RGBD Occlusion Detection via Deep Convolutional Neural Networks

Soumik Sarkar[1,2], Vivek Venugopalan[1], Kishore Reddy[1],
Michael Giering[1], Julian Ryde[3], Navdeep Jaitly[4,5]

[1]*United Technologies Research Center, East Hartford, CT*
[2]Currently with *Iowa State University, Ames, IA*
[3]*United Technologies Research Center, Berkeley, CA*
[4]*University of Toronto, ON, Canada*
[5]Currently with *Google Inc., Mountain View, CA*

United Technologies

UTC Climate, Controls & Security

Otis

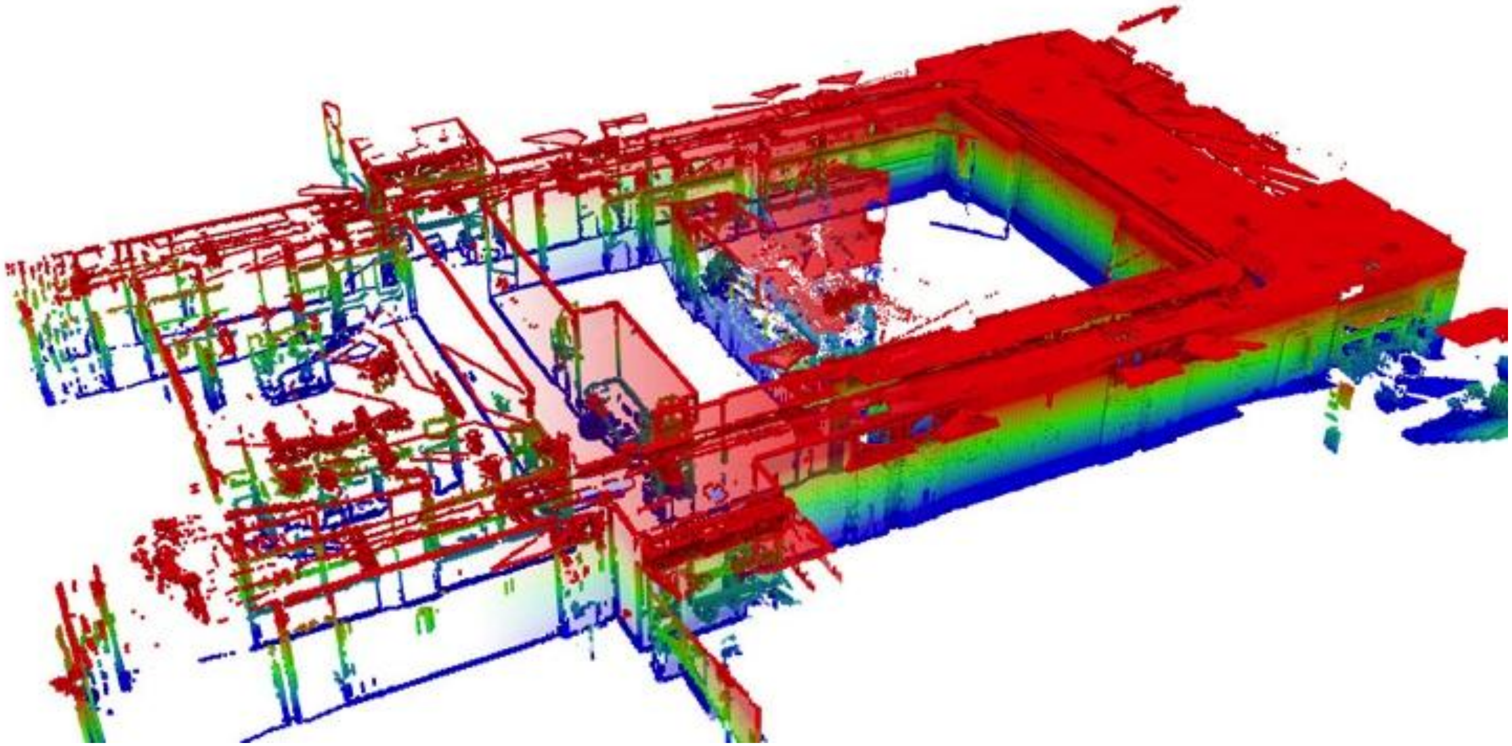
Sikorsky

UTC Propulsion & Aerospace Systems

Pratt & Whitney

UTC Aerospace Systems

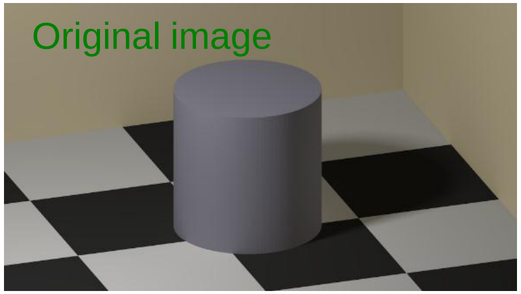
Occlusion detection

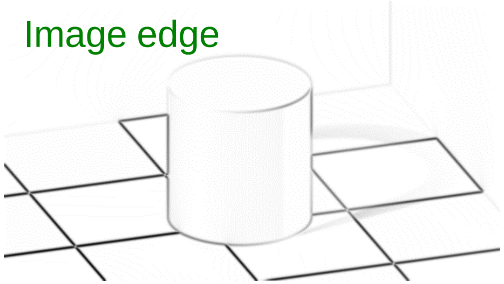


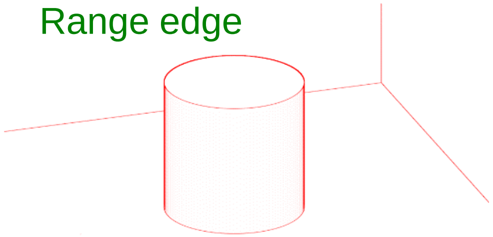
A voxel map and the corresponding geometric edges for a hallway

- Occlusion edges help image feature selection, once occlusion boundaries are established – the depth of the region can be determined
- This is very useful in Simultaneous localization and mapping (SLAM) problems in robotics applications for indoor environments, object recognition, grasping, obstacle avoidance in UAV applications, etc.

