AFRL-AFOSR-JP-TR-2024-0050

Semantic Mapping for Disaster Response

Hebert, Martial
CARNEGIE MELLON UNIVERSITY
5000 FORBES AVE
PITTSBURGH, PA, 15213
USA

02/07/2024
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20240207	Final	START DATE 20170421	END DATE 20210420

4. TITLE AND SUBTITLE
Semantic Mapping for Disaster Response

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA2386-17-1-4660		61102F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Martial Hebert

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CARNEGIE MELLON UNIVERSITY
5000 FORBES AVE
PITTSBURGH, PA 15213
USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD UNIT 45002 APO AP 96338-5002	AFRL/AFOSR IOA	AFRL-AFOSR-JP-TR-2024-0050

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
One of the key challenges in the semantic mapping problem in postdisaster environments is how to analyze a large amount of data efficiently with minimal supervision. To tackle this challenge, we investigate the following research questions: 1) how to train semantic classifier when we have only scarcely labeled training data, 2) how to quickly learn if a new class is introduced to a classifier that has been already trained for other classes, and 3) how to generate navigation paths when a map is no longer consistent with an environment and the only available information is a set of aerial images. In this report, we describe our semantic mapping approach, focusing on three main subtasks. First, we develop learning algorithms that
can be quickly trained to generate traversability cost maps using only raw sensor data such as aerial view imagery. Second, we investigate on the problem of learning to detect a new class of object using only a few training examples. Third, we develop a new dataset for rare scenes
focusing on postdisaster scenarios, DIsaster SCenarios (DISC) Dataset. This report includes the technical details of our approaches and the evaluation results that show state-of-the-art performance in our experiments. In all subtasks, we collaborate closely with the KAIST team
in terms of data sharing as well as technical collaboration.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	55

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
JERMONT CHEN	315-227-7003

Standard Form 298 (Rev.5/2020)
Prescribed by ANSI Std. Z39.18

Final Report: FA2386-17-1-4660: Semantic Mapping for Disaster Response

PI: Martial Hebert, co-PI: Jean Oh
The Robotics Institute, Carnegie Mellon University

May 5, 2022

Abstract

One of the key challenges in the semantic mapping problem in postdisaster environments is how to analyze a large amount of data efficiently with minimal supervision. To tackle this challenge, we investigate the following research questions: 1) how to train semantic classifier when we have only scarcely labeled training data, 2) how to quickly learn if a new class is introduced to a classifier that has been already trained for other classes, and 3) how to generate navigation paths when a map is no longer consistent with an environment and the only available information is a set of aerial images. In this report, we describe our semantic mapping approach, focusing on three main subtasks. First, we develop learning algorithms that can be quickly trained to generate traversability cost maps using only raw sensor data such as aerial view imagery. Second, we investigate on the problem of learning to detect a new class of object using only a few training examples. Third, we develop a new dataset for rare scenes focusing on postdisaster scenarios, DIsaster SCenarios (DISC) Dataset. This report includes the technical details of our approaches and the evaluation results that show state-of-the-art performance in our experiments. In all subtasks, we collaborate closely with the KAIST team in terms of data sharing as well as technical collaboration.

1 Introduction

In recent years, fast-growing interests in self-driving cars have contributed to various technological advances in autonomous navigation on the roads. Whereas highly accurate prior maps are generally available for path planning in such urban environments [137, 141, 101], the information on drivable surfaces in postdisaster environments is generally unavailable and also subject to change frequently. In such scenarios as disaster relief operations, it is needed to predict traversability using visual data such as aerial view images and data from on-board sensors. With the advent of low-cost cameras and drones [146] as well as several satellite imagery service providers providing near real-time imagery for

(a) Apr. 29, 2013 (Before) *(b) May 22, 2013 (After)*

Figure 1: Aerial images taken *(a)* before and *(b)* after a tornado in Moore, Oklahoma, on May 20 2013 [33].

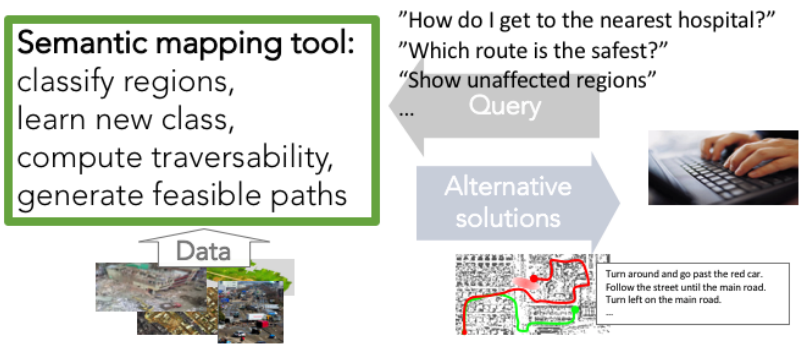


Figure 2: A scenario showing a use case of the proposed semantic mapping tool where the system assists a first responder by providing route options for dispatching and medical transportation operations in the affected areas.

disaster zones at low or no cost to disaster relief units [4, 108, 31, 55], a huge amount of high-resolution overhead imagery data can be collected in a short time; however, it is challenging to utilize such a gigantic amount of raw data to produce useful information for operators. As Kim Scriven, the manager of the Humanitarian Innovation Fund, noted in his BBC interview [79], "trying to harvest and filter the vast amounts of data generated by a disaster or conflict is the big nut that people are trying to crack, with the real challenge being to turn all of that data into information that humanitarian agencies can actually act on." Thus, the current challenge lies in semantic analysis of the large quantity of imagery data such as those shown in Figure 1. For example, for the navigation support, depending on the type of operations, a different path can be chosen according to the criteria beyond basic traversability, *e.g.*, in order to deliver time-critical supplies and medicines, it is desired to find a quickest path by cutting through vegetation, unpaved surfaces, or destructible objects.

In this context, we address the following research questions: given aerial-view images and/or other raw sensor data, how can we generate customizable traversability cost maps to accommodate various requirements and preferences? How can we train a system that can classify new types of objects using only a few examples that the system has not seen previously? How can we make the system intuitively interact with non-technical end users such as the first responders without the need of special training? In this document, we report on our work towards answering these questions.

To tackle the challenges, we develop a semantic mapping system that can assist first responders as illustrated in Figure 2. The idea is to classify and analyze new semantic information about the affected areas from a large collection of aerial imagery data, and provide mapping and navigation services using the analyzed information. Here, we focus on the requirements that are specific to the disaster response case. The key objectives of our technical approach reflect the urgent needs of the first responders–namely, data-efficient training, adaptive learning, and intuitive interaction.

2 Semantic navigation using aerial images

In this subtask, we explore the technologies for navigation in disrupted areas using aerial imagery when a prior map is no longer valid. We develop deep Inverse Reinforcement Learning (IRL) algorithms to learn various traversability cost maps from large-sized, high-resolution aerial/satellite imagery. Figure 3 shows the aerial views before and after the Fukushima tsunami in March 2011. Figure 4 shows how semantic navigation using deep IRL can be used in the proposed semantic mapping services. In this example, our approach generates a feasible path given an aerial image and an end-user's navigation query "find a safe route to transport patient 1 to the nearest hospital." The system explains that the path was chosen over a possibly shorter route due to the flooded area in the shortest path.

(a) Before (b) After

Figure 3: Aerial-view images of the same area before and after the Fukushima tsunami in March 2011.

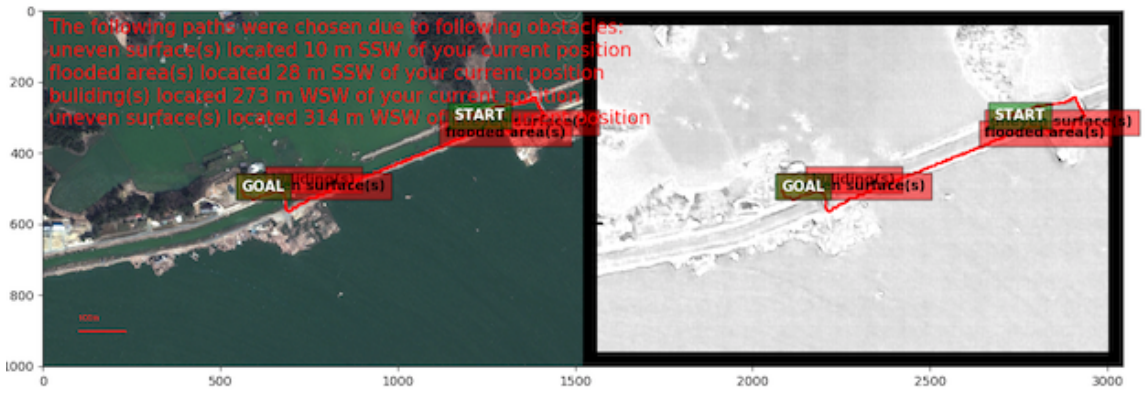


Figure 4: Given a user query "Find a safe route to transport patient 1 to the nearest hospital," our system generates a safe traversability map (right) to find a path and explains why this path has been chosen.

This subtask involves multiple technical challenges. While there exist algorithms that allow autonomous agents to perform path planning in outdoor environments, the majority of such algorithms requires a significant amount of engineering efforts such as handcrafting domain-specific features for cost functions. In this project, we seek an approach that can reduce the burden of feature engineering. The motivation for using deep IRL here is, therefore, to reduce such human engineering efforts and have the network learn useful features instead.

Most of existing IRL algorithms require solving a Markov Decision Process (MDP) at every iteration to update cost functions. Computing a full policy, however, can be costly as the state space grows, imposing a scalability limit on the IRL approach. In this work, we explore how we can efficiently approximate a solution given an MDP to address the scalability issue. We describe our approach for training multiple navigation modes using the idea of conditional network in Section 2.3.1 and the multimodal network that can also utilize 3D data in addition to the imagery data in Section 2.4. We report on the experimental results on multiple datasets.

2.1 Related Works

Our approach is related to existing works in representation of cost functions, conditional learning, and multimodal deep learning.

Representation of Cost Functions: The IRL approaches [118, 164] have shown great successes in learning human driving behaviors. The cost functions in early works are linear combinations of input features that are elaborately engineered. To support complex control problems, nonlinear cost functions have been studied, *e.g.*, using Gaussian Processes [83], Dirichlet Processes [27], or function approximation for large state-spaces [86]. The work that is most closely related to our research is Maximum Entropy Deep IRL (MEDIRL) [149] where Maximum Entropy IRL [164] was generalized to a deep learning networks; such an end-to-end learning idea has been applied to generate a costmap using 3D data from a vehicle-mounted LIDAR sensor in a semi-urban environment [150].

Conditional Imitation Learning: Conditional Imitation Learning (CIL) [30] is a unique form of multimodal learning that by giving an expert demonstration and a descriptor explaining the behavior of the demonstration, a deep neural network can learn multiple navigation behaviors at the same time on a single set of network weights. In [30], CIL was trained with expert driving demonstration comprised of ground level images, various sensor data to learn three different behaviors of going straight, turning left, and turning right at an road intersection by using an extra input to describe the behavior of the expert demonstration. However, CIL's main difference from most other multimodal learning algorithms is how the descriptor is merged with the main data. In the case of [44], [102], [16], and [159] was concatenated with the main data in the form of fully connected layers. Branched approach of CIL [30] uses first part of the network to first process image and pass the extracted deep network to corresponding deep action network for going straight, turning left, and turning right depending on the descriptor input. Such approach showed better results than Command Input approach for CIL [30] where descriptor was simply concatenated as in other multimodal learning networks [44], [102], [16], and [159].

Multimodal Deep Learning: With a success in [109] that used two different input modalities of audio and video for better speech classification, Deep Multimodal Learning has gained attention in recent years on how deep neural network can utilize more than one modalities of observation to understand an environment. With a diverse field of applications from using MRI and PET for medical images [128] to autonomous navigation using images and LIDAR data [90], deep learning with Earth Observation (EO) data is one major field that can greatly benefit from deep multimodal learning. For instance, while a grassy hill and a swamp covered with dense vegetation may look similar as featureless grass area, and concrete structures, pavement, and roads may all look like square grey objects, deep networks using 3D information such as LIDAR data can disambiguate them. As such, Multimodal Deep Learning for better semantic segmentation using the EO data showed great promises in recent

years such as [11] and [110] that used Infrared-Red-Green Imagery, Digial Surface Model (DSM), Normalized Digial Surface Model (NDSM) and Normalized Difference Vegetation Index (NDVI) to better understand an environment.

2.2 Preliminaries

Our approach generally follows the principle of maximum entropy as in [164] where we maximize the likelihood of demonstrations with the highest entropy. Following the derivations directly from [149], this problem can be defined as computing the joint probability of observing the demonstrations $\mathcal{D}$ under parameters θ given cost r as follows:

$$\mathcal{L}(\theta) = logP(\mathcal{D}, \theta|r) = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\theta} = logP(\mathcal{D}|r) + logP(\theta)$$

By defining the cost function r as a parameterized function over features f, such that $r = g(f, \theta)$, we develop a neural network representing the function. To compute the gradient of the loss of demonstration $\mathcal{L}_{\mathcal{D}}$, the backpropagation algorithm can be applied and the gradient can be rewritten as the product of the gradient of the loss with respect to the cost, $\frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial r}$, and the gradient of the cost with respect to the network parameters, $\frac{\partial r}{\partial \theta}$. In the case of a linear function, the gradient of demonstration with respect to the cost is the difference in feature frequencies between empirical counts and the learner's expectation [164] as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}_{\mathcal{D}}}{\partial r}\frac{\partial r}{\partial \theta} = (\mu_{\mathcal{D}} - \mathbb{E}[\mu])\frac{\partial}{\partial \theta} r(\theta)$$

where $\mu_{\mathcal{D}}$ is the State Visitation Frequency (SVF) of the expert demonstration and $\mathbb{E}[\mu]$ is the expected SVF of the learner. With such derivations, we backprogate the neural network parameters θ to minimize loss, $\mathcal{L}$

2.3 Approximation of Forward Solver

Solving the forward problem during IRL can be expensive [47]. Following ideas in [125, 46, 97] that MDPs can be solved more efficiently with path planners, we use A* planner [58] to approximate forward solver during IRL. This approach was also implemented successfully with the case of Maximum Margin Planning [118].

In 2D navigation, the trajectory generated via A* [58] can be viewed as sampling one of the most probable trajectories in the SVF distribution. While this is an approximation of the true SVF at a given iteration, such an approximation is acceptable as A* is likely to visit all states with a distribution similar to the actual SVF during the long training process of deep networks that involves numerous iterations and updates. Marking only the states with highest SVF that would dominate the training process, A* can greatly enhance the speed of training compared to other methods such as Value Iteration [15] used in MEDIRL [150]. As deep networks generally need a large amount of data for training, this speeding up is crucial in time-constrained applications such as post-disaster assistance.

2.3.1 Conditional Network Architectures

To extract high-level feature information directly from aerial and satellite imagery for deep IRL, we design network architectures to utilize multiple pooling operations to allow the neural networks to process an input image at multiple resolutions.

The idea is based on the assumption that certain features may appear more pronounced to neural networks (and can thus be more easily trained) depending on two factors: the resolution of an input image and the size of a kernel. For instance, to differentiate forest from grass fields, we need the texture information that may only be available in a high-resolution view.

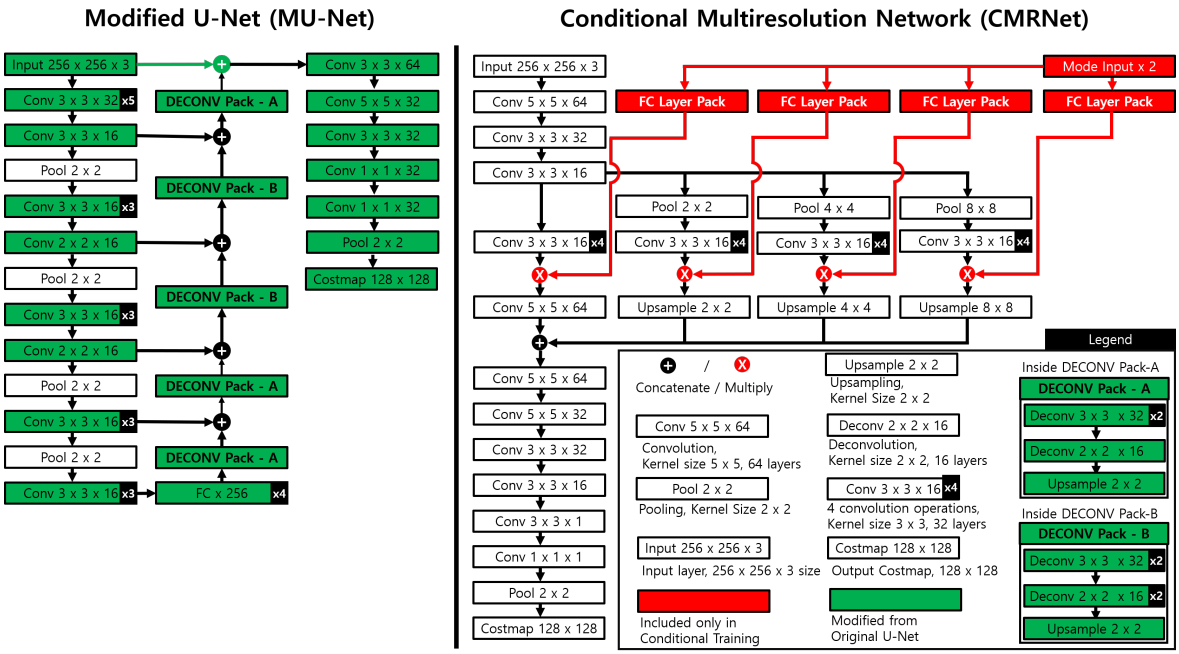


Figure 5: Proposed networks: Modified version of U-Net[122] (MU-Net)(left) and Conditional Multiresolution network (CMRNet)(right). Modifications from original U-net[122] is marked in green. Note that Multiresolution Network(MRNet) in this report refers to CMRNet Network without the conditional parts outlined in red.

Tangentially, objects that look similar in terms of color and texture can be indistinguishable in a small, isolated patch, and require wider context information to discern, *e.g.*, a grey patch of rough texture can be either concrete road surfaces or flat-top grey building roofs depending on size, shape, and what surrounds it. To learn such context features, we need larger kernels in the convolutional layers.

Based on this intuition, we propose two networks that utilize multiple pooling operations and concatenations in order to capture contexts such as texture, geometric, or spatial information, within a reasonable network size. By varying the resolution of an image within the network through the use of multiple pooling operations, the part of the network with higher-resolution images is responsible for picking up texture-related information, while that with lower-resolution images is responsible for geometric, spatial and contextual features.

Under the idea of varying resolutions within networks, we modify the U-Net architecture [122] and introduce Multi-resolution networks (MRNet) as shown in Figure 5.

Modified U-Net (MU-Net): Originally used for medical imagery, U-Net [122] is a Full Convolution neural network using multiple pooling operations that showed good performance even with small datasets in the field of medical imaging as well as versatility of being applied in various different fields, such as radio astronomy. [5]. The original U-Net [122] starts with 64 channels per convolutional layer; the number of channels doubles every time the network pools down to a smaller height and width. As shown in Figure 5, in an attempt to decrease the number of parameters and introduce constraints in the network for simpler training, we reduce the number of channels to 16 or 32, while increasing the number of convolution operations at each resolution from 2 to 3.

In addition, we use a Fully Connected (FC) layer at the lowest resolution since the layer is only 4x4x16 in size; when compared to original U-net's size of 30x30x1024, it is small enough for FC operations to be used. Whereas U-Net uses a larger size input to utilize surrounding pixels in place of zero-padding, we match our MU-Net network output to have the same height and width as a network

input such that MU-Net can be used interchangeably with other networks in our deep IRL framework depending on the problem setup and performance requirements.

Multiresolution Networks (MRNet): We introduce Multiresolution networks (MRNet) where a set of pooling operations is performed after the first convolutional layer to extract the features at a gradually lower resolution simultaneously as shown in Figure 5. The main difference between MRNet and the MU-Net is that, whereas MRNet has a *parallel* structure where input is processed at multiple resolutions along separate pathways and than later concatenated all at once, MU-Net has a *serial* structure where an input is processed at multiple resolutions along a single pathway in a sequential order and a concatenation operation is done for each resolution at a time.

