Deep Learning Application to 3D Vision: Volumetric Medical Image Analysis and Camera Pose Estimation

Ziming Qiu

December 21, 2021

Deep Learning Application to 3D Vision: Volumetric Medical Image Analysis and Camera Pose Estimation

DOCTORAL DISSERTATION

Submitted in Partial Fulfillment

of the Requirements for the

Degree of

DOCTOR OF PHILOSOPHY (Electrical Engineering)

at the

NEW YORK UNIVERSITY

TANDON SCHOOL OF ENGINEERING

by

Ziming Qiu

January 2022

Approved:

Ivan Selesnick

Department Head

Dec 21 2021

Copy No. ____________________

Approved by the Guidance Committee :

Major : Electrical Engineering

Wang Yao

Yao Wang (Advisor)

Professor of
Electrical Engineering

ACh

Anna Choromanska

Assistant Professor of
Electrical Engineering

冯晨

Chen Feng

Assistant Professor of
Mechanical Engineering

Vita

Ziming Qiu was born in Qingyuan, Guangdong, China in 1995. He received his B.S. degree in Biomedical Engineering from Beihang University, Beijing, China in 2017. Then, he entered Tandon School of Engineering of New York University to pursue a doctoral degree in Electrical Engineering in Fall 2017. His PhD research focuses are computer vision, deep learning and medical image analysis. During his PhD years, he also interned at Siemens, Nokia Bell Labs, and Facebook on machine learning related projects.

Stay simple, stay patient, stay happy and be grateful!

Acknowledgements

I am very lucky and grateful to receive a lot of help during my PhD study from different people, without whom I could not achieve what has been done in this thesis.

First of all, I would like to thank my advisor, Prof. Yao Wang, for her invaluable guidance, support and help throughout the course of my PhD degree. If someone asks me: ”Do you recommend Prof. Yao Wang as my PhD advisor”, my answer would be a definite yes without second thought.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Anna Choromanska and Prof. Chen Feng for their insightful comments and suggestions.

Furthermore, I would also like to thank my research collaborators: Jen-wei Kuo, Jeffrey A. Ketterling, Orlando Aristizabal, Daniel H. Turnbull, Jonathan Mamou, Nitin Nair, Jack Langerman, Tongda Xu, Howard Huang and Gabor Soros.

Last but not least, I would like to express my sincere gratitude to my parents for their support throughout my life.

Ziming Qiu, New York University, Tandon School of Engineering
December 21, 2021

ABSTRACT

Deep Learning Application to 3D Vision: Volumetric Medical Image Analysis and Camera Pose Estimation

by

Ziming Qiu

Advisor: Yao Wang

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy (Electrical Engineering)

January 2022

With the availability of powerful modern computation resources and large scale labeled data, deep learning has shown enormous success in various computer vision tasks, including medical and natural image analysis. In this thesis, deep learning methods are specifically applied to two 3D vision tasks: volumetric medical image analysis and camera pose estimation.

For volumetric medical image analysis, segmentation and mutant classification of high-frequency ultrasound (HFU) mouse embryo images can provide valuable information for developmental biologists. However, manual segmentation and identification of brain ventricle (BV) and body requires substantial time and expertise. This thesis proposes an accurate, efficient and explainable deep learning pipeline for automatic segmentation and classification of the BV and body. For segmentation, a two-stage framework is implemented. The first stage produces a low-resolution segmentation map, which is then used to crop a region of interest (ROI) around the target object. The second stage fine-resolution refinement network acts on the ROI of each object and uses the segmentation probability map generated by the first stage as its auto-context. The proposed segmentation method significantly reduces inference time while maintaining high accuracy comparable to previous sliding-window approaches. Based on the BV and body segmentation map, a volumetric convolutional neural network (CNN) is trained to perform a mutant classification task. Through backpropagating the gradients of the prediction to the input BV and body segmentation maps, the trained classifier is found to largely focus on the region where the *Engrailed-1 (En1)* mutation phenotype is known to manifest itself. This suggests that gradient backpropagation of deep learning classifiers may provide a powerful tool for automatically detecting unknown phenotypes associated with a genetic mutation.