Occlusion detection

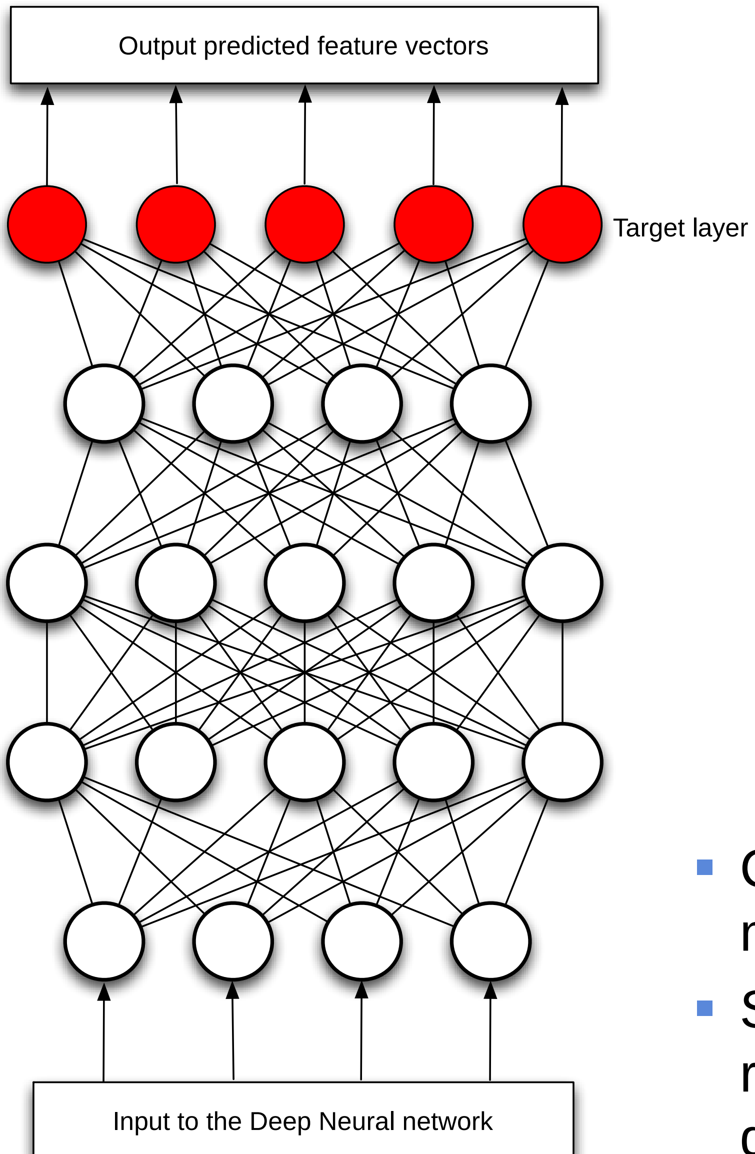


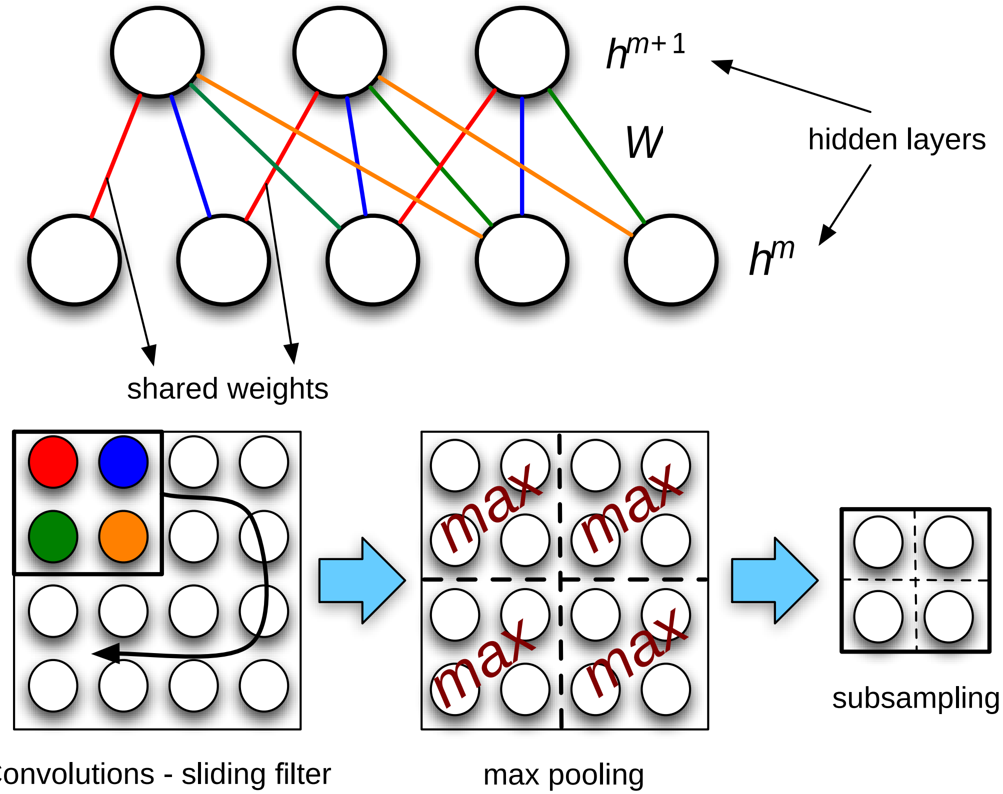




- Occlusion edges depend on the gradient of the depth image which is very sensitive to noise in the depth map
- The depth map derived from a single image is very noisy and has large errors.
- In our work, we are estimating the occlusion edges directly rather than estimating depth first and then calculating occlusion edges. Secondly there are additional cues other than depth which contribute to establishing occlusion edges that our technique is taking advantage of.

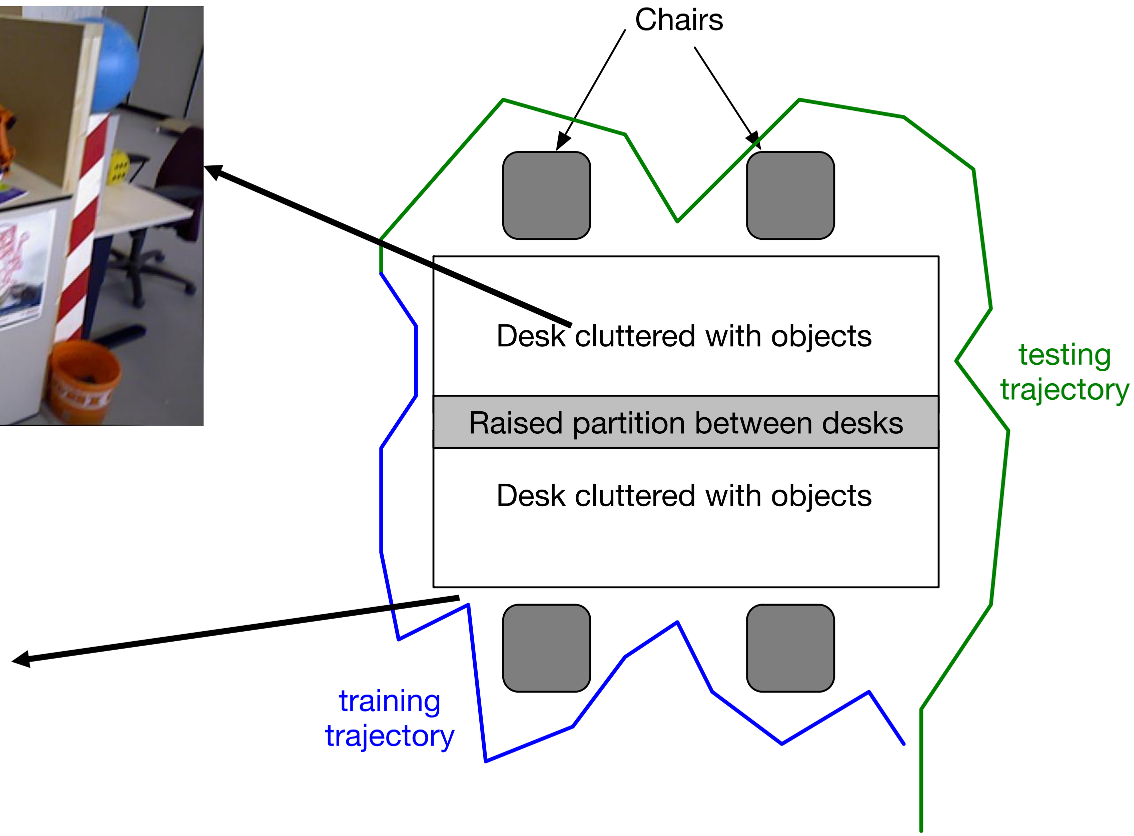
Deep Neural Nets and Convolutional Neural Nets





- Convolutional filters to generate feature maps from data
- Subsampling or pooling for dimension reduction and higher order feature generation

Occlusion detection from Freiburg dataset

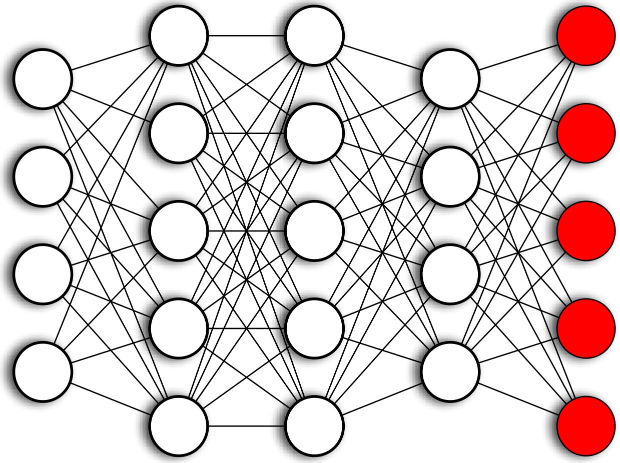


Reference: http://vision.in.tum.de/data/datasets/rgbd-dataset/download

- Use readily available dataset for demonstrating occlusion edge detection from Computer Vision Group at Technische Universität München (TUM)
- Partition the trajectory into training and test datasets for the neural nets

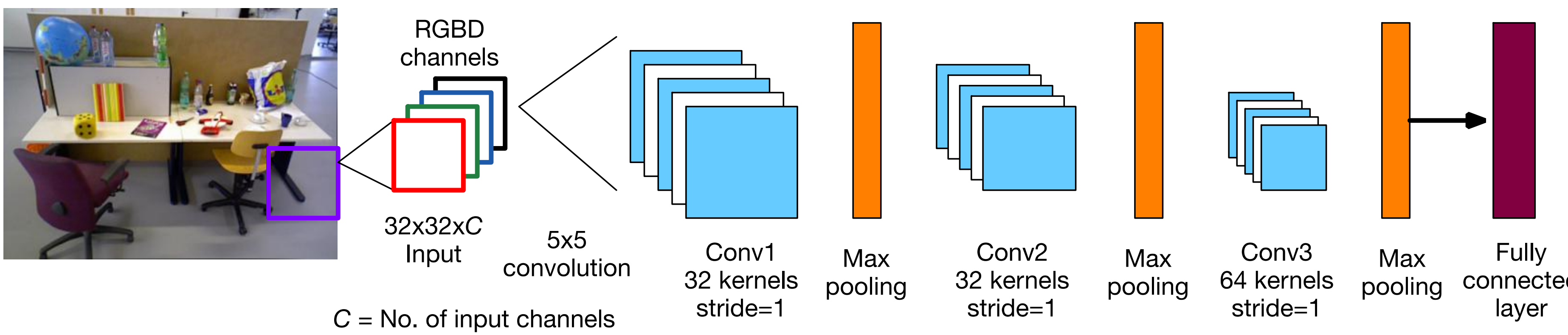
Problem setup

Input: RGB+D information from consecutive video frames (640x480) captured by mobile sensor



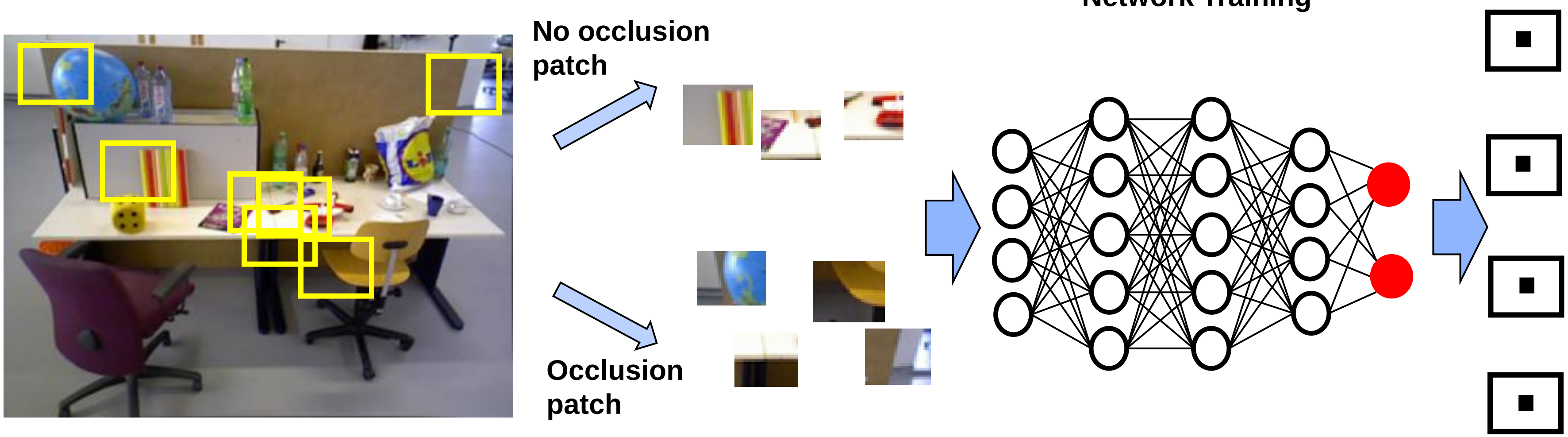
Deep Convolutional Neural Network

Output: Occlusion edges

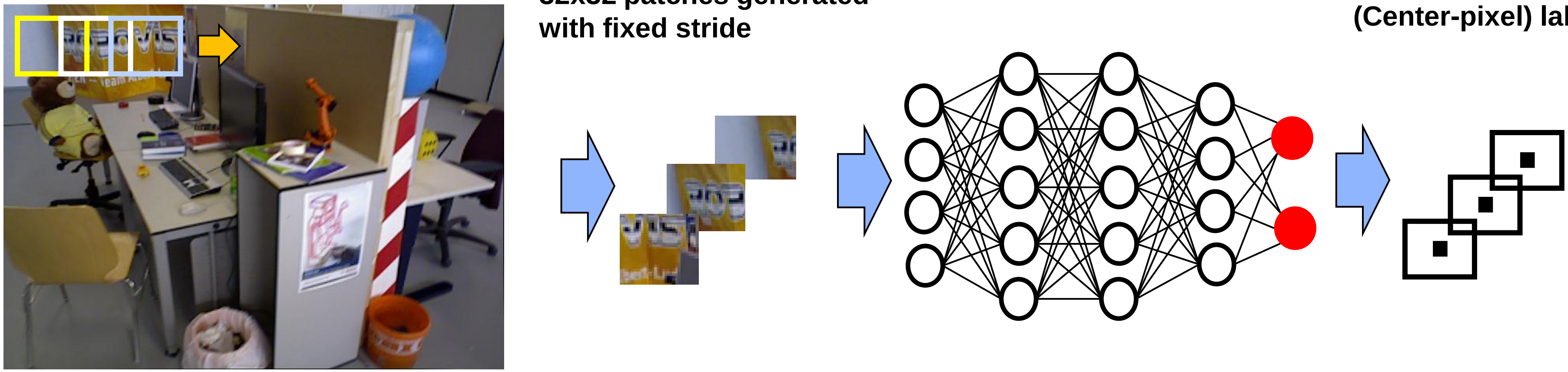


Training and Testing processes

Training process

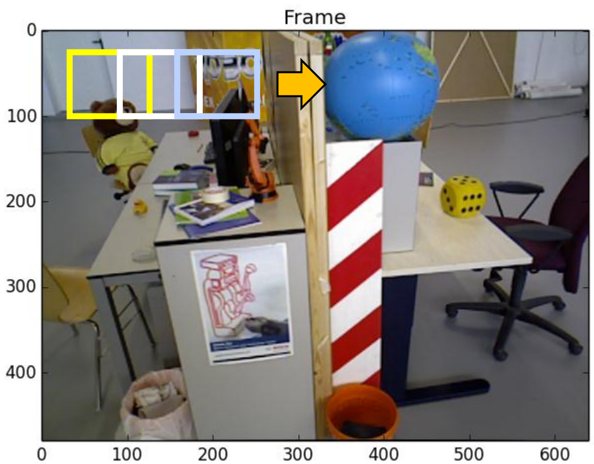


Testing process



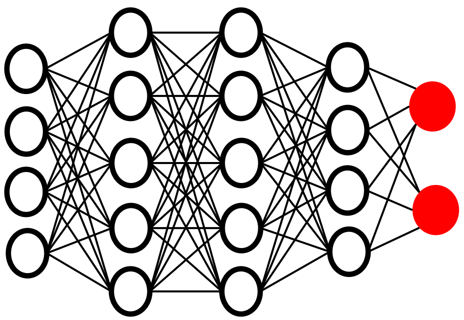
Post-processing for Occlusion edge reconstruction

Testing and post-processing

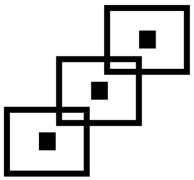


32x32 patches generated with fixed stride

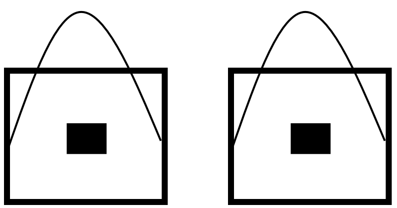
Trained Network



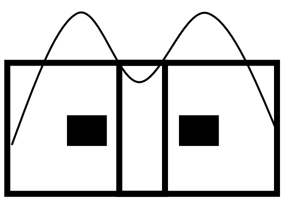
Prediction of patch (Center-pixel) label



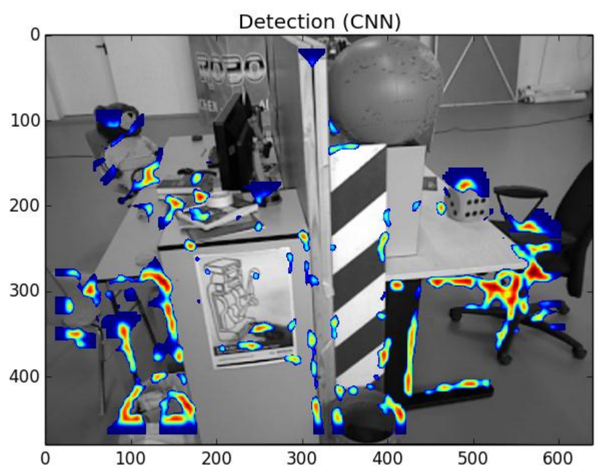
Prediction confidence from softmax posterior



Prediction confidence converted to patch-wide label using a Gaussian kernel (with Full Width at Half Maximum - FWHM)



Gaussian labels are fused in a mixture model to generate smooth occlusion edges



Experimental setup

- Nvidia Tesla K40 GPU with 2880 cores and 12 GB device RAM
- Initial pre-processing for dividing dataset into training and test and extracting small images (32x32) from large frames (480x640)
- Image size fixed at 32x32 with number of channels depending on the experiment
  - 4 channels for RGBD
  - 3 channels for RGB
  - 6 channels for RGBD + optical flow (UV)
- Ground truth consists of labelled edges by using only the depth sensor

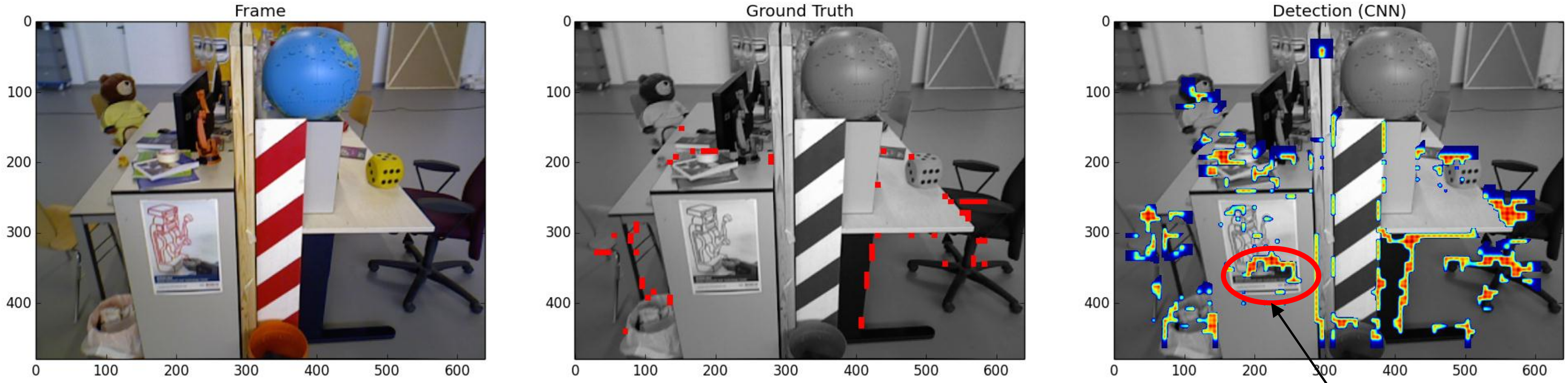
Optical flow pre-processing

Results

Data	Channels	Patch stride	Training dataset	Testing dataset	Test error (averaged over 80-100 epochs)	Computation time/epoch
RGBD (1 frame)	4	4	56354	500000	15.35	1m 21s
	4	8	14278	316167	18.76	2m 17s
RGB (1 frame)	3	4	56354	500000	16.43	1m 2s
	3	8	14278	316167	18.72	1m 42s
RGBDUV	6	4	56354	500000	15.18	1m 22s

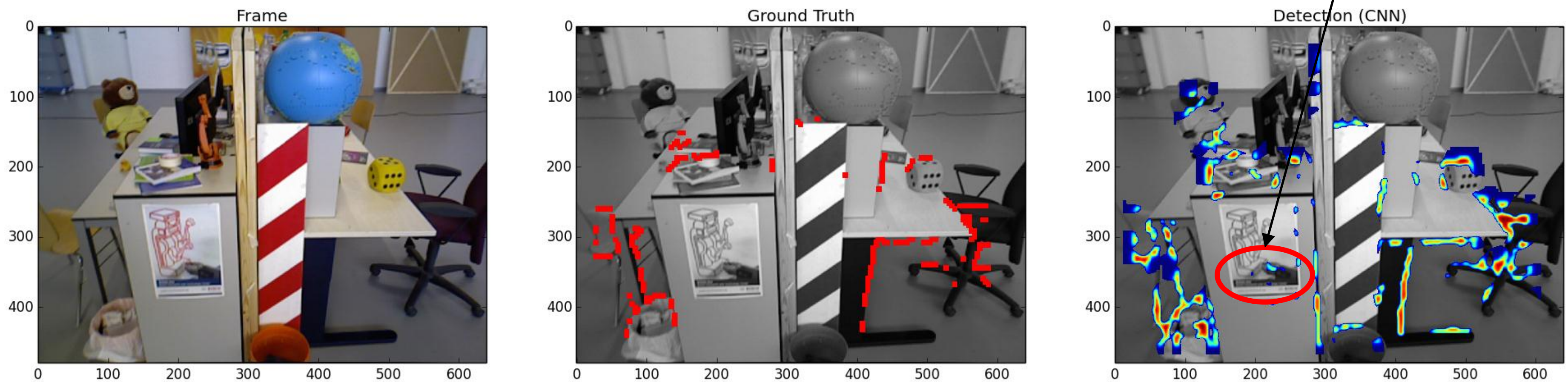
Post-processing Results

- Input: RGBD image (32x32x4), stride 8



Performance improves with higher granularity of fusion

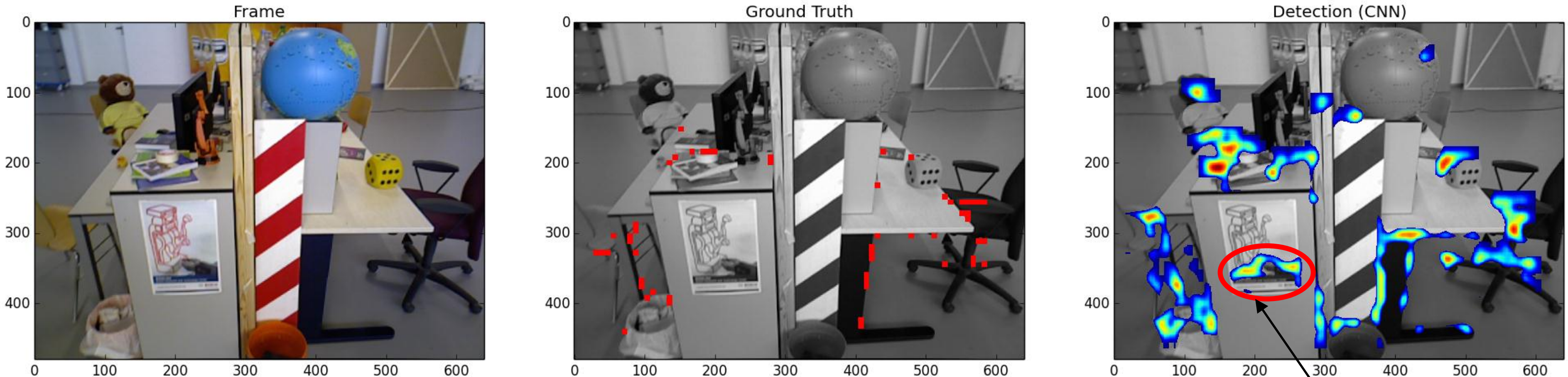
- Input: RGBD image (32x32x4), stride 4



Post-processing Results

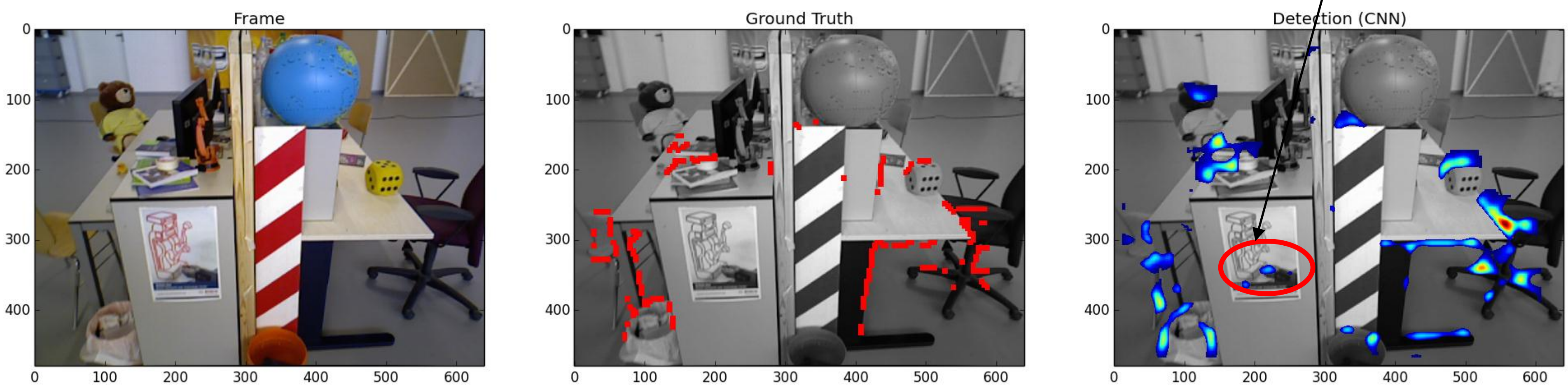
- Input: RGB image (32x32x3), stride 8

Overall detection confidence deteriorates without D channel

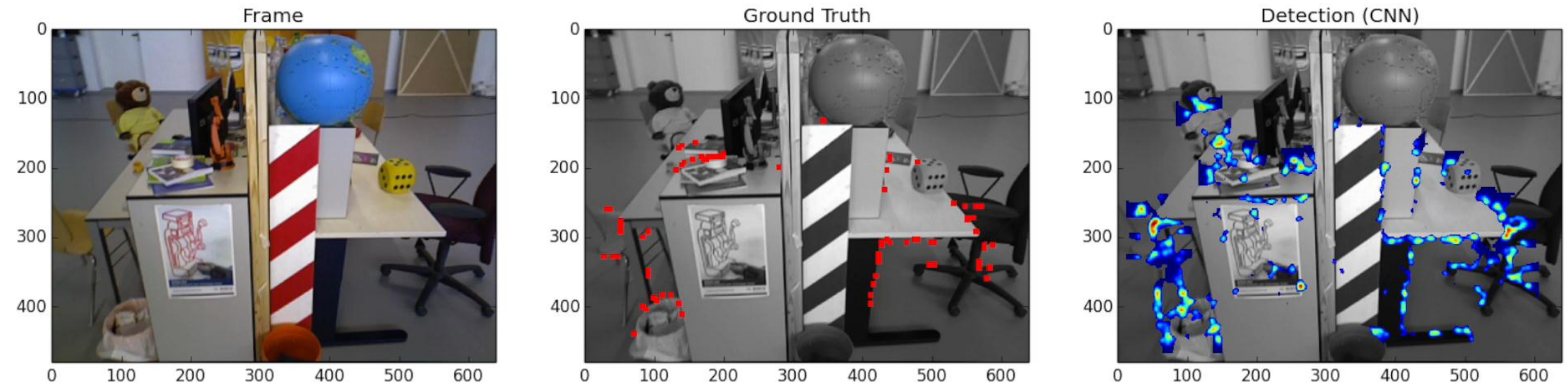


- Input: RGB image (32x32x3), stride 4

Performance improves with higher granularity of fusion



RGBD and optical flow (RGBDUV) Results



Conclusion

- Deep CNN can extract significant occlusion edge features from only RGB channels (i.e., without the depth sensor information). Occlusion detection accuracy increases when we introduce optical flow.
- Deep Convolutional Neural Nets (Deep CNN) for multi-modal fusion applied to occlusion detection
- The trade-off between high resolution patch analysis and frame-level computation time is critical for real-time robotics applications
- Currently investigating multiple time-frames of RGB input in order to extract structure from motion

Questions