When compared to sequential pooling operations, parallel pooling models tend to be less deep; therefore, parallel models might have issues with supporting complex cost function shapes. At the same time, this feature also helps the models to avoid overfitting. Intuitively, the parallel model has fewer operations for the extracted features to go through before the final output layer compared to the sequential model, potentially reducing the chance of being affected by additional operations.

Pooling Operations for Resolution Discrepancy: Our proposed methods have pooling operation at end of the network for faster training. For input layers, high resolution imagery is desired to have more information about an environment. However, such high resolution often leads to large search space, making it computational expensive to solve MDP. As a technical solution to solve the discrepancy between resolution requirements, our proposed methods have pooling methods such that input and output resolutions can be tuned separately to maximize speed and performance of training. While such pooling operation results in loss of information for cost maps, such loss is acceptable as global planners has much coarser resolution requirement than those required from input images.

Conditional Multiresolution Network: In traditional linearl IRL architectures based on handcrafted feature inputs, each navigation behavior is trained independently [17]. As shown in Figure 6, these linear IRL algoirthms take an input of handcrafted feature layers, multiply each layer with a corresponding trained weight, and add the results to generate a costmap.

Similarly, Conditional Multiresolution Network (CMRNet) first passes the input image through the multiresolution structure to generate multiple deep feature layers. Each layer would then be multiplied by a set of weights, concatenated, and passed through another set of convolutional layers to generate a costmap. The set of weights for the case of Conditional Multiresolution Network would be generated using a set of Fully Connected Layers that takes an input as a 1-D vector of numbers describing each driving behavior in the dataset. Such a structure is analogous to the structure used in Conditional Imitation Learning (CIL) [30] in a sense that the mode input dictates which part of the network would be used to generate a desired behavior. As illustrated in Figure 6, whereas the CIL approach hard-wires each navigation mode to which branch of the network to use, CMRNet represents the mode decision as a weight vector and *learns* the weights as opposed to having a separate branch for each navigation mode as in CIL.

Figure 5 shows the resulting CMRNet that can be trained to extract texture, spatial, and contextual information from images and learns how to weight each deep feature layer and look at the concatenated result of multiplication operations to recover a costmap according to each different type of expert behaviors.

2.4 Multimodal Multiresolution Network

In this section, we describe how we extend our approach to utilize both aerial imagery and 3D data. This work is a result of our close collaboration with the KAIST team and a postdoctoral researcher from KAIST.

To utilize both imagery and 3D data, we propose a multimodal multiresolution networks architecture for MaxEnt Deep IRL. Our network structure is inspired from MRNet [7] that offers the state-of-the-art performance on learning complex navigation behaviors using aerial and satellite im-

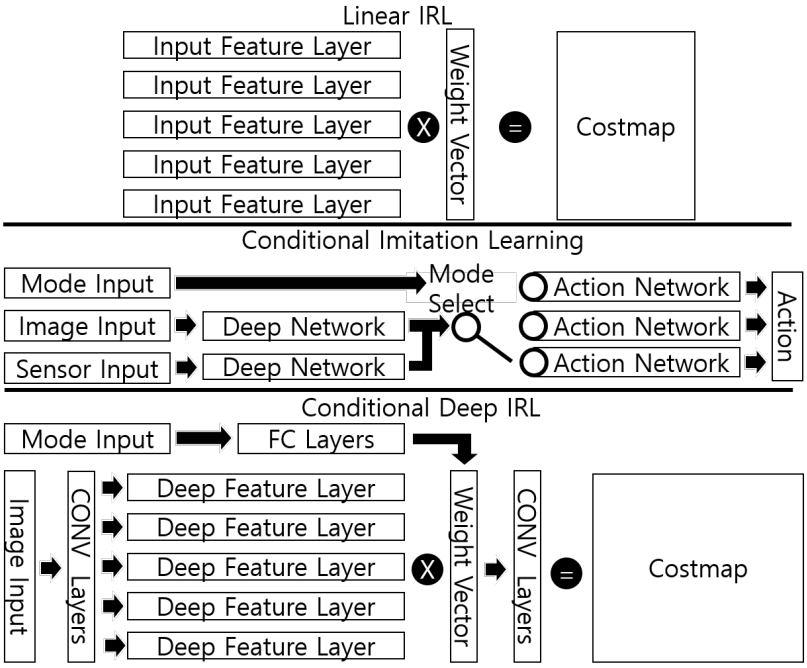


Figure 6: Conceptual comparison of Linear IRL, Conditional Imitation Learning, and Conditional Multiresolution Network.

agery data in a variety of urban, outdoor, and post-disaster scenarios. In MRNet, whose structure follows the similar principle to those found in [162], the input data is first passed through a set of initial convolutional layers, the results from which are then pooled at different resolutions to be processed in parallel through a set of convolutional layers. The deep features extracted from this operation is concatenated and passed through a series of convolution operations to generate a costmap.

Whereas MRNet was designed for taking large-sized, high-resolution images as inputs, the proposed architecture is specifically focused on how to fuse the 3D data with 2D image by varying the network structure, fusion techniques, and grid resolution of the 3D. To verify the pros and cons of each methods and network structure, the algorithms were kept as modular as possible, switching out parts and data as need to find the correlation between algorithms and their performances.

Early vs Intermediate Fusion: The simplest form of multimodal network implemented in this report is the Early Fusion network where two modalities are merged at before the start of the neural network operations, also commonly referred to as early fusion technique [117]. However, recent advances multimodal learning shows how change in method of fusion can affect network performance [3]. As such, we also try intermediate fusion where modalities are fused as deep features rather than raw data. Emerging as a popular option in deep multimodal learning [82, 91, 67, 90, 110], later fusion can be especially beneficial if modalities differ greatly [117]. Our implementation of Intermediate Fusion is highlighted in blue box in Figure 7.

Multimodal Image Prioritized Network (MIPNet) and Multimodal 3d Prioritized Network (M3PNet): In our multimodal setup, one modality passes through a multiresolution structure while other modality passes through simple convolution operations before the extracted deep features are merged. To see which mode of data benefits from having a multiresolution structure, we compare results between Multimodal Image Prioritized network which passes image data through multiresolution structure with Multimodal 3D Prioritized network which passes 3D data through multiresolution structure, with the network structures marked in Blue in Figure 7.

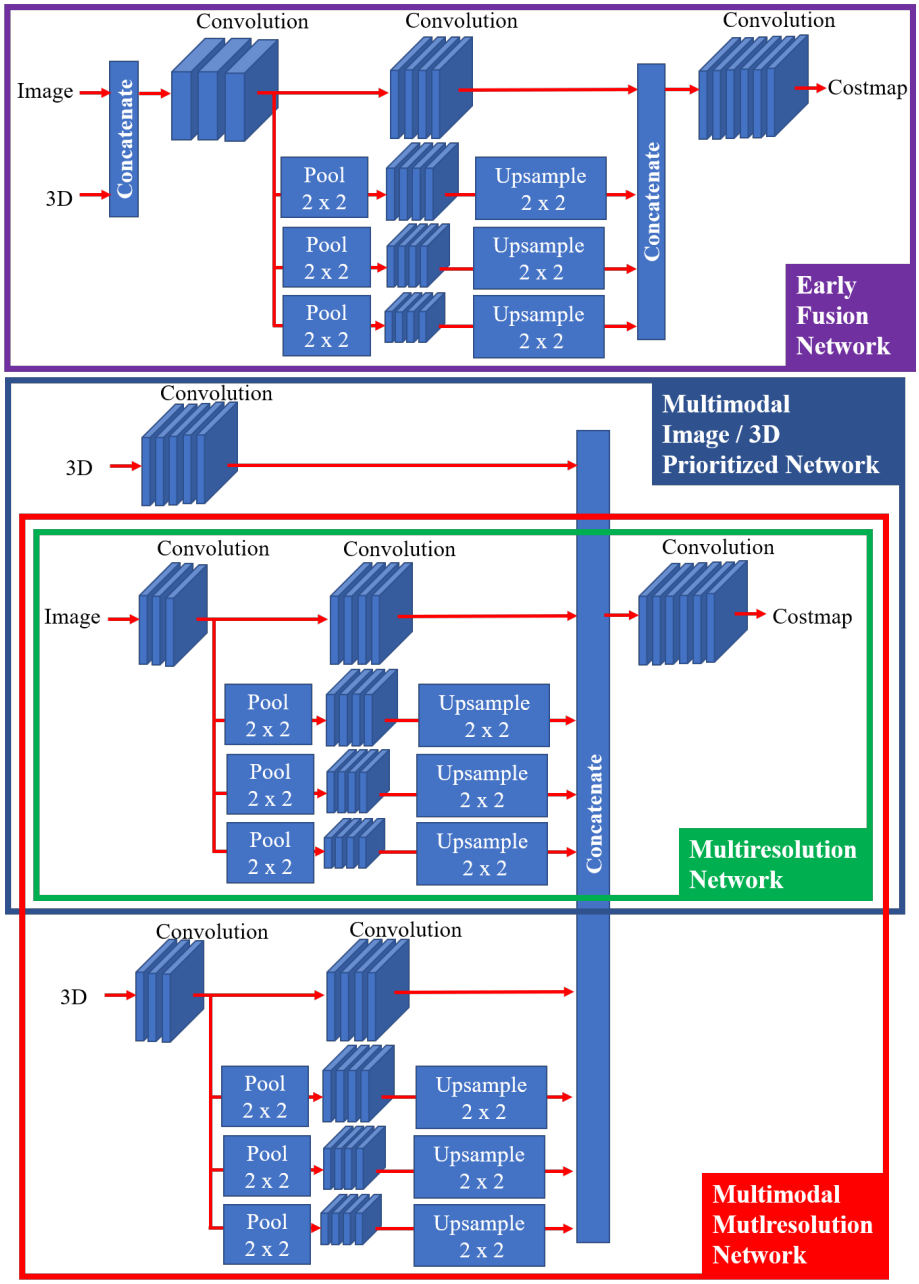


Figure 7: Diagram showing network structures used in the report. Blue box marks the Multimodal Image / 3D Prioritized Network, green marks Singlemodal Multiresolution Network, purple marks the Early Fusion Network, and red box marks the Multimodal Multiresolution Network

Multimodal Multiresolution Network (M2RNet): As multiresolution network uses parallel convolution operations at multiple resolutions to find deep features, we believe that both 3D and image information can benefit from multiresolution operation to improve network performance. Based on this idea, we test Multimodal Multiresolution Network which uses multiresolution operation for both modalities before concatenating the deep features, as marked in Red in Figure 7. Through the effective incorporation of multimodality and multiresolution concepts, our network achieves state-of-

the-art path planning results, which will be validated in Section 2.8.

2.5 Datasets

Image data: We collected a set of aerial and satellite imagery data in both real and synthesized environments from the following publicly available sources: ISPRS [66], Google Earth Pro and associated image providers [54][53][42][41], and CARLA [38]. The ISPRS set covers various locations in Vaihingen, Germany and provides red, green, and infrared channel inputs at 0.4 m / pixel resolution. All other images were captured with red, green, and blue channels at 0.8 m / pixel. The two Fukushima (FKS) sets were taken in the same locations within 10 km of Fukushima Daiichi Nuclear Power plant in Japan before and after the 2011 Tohoku Earthquake and the subsequent tsunami; FKS-1 contains the images from November 2009 before the earthquake, and FKS-2, March 2011 after the earthquake. All images were captured at bright lighting conditions. Two exceptions to that are: 1) FKS-1 was taken with the dark lighting conditions and 2) CARLA data was taken under 6 different weather conditions supported by the simulator such as rainy day, late after noon, sunny noon, etc. Each image was cropped to 256 x 256.

3D LIDAR data: To compare with MaxEnt IRL[164] that uses feature inputs, semantic labeled version of ISPRS datasets was also created with 0.8m / pixel 256 x 256 size. The 6 semantic labels included in the dataset are Impervious Surfaces, Building, Low Vegetation, Tree, Car, and Clutter / Background [66].

Multimodal data: We created two multimodal datasets for real and synthesized environments using the data from two publicly available sources: ISPRS [66] and CARLA [38]. The ISPRS [66] dataset provides observation data over various areas in Vaihingen, Germany in the form of high-resolution images in Infrared-Red-Green (IR-R-G) and the airborne laser scan data collected at a point cloud density of around 4 points per m^2. CARLA [38] is a driving simulator that provides high-fidelity camera sensor data in Red-Green-Blue (R-G-B) and LIDAR sensor data in ground-based point clouds. The ground-truth semantic labels are also available, *e.g.*, roads, buildings, etc.

Expert demonstrations: We collected expert demonstrations for the following navigation behaviors: normally (default), safely or quickly (fast), driving on the center of the road or on the edge of the road. The default mode, driving normally, represents the behavior of staying on the center of the road (or other types of drivable surfaces) to maximize the distance from potential obstacles. The default mode was collected for all of the sets. The customized behaviors were collected in the ISPRS set to evaluate the learning of multiple navigation behaviors, such as *safely* staying on paved surfaces while moving through *center* of such areas to keep a distance from obstacles (SafeCenter), *safely* staying on paved surfaces and hugging *edges* of obstacles in such areas (SafeEdge), cutting through unpaved surfaces such as vegetation to get to the goal *fast* while moving through *center* of such areas to keep a distance from obstacles (FastCenter), and cutting through unpaved surfaces such as vegetation to get to the goal *fast* and hugging *edges* of obstacles in such areas (FastEdge). As for the CARLA dataset, the driving behavior was to stay on one side of the road without crossing the center line. For each image, given a start and an end positions and a navigation mode, a human expert drew a desired path over the image.

Data augmentation: We augmented the core dataset using the process of flipping and rotating, resulting in the dataset increased in the size by the factor of 8.

2.6 Evaluation Metric

We use Modified Hausorff Distance (MHD), an evaluation metric for comparing trajectories generated by planners learn from demonstrations [148, 40, 127], which is defined as follows:

$$\texttt{MHD}(\zeta_D, \zeta_L) = [\sum_{i=1}^{N} d(\zeta_L(i), \zeta_D) + \sum_{i=1}^{N} d(\zeta_D(i), \zeta_L)] * \frac{1}{2N} \tag{1}$$

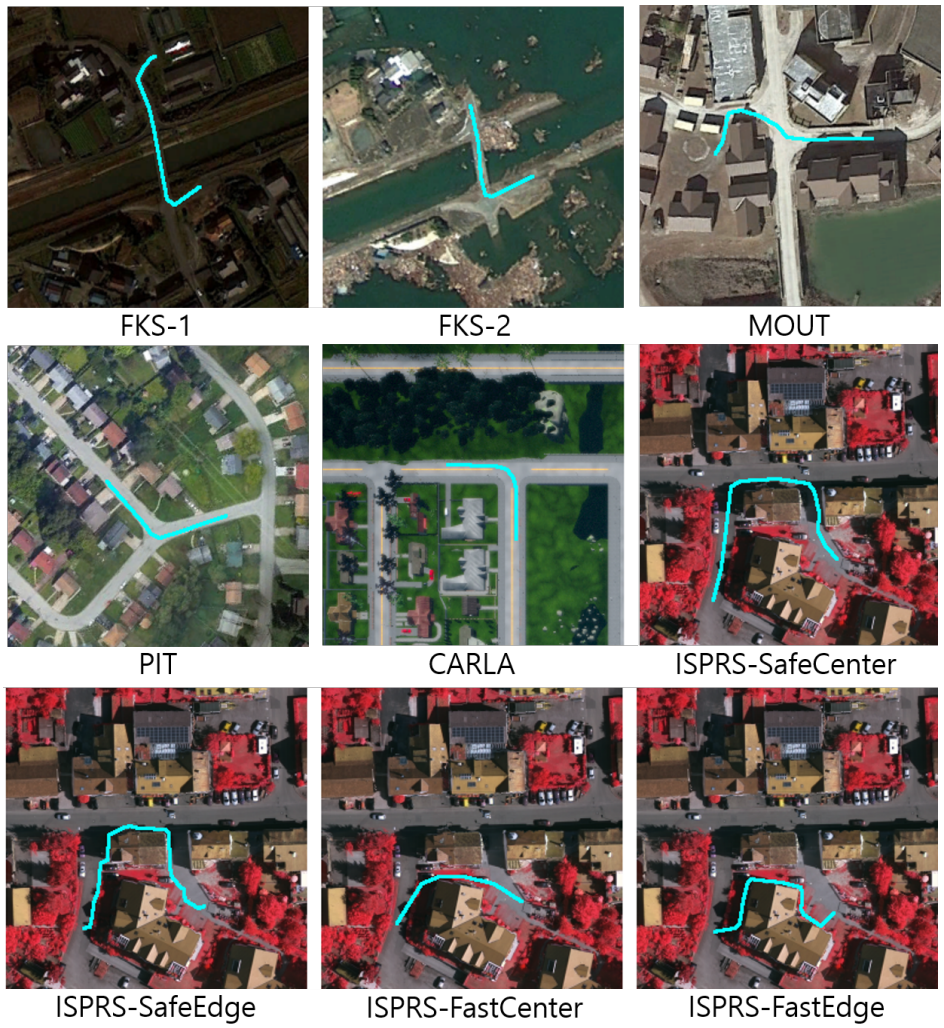


Figure 8: Characteristic samples of datasets used in this report. Note how FKS-1 and FKS-2 covers same location before and after tsunami, while ISPRS datasets show different behaviors on a same location.

where $(d(\zeta_L(i), \zeta_D))$ is the minimum Euclidean distance from point i on the learner's trajectory ζ_L to the closest point on the expert's trajectory ζ_D, and vice versa.

2.7 Baselines

To evaluate our proposed approach, we compare the experimental results on the datasets for Maximum Entropy IRL [118] and MEDIRL [150]. The former represents a well-known classical IRL algorithm while the later represents a state-of-the-art deep learning based IRL algorithm.

For a fair comparison, our implementation of Maximum Entropy IRL [118] was fed with ground-truth semantic labels as inputs as the algorithm was primarily designed to use labeled inputs. It is fair since algorithms exist to classify terrain types given aerial imagery, *e.g.*, achieving 88.5% accuracy for the images of ISPRS dataset [94]. Also, since MEDIRL [150] was originally designed for LIDAR data only, we use the CARLA simulator data where LIDAR data was also available. Our implementation of the algorithm was also used with 3-channel aerial / satellite imagery to show compare and contrast

Table 1: Size of Datasets and Area of Imagery

Name	Size of Dataset				Location
	Source	Augmented	Train	Test	
CARLA	150	1200	1000	200	CARLA Town 01
ISPRS	144	1152	960	192	Vaihingen, Germany
PIT	144	1152	960	192	Pittsburgh, US
FKS-1	144	1152	960	192	Fukushima, before tsunami
FKS-2	144	1152	960	192	Fukushima, after tsunami
MOUT	128	1028	800	224	MOUT* Training Facility in the US

the pros and cons of LIDAR input against image input.

2.8 Results

Table 2: Average MHD of Various IRL Methods. Lower is Better

Dataset	MaxEnt [164]	MEDIRL [150]	MU-Net	MRNet
SafeCenter	3.844	3.407	2.512	1.783
SafeEdge	2.994	4.064	2.855	2.960
FastCenter	3.350	7.055	3.822	2.694
FastEdge	3.644	5.812	5.821	4.616
CARLA	2.382	RGB: 1.792, LIDAR: 2.690	1.029	1.277
PIT	N/A	2.698	1.224	2.005
FKS-1	N/A	1.955	1.681	1.721
FKS-2	N/A	6.665	1.974	2.703
MOUT	N/A	6.806	6.167	3.298

*MEDIRL was run with 3-channel imagery for all dataset, with another run for CARLA dataset using 3D LIDAR Data.

Table 2 shows the evaluation results for the two baseline approaches and the proposed deep IRL approach using two new networks, namely MU-Net and MRNet. Our proposed methods outperform the state-of the art baseline approaches across all 9 datasets that include European cities, US suburbs, a synthesized environment, a simulated town, undeveloped villages, and an area struck by a natural disaster. Comparison with benchmark approaches and their respective results to datasets highlights the strengths of our proposed methods as follows:

Comparison with MaxEnt IRL algorithm

Table 2 shows traditional IRL method such as MaxEnt IRL [164] can perform comparablely to MEDIRL [150] and match the performance of MU-Net for FastCenter. We note that such performance is only possible if appropriate feature input can be provided. For instance, MaxEnt IRL [164] does worse in SafeCenter compared to SafeEdge as while given semantic labels are suitable for supporting edge following behaviors, it doesn't mark where the center of the road is, making beahvior in the former dataset difficult to replicate.

Across all dataset, MRNet outperforms MaxEnt IRL [164]. This shows deep IRL's ability to extract features and complex cost functions through learning.

Comparison with LIDAR based IRL algorithm

In the CARLA dataset, designed to contrast the strengths and weakness of 3D LIDAR data and 2D image data, we can see that the 2D images offers additional information not easily conveyed

Figure 9: Results of SafeCenter training for MAXENT IRL [164] and MRNET. With high cost regions marked in black, we can see that unlike MAXENT where transition between obstacles and driveable space is sharp, MRNET has gradual transition between the two regions, favoring center of the road then edge of roads.

in 3D point cloud data. While the former can mark size and location of objects in the object, the later also shows texture such as yellow center line marked in the center of the road. In the case of CARLA dataset where expert behavior was to stay on one side of a road, such additional information was beneficial in allowing image-based algorithm to easily assign high cost to center of the road as shown in Fig.10 much easier than LIDAR based methods which has no such cues.

Network Structure Comparison

The overall results as shown in Table 2 implies that a deep, carefully designed network architecture is needed to learn a good costmap from high-resolution imagery. When compared to the network architecture used in MEDIRL [150], the proposed models use multiple pooling and upsampling layers to utilize both low-level local features as well as high-level contextual features. On one hand, the network needs the texture information from low-level layers to distinguish whether an object is vegetation or roads. On the other hand, the network needs to pay attention to size and shape information to determine if a green patch is a grass field or a tree. Sometimes, the network even needs to look at the surrounding context to see if a tree is standing in the middle of a forest or next to a road covering a traversable road underneath. By using multiple pooling, upsampling, and concatenation operations, the proposed networks can run convolution operations at a high resolution with a narrow viewing window to extract texture information, while running convolution operations at low resolution with a wide viewing angle to extract the spatial and contextual information.

While both MRNet and MU-Net are similar in their network sizes, their differences in performance show pros and cons of how pooling, upsamplig, and concatenation operations can be structured for an enhanced performance. In the case of MU-Net, the deep features extracted at the high-resolution layers are fed as inputs to the consecutive lower-resolution layers. Such *linear* structure induces high-resolution features to be weighted more than lower-resolution features. At the same time, the path that an input image goes through in the network pipeline can be extended, possibly leading the model to overfit the training set. Conversely, MRNet has a *parallel* structure where both high- and low-resolution convolution operations share the same input, and their outputs are concatenated together. In this way, the spatial and contextual information extracted from the low-resolution layers can be weighted fairly as the texture information extracted from the high-resolution layers. Due to this parallelism, MRNet performs better than MU-Net for those cases where the contextual information is more critical in generating a costmap, *e.g.*, Figure 12 shows that MRNet discerns various types of vegetation that share similar color texture but exhibit different spatial and contextual features.

Results of Conditional Training

To evaluate the conditional learning performance of the proposed approach, the models were trained using the ISPRS series datasets in three experimental settings: 1) Independently: where each navigation behaviors were trained and their weights stored separately as a benchmark, 2) Combined: where all behaviors were trained at the same time on single set of weights. For combined training,

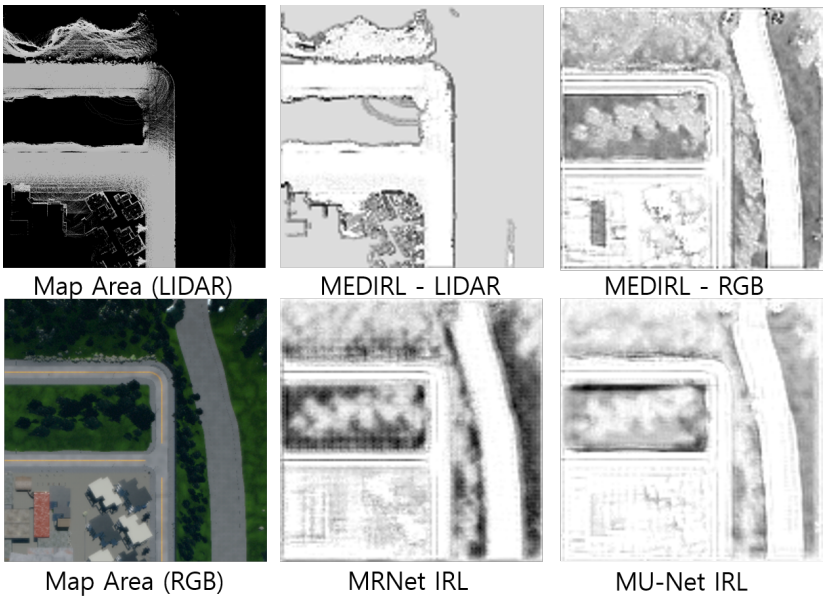


Figure 10: Results of CARLA Dataset with RGB and LIDAR input on MEDIRL [150], MRNet, and MU-Net. With high cost regions marked in black, we can see that LIDAR based method cannot detect center line easily as those based on RGB images.

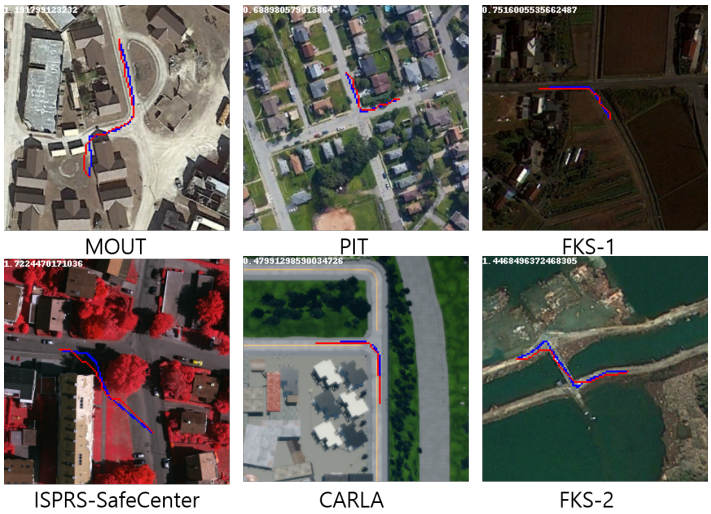


Figure 11: Results of MRNet on various datasets. Expert demonstration is marked in Blue and trajectory generated from IRL's recovered costmap is marked in Red, with MHD value marked on top left corner of each images.

each driving behavior was assigned with descriptors to show preference for safe / fast driving and center / edge of traversable space, 3) One-Out: where CMRNet was trained for a combined dataset excluding the SafeEdge dataset and later tested with the excluded dataset to verify if the network can predict an unseen navigation behavior based on descriptions on behaviors included in the training set.

As shown in Table 3 CMRNet's architecture of using Fully Connected Network to learn weights assigned to each deep features extracted during training allows the network to train multiple navigation behaviors with around on average of 0.663 % on MHD compared to training each behaviors separately.

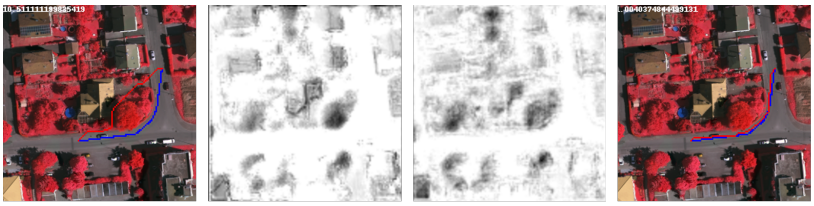


Figure 12: Results of MRNet and MU-Net on ISPRS - FastCenter. Expert demonstration is marked in Blue and trajectory generated from IRL's recovered costmap is marked in Red, with MHD value marked on top left corner of each images. Dark regions in costmaps refer to high cost regions.

Table 3: MHD Results for Conditional Training

Dataset	Independent	Combined	One-Out
SafeCenter	1.783	1.968	2.402
SafeEdge	2.960	2.950	2.980
FastCenter	2.694	2.951	3.934
FastEdge	4.616	4.104	4.878
Average	2.993	3.013	3.5486

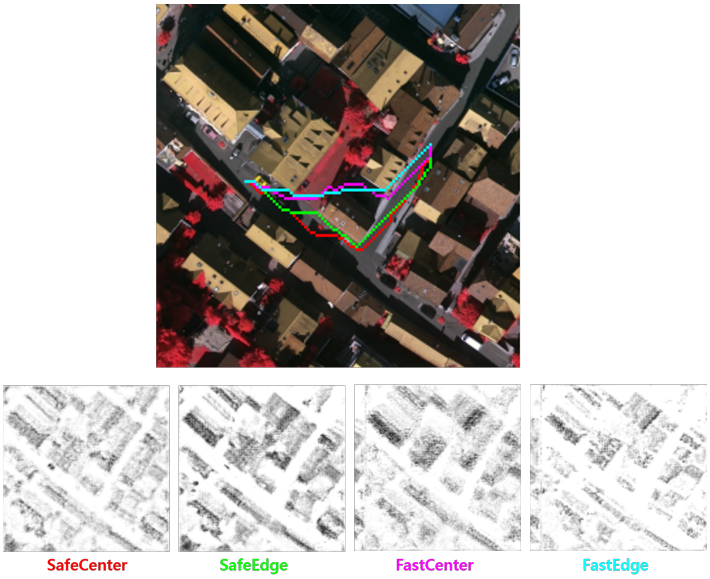


Figure 13: Results from Combined Training on ISPRS Dataset. Top row shows recovered trajectory for each navigation behavior colored in red (SafeCenter), green (SafeEdge), pink (FastCenter) and cyan (FastEdge). Images in the lower rows shows recovered costmaps for the navigation behaviors, with higher cost regions marked in darker colors.

Fig. 13 shows CMRNet generating appropriate costmaps for each navigation behaviors, such as showing gradual transition of high cost buildings to low-cost center of roads for navigation behaviors which favors center of the roads, and assigning cost slightly higher than those of roads to grass fields for where navigation behaviors allows crossing over such areas to cut travel such areas for shortest trajectory.

The last column of Table 3 shows the preliminary results for interpolating behaviors in One-

Out scenario where a unseen navigation behavior of FastEdge was recovered with 5.675% loss in MHD compared to case in which the behavior was trained separately. While One-Out scenario shows slight performance drop for seen and unseen navigation behaviors, this is likely from bigger combined training making use of common navigation features across the behaviors. For example, all ISPRS series dataset provides examples of expert avoiding buildings, allowing CMRNet to exploit the larger size of training dataset for learning such common behaviors.

Effect of Data Augmentation Process

Table 4: MHD Results for Effect of Data Augmentation for SafeCenter Dataset

Dataset	Training Set Size	MU-Net	MRNet
Without Augmentation	118	3.705	3.336
Sample of With Augmentation	118	2.571	2.759
With Augmentation	960	2.512	1.9681

Table 4 shows the impact of data augmentation via 1) increased size of training data and 2) reducing the risk of dataset biases such as sun casting a shadow towards a certain direction etc. Augmentation operation of rotating and flipping images help reducing such biases, improving overall performances.

Efficiency of the forward solver

Table 5: Iteration Speed of MDP solvers during training*

Results, Algorithm	MEDIRL[150]	Proposed
Time Per Task(s)	660.0	0.1967

*Average result of generating 50 trajectories in 256x256 gridworld using Intel(R) Xeon(R) CPU X5690 @ 3.47GHz

Table 5 shows that our proposed forward solver can find solution roughly 3,000 times faster than the value iteration solver used in [150] for every iteration. This is because unlike the former, the later needs to calculate and update values for all cells until convergence (average of 95.5 updates per map of size 256 x 256), uses computational expensive exponential operations to calculate probability distribution of policies for all states, our proposed method only explores, calculates, and updates states just enough to find a trajectory that, on average, consists of 178.4 states in a 256 x 256 grid.

Evaluation on multimodal approaches

The results shown in Table 6 demonstrate that the proposed multimodal approach outperforms all of the single modal approaches including the state-of-the-art LIDAR-based model [150]. The results are not surprising because more context can be captured through multimodal sources than any single modality, subsequently reducing the level of ambiguity in understanding environments. For example, short bushes in 2D images may appear very similar to tall tree branches in terms of texture, shape, and color. Concrete buildings may look similar to concrete pavements, both of which share the same texture of grey concrete and similar shapes such as rectangles. Such ambiguity can be resolved by incorporating height information in 3D data. Likewise, while the 3D data generally lack semantic information to distinguish certain objects of a similar shape. For instance, the center or side of the road both of which appear flat in 3D; using color or texture information available in images such ambiguity can be resolved.

Among the multimodal models tested, the M2RNet outperformed all other approaches in the experiments. The M2RNet maximally utilizes both input modalities with multiresolution parallel convolution operations. In this way, various types of information can be extracted from both 2D and 3D data, including texture information coming from the difference between a cell and its immediate

surroundings, geometric information of how the cells of similar texture can be grouped together, and spatial information on how such groups are located in relationship to each other.

Table 6: The performance of multimodal approaches in MHD; the lower, the better (Bold: best, Underline: second)

Model	MHD (CARLA)	MHD (ISPRS)	Resolution (3D)
MRNet (Image)	1.281	3.822	N/A
MRNet (3D)	5.654	N/A	0.8m
MRNet (3D)	N/A	11.005	0.2m
MEDIRL [150]	2.690	11.005	0.2m
MAXENT [164]	2.382	N/A	N/A
MIPNet	0.906	<u>2.523</u>	Same as Image
M3PNet	**0.891**	3.624	Same as Image
Multimodal Early Fusion	5.654	3.994	Same as Image
M2RNet	<u>0.894</u>	**2.229**	Same as Image

2.9 Summary

We propose a Conditional Deep Inverse Reinforcement Learning framework for generating cost maps from aerial imagery for ground vehicle navigation. Our main accomplishments are:

- We propose an efficient deep learning pipeline for using high-resolution aerial and satellite imagery for navigation in a normally computationally expensive IRL framework;
- We introduce two new networks for visual learning from high-resolution aerial-view imagery and share the lessons learned from our experiments on how to design a network architecture for effective cost map learning for navigation;
- We propose a conditional inverse reinforcement learning that can be trained for multiple navigation behaviors, with a potential for generalization such that a cost map can be predicted for a new behavior as long as it is related to known behaviors;
- We create a dataset consisting of 9 different environmental settings including pre- and post-disaster scenarios and multiple navigation behaviors;
- Research into effects of network structures and fusion techniques on the performance of our framework; and
- A paired dataset including multimodal input and expert demonstrations that can be used to train IRL algorithms for autonomous navigation behaviors in challenging environments such as postdisaster scenarios.

3 Learning new class in unseen condition

3.1 Problem definition

Semantic Segmentation in computer vision refers to a task of densely classifying each pixel to a pre-defined known class. Due to its fine-grained labeling procedure, segmentation often does not have as many training images and datasets as image classification and object detection. However, it provides an interpretable output that can be used for many applications such as path planning, scene understanding and question and answering. Therefore, we aim to tackle the problem of learning

to segment unseen classes from only a few training samples, also known as few-shot segmentation learning.

The goal of few-shot semantic segmentation is to learn a semantic map of the interested and unseen class given a few support images, their corresponding ground truth semantic map, and a query image to be predicted. Commonly there are two types of data partitioning, one where training set contains the unseen classes but are labeled as background, and the other excluding all images containing unseen classes. The former suits the scenario where annotator becomes interested in sub-class of existing class, e.g. green breaks into high vegetation and grass, or annotators starts with large background class and incrementally segments out classes of interest, e.g. adding a class of buildings out of background. In our experiments, disaster scenario typically does not have pre-existing training samples. Therefore, we choose the latter where training set excludes all images containing test (query) classes.

3.2 Related work

Current breakthrough in deep neural networks, in particular convolutional neural network (CNN) has led to a series of model conception such as U-net [122], FCN [92], and Mask R-CNN [59]. These models enabled convenient end-to-end feature extraction and performed exceptionally well on various visual tasks such as recognition, detection and segmentation. However, when extending the model to a different task or even learning a new class within the same task, these models require a large amount of data, which in real-world results in effort and time in collecting and annotating samples. Such problem becomes even more significant in the tasks of segmentation, where dense and fine labels are required in training.

To address the common issue of long tail distribution with insignificant data, several methods were proposed. One direction is to reduce annotating time by using weak supervision signal such as image-level label [72, 63], bounding boxes [34, 64], and even points [14]. Even though these algorithms reduce annotating time, they still require a large pool of training samples. Moreover, often due to using many training data to fine-tune the last few layers, these methods quickly overfit the training and become insensitive to new incoming data, yielding great performance on learned classes but poor result in newly unseen classes with only a few training samples.

Therefore, to directly tackle the issues of limited annotation and samples for a given unknown concept, we start with one-shot segmentation. The goal is to output a binary mask of an unseen class given a pair of support and query images containing the same unseen class and the binary ground truth mask for the support image. One simple approach, as mentioned above, is to fine-tune on the pre-trained segmentation network. However, such technique is prone to overfitting due to potentially millions of parameter updates. In addition, tuning hyperparameters such as learning rate, momentum, number of iterations and layers often resorts to heuristics and can be hard to determine. In contrast, meta-learning [78] abstracts such parameter tuning away by letting the meta-learner infer base classification or segmentation model parameters [104, 28, 143]. This approach often combines base network with metric learning loss to back propogate, aiming to learn the projection function to transform feature embedding space for distance comparison using simple nearest neighbor. Due to easily overfitting the training set, the meta-learner often uses a shallow network to regress the projection function.

Alternatively, inspired by the meta-learning parameter generating approach, the two-branch models [126, 116] replace meta-learner with another similarly deep network to directly infer feature from the support image. The assumption is that the new class in support and query images should have similar appearance, therefore, the base segmentation model and the conditioning model should produce similar features. Although less proned to overfitting and much easier to train, two-branch model significantly increases the model parameters, by the nature of two separate networks.

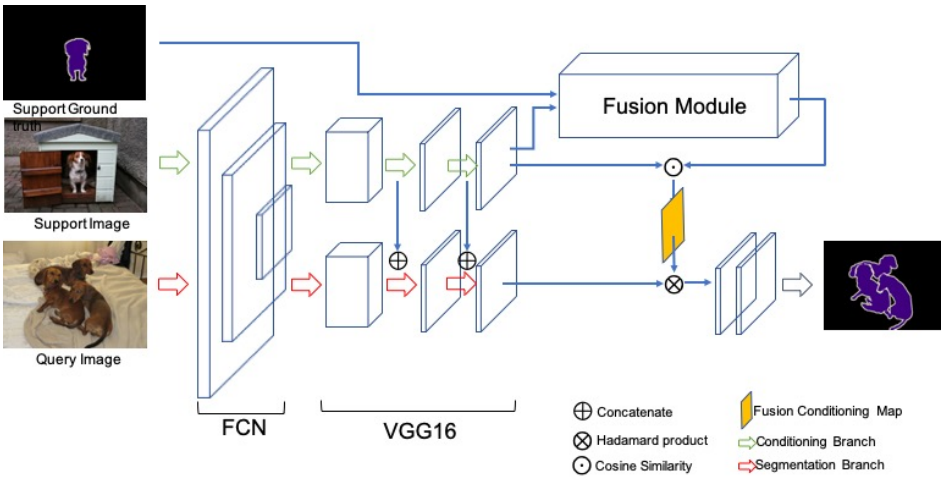


Figure 14: FuseNet: overall architecture of one-shot example consisting of segmentation and conditioning branches

3.3 Our Approach

Our work builds on the two-branch conditioning network and aims to tackle the above mentioned problems of overly-strict conditioning by extending one-shot to few-shot with more variety of support samples.

First, by combining the base segmentation network and the conditioning network into one common fully convolutional network [92], we gain the benefit of the deep feature extraction and the reduced total number of parameters. Since the query and support samples come from similar distribution within the same dataset, we hypothesize that it is sufficient to train one segmentation network for both branches. To ensure similar distribution, we add information flow from conditioning branch to segmentation branch by concatenating support image feature with query feature in the last 2 layers of VGG16 as illustrated in Figure 14.

Second, we create Fusion Module, a plug-and-use module for masking intermediate feature with binary ground truth map. Instead of masking out background from the input of support image, leaving out other close-to-boundary information that may be helpful and creating unnatural RGB image, we think the local context around the new class can be used as a guidance. By fusing the mask with intermediate feature out of VGG, the network is able to look at local contextual feature of the new class.

$$f_i = \frac{\sum_{w,h} Y \circ F_i}{\sum_{w,h} Y}$$

where $\circ$ is element-wise matrix multiplication, f_i represents the ith value along final output vector, f, from Fusion Module, Y is the $w \times h$ ground truth binary map, and F_i is the ith slice of feature map after bilinear interpolation.

Third, we extend from one-shot learning to few-shot learning by guiding with the sum of multiple conditioning maps weighted by pairwise cosine similarity with the query feature map (see Figure 15). One-shot often performs poorly when the support and query images have vastly different perspective or background such as clutter. To improve robustness and increase mean Intersection over Union (IoU), we compute the fusion conditioning map for each support image. Unlike [126] and [116] where they take the union of all K outputs of semantic maps,

$$\hat{Y} = max(\hat{Y}^1, \hat{Y}^2, ..., \hat{Y}^K)$$

, where Y is the fusion conditioning map for kth support image,

or simply taking the average of fusion conditioning maps to use centroid as representation, we think the conditioning should not be decoupled from the query image. Therefore, we weigh each fusion map according to the distance, measured by cosine similarity, between their global feature map and the query global feature map.

$$s_i = \frac{v^{query} \cdot v_i^{support}}{||v^{query}||_2 \cdot ||v_i^{support}||_2}$$

$$\hat{s_i} = \frac{s_i}{\sum_i s_i}$$

where $\hat{s_i}$ is the scalar weight for ith fusion conditioning map, v^{query} is the vectorized query feature map from VGG, and $v_i^{support}$ is the ith vectorized support conditioning feature map.

We hypothesize that the coarse label from image classification can be helpful for dense segmentation. Inspired by image classification, we couple query and support image pair through their respective feature maps from two branches, and exploit image-level feature, mimicking the classification label by using the same high-level features out of VGG. Such pairwise distance represents how similar the query and the ith support image are. The intuition is to weigh the local contextual features, i.e. fusion conditioning map, from support images alone more heavily when they resemble the query image globally, i.e. similar background and scenery out of entire image.

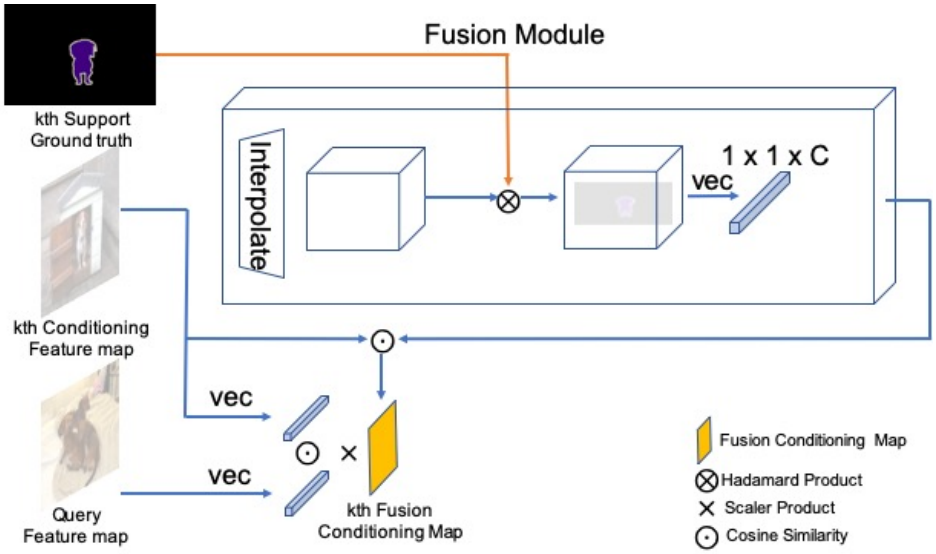


Figure 15: Fusion Module: detailed illustration multiple support images. Note that we normalize the sum of all cosine similarity scalar weight to 1, hence one-shot case would be identical to few-shot with scalar $s_1 = 1$.

3.4 Experiments setup

We follow the standard training setup and data partition for PASCAL VOC 2012 as [126]. Specifically, we partition datasets into a training set, D_{train}, a support set, $D_{support}$, and a testing set, D_{test}, where support and testing share the same classes disjoint from training. Our network learns from the training set by further dividing D_{train} into D_{train}^{sup} and D_{train}^{test}, to simulate $D_{support}$ and D_{test} in testing time.

Dataset	Test classes
PASCAL-A	aeroplane, bicycle, bird, boat, bottle
PASCAL-B	bus, car, cat, chair, cow
PASCAL-C	dining table, dog, horse, motorbike, person
PASCAL-D	potted plant, sheep, sofa, train, tv/monitor

Table 7: 4-fold cross validation for PASCAL VOC 2012

As illustrated in Table 7, we use 4-fold cross validation and average the test samples within the class to get class IoU. We then compute the mean across 5 classes to get the mean IoU for that fold.

For KAIST dataset, we have two types of locations, city and suburb, in which contains normal scenario and disaster scenario, e.g. fire. The fire scenario is produced using image stylization from the orignal normal scenario. Therefore, the training set normal does not contain any fire class. We randomly select a pair of support and query image (I_{sup}, I_{query}) out of fire set, and ensure that the two images are different enough by thresholding the time difference according to frame number.

3.5 Results

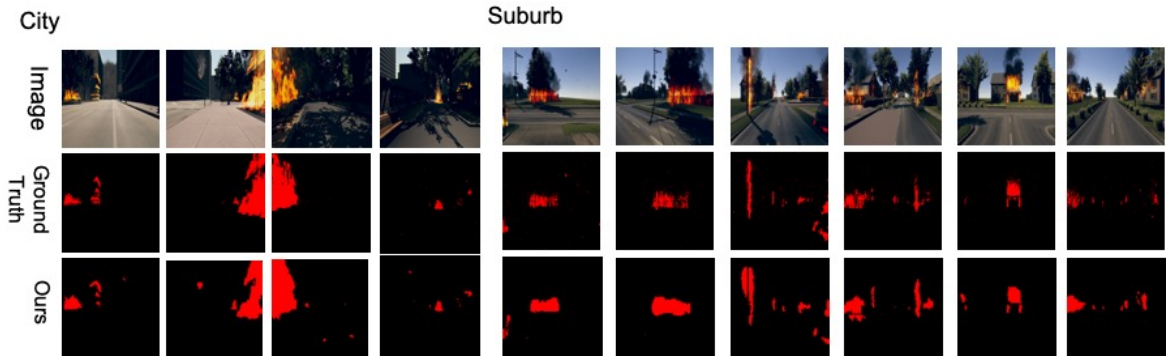


Figure 16: Visualization: One-shot semantic segmentation using FuseNet on KAIST city and suburb fire scenario

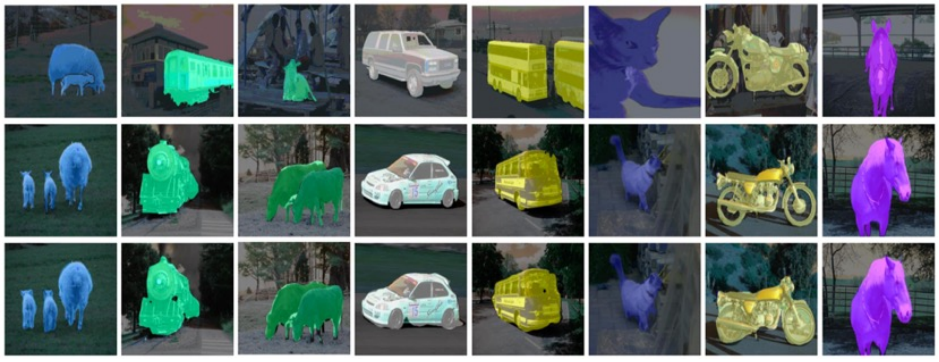


Figure 17: Visualization: One-shot semantic segmentation using FuseNet on PASCAL VOC 2012, visualization tool from [160]

Methods(1-shot)	PASCAL-A	PASCAL-B	PASCAL-C	PASCAL-D	Mean
1-NN	25.3	44.9	41.7	18.4	32.6
LogReg	26.9	42.9	37.1	18.4	31.4
Siamese	28.1	39.9	31.8	25.8	31.4
Fine-tuning	24.9	38.8	36.5	30.1	32.6
OSLSM	33.6	55.3	40.9	33.5	40.8
co-FCN	36.7	50.6	44.9	32.4	41.1
Ours	**40.3**	**58.6**	**47.9**	**38.7**	**46.4**

Table 8: Mean IoU for one-shot semantic segmentation given class partition from Table 7

Methods(5-shot)	PASCAL-A	PASCAL-B	PASCAL-C	PASCAL-D	Mean
1-NN	34.5	53.0	46.9	25.6	40.0
LogReg	35.9	51.6	44.5	25.6	39.3
OSLSM	35.9	58.1	42.7	39.1	43.9
co-FCN	37.5	50.0	44.1	33.9	41.4
Ours	**45.3**	**61.6**	**52.1**	**41.7**	**50.2**

Table 9: Mean IoU for 5-shot semantic segmentation given class partition from Table 7

Methods	1-shot	5-shot
FG-BG	55.1	55.6
OSLSM	55.2	-
co-FCN	60.1	60.8
Ours	**63.3**	**69.3**

Table 10: Mean IoU across all classes by [116] evaluation metric

We found that the alphabetical partitioning order according to [126] results in significant performance variance across folds. From Table 8, partition B performs well primarily due to those images are predominant in ImageNet [35]. Therefore, the initialization using pre-trained model of VGG16 implies that the model and weights have seen many samples and image level labels, i.e. classification, of those classes, whereas potted plant from partition D yields intricate boundaries that are hard to capture.

From Table 8 and 9, we can see that our fusion module significantly outperforms the second place co-FCN [116] in the basic one-shot case. In addition, our 5-shot model, employing global similarity weighting of fusion map, also achieves the state-of-art result by large margin.

From the visualization of KAIST dataset, we observe that in general, our model predicts reasonably accurately for most fire within the image. It can also handle perspective shift given the same fire. For example, suburb first and second images are the same fire but seen from different angles.

However, similar to potted plant in PASCAL, fire has intricate boundary whose shape can be hard to predict. Moreover, due to stylization, there are small clusters of dotted fire across the scenes based on ground truth. Yet, our model fails to capture those clusters most likely due to multiple layers of convolution filter of size 3, which averages out the minute but possibly salient feature and treating those as noise. Moreover, the high contrast due to significant intensity change in-and-out of shadow, within an image and across the two types of locations, produces some false positives prediction, as evident in the third and fourth sample in city.

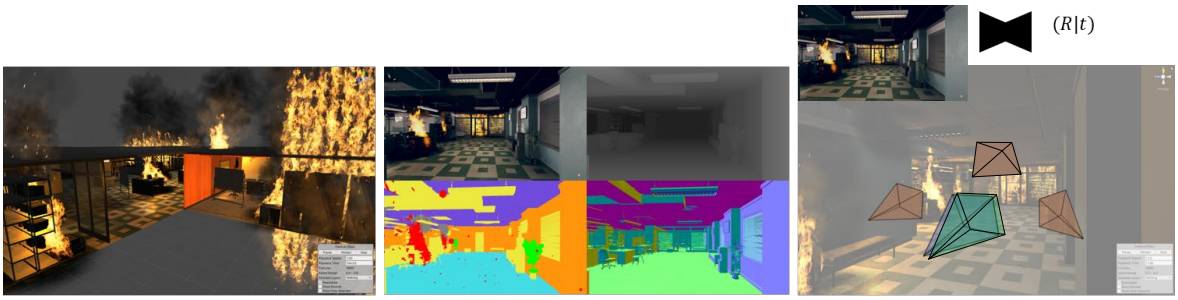
We plan to employ pre-processing such as color temperature transform to reduce simulation artifacts, e.g. abrupt shadow change, to further improve one-shot segmentation performance. Furthermore, we think fusing another boundary map of that unseen class along with the ground truth map can be helpful in predicting query image boundary. Given our general purpose fusion technique, we can weigh the boundary more heavily by producing a boundary fusion map in addition to the original fusion conditioning map for each support image. Common techniques such as errosion and dilation can be applied to auto-extract boundary in the pre-processing stage.

3.6 Summary

Learning-based method has become the predominant feature extractor in the structured input such as images. However, most architectures require significant amount of data and annotations, especially in the task of semantic segmentation, where dense pixel-level label is required. Few-shot semantic segmentation aims to replace large amount of training data with only a few training samples, namely support samples. In this report, we extend one-shot learning to few-shot and propose a new network, FuseNet, that merges multiple feature vectors from support images and leverages cosine similarity as guidance to predict segmentation mask. We also explore the effects of number of support images quantitatively on Intersection over Union(IoU) and qualitatively on visualization. We collaborated with KAIST project by applying our FuseNet to few-shot learn from disastrous scenarios such as fire. The visualization shows that the semantic map closely matches the ground truth generated by the original stylization. In addition, we also achieve state-of-the-art result on standard segmentation benchmark, PASCAL VOC 2012, for both 1-shot and 5-shot semantic segmentation.

4 DIsaster SCenarios (DISC) Dataset

In disaster scenarios such as buildings on fire and earthquakes, the local environment can be seriously damaged, and this has led to growing demand for an automatic sensor fusion system that can visually survey the affected areas. Sensor fusion systems can perform various visual survey tasks, including visual perception, mapping, and localization, from an egocentric viewpoint, which is needed by the first responders. However, few works have addressed the unique challenges posed by the extreme conditions of disaster environments. One reason for the lack of existing works on disaster conditions is



(a) Virtual 3D scene with disaster effects (b) Rendered Dataset of Fully-labeled Images (c) CNN-based visual localization

Figure 18: Examples of the DISC: We provide stereo image sequences with corresponding ground-truth data including depth map, surface normal, optical flow, semantic label and camera poses for both before and after disaster scenarios. With DISC, we propose a CNN-based visual localization to infer 6-DoF camera poses, which is robust to extremely changing conditions. The visual localization is trained on pre-disaster scenarios and is then tested on post-disaster scenarios. Figure (c) elaborates the localization process where among 4 pose queries (red and green), the green estimated pose is the best candidate compared to the ground truth pose (blue).

the scarcity of suitable datasets. For instance, it can be difficult to obtain a pair of accurately labeled images of the exact same place before and after a disaster.

In the past several decades, research on computational photography has focused on high-quality image acquisition and recovery in bad weather conditions such as low-light and haze conditions. In particular, convolutional neural networks (CNNs) trained with large-scale datasets can turn images captured under bad weather conditions into high-quality images [48, 153]. Such accomplishments are possible because weather effects, unlike other extreme cases such as disaster scenarios, can be successfully modeled mathematically, and this has allowed the generation of large-scale datasets of various weather conditions which can be used as needed. As an attempt to fill the dataset gap, one research direction with growing interests involves generating synthetic images with the corresponding ground truth images using 3D computer graphics [120, 49, 114, 36, 84, 57, 96, 103, 73]. Various features of graphics tools, such as texturing and shading effects, allow full control over the virtual 3D environment, ensuring low cost, great flexibility, limitless variety, and quantity. Moreover, the physics engine built into the tools supports an approximate simulation of certain physical systems, such as fire, smoke, and fluid dynamics. This synthetic datasets, simulated using physical phenomena, can extend the range of visual perception to include severe conditions that are rarely found in existing datasets [120, 49].

In this project, we develop a large-scale disaster scene dataset and showcase a visual localization approach that has been developed using our synthesized dataset. First, we present a large-scale synthetic dataset for DIsaster SCenarios (DISC), simulated using a physics engine of both normal and post-disaster scenarios as shown in 18(a). We introduce a dataset with 300,000 images with two types of damage scenarios: 1) collapsing structure and 2) building on fire scenarios. The input modality is high-resolution stereo video data. Its well-annotated ground-truth labels are provided for scene understanding and visual perception tasks as shown in 18(b). We present a manual process for augmenting and retouching disaster effects to achieve photo-realistic disaster effects, which cannot be performed automatically. In addition, we show a technique for extracting a mask for fire and fog, and labels for the rubble of collapsed buildings in disaster scenes (see 4.2). We perform extensive experiments to evaluate the performance of the state-of-the-art methods for several vision tasks including semantic segmentation, surface normal estimation, depth estimation, and optical flow estimation for before and after disaster scenarios in DISC. In particular, the state-of-the-art methods fine-tuned on DISC show significant performance improvements for disaster scenes, as well as with real-world disaster imagery

(see 4.4).

Second, to showcase the successful use of the DISC dataset, we introduce a learning-based egocentric localization, where the goal is to infer the 6 degree-of-freedom (DoF) camera poses consisting of translational and rotational information from a query image with respect to a scene model in 18(c). Conventional visual localizations [129, 20, 142, 56] establish scene coordinates [129] that map image patches to corresponding dense 3D points in a scene model. The studies show that camera pose estimation based on local features works well under the constraint of similar viewing conditions. For advanced cases, high-level scene abstraction from semantic information is utilized to find the matches between a query image and corresponding points [105, 124, 138]. Such approaches can handle appearance and illumination changes over time. These approaches still suffer from the scene geometric changes, which can increase the variations of semantic labels. This bias can be a more serious issue in certain problem domains where there are large changes in appearance, for instance, in post-disaster environments.

Motivated by the idea of reducing the texture bias in [52], we develop a CNN architecture that is robust to appearance variances, and robust to scene appearance and geometry changes. Our ideas in building the CNN are twofold: 1) increasing shape bias, and 2) estimating the dominant planes of scenes such as road, floor, building, wall, and ceiling. Our CNN is trained on both reference images and stylized images to increase shape bias as in [52]. In addition, our CNN computes the dominant planes of scenes using 3D depth information, which acts as a simple version of scene coordinates and can handle scene geometry changes. This is made possible by the use of a plane structure-induced loss represented by 3D information [154] to estimate the planes. With this end-to-end network for visual localization, we are able to reliably localize a camera in a known scene. While comparison methods demonstrate an accurate pose estimation, our technique has the ability to robustly infer the camera poses and achieves better scores in coarse pose estimation. In particular, our network shows promising performance on a challenging synthetic dataset representing disaster scenarios such as buildings on fire and destruction from an earthquake using our synthetic dataset. A set of ablation studies also indicates that each of these technical contributions leads to appreciable improvements in camera poses prediction.

Our work presents a unified version, incorporating our previous works on generating a large-scale virtual dataset for disaster scenarios in [70], and an end-to-end learning-based visual localization that is robust to vastly changed conditions in [71]. The effectiveness of the unified method is extensively examined through both qualitative and quantitative evaluations.

4.1 Related Works

4.1.1 Large-scale Dataset Generations

Real-world Dataset Recent progress in CNN-based computer vision applications has been made possible by the introduction of new datasets. These include large-scale datasets with annotated ground-truths such as ImageNet [123], Pascal VOC [45], Microsoft COCO [89], SUN [151] and Cityscapes [32].

Technological advances in capturing devices now make it possible to acquire additional information about scenes, such as depth, optical flow, and camera motions: RGB-D sensors support research on indoor 3D estimation and semantic segmentation for scene understanding [107, 131]. Sensor fusion systems consisting of cameras, 3D LiDAR, and inertia sensors on automobiles [51, 100, 93, 12] play an important role in the development of autonomous driving applications such as depth, optical flow, visual odometry, object detection, and classification. Datasets collected from unmanned aerial vehicles [121, 62, 39] offer new viewing perspectives, enabling new computer vision applications. Recently, large-scale datasets including aerial and street views with systematical annotation have been presented [147, 145]. However, most of these datasets rely on the costly manual annotation of labels, which does not only ensure high-fidelity annotation but also requires considerable efforts.

Synthetic Dataset Virtual worlds can be useful for generating large-scale datasets with well-annotated ground-truths. These datasets have been used to evaluate the performance of a variety of computer vision algorithms and to train CNNs for visual perception tasks such as surface normal [114], optical flow [36, 120], scene flow [96], disparity [114], multi-view stereo [84, 57], visual odometry [120, 57], feature descriptors [73], visual surveillance system [135], pedestrian detection [120], semantic segmentation [114, 120], multi-object tracking [49, 120] and UAV tracking [103]. Such approaches not only improve the performance of the CNNs, but also contribute to enlarging its range of applications. In particular, since synthetic data comes with annotations for free, these datasets potentially enable training on a much larger scale than would normally be possible with real data. In addition, domain adaptation techniques that either learn an extra mapping step to reduce the domain representation gap [95] or domain invariant representations [140] support the effectiveness of generating datasets in the virtual world and using them on real data.

Compared to these datasets, to the best of our knowledge, our work is the first one to provide a synthetic dataset which simulates before and after disaster scenarios like fire and collapse. Moreover, our dataset contains changes depending on the severity of the disaster. We validate the realistic nature of the datasets by applying state-of-the-art CNNs trained on DISC in real-world disaster data.

4.1.2 Egocentric Localization

The position and orientation estimation of a mobile object (e.g., a robot) or wearable devices from egocentric images is crucial to many industrial applications [132, 156, 115]. Spera *et. al.* [132] introduces a large-scale dataset for egocentric images in retail stores and proposes image-based egocentric shopping cart localization. A wearable-assisted navigation system is also studied for task-specific route guidance. Qian *et. al.* [113] proposes to learn a global metric-topological abstract map of outdoor workspace using a combination of stereo cameras and inertial measurement units. RAGUSA [115] has investigated the problem of localizing visitors in a cultural site using an egocentric viewpoint. We also predict a 6-DoF camera pose from egocentric images taken across large appearance variations. We refer the readers to the following papers for a comprehensive review of image-based localization.

Regression-based Localization Starting with PoseNet [76] which regresses 6-DoF camera poses directly from GoogLeNet [134], visual localization has been actively studied in the robotics field. This work is extended by adopting a stochastic Dropout [50] in [74], and geometric loss minimizing reprojection errors in [75]. Long Short-Term Memory units are incorporated into CNNs to learn structured feature correlations in [144], and to exploit temporal dependencies in input video streams [29]. MapNet [23] takes a data-driven approach to bring geometric constraints used in SLAM or structure from motion into a CNN framework.

For more reliable camera poses prediction, recent works have taken advantage of the concept of scene coordinates [87, 19, 21]. In [87], a dual-stream CNN encodes input color and depth images separately to minimize a weighted Euclidean distance between ground-truth (GT) and the predicted camera poses. A differentiable RANSAC (DSAC) proposed in [19] enables an end-to-end CNN with the scene coordinates from a single input image assumption to optimize its initial prediction. In [21], a CNN regresses the scene coordinates using RGB-D images. Initial predictions of the CNN are refined using a differentiable module for counting the number of inliers of 2D-3D matches. CNN-based approaches, however, can exhibit strong texture bias of images [52].

Retrieval-based Localization Instead of directly regressing the absolute pose of the query image in the scene, the localization problem can be solved in two distinct stages, known as hierarchical localization. First, given a query image, an image retrieval based technique [9] is used to find the most similar images in the database. These images are then utilized to accurately estimate the pose of the camera via a sparse keypoints matching strategy, followed by a robust perspective-n-point technique [119]. We note that the second stage of this pipeline does not necessarily need to be achieved by sparse keypoint matching. For instance, in DenseVLAD [139], the camera pose is estimated via an image synthesis strategy given an approximated depth map [139]. These retrieval-based localization techniques are particularly effective under challenging conditions (see Section 4.8); however, additional

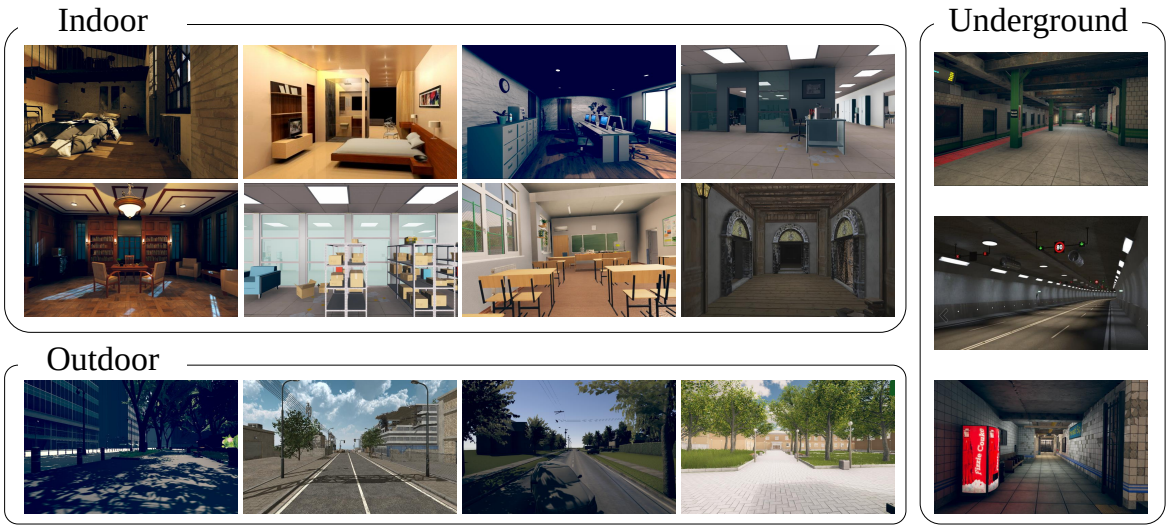


Figure 19: Virtual 3D models to generate the DISC with various scene contexts, light conditions and materials. (Indoor) furniture shop, living room, office, police station, residence, warehouse, school and old castle. (Outdoor) city scape1, city scape2, suburban and park. (Underground) subway station, tunnel and underpass.

information such as prior sparse 3D map of the scene or depth maps would be required to estimate the absolute pose of the camera.

Visual Localization in Changing Conditions A key issue of visual localization in changing conditions is to match new (query) images to previously recorded (database) ones. In [106], a directed acyclic data association graph with a hand-crafted descriptor is built to formulate this problem as a minimum cost network flow problem. This network flow has been shown to provide a higher gain when integrated with more robust image descriptions from CNNs. In [10], an illumination invariant binary description of image sequences over seasonal changes is utilized and its similarity is computed in terms of the Hamming distance between the binary descriptors of the database and query images.

The use of semantic information is another approach to predict the location information of a model [105, 124, 138]. In [105], a CNN model achieves robust global localization by learning visually discriminant regions of interest. The CNN then aggregates the output features from both a reference image and a query image prior to matching them. Using a depth map and a semantic map of a query image, variational CNN captures high-level geometry and semantic information for scene completion [124]. A descriptor from the high-level information is embedded in the Euclidean space and can be matched efficiently. In [138], a semantic consistency score is proposed to match semantic labels between a scene model and a query image. As a last step, a weighted RANSAC refines an initial estimated poses from the score.

4.2 Data Generation

We first simulate and render before and after disaster scenarios of 15 virtual places including indoor, outdoor and underground scenes in 19. We use public 3D models to ensure the scalability of our synthetic data. For each 3D model, we capture stereo video sequences following pre-defined camera paths in normal situations. We then create realistic composite disaster effects on the 3D model and re-capture them in the same paths.

(a) (b) (c)

Figure 20: An example of simulating fire scenarios. (a) Soot image samples. (b) Soot patch composition. (c) Adding light sources

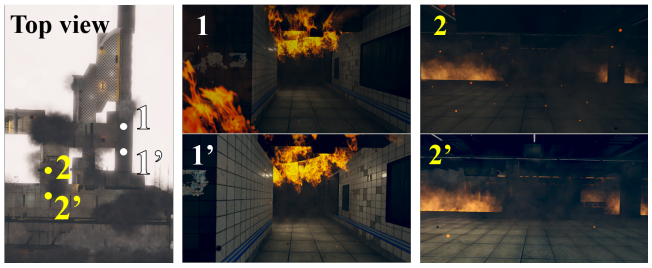


Figure 21: Examples of reflection changes over time in DISC.

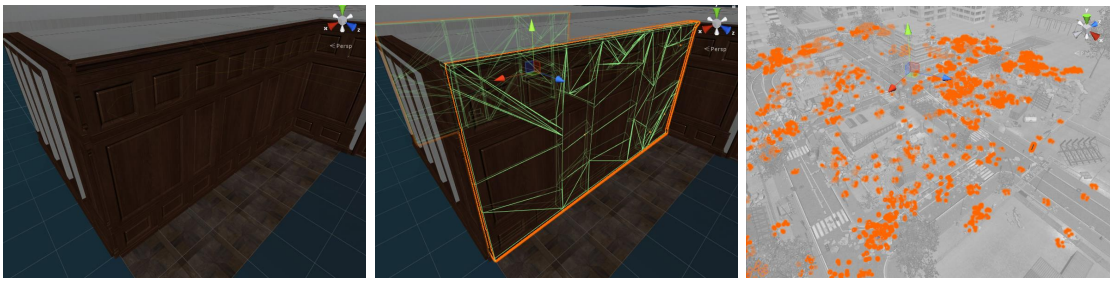
4.3 Simulation of disaster effects

To generate realistic disaster scenarios, we introduce efficient ways to composite the disaster effects on normal 3D models. The disaster effects are inevitably composed manually because there are no public 3D models with realistic disaster effects. We then prepare an approach for rapidly producing 3D models with the corresponding ground-truth data. For this task, we utilize Unity [2] which is widely used for modern 3D computer games.

Fire cases We define fire scenes as combinations of smoke and flame, and soot. The colors of flame and smoke depend on several factors, such as the materials being burned and the ambient temperature. In addition, smoke from a fire in an indoor environment significantly reduces visibility. With this observation, we create 3D models including flames, soot on object surfaces and shorten visibility distances of cameras due to smoke. To produce these effects, however, a great deal of manual work is required. We introduce efficient ways to reduce the amount of manual work.

Instead of making flames with the corresponding smoke for every 3D model, we generate a number of flame samples and smoke, and randomly augment them in the 3D models. Using particle effects in Unity, we design 25 different flames and synthesize them with various colors on the 3D models. The shapes of the flames are determined by the direction of the surface from which they originated, such as walls, floors, or ceilings. Note that we can generate more diverse flames if we employ a variety of colorful flame images, such as with green or blue. Smoke effects are generated by the particle tools, and their colors are randomly selected from black to white. The scale of the effects is also determined by manually tuning the starting points and end points of the flame.

Before compositing the fire effects, we synthesize the smoke images in advance with ignition spots in the 3D model in 20(b). Using the spray effect in Unity, we generate diverse smoke and soot images, as shown in 20(a). Their sizes depend on the scale of the fire effects. For indoor scenes, we additionally simulate thick smoke in rooms with reduced camera visibility. We utilize a particle tool for fog, which involves overlaying a color onto objects based on their distance from the camera. The



(a) (b) (c)

Figure 22: An example of simulating collapse scenarios. (a) Before collapse. (b) Cracking 3D model. (c) Scattering small debris.

fog color is randomly augmented, and the fog density is manually set. As the last step for realistic fire effects, we add emissive lighting sources at each of the ignition spots, as shown in 20(c). The intensity of the emitted light is also set in proportion to the flame scale. This approach enables changes in illumination to be represented over time. To demonstrate it, we display examples of the illumination changes by taking two sequential images with 10 frame intervals in 21. Although these images are captured within a few meters of each other, illuminations changes are present.

Collapse Collapse is a phenomenon commonly observed in disasters such as earthquakes and hurricanes. In this case, it is necessary to bring an object down in the 3D model or to disintegrate a wall or ceiling into debris and then scatter it onto the floor. We use destructive models in the physics engine of Unity to produce various collapse scenarios.

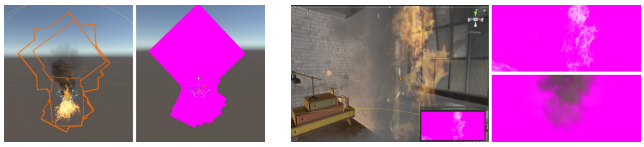
We use a mesh collider in Unity with a manual fracture as the destructive method. We directly apply the mesh collider into target objects or 3D meshes such as walls and ceilings. We are able to control the strength of the collider and then generate randomly distributed fragments on a scene floor. More diverse collapse scenarios which cannot be achieved with the mesh collider can be generated using a manual fracture. For this, we select a particular 3D model in 22(a) and create cracks using the fracture model in Unity in 22(b). These cracked 3D objects are then manually spread over the 3D space. We perform rotation and flipping of the cracked objects to create a greater variety of screen configurations.

Lastly, we produce the effect of debris objects scattered from the building onto the floor. Small objects such as sample building debris provided by Unity are sprinkled around the collapsing 3D object in the scene (see 22(c)). The arrangement, rotation, and size of each piece of the debris are manually determined to reflect the position and type of destruction of the collapsed 3D model. Because the types of building debris provided in Unity are limited, this augmentation is very useful for creating diverse scenarios without sacrificing the realistic nature of the result.

4.3.1 Acquisition of ground-truth data

A virtual environment has the benefit of easily obtaining ground truth data such as depth, optical flow, surface normal and camera poses. Unity supports depth, optical flow and surface normal information with sub-pixel precision, but camera parameters for computer vision tasks cannot be directly recovered from this graphics tool. Moreover, although the semantic segmentation data is also directly generated by outputting a unique color on the surface of an object, accurately labeling materials such as flame and smoke using only graphics tools, including Unity, is not feasible. We present two ways to solve these problems.

Camera Setting We set a fixed vertical field-of-view (FOV) for the stereo camera to capture stereo video sequences to 60° (horizontal FOV varies depending on the aspect ratio a). In out setup, we set the optical axis of the right camera to be parallel to that of the left camera with $0.2m$ baseline on a



(a) Direct labeling in Unity (b) Our labeling strategy

Figure 23: Comparison labeling fluids such as flame and smoke in Unity with in the video editing program.

Figure 24: An example of provided labels for fire cases. Dark gray: smoke (soft label), light yellow: fire (soft label), gray: smoke, red: fire, yellow: furniture, orange: wall, purple: ceiling. The soft label represents a highly detailed transparency-preserving segmentation of flame and smoke.

horizontal axis. The focal length f of the cameras can be simply computed as:

$$f_x = \frac{\hat{f} \cdot p_x}{a_x}, f_y = \frac{\hat{f} \cdot p_y}{a_y}, \text{ where } \hat{f} = \frac{a_y}{2\tan(\frac{\text{FOV}}{2})}, \tag{2}$$

where a_x, a_y are the sizes of the sensor in the x, y axes. The stereo cameras used have the same intrinsic parameters as each other, do not suffer from optical distortions. The principal point of the camera is located in the image center.

Fluids labeling strategy Because fluids such as flames and smoke are created by particle effects, directly labeling each particle is the most intuitive way to obtain the ground truth labels for a fluid. We paint each particle and render label images for these types of fluids as an example. However, this approach fails to obtain the correct labels as shown in 23(a). This problem may be due to a resolution issue related to labeling functionality for particles in Unity or Blender. To address this, we introduce a fluid labeling technique using a video editing program (Adobe Premiere).

First, we assign color labels to objects in the 3D models, but not for the fluids. In this state, we render the 3D model as sequential images. We select each colored region and remove all of them using the background removal functionality of the video editing program. After removing the colored regions, only the pixels corresponding to the flames are left. By colorizing the remaining pixels in the video editing program, we are able to obtain an accurate label for the flames as shown in 23(b). This process is then equally applied to the smoke. Finally, we synthesize the labels of the scene for normal situations. Thanks to the annotation process, we can annotate soft labels representing semi-transparent regions in the fire and smoke well. In total, we provide three types of semantic labels: object-smoke, object-fire and object-smoke-fire as shown in 24.

4.3.2 Dataset

We generate a total of 300,000 stereo image pairs with 1280×720 resolution as video sequences. We create a synthetic dataset suite consisting of 15 subsets and provide complete ground-truth depth in metric scale, optical flow, semantic label, surface normal and camera poses. We render all image data using a virtual FOV of 60°. DISC provides 15 semantic labels (roads, sidewalks, cars, walls, ceilings, furniture, trees, sky, buildings, electric devices, fences, floors, stairs, fire and smoke) in 25. For applications to indoor navigation systems, we individually label fences and stairs.

A machine equipped with an Intel i7 3.40GHz CPU, 32GB RAM and GTX 1070 Ti GPU is used for the rendering step. The rendering time on average is 15 seconds for one stereo pair with the

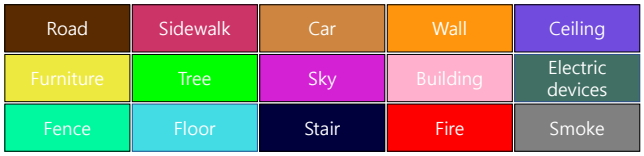


Figure 25: Color code for label in DISC.

Table 11: Semantic segmentation for 15 classes without and with finetuning (FT) on the DISC. (Measure: mean IoU)

Method	Normal	Fire		Collapse	
		w/o FT	w/ FT	w/o FT	w/ FT
FC-DenseNet	0.573	0.427	**0.500**	0.481	**0.540**
SegNet	0.706	0.466	**0.574**	0.517	**0.687**
DeepLab	0.549	0.420	**0.527**	0.506	**0.521**
PSPNet	0.636	0.458	**0.593**	0.478	**0.590**
DenseASPP	0.560	0.471	**0.533**	0.431	**0.499**

corresponding ground truth. When rendering scenes in the presence of particle effects, the average rendering time is significantly increased. To accelerate the rendering speed, we deactivate the physical engine during the rendering procedure, after which we are able to achieve a time of approximately 20 seconds per one stereo pair.

4.4 Benchmarks

In order to demonstrate the usefulness of DISC, we set up several experiments on semantic segmentation, surface normal estimation, depth estimation and optical flow. To establish concrete benchmarks, we split the DISC dataset into a BEfore Disaster set (BE-D) and an AFter Disaster set (AF-D) such as fire and collapse scenarios. In particular, we show that the CNNs used for these experiments, which is firstly trained on BE-D from scratch, achieve remarkable results in disaster scenarios after training on AF-D. The performance improvements are indicated in bold.

For qualitative evaluations on real-world disaster imagery, we downloaded real-world images for the semantic segmentation and surface normal from the Internet using the Google image search tool. We captured real-world smoke scenes in the public safety training center for firefighters to acquire calibrated stereo pairs and sequential images. These images and videos were used for stereo matching and optical flow estimation, respectively.

4.5 Semantic segmentation

We evaluated five semantic segmentation networks on the DISC dataset: FC-DenseNet [68], SegNet [13], DeepLab [25], PSPNet [163] and DenseASPP [155], and we report the mean intersection over union (IoU) as a performance measure. We used approximately 20K images for training and allocated 3K images from eight indoor scenes for testing and validation. There was no temporal overlap between the training and test splits. For the training set, we randomly sample 20K images from the first half of the entire sequence, and the test set is chosen from the last 15% of the entire sequence.

As shown in 11, we are able to draw several conclusions. First, we observed that performance degrades in disaster scenes. In particular, reduced performance is particularly remarkable in dense smoke regions as compared to other disaster scenarios. To achieve a robust solution for smoke conditions, we trained state-of-the-art networks on AF-D. The training sets (about 5% of AF-D) and

Table 12: A quantitative evaluation of semantic segmentation on real-world images without and with finetuning (FT) on the DISC. (Measure: mean IoU)

Method	w/o FT	w/ FT
FC-DenseNet	0.249	0.315
SegNet	0.278	0.350
DeepLab	0.253	0.342
PSPNet	0.277	0.312
DenseASPP	0.266	0.281

the test sets for disaster scenes were temporally separated[1]. 26(a) shows that AF-D helps the networks to recognize scene conditions, and improves the performance for both disaster cases in 26(a). The networks for semantic segmentation learn about the geometric transformations that occur when collapse scenarios happen. When scene layout transformations that are not seen in BE-D occurs in AF-D, the fine-tuned networks infer object classes based on AF-D. As shown in 26(a), the network without fine-tuning classifies the collapsed structure as wall and furniture, but after fine-tuning, the same network classifies it correctly.

We also show real-world results of fire scenes from SegNet, which achieved the best performance in our experiment. As shown in 26(b), SegNet finetuned on AF-D works well with real-world fire scenes. For completeness, we prepare a short quantitative analysis using ten manually annotated images containing fire (randomly scraped from the internet). The results from each network (before and after fine-tuning on AF-D images) are available in Table 12. We also experimented to train the networks directly from AF-D images. Overall, the performance difference - between the fine-tuned networks (using on 5% of the AF-D) and a direct training using all the available data - is marginal. For certain networks (i.e. PSPNet) the pre-training stage has even been beneficial despite the very limited quantity of data used for the fine-tuning. As in our experiments using DISC in 11, the networks finetuned on disaster images systematically demonstrate better performance, compared to the networks trained on BE-D only. However, we observe a performance gap between the results on DISC and real images. This performance decay is a well-known phenomenon caused by the domain shift between the training data (synthetic) and the testing data (real images).

In particular, the images selected for this experiment are very different from our dataset with different viewpoint, semantic context, illuminations and aspect ratio, etc. Recent works have attempted to reduce the domain gap for semantic segmentation in [18, 61, 81] translating source data into the "style" of a target domain. 27 shows an example of unsupervised (pixel-level) domain adaptation (UDA) [81] that transfers the domain in DISC to the real image domain. The result shows that the domain gap between DISC to real images can be reduced using UDA. Thus, we believe that DISC can be a valuable dataset for unsupervised domain adaptation in the context of disaster scenarios.

4.5.1 Surface Normal Estimation

Next, we evaluated state-of-the-art surface normal estimation using single images, specifically in relation to VGG-Multiscale [43], FCN-Skip [161] and HourglassNet [26]. Like 4.5, we show that the performance capabilities of networks trained on BE-D degrade and that these methods yield reliable results after finetuning on AF-D in disaster scenarios.

Our benchmark in 13 indicates that even though the collapse situations change the compositions of scenes, the performance degradation of the networks is relatively minor. On the other hand, fire situations produce limited visibility and irregular lighting changes, which are the main causes of inaccurate surface normal estimations. Another cause of prediction error is the soot, which makes the surfaces of a scene black. In this situation, FCN-Skip and HourglassNet work well after finetuning on

[1]The same experimental strategy employed in [120] is adopted for our benchmarks.

Table 13: Surface normal from single images. Mean and median of angular error (the lower the better), and percentage of pixels with error smaller than 11.25 (the higher the better).

Method	BE-D	Fire (° / ° / %)		Collapse (° / ° / %)	
		w/o FT	w/ FT	w/o FT	w/ FT
VGG-Multiscale	22.19/19.57/30.12	40.03/35.14/11.19	**27.61/22.59/25.45**	28.90/22.07/22.31	**25.42/20.36/27.09**
FCN-Skip	13.92/8.78/57.23	28.87/25.14/19.42	**22.24/16.30/31.37**	16.65/14.09/41.88	**14.20/10.18/45.67**
HourglassNet	19.75/13.02/39.34	38.07/31.92/12.48	**24.58/18.37/28.75**	22.71/16.35/30.02	**20.53/15.55/38.58**

Table 14: Stereo matching. Averaged bad pixel rate as the disparity error larger than 5 pixels (BPR5) and 7 pixels (BPR7), and RMSE in smoke scenes (the lower the better).

Method	BPR5 (%)		BPR7 (%)		RMSE (pixel)	
	w/o FT	w/ FT	w/o FT	w/ FT	w/o FT	w/ FT
MC-CNN	20.02	**15.97**	12.49	**9.41**	5.37	**4.38**
DispNet	25.26	**24.26**	16.92	**16.65**	5.15	**4.60**
PSMNet	18.03	**17.94**	10.75	**8.99**	4.80	**4.17**

AF-D as shown in 28. Both methods use skip links between each pair of corresponding convolution layers in the encoder and decoder of the networks. In particular, the multi-scale feature maps in VGG-16 used in FCN-Skip appear to be advantageous when used to extract informative features in these situations.

The scene understanding benchmarks in 4.5 and 4.5.1 indicates that its performance is mainly determined by how both high-level and local information are preserved, and how context information is aggregated. The skip link of SegNet and the multi-scale pooling modules in PSPNet are effective for dealing with the disaster effects.

4.5.2 Stereo Matching

We evaluate CNN-based stereo matching methods for fire scenarios, especially those with smoke. First, we benchmarked MC-CNN [158] based on a local window matching-based method, DispNet [96] using semantic information from stereo images, and PSMNet [24] which is an end-to-end network exploiting the global context information of input images and a cost volume regularization. We subsequently compared the baseline networks with the finetuned networks to validate the effectiveness of the DISC dataset. In 14, we report the results of quantitative evaluations using the bad pixel rate (BPR) and the root mean square error (RMSE) as performance measures.

We are able to draw two conclusions from this experiment. First, finetuning the networks using AF-D worked well in smoky conditions. In particular, the performance improvement is apparent on all baseline approaches for all error measurements. We found that the depth map accuracy increased the most for poorly visible pixels, as shown in 29. Second, we observed that the performance drop of DispNet was more drastic on disaster scenes. The performance degradation of DispNet is likely due to its high-level feature matching strategy. Direct matching with local windows of the MC-CNN or the spatial pyramid pooling strategy of PSMNet are suitable for depth estimation in smoky areas, and training on AF-D allows the performance to be improved in challenging conditions.

We also conducted a qualitative experiment on real-world datasets, and compared the performance of PSMNet with [88] which is a specially designed stereo matching method for defogging. As shown in 30, the real-world results indicate that the CNN-based stereo matching with training on the DISC dataset works well in practice. In particular, the PSMNet finetuned on AF-D in 30(d) shows promising results over [88] even with little computation (1s vs. 10min). In contrast, finding correspondences in scattering media [88] (e.g. fog, haze, or turbid water) causes a heavy computational burden, and spatially-variant smoke density levels in the real-world data complicate the matching step

Table 15: Optical flow. Averaged BPR and EPE in smoke scenes (the lower, the better).

Method	BPR3 (%)		BPR5 (%)		EPE (pixel)	
	w/o FT	w/ FT	w/o FT	w/ FT	w/o FT	w/ FT
FlowNet2	20.44	**4.53**	8.25	**1.04**	2.81	**0.69**
DCFlow	19,83	**2.25**	10.10	**0.73**	2.17	**0.30**
PWCNet	18.88	**1.30**	9.15	**0.37**	2.37	**0.31**

in 30(b).

4.5.3 Optical Flow

We also evaluated state-of-the-art optical flow estimation methods: FlowNet2 [65], DCFlow [152] and PWCNet [133]. Similar to 4.5.2, we compared these networks trained on BE-D and networks finetuned on AF-D. Errors are measured regarding the average bad pixel rate and end-point error (EPE) in 15.

Fires in indoor environments change lighting conditions, which causes inaccurate optical flows. As a result, inferred optical flows exhibit texture-copy artifacts in 31(b), and this tendency could be seen in the real-world imagery as well in 32(b). In this experiment, we observe significant performance improvements when examples are available for training fire situations. The networks trained on AF-D predict accurate optical flows that are robust to lighting changes and scattering media in 31(c). As shown in 32(c), the realistic disaster simulations in AF-D alleviate the negative influence of varying levels of illumination in the real-world scenes.

Note that a noticeable performance improvement in PWCNet performance is achievable by using a pyramid feature extractor and a context network based on dilated convolutions [157]. We found that multi-scale feature extraction and context information from CNNs are beneficial for correspondence estimation, particularly in reduced visibility conditions.

4.6 Dominant Plane Estimation

Previous visual localization methods for changing conditions assume that the scene geometry of the database and the query images is consistent. Since the positions of buildings and vegetation do not change, counting feature matches between its semantic labels produces good results [124, 138] in 34(a). However, severe structural changes in the same places might cause low semantic consistency matches as shown in 34(b).

Our idea to solve this problem is to relax the assumption such that, although the positions of semantic features at the fine-grained level can change, a high-level, abstract view may still be retained between the database and the query images. This intuition leads us to use the dominant plane information of the scenes as input to the pose estimation network in 33. According to a recent work in SLAM in [85], camera poses are computed by using only structural features that have useful geometric information of parallelism, orthogonality, and/or coplanarity in the scene.

To extract the structural features–i.e., in our case, a family of salient planes in the scene– we use an encoder-decoder network with skip connections and multiple levels of scales for plane estimation [154]. This network predicts plane segmentation maps and plane parameters. As in [37], an input image is passed through the encoder to produce a set of high-level feature maps and then the decoder upsamples the feature maps via a series of deconvolution layers to infer the plane segmentation maps with $m + 1$ channels including the non-planar class where m is a hyper-parameter specifying the number of planes. The multiple scales allow the network to abstract feature maps and the skip connections help preserve high-level information. Another branch of the network infers plane parameters $\mathbf{p}_n = \{a_n, b_n, c_n\}$ where n is the plane index. A 3D point $\mathbf{X}$ belongs to plane n if $\mathbf{p}_n^T X = 1$. This branch shares the same encoder for the plane segmentation maps and has two fully connected layers whose output is $1 \times 3m$ of the m planes.

Table 16: Comparison results on 4.2 for simulating severe conditions. The first row of each block is a position (m) and a rotation ($^\circ$) error, and the second row means the accuracy measures with the high-/medium-/low- precisions. Database for each situation of House, Office1, Office2, and School contains 118, 216, 515, and 2123 images, respectively. The number of query images is 41, 72, 180, and 746, respectively. (**Bold**: Best, Underline: Second best)

	Average	Home (10m × 10m)		Office 1 (5m × 9m)		Office 2 (18m × 20m)		School (90m × 80m)	
		Fire	Collapse	Fire	Collapse	Fire	Collapse	Fire	Collapse
DenseVLAD + SIFT	19.57m ± 15.97, 116.98° ± 4.18 3.57/4.93/6.14	3.03m, 118.42° 2.43/2.43/4.87	2.98m, 119.35° 10.00/17.50/20.00	7.64m, 111.64° 5.70/6.81/7.38	8.80m, 114.11° 1.71/1.71/2.28	22.13m, 116.76° 0.59/1.78/2.97	24.30m, 123.22° 1.46/2.35/4.40	42.99m, 110.89° 3.77/3.77/3.91	44.72m, 121.43° 2.89/3.10/3.31
DenseVLAD + D2Net	3.89m ± 5.81, 9.47° ± 13.11 27.69/40.65/55.02	0.62m, 9.86° 0.00/21.95/51.22	0.04m, 0.76° 65.00/82.50/90.00	0.40m, 3.63° 35.22/50.56/55.11	0.44m, 4.05° 28.00/47.43/62.28	0.53m, 1.78° 35.90/47.18/64.09	2.06m, 3.81° 24.92/35.19/55.42	16.74m, 43.29° 11.59/16.62/26.67	10.32m, 8.61° 20.91/23.80/35.40
NetVLAD + SIFT	1.49m ± 2.36, 18.63° ± 30.25 47.41/55.04/59.35	2.52m, 39.27° 9.75/19.51/24.39	0.03m, 0.53° 68.50/72.50/72.50	0.29m, 3.88° 38.63/50.56/58.52	7.29m, 91.62° 9.14/17.14/20.00	0.03m, 0.17° 88.42/93.17/94.65	0.12m, 1.44° 52.78/60.41/67.15	1.65m, 12.10° 31.56/40.92/48.18	0.01m, 0.05° 80.53/86.12/89.44
NetVLAD + D2Net	0.63m ± 1.24, 11.74° ± 26.09 **52.76**/**64.85**/73.90	0.68m, 8.11° 0.00/21.95/51.21	0.03m, 0.69° 82.50/95.00/97.50	0.13m, 1.85° 50.56/62.50/68.75	3.87m, 80.48° 14.85/28.57/33.14	0.04m, 0.28° 87.83/93.17/94.36	0.10m, 0.97° 57.18/68.91/79.17	0.21m, 1.43° 45.94/61.31/75.97	0.01m, 0.08° 83.22/87.37/91.09
NetVLAD	0.62m ± 1.18, 11.20° ± 26.36 50.93/63.81/74.50	0.66m, 6.73° 7.31/31.70/58.53	0.03m, 0.00° 80.00/92.50/92.50	0.16m, 0.96° 44.88/61.36/68.75	3.70m, 80.71° 14.85/22.28/32.57	0.053m, 0.00° 81.89/91.39/94.36	0.132m, 0.00° 54.83/65.39/79.18	0.23m, 1.16° 42.3/58.59/78.21	0.00m, 0.00° 81.36/87.16/91.92
DSAC	23.84m ± 20.58, 64.87° ± 5.00 0.00/0.00/10.97	8.19m, 75.43° 0.00/0.00/24.39	4.83m, 62.21° 0.00/0.00/33.65	17.17m, 67.69° 0.00/0.00/0.00	11.16m, 60.06° 0.00/0.00/19.44	14.60m, 61.93° 0.00/0.00/0.55	17.29m, 61.75° 0.00/0.00/9.44	61.50m, 68.96° 0.00/0.00/0.00	55.97m, 60.94° 0.00/0.00/0.26
DSAC ++	23.33m ± 16.60, 147.12° ± 9.26 0.00/0.00/0.00	4.87m, 150.59° 0.00/0.00/0.00	4.28m, 167.90° 0.00/0.00/0.00	10.31m, 150.53° 0.00/0.00/0.00	9.85m, 143.01° 0.00/0.00/0.00	32.90m, 139.54° 0.00/0.00/0.00	35.14m, 147.30° 0.00/0.00/0.00	46.94m, 135.40° 0.00/0.00/0.00	42.34, 142.67° 0.00/0.00/0.00
PoseLSTM	4.77m ± 4.67, 34.16° ± 21.10 0.723/3.28/24.98	1.25m, 34.96° 0.00/0.00/21.95	0.61m, 10.86° 0.00/2.50/40.00	6.50m, 62.44° 0.00/0.56/3.40	2.28m, 46.34° 4.00/6.85/16.57	1.34m, 8.20° 1.78/12.46/55.19	14.85m, 58.64° 0.00/0.00/0.00	8.91m, 44.20° 0.00/0.13/12.01	2.38m, 7.67° 0.00/3.72/50.72
PoseNet	9.41m ± 9.51, 52.48° ± 34.19 1.01/6.21/18.48	2.60m, 75.85° 0.00/0.00/2.43	0.38m, 5.16° 7.5/45.00/70.00	5.51m, 51.89° 0.00/0.00/0.56	5.54m, 111.41° 0.00/1.14/10.85	2.77m, 4.34° 0.59/3.56/49.85	9.90m, 36.39° 0.00/0.00/12.32	30.57m, 67.78° 0.00/0.00/0.00	18.04m, 67.04° 0.00/0.00/1.86
PoseLSTM + Style	2.90m ± 2.58, 23.68° ± 13.64 1.79/8.93/33.80	0.38m, 8.17° 10.00/37.50/57.49	0.38m, 8.17° 10.00/37.50/57.49	3.86m, 33.33° 0.00/1.13/19.88	2.37m, 46.95° 1.71/13.14/31.42	1.34m, 8.20° 1.78/12.46/55.19	2.98m, 18.51° 0.58/2.63/27.89	8.98m, 23.86° 0.27/0.83/20.53	2.38m, 7.68° 0.00/3.72/50.72
PoseNet + Style	8.12m ± 7.31, 49.21° ± 37.36 0.60/3.00/17.86	2.34m, 113.00° 0.00/0.00/0.00	0.53m, 11.60° 2.50/12.50/40.00	4.71m, 19.94° 1.70/7.95/28.40	4.96m, 97.22° 0.00/.00/5.15	2.76m, 4.35° 0.59/3.56/49.85	9.89m, 36.38° 0.00/0.00/12.31	21.71m, 44.11° 0.00/0.00/5.30	18.04m, 67.05° 0.00/0.00/1.86
SCORF	18.94m ± 13.84, 96.68° ± 14.60 0.00/0.00/0.00	2.74m, 126.92° 0.00/0.00/0.00	3.32m, 93.49° 0.00/0.00/0.00	8.08m, 112.53° 0.00/0.00/0.00	8.61m, 98.87° 0.00/0.00/0.00	27.71m, 85.27° 0.00/0.00/0.00	26.74m, 82.15° 0.00/0.00/0.00	37.40m, 87.73° 0.00/0.00/0.00	36.93m, 86.49° 0.00/0.00/0.00
Ours	**0.58m** ± 0.44, **3.40°** ± 2.68 21.61/54.29/**88.45**	0.36m, 1.43° 34.72/69.44/95.83	0.22m, 1.29° 33.33/65.71/94.00	0.29m, 2.96° 18.82/66.08/91.94	0.49m, 1.87° 15.44/60.37/89.65	0.33m, 2.69° 19.00/67.80/91.36	0.29m, 1.18° 42.58/70.74/98.02	1.38m, 8.59° 3.62/15.82/74.13	1.30m, 7.16° 5.36/18.36/72.65

For this network, we use a depth-induced loss L_P in [154]:

$$L_P = \sum_{n=1}^{m} \sum_{\mathbf{x}} \Psi_n(\mathbf{x}) \Big[\Big] \mathbf{p}_n \cdot \mathbf{X} - 1_2 + R(\Psi_n(\mathbf{x})), \tag{3}$$

$$\text{where} \quad \mathbf{X} = D(\mathbf{x})\mathbf{K}^{-1}\mathbf{x} \quad \text{and}$$

$$R(\Psi_n) = \sum_{\mathbf{x}} -H(\mathbf{x}) \cdot \log \left(\sum_{n=1}^{m} \Psi_n(\mathbf{x}) \right) -$$

$$(1 - H(\mathbf{x})) \cdot \log \left(\sum_{n=1}^{m} \Psi_n(\mathbf{x}) \right),$$

where $\mathbf{x} = [x, y, 1]^T$ is a pixel of input images in homogeneous coordinates, D is a depth map corresponding to an input image, $\mathbf{X}$ is a 3D point, and Ψ indicates the probability of pixel $\mathbf{x}$ belonging to the n-th plane. $\| \cdot \|_2$ is an L_2 norm. $\mathbf{K}$ is an intrinsic matrix of the camera used[2]. R is a cross entropy-based regularization term which is minimized with constant label 1 at each pixel. We define the dominant planes including building, road, sidewalk, wall and ceiling as planar class, and $H(\mathbf{x}) = 1$ if a pixel $\mathbf{x}$ belongs to the planar class.

The predicted plane segmentation maps and parameters represent the 3D information of scenes. Since the descriptors allow the CNNs to consider large plane parts of scenes that are less likely to suffer from minor geometric changes, e.g., the floor plane in 35, we hypothesize that the descriptors based on dominant plane information can make the pose-prediction CNN more robust to geometry changes.

4.7 6-DoF Camera Pose Prediction

With the estimated plane segmentation maps and its parameters, we predict 6-DoF poses from the pose network. The pose network has a similar structure as the encoder of the plane network, but takes the image and the plane segmentation map as inputs. We reduce $m + 1$ channels of the plane segmentation map to one without loss of information by using the softmax and argmax operations.

[2]We assume that $\mathbf{K}$ is known in this application.

We then pass a concatenation of the image and that one-channel plane segmentation map through the convolution layers.

As illustrated in 35, the 2D representation of the plan segmentation map cannot fully capture the scene geometry errors of the 3D space, which can cause structural ambiguities in scenes. In order to augment the 2D segmentation map with additional 3D information, we embed the plane parameters $\mathbf{p}_n = \{a_n, b_n, c_n\}$ for each plane n into the encoded feature vector in a similar manner presented in [60].

The encoded feature is passed to the two separate, fully-connected pose-prediction layers to generate the position of the camera $\mathbf{t}$ and its orientation $\mathbf{r}$ encoded as the Euler angles [144]. We minimize the pose loss function L_{RT} as below:

$$L_{RT} = \mathbf{t} - \hat{\mathbf{t}}_2 + \gamma \mathbf{r} - \hat{\mathbf{r}}_2, \tag{4}$$

where $\hat{\mathbf{t}}$ and $\hat{\mathbf{r}}$ are ground-truth position and orientation, respectively. γ determines the relative weight of the orientation error with respect to the positional error.

Finally, the loss function L of the whole network is defined as the weighted sum of the plane loss and the pose loss:

$$L = L_P + \lambda L_{RT}, \tag{5}$$

where λ is a scale factor between the plane estimation error and the pose estimation error.

4.8 Evaluations of Visual Localization

In this section, we propose a large series of experiments to underline the effectiveness and relevance of our end-to-end absolute camera pose estimation strategy in the context of disaster environments. For the sake of completeness, we compare our technique against a set of widely recognized methods. These approaches can be divided into multiple categories: hierarchical localization, absolute pose regression, differentiable robust estimator and scene coordinates regression.

Direct absolute pose regression Absolute camera pose regressions allow the 6-DoF pose of the camera to be estimated in a single step. A pioneering approach for direct pose regression is PoseNet [77] which has been followed by a plethora of follows-up such as PoseLSTM [144]. These two approaches are tested on DISC (see Table 16), they have been trained with BE-D and tested on the AF-D.

We observe that these approaches have difficulty predicting accurate camera poses. CNN features from PoseNet are useful for scene discrimination in texture-less regions, but are biased toward scene textures. In comparison, our strategy, which explicitly includes the geometric structure of the scene, is more reliable and accurate regardless the sequence or the type of disaster. Our technique has been trained with style transferred images which can be regarded as data augmentation. For the sake of fairness, we also conduct this test with these two direct regression techniques. Overall, the style transfer improves the results of both techniques without leading to competitive accuracy levels.

Under severe changing conditions, SCoRF [129] fails to produce reliable results even when ground-truth depth maps are used as input. We observe that the initial set of camera pose hypotheses was not generated due to the difficulty in matching 2D-3D correspondences. Finally, the differentiable robust estimation such as DSAC [19] and DSAC++ [22] tend to perform poorly on the entire dataset. One reason for this low performance is that DSAC fails to generate a good initial scene coordinate of query images from a CNN. We observe that the scene coordinates between the database and query images are unmatched. As a result, computing a reliable geometry of scenes is necessary to account for changes in the scene structures.

Our network tackles this problem by taking advantage of the CNN's feature and scene geometry with dominant planes. Particularly, since our network has high shape representation of scenes, it is robust to severe changing conditions. In 39, we show pose prediction results from our network with input database and style transferred images. Although the database images share only small scene structures with the query images, the style transferred images enable the shapes of the entire images to be learned. However, there are several inaccurate prediction points in 39. The errors come from large occlusion and texture disappearances as shown in 37.

Table 17: Ablation Experiment on our buildings on fire and destruction dataset. ST means a style transferred image.

	Buildings on fire	Destruction
# of ST: None	$9.80m$, 26.97° 0.0 / 9.64 / 20.71	$10.73m$, 29.68° 0.0 / 4.19 / 28.15
# of ST: 2	$1.57m$, 9.14° 7.02 / 48.76 / 78.55	$1.62m$, 10.36° 6.79 / 45.27 / 76.04
# of ST: 3	$1.02m$, 6.84° 10.52 / 55.43 / 83.61	$0.98m$, 6.71° 12.82 / 54.39 / 86.12
# of ST: 4	$1.06m$, 6.77° 10.11 / 56.03 / 83.99	$0.98m$, 6.87° 13.09 / 55.63 / 87.01
# of ST: 5	$0.98m$, 6.50° 11.08 / 56.01 / 84.90	$0.97m$, 6.73° 12.93 / 54.27 / 86.88
w/o plane information	$1.51m$, 10.03° 7.69 / 47.99 / 78.34	$1.52m$, 10.88° 6.20 / 46.83 / 74.75
w/o plane map	$2.03m$, 11.58° 8.14 / 40.20 / 79.81	$2.24m$ / 14.39° 3.95 / 37.26 / 65.33

Ablation Study An extensive ablation study was carried out to demonstrate the effect of different component on our network. We summarize the results in 17.

We first examine the performance of our network with respect to the number of style transferred images for one database image. We set the same number of epochs. As shown in tab:ablation, the greater the number of images is, the better the results are. This is because the network learns various situations through more style transferred images. Even though the performance improvement plateaus when three or more style transferred images are used, more images will be beneficial if more various situations exist in the query images. Note that we used three seed images to generate the style transferred images for all experiments in this work.

We next compare our network with and without the plane information including plane segmentation and its corresponding parameters. We found that including plane information leads to significant improvements in performance. In particular, the plane segmentation maps remove unnecessary features of the query images by focusing on regions that are locally preserved from destruction. The plane parameters also help to enhance performance because the plane parameters account for the geometric uncertainty in the predicted plane segmentation maps, learning 3D scene information from the tensor of embedding features.

Lastly, we validate the sensitivity of the prediction accuracy to the types of seed images. Intuitively, our network performs more effectively if we account for prior knowledge of variations between the database and the query image. For example, using night-time images as a seed for the style transfer would be more beneficial for day-night changes as compared to random images [8, 112]. Although selecting seed images based on prior knowledge of a target domain is useful, we also note that it is not required to reduce texture bias.

We categorize the seed images to be used for stylization based on the contextual loss [99] between seed and reference images, which measures the similarity between learned features of non-aligned images extracted from VGG-Net [130] as shown in 38(b). The contextual loss is based on both the semantics of images to compare regions with similar semantic meaning and the context of the entire image. Seed images in the first row in 38(b) are searched based on prior knowledge of the query scene[3], and the other images are randomly selected. With these stylization seed images, we generate 12 different combinations and train our network using each of the combinations as seeds for the stylization. As shown in 38(c), we observe that our network achieves significant improvement in performance with any combination of seed images, compared to the baseline that does not utilize stylized images during training (blue and orange color bars). In addition, since the images with low contextual loss help capture the variation loss of query images as well as increasing shape representations, they contribute to the performance improvement, but the gain is not significant.

[3]We use a keyword "building on fire" for the Google image search.

4.9 Main Contributions

We present a new dataset for simulating disaster scenarios. The dataset includes ground-truth data covering low-level to high-level computer vision tasks. The total number of images exceeds 300K sequential images for 15 different places. As demonstrated by the experiments, the state-of-the-art CNNs trained using our dataset performed well on the disaster conditions, especially when tested on real-world disaster imagery.

With this dataset, we introduce a convolutional neural network for visual localization that is robust to extremely changed conditions. To do this, we have increased the shape representations of the proposed network by augmenting the database images with various textures generated via style transfer. In addition, our network predicts dominant planes using corresponding parameters to establish the image-to-3D correspondences following severe geometric changes. We have then demonstrated the effectiveness of the proposed network and its components through several evaluations of various scenes.

We expect that the release of our challenging datasets with various scenes and realistic disaster effects will stimulate the development of new computer vision tasks, such as obstacle avoidance and trip hazard avoidance [98]. In addition, we show that our visual localization is useful for applications to disaster relief from the perspective of victims who have been affected. As a future work, with DISC datasets, we will study data scheduling methods and training techniques that can build well-generalized visual perception models for both before and after disaster scenarios. Moreover, we plan to simulate more challenging disaster conditions such as underwater conditions [111, 6] and to generate multi-spectral images in virtual worlds for fire-fighting equipment [1].

5 Summary

In this project, we develop a semantic mapping approach to assist first responders who would need to operate in harsh, challenging, postdisaster environments. We conduct research on the semantic analysis of large, high-resolution images to generate navigation plans to meet urgent, dynamically changing needs. Specifically, we design conditional deep inverse reinforcement learning algorithms that can utilize high-resolution aerial-view images to quickly generate traversability cost maps for ground vehicle navigation. Our experiments show that the proposed approach outperforms the state-of-the-art in several datasets including pre- and post-disaster dataset that we created. We develop a learning algorithm for learning a new class of object using only a few training examples through a technique known as a few-shot learning. Our approach is focused on the types of objects that are ill-shaped, unstructured as common in postdisaster environments.Our approach shows the state-of-the-art performance in our experiments on large object recognition datasets.

Our collaboration with the KAIST team has enabled the use of simulation data for such scenarios as postdisaster environments. With the generous support for international students, we were able to host three PhD students from KAIST. Collaboratively, we conduct research on how we can leverage their expertise on 3D reconstruction in learning an improved-quality cost maps and the camera relocalization problem. Based on a recent study that CNN-based methods have stronger bias towards textures than shapes, we used the idea of style transfer toreduce the texture bias in our relocalization problem to improve performances. The collaboration with Dr. Hae-Gon Jeon, a postdoctoral researcher from KAIST, has been particularly fruitful, resulting in the development of DIsaster SCenarios (DISC) Dataset (DISC), a new dataset for disaster scenarios [71, 69]. Collaboration with KAIST also includes the multimodal deep learning where using both imagery and 3D data further improves the quality of generated cost maps [80, 136]. The DISC dataset that the team has produced provides a training and evaluation testbed in disaster response including rare scenes found in other datasets, promoting future research in this challenging problem domain.

References

[1] Flir k2 thermal imaging camera. http://ww4.flir.com/k2/.

[2] Unity. https://unity3d.com/.

[3] Mahdi Abavisani and Vishal M Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.

[4] Airbus. Airbus to provide near real-time access to it's satellite data. https://www.airbus.com, January 2018.

[5] Joel Akeret, Chihway Chang, Aurelien Lucchi, and Alexandre Refregier. Radio frequency interference mitigation using deep convolutional neural networks. *Astronomy and computing*, 18:35–39, 2017.

[6] Cosmin Ancuti, Codruta Orniana Ancuti, Tom Haber, and Philippe Bekaert. Enhancing underwater images and videos by fusion. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Anonymous. Conditional deep inverse reinforcement learning for multiple navigation behaviors. In *Under Review*.

[8] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[9] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[11] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision*, pages 180–196. Springer, 2016.

[12] Hernán Badino, Daniel Huber, and Takeo Kanade. Real-time topometric localization. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[13] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017.

[14] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[15] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[16] Paulo Blikstein and Marcelo Worsley. Multimodal learning analytics and education data mining: using computational technologies to measure complex learning tasks. *Journal of Learning Analytics*, 3(2):220–238, 2016.

[17] A. Boularias, F. Duvallet, J. Oh, and A. Stentz. Learning to ground spatial relations for outdoor robot navigation. In *Proc. of IEEE Conference on Robotics and Automation (ICRA)*, 2015.

[18] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018.

[26] Weifeng Chen, Donglai Xiang, and Jia Deng. Surface normals in the wild. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] Jaedeug Choi and Kee-Eung Kim. Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.

[28] Ronald Clark, John McCormac, Stefan Leutenegger, and Andrew J Davison. Meta-learning for instance-level data association. In *Neural Information Processing Systems (NIPS)*, 2017.

[29] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, May 2018.

[31] Copernicus. Copernicus emergency management service - mapping. https://emergency.copernicus.eu, January 2018.

[32] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] Google Crisis. https://google.org/crisismap/2013-oklahoma-tornado, 2013.

[34] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[36] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[37] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[38] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[39] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[40] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, pages 566–568. IEEE, 1994.

[41] Google Earth. Imagery of mout training facility.

[42] Google Earth. Imagery of pittsburgh, pa, usa.

[43] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[44] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[46] Dave Ferguson and Anthony Stentz. Focussed processing of mdps for path planning. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 310–317. IEEE, 2004.

[47] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58, 2016.

[48] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[50] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. of International Conference on Machine Learning (ICML)*, 2016.

[51] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[52] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Felix Wichmann, Wieland Brendel, and Matthias Bethge. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proc. of International Conference on Learning Representations (ICLR)*, 2019.

[53] Digital Globe. Imagery of fukushima, japan after 2011 tohoku earthquake.

[54] Digital Globe. Imagery of fukushima, japan before 2011 tohoku earthquake.

[55] Digital Globe. Open data program. https://www.digitalglobe.com, January 2018.

[56] Abner Guzman-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew Fitzgibbon, and Shahram Izadi. Multi-output learning for camera relocalization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[57] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[58] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[59] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[60] Lei He, Guanghui Wang, and Zhanyi Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing (TIP)*, 27(9):4676–4689, 2018.

[61] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proc. of International Conference on Machine Learning (ICML)*, 2018.

[62] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[63] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.

[64] Mostafa S Ibrahim, Arash Vahdat, and William G Macready. Weakly supervised semantic image segmentation with self-correcting networks. *arXiv preprint arXiv:1811.07073*, 2018.

[65] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[66] ISPRS. International society of programmetry and remote sensing. http://www.isprs.org.

[67] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3118–3125. IEEE, 2016.

[68] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2017.

[69] H.-G. Jeon, S. Im, B.-U. Lee, D.-G. Choi, **J. Oh**, I. S. Kweon, and M. Hebert. A Large-sacle Virtual Dataset and Egocentric Localization for Disaster Responses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[70] Hae-Gon Jeon, Sunghoon Im, Byeong-Uk Lee, Dong-Geol Choi, Martial Hebert, and In So Kweon. Disc: A large-scale virtual dataset for simulating disaster scenarios. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[71] Hae-Gon Jeon, Sunghoon Im, Jean Oh, and Martial Hebert. Learning shape-based representation for visual localization in extremely changing conditions. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[72] Longlong Jing, Yucheng Chen, and Yingli Tian. Coarse-to-fine semantic segmentation from image-level labels. *arXiv preprint arXiv:1812.10885*, 2018.

[73] Biliana Kaneva, Antonio Torralba, and William T Freeman. Evaluation of image features using a photorealistic virtual world. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[74] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[75] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[76] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[77] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[78] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

[79] Edwin Lane. How technology is changing disaster relief. http://www.bbc.com/news/technology-29149221, 2014. BBC News.

[80] B.-U. Lee, K. Lee, **J. Oh**, and I. Kweon. Cnn-based simultaneous dehazing and depth estimation. In *Proc. of IEEE Conference on Robotics and Automation (ICRA)*, 2020.

[81] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[82] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[83] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 19–27, 2011.

[84] Andreas Ley, Ronny Hänsch, and Olaf Hellwich. Syb3r: A realistic synthetic benchmark for 3d reconstruction from images. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[85] Haoang Li, Yazhou Xing, Ji Zhao, Jean-Charles Bazin, and Yunhui Liu. Leveraging structural regularity of atlanta world for monocular slam. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[86] Kun Li and Joel W Burdick. Inverse reinforcement learning in large state spaces via function approximation. *arXiv preprint arXiv:1707.09394*, 2017.

[87] Ruihao Li, Qiang Liu, Jianjun Gui, Dongbing Gu, and Huosheng Hu. Indoor relocalization in challenging environments with dual-stream convolutional neural networks. *IEEE Transactions on Automation Science and Engineering (TASE)*, 15(2):651–662, 2018.

[88] Zhuwen Li, Ping Tan, Robby T Tan, Danping Zou, Steven Zhiying Zhou, and Loong-Fah Cheong. Simultaneous video defogging and stereo reconstruction. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[89] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. of European Conference on Computer Vision (ECCV)*, 2014.

[90] Guan-Horng Liu, Avinash Siravuru, Sai Prabhakar, Manuela Veloso, and George Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. *arXiv preprint arXiv:1705.10422*, 2017.

[91] Siqi Liu, Sidong Liu, Weidong Cai, Hangyu Che, Sonia Pujol, Ron Kikinis, Dagan Feng, Michael J Fulham, et al. Multimodal neuroimaging feature learning for multiclass diagnosis of alzheimer's disease. *IEEE Transactions on Biomedical Engineering*, 62(4):1132–1140, 2015.

[92] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[93] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.

[94] Dimitrios Marmanis, Jan D Wegner, Silvano Galliani, Konrad Schindler, Mihai Datcu, and Uwe Stilla. Semantic segmentation of aerial images with an ensemble of cnns. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:473, 2016.

[95] Francisco Massa, Bryan C Russell, and Mathieu Aubry. Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[96] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[97] H Brendan McMahan and Geoffrey J Gordon. Fast exact planning in markov decision processes. In *ICAPS*, pages 151–160, 2005.

[98] Sean McMahon, Niko Sünderhauf, Ben Upcroft, and Michael Milford. Multimodal trip hazard affordance detection on construction sites. *IEEE Robotics and Automation Letters*, 3(1):1–8, 2018.

[99] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[100] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[101] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

[102] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2130–2134. IEEE, 2015.

[103] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[104] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2554–2563. JMLR. org, 2017.

[105] Tayyab Naseer, Gabriel L Oliveira, Thomas Brox, and Wolfram Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[106] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

[107] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. of European Conference on Computer Vision (ECCV)*, 2012.

[108] United Nations. Data application of the month: Free satellite data. http://www.un-spider.org, January 2018.

[109] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[110] Xuran Pan, Lianru Gao, Andrea Marinoni, Bing Zhang, Fan Yang, and Paolo Gamba. Semantic labeling of high resolution aerial imagery and lidar data with fine segmentation network. *Remote Sensing*, 10(5):743, 2018.

[111] Clment Petres, Yan Pailhas, Pedro Patron, Yvan Petillot, Jonathan Evans, and David Lane. Path planning for autonomous underwater vehicles. *IEEE Transactions on Robotics (TRO)*, 23(2):331–341, 2007.

[112] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[113] Kun Qian, Wei Zhao, Zhewen Ma, Jiale Ma, Xudong Ma, and Hai Yu. Wearable-assisted localization and inspection guidance system using egocentric stereo cameras. *IEEE Sensors Journal*, 18(2):809–821, 2017.

[114] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *ACM International Conference on Multimedia*, 2017.

[115] Francesco Ragusa, Antonino Furnari, Sebastiano Battiato, Giovanni Signorello, and Giovanni Maria Farinella. Egocentric visitors localization in cultural sites. *Journal on Computing and Cultural Heritage (JOCCH)*, 12(2):1–19, 2019.

[116] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018.

[117] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[118] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006.

[119] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[120] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[121] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proc. of European Conference on Computer Vision (ECCV)*, 2016.

[122] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[123] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[124] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[125] Pierre Sermanet, Raia Hadsell, Marco Scoffier, Matt Grimes, Jan Ben, Ayse Erkan, Chris Crudele, Urs Miller, and Yann LeCun. A multirange architecture for collision-free off-road robot navigation. *Journal of Field Robotics*, 26(1):52–87, 2009.

[126] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017.

[127] Fei Shao, Songmei Cai, and Junzhong Gu. A modified hausdorff distance based algorithm for 2-dimensional spatial trajectory matching. In *Computer Science and Education (ICCSE), 2010 5th International Conference on*, pages 166–172. IEEE, 2010.

[128] Jun Shi, Xiao Zheng, Yan Li, Qi Zhang, and Shihui Ying. Multimodal neuroimaging feature learning with multimodal stacked deep polynomial networks for diagnosis of alzheimer's disease. *IEEE journal of biomedical and health informatics*, 22(1):173–183, 2018.

[129] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[130] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[131] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[132] Emiliano Spera, Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Egocentric shopping cart localization. In *Proc. of International Conference on Pattern Recognition (ICPR)*, 2018.

[133] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[134] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[135] Geoffrey R Taylor, Andrew J Chosak, and Paul C Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[136] **J. Oh**, M. Hebert, H.-G. Jeon, C. Dai, and Y. Song. Explainable Semantic Mapping for First Responders. In *Artificial Intelligence for Humanitarian Assistance and Disaster Response Workshop at the Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[137] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[138] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[139] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[140] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[141] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[142] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[143] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[144] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[145] Shenlong Wang, Min Bai, Gellert Mattyus, Hang Chu, Wenjie Luo, Bin Yang, Justin Liang, Joel Cheverie, Sanja Fidler, and Raquel Urtasun. Torontocity: Seeing the world with a million eyes. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[146] Olivia Ward. How a canadian team's drones are changing search and rescue. https://www.thestar.com, May 2016.

[147] Jan D Wegner, Steven Branson, David Hall, Konrad Schindler, and Pietro Perona. Cataloging public objects using aerial and street-level images-urban trees. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[148] Maggie Wigness, John G Rogers III, and Luis E Navarro-Serment. Robot navigation from human demonstration: Learning control behaviors. *IEEE ICRA 2018*.

[149] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Deep inverse reinforcement learning. *CoRR, abs/1507.04888*, 2015.

[150] Markus Wulfmeier, Dominic Zeng Wang, and Ingmar Posner. Watch this: Scalable cost-function learning for path planning in urban environments. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2089–2095. IEEE, 2016.

[151] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[152] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[153] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[154] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proc. of European Conference on Computer Vision (ECCV)*, 2018.

[155] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[156] Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[157] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. of International Conference on Learning Representations (ICLR)*, 2016.

[158] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17(1):1–32, 2016.

[159] Wei Zhang, Youmei Zhang, Lin Ma, Jingwei Guan, and Shijie Gong. Multimodal learning for facial expression recognition. *Pattern Recognition*, 48(10):3191–3202, 2015.

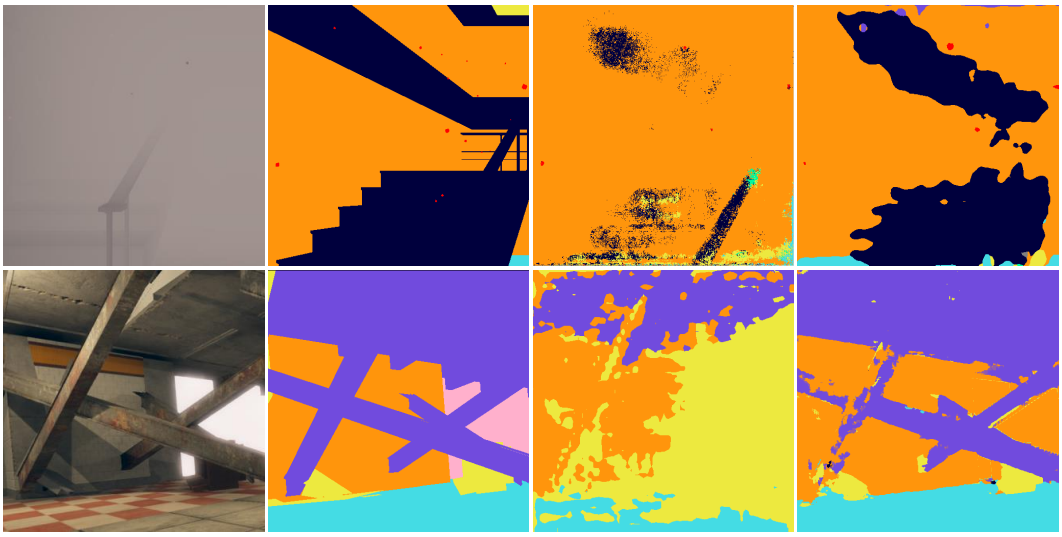
[160] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *arXiv preprint arXiv:1810.09091*, 2018.

[161] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

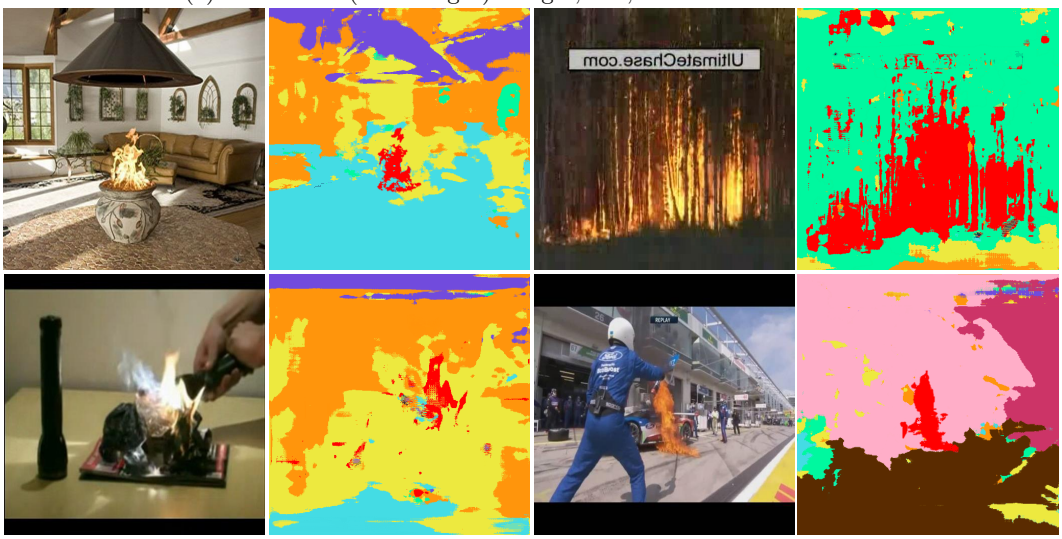
[162] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[163] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[164] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.



(a) The DISC. (left to right) Images, GT, without and with FT.



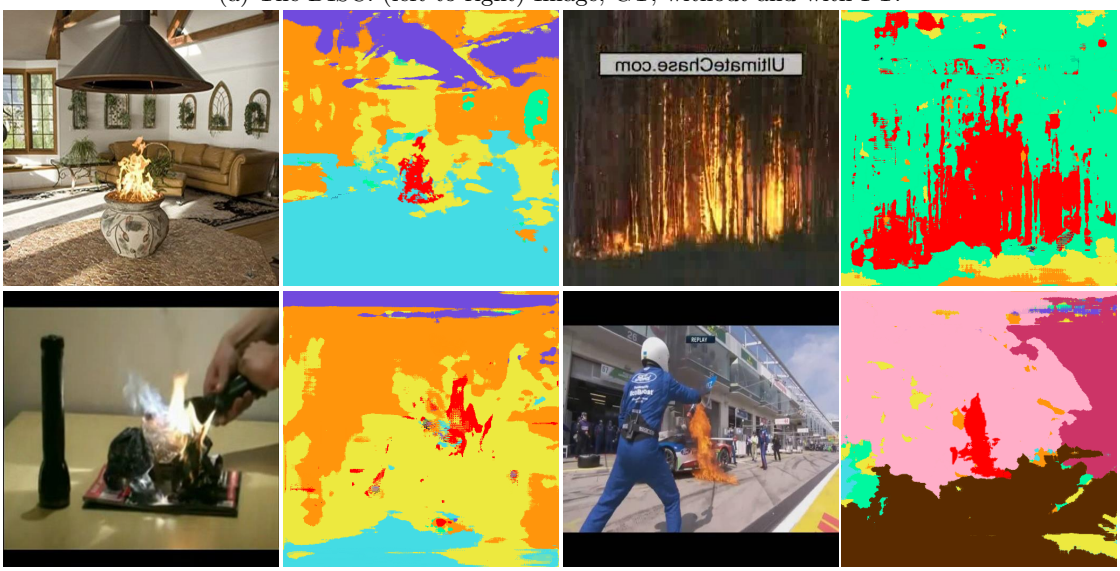
(b) Real-world results

Figure 26: Examples of semantic segmentation benchmark.

(a) DISC image (Source) (b) Real image (Target) (c) DISC-to-Real (Source-to-Target)

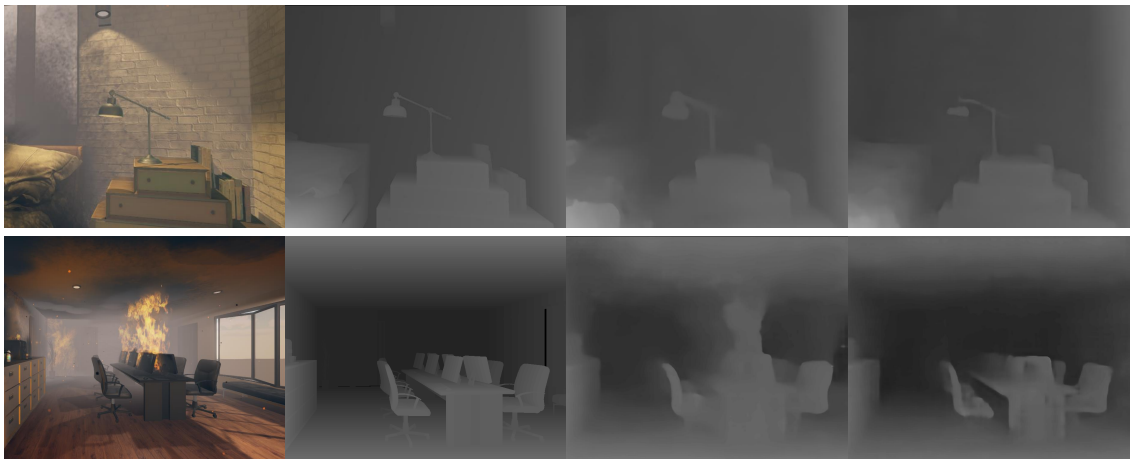
Figure 27: An example of unsupervised (pixel-level) domain adaptation using [81] from DISC to a real image domain.

(a) The DISC. (left to right) Image, GT, without and with FT.



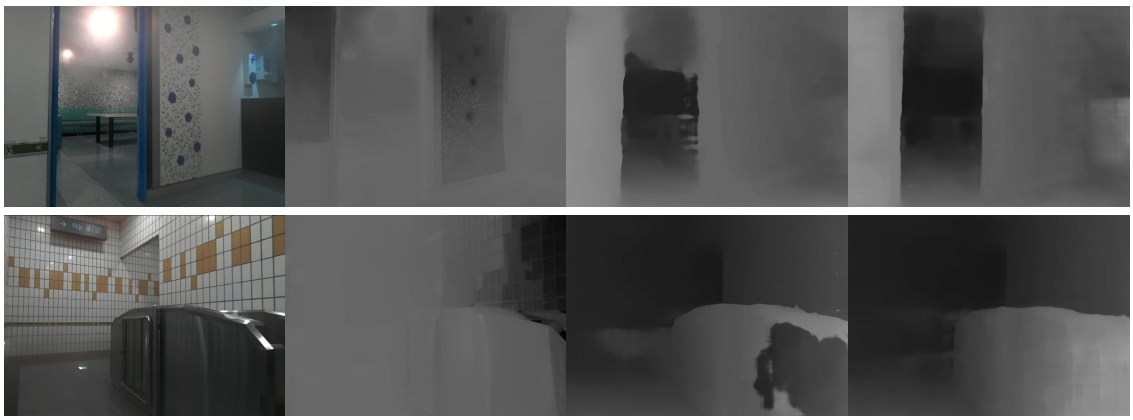
(b) Real-world results. (left to right) Images, without and with FT.

Figure 28: Surface normal estimation. Baseline: FCN-Skip



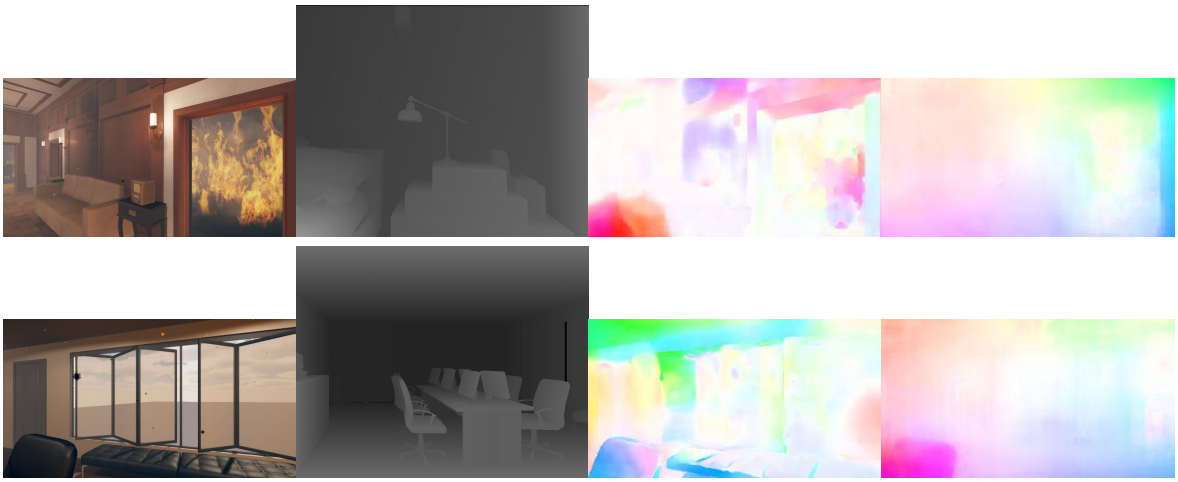
(a) Input (b) GT (e) w/o FT (f) w/ FT

Figure 29: Stereo matching results on DISC. Baseline: PSMNet.



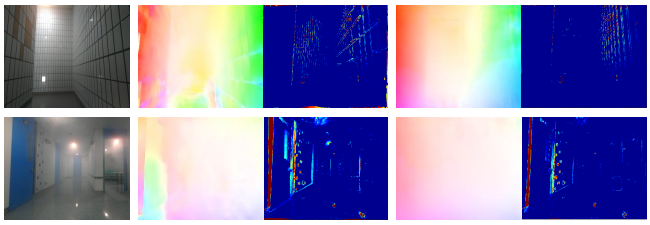
(a) Input (b) [88] (c) w/o FT (d) w/ FT

Figure 30: Stereo matching results on real-world scenes.



(a) Input (b) GT (c) w/o FT (d) w/ FT

Figure 31: Optical flow results on DISC. Baseline: PWCNet

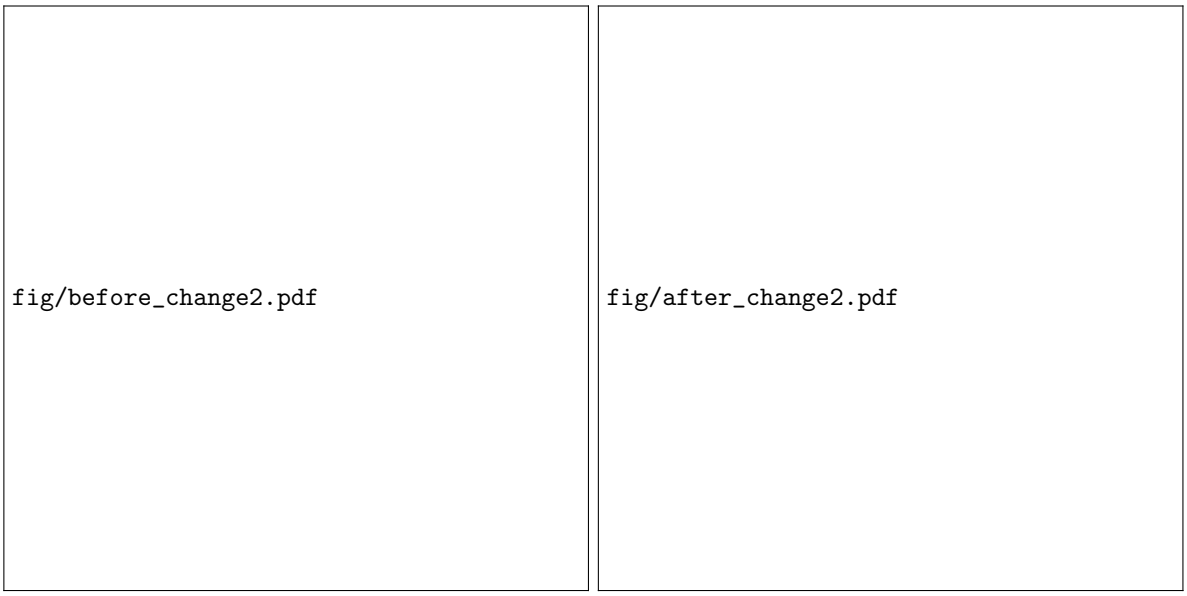


(a) Input (b) w/o FT (c) w/ FT

Figure 32: Results from PWCNet and interpolation errors between the reference images and warped images on real world scenes.

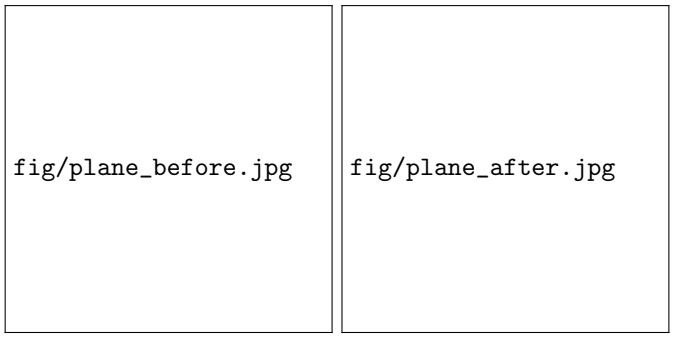
fig/Network.pdf

Figure 33: An overview of the proposed visual localization network. All convolution and deconvolution layers have ReLU except for the prediction layers. The numbers in each block represent a filter size, a channel size, and stride.



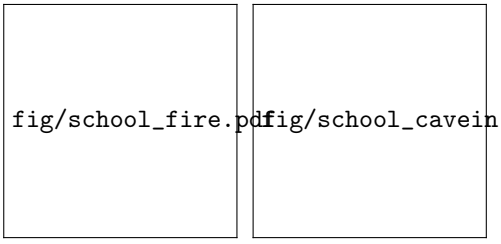
(a) (b)

Figure 34: An example of semantic label changes caused by geometry changes.



(a) (b)

Figure 35: (a) Before geometry changes. (b) After geometry changes with plane estimation results. Each estimated plane is colorized for visualization.



(a) Building on fire (b) Destruction

Figure 36: Results from our network on School scene in 4.2. In the maps, red and blue dots represent ground-truth positions and the estimated positions, respectively. The red arrows mean positions of the database, query and style transferred images.

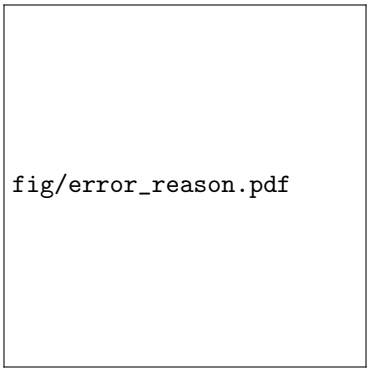
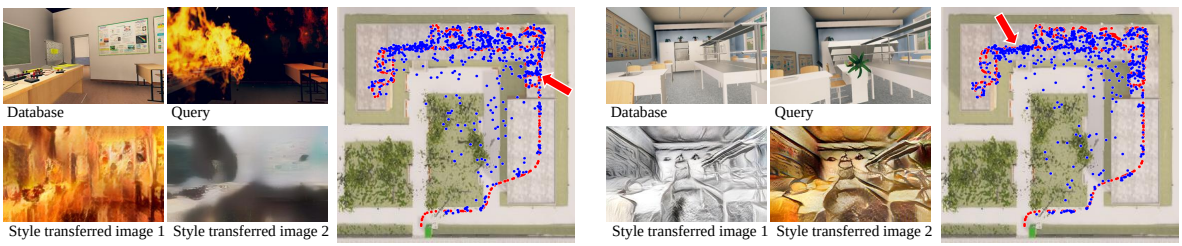


Figure 37: Failure cases of our network.

Figure 38: Performance changes according to the seed image selections. (a) Examples of DISC dataset for simulating before and after building on fire. (b) Seed images with its corresponding contextual loss. The images are categorized based on averaged contextual loss between each seed image and all query images. (c) The x-axis represents combinations of seed images in (b) and the y-axis are units for translation (m) and rotation error (degree), respectively. The horizontal bars represent the errors when the seed images are not used.



(a) Building on fire (b) Destruction

Figure 39: Results from our network on School scene. In the maps, red and blue dots represent ground-truth positions and the estimated positions, respectively. The red arrows mean positions of the database, query and style transferred images.