEP-4 SALIENCY PREDICTION

Saliency prediction deals with the detection of important objects and events in a scene. There are many evaluation metrics for saliency prediction including Similarity, Normalized Scanpath Saliency (NSS) and F-measure (F-beta).

$$\text{Similarity} = SM - FM$$

$$NSS(p) = \frac{SM(p) - \mu_{SM}}{\delta_{SM}}$$

mu is the mean value of predicted saliency map and delta is the standard deviation of the predicted saliency map

$$F_\beta = \frac{(1+\beta^2) * (percision) * (recall)}{\beta^2 * (percision) + (recall)}$$

Beta -Square is a hyper parameter set to normally 0.3

STEP-5: SEGMENTATION EVALUATION

Semantic segmentation is the task that involves labeling each pixel in a given image by its corresponding class:

$$\text{Pixel Accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i}$$

$$\text{Mean Accuracy} = \left(\frac{1}{n_{cl}}\right) \sum_i \frac{n_{ii}}{t_i}$$

$$\text{MIoU} = \left(\frac{1}{n_{cl}}\right) \sum_i \frac{n_{ii}}{\left(t_i + \sum_j n_{ji} - n_{ii}\right)}$$

$$\text{FIoU} = \left(\sum_k t_k\right)^{-1} \sum_i \frac{t_i n_{ii}}{\left(t_i + \sum_j n_{ji} - n_{ii}\right)},$$

Where, MIoU stands for Mean Intersection over Union, FIoU denotes Frequency weighted Intersection over Union, n_cl are the number of different classes, n_ii is the number of pixels of class i predicted to belong to class i, n_ji is the number of pixels of class i predicted to belong to class j and t_i is the total number of pixels belong to class i.

STEP-6: AFFORDANCE PREDICTION

Affordance is the ability of a robot to predict possible actions that can be performed on or with an object. The common evaluation metric for affordance is the accuracy:

$$\text{Accuracy} = \frac{\text{Number of affordances correctly classified}}{\text{Total number of affordances}}.$$

STEP-7: 3D RECONSTRUCTION

3D reconstruction is a task of recovering full 3D shape from a single or multiple RGB-D images. Intersection over union is commonly used as an evaluation metric for the 3D reconstruction task:

$$\text{IoU} = \sum_i \frac{v_{ii}}{\left(T_i + \sum_j v_{ji} - v_{ii}\right)}$$

where v_ii is the number of voxels of class i predicted to belong to class i, v_ji is the number of voxels of class I predicted to belong to class j and Ti is the total number of voxels belong to class i.

CONCLUSION

Computer vision and Deep vision is a branch fairly in its infancy and have evolved considerably in past decade itself. The approaches outlined are the ones that have been recent and are being used generally in 2014-2020. These approaches might be replaced with better algorithms as more and more research is being carried out in these areas. If you are planning to build an application or model which needs to detect indoor environment, then approach outlined above can be especially useful for preliminary design and can be expanded from there on itself based on specific use case for your application or Robotic agents.

Further Reading: https://github.com/kjw0612/awesome-deep-vision

REFERENCES:

Original Paper Link: https://arxiv.org/abs/1803.03352

Supported Link-1: http://3ddl.cs.princeton.edu/2016/slides/funkhouser.pdf

Link-2 : https://www.researchgate.net/publication/276916013_Indoor_Scene_Understanding_with_RGB-D_Images_Bottom-up_Segmentation_Object_Detection_and_Semantic_Segmentation

Link-3: https://github.com/suhangpro/mvcnn

Link-4: https://github.com/shurans/DeepSlidingShape

Link-5: https://github.com/shurans/sscnet

Link-6: http://segcloud.stanford.edu/

Link-7: https://github.com/yuxng/PoseCNN

Link-8: https://github.com/sshige/rgbd-saliency

Link-9: https://blesaux.github.io/files/talk-BLS-2020-esrin-deeplearn4scene.pdf

Further reading: https://github.com/kjw0612/awesome-deep-vision