One of the key criticisms of deep learning is that large amounts of expensive and difficult-to-acquire training data are required in order to train models with high performance and good generalization capabilities. Focusing on the task of monocular camera pose estimation via scene coordinate regression (SCR), we describe a novel method, Domain Adaptation of Networks for Camera pose Estimation (DANCE), which enables the training of models without access to any labels on the target task. DANCE requires unlabeled images (without known poses, ordering, or scene coordinate labels) and a 3D representation of the space (e.g., a scanned point cloud), both of which can be captured with minimal effort using off-the-shelf commodity hardware. DANCE renders labeled synthetic images from the 3D model, and bridges the inevitable domain gap between synthetic and real images by applying unsupervised image-level domain adaptation techniques (unpaired image-to-image translation). When tested on real images, the SCR model trained with DANCE achieved comparable performance to its fully supervised counterpart (in both cases using PnP-RANSAC for final pose estimation) at a fraction of the cost.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

With the availability of powerful modern computation resources and large scale labeled data, deep learning has shown enormous success in a wide variety of computer vision tasks, including medical and natural image analysis. As the actual world is essentially 3D, a vast majority of 3D vision tasks have been attracting increasing attention from the deep learning community. In this thesis, deep learning methods are specifically applied to two 3D vision tasks: volumetric medical image analysis and camera pose estimation. For the task of volumetric medical image analysis, we focus on developing a deep learning approach for the segmentation, classification, and visualization of 3D high-frequency ultrasound (HFU) images of mouse embryos. For the task of camera pose estimation, we focus on estimating the camera pose without pose labels by using labeled synthetic data and domain adaptation techniques. It is worth noting that not only these two applications belong to the field of 3D vision, but also their key common component in the pipeline is an image-to-image translation module (i.e. ultrasound images to segmentation label images and camera images to scene coordinate images).

1.2 Problem Statement

In this section, we will define the problems of volumetric medical image analysis and camera pose estimation separately.

- The volumetric medical image analysis project is aiming to phenotype early- to mid-gestational mouse embryos by segmenting brain ventricle (BV) and body in 3D images acquired in-utero with HFU. Around 20,000 NIH Knockout (KO) mouse strains will be generated, 25% of which are expected to be embryonic or perinatal lethal, including many important models of human structural birth defects and congenital diseases. The development of phenotyping methods for embryonic lethal mice that enable efficient analysis of defects in embryonic growth in the KO mouse strains is highly demanded. In-utero 3D HFU image acquisition protocols and image processing methods that permit real-time, noninvasive and longitudinal studies of the embryonic development has been developed and validated. Volumetric HFU data will be collected in-utero from mouse embryos staged between E9.5 to 15.5 in order to establish a database of normal development. With provided HFU images of mouse embryos, the research focus of this thesis is developing advanced image analysis and deep learning methods for analyzing brain development in mouse embryos and characterizing defects caused by mutations.

- Estimating the 3D position and 3D orientation (6 degrees of freedom pose) of an agent or an object with respect to a reference coordinate frame is a fundamental requirement in robotic applications, such as robot navigation. One of the existing and accurate deep learning camera pose estimation pipeline uses a deep neural network to predict the scene coordinates followed by PnP-RANSAC. The problem for this pipeline is that the training of the deep neural network requires large amounts of labeled data, which is expensive and difficult to acquire. To ease the burden of collecting training data, a simple and effective data collection pipeline will be proposed and demonstrated in this thesis.

1.3 Contributions

Because deep learning methods are specifically applied to two 3D vision tasks: volumetric medical image analysis and camera pose estimation, it will be better to discuss the contributions of each project separately.

- For volumetric medical image analysis, an accurate, efficient and explainable deep-learning-based pipeline for the segmentation and mutant classification of the brain ventricle and body from high-frequency ultrasound mouse embryo images has been developed. Segmentation and mutant classification of HFU mouse embryo images can provide valuable information for developmental biologists. However, manual segmentation and identification of the images requires substantial time and expertise.

  For segmentation, a two-stage framework is implemented. The first stage produces a low-resolution segmentation map, which is then used to crop a region of interest (ROI) around the target object. The second stage fine-resolution refinement network acts on the ROI of each object and uses the segmentation probability map generated by the first stage as its auto-context. The proposed segmentation method significantly reduces inference time while maintaining high accuracy comparable to previous sliding-window approaches.

  For mutant classification and visualization, a volumetric convolutional neural network (CNN) is trained to perform a mutant classification task based on the BV and body segmentation map. Through backpropagating the gradients of the prediction to the input BV and body segmentation maps, the trained classifier is found to largely focus on the region where the Engrailed-1 (En1) mutation phenotype is known to manifest itself.

In summary, the proposed pipeline has the potential to uncover unknown phenotypes manifested as shape changes associated with different gene mutations. Moreover, our segmentation, mutant classification and visualization algorithms may be applicable and invaluable in streamlining developmental biology studies.

- For camera pose estimation via scene coordinate regression (SCR), a novel pipeline, Domain Adaptation of Networks for Camera pose Estimation (DANCE), is developed in order to ease the burden of collecting labeled data to train the deep SCR network. Though without access to any labeled camera images, the SCR network trained with DANCE achieved comparable performance to its fully supervised counterpart. DANCE requires unlabeled images (without known poses, ordering, or scene coordinate labels) and a 3D representation of the space (e.g., a scanned point cloud), both of which can be captured with minimal effort using off-the-shelf commodity hardware. DANCE renders labeled synthetic images from the 3D model, and bridges the inevitable domain gap between synthetic and real images by applying unsupervised image-level domain adaptation techniques (unpaired image-to-image translation). One of the key criticisms of deep learning is that large amounts of expensive and difficult-to-acquire training data are required in order to train models with high performance and good generalization capabilities. Our proposed pipeline demonstrates a possible solution: the deep neural networks could be trained at a lower cost with synthetic labeled data and a pool of unlabeled samples if the generation of synthetic labeled data is easier than the direct gathering of labeled data.

1.4 Organization of the Thesis

This dissertation is organized as following: Chapter 1 has introduced both 3D vision tasks; Chapter 2 will discuss the 3D vision task 1: volumetric medical image analysis; Chapter 3 will discuss the 3D vision task 2: camera pose estimation. Finally, Chapter 4 will summarize both 3D vision application works.

Chapter 2

Volumetric Medical Image Analysis

2.1 Introduction

The mouse is a commonly used animal model in the study of mammalian embryo development due to its high degree of homology with the human genome. Along with complete knowledge of the mouse genome, a wide variety of gene editing tools have enabled the creation of genetic modifications in mice, including many mutations that are lethal *in utero* [14]. For instance, *En1* homozygous mutants exhibit early embryonic deletion of the mid-hindbrain region in the developing central nervous system that leads to a thickening of the BV and subsequent death at birth [86]. Observing variations in the shape of the BV and body is an effective way to study how genetic defects, such as the *En1* mutation, are manifested during embryonic development [36, 44].

High-throughput HFU has proven to be an effective imaging modality to generate high-resolution volumetric datasets of mouse embryos *in utero* over mid-to-late gestational stages [3]. Accurate delineation of anatomical structures from HFU images can provide valuable structural information and enable downstream analysis of complex biomedical image data [46]. As such, accurate and time-efficient BV and body segmentation in HFU data can substantially aid biologists in observing and understanding the development of mouse embryos.

Manual segmentation by imaging experts has long been considered the gold standard in the field of biomedical image analysis. However, manual segmentation of the BV and body (Fig. 2.1) from 3D HFU volumes is time-consuming, requiring half an hour or more for each volume, which increases considerably with image quality decay. Additionally, the large and ever increasing quantity of HFU images typically used in developmental studies makes manual labelling impractical in the long run. Therefore, it is necessary to develop fully automatic segmentation and classification algorithms to optimize this process [24]. Such an algorithm must overcome five primary challenges related to the image data: (1) extreme imbalance between background and foreground (i.e., the BV makes up only 0.367% of the whole volume, on average, while the body is around 10.6%); (2) differing shapes

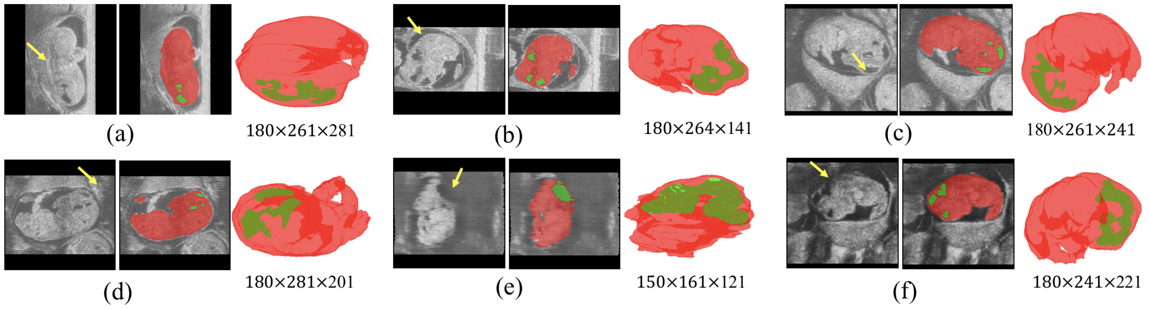

Figure 2.1: (a-f) 6 embryonic mice HFU volumes are shown with three views each: a B-mode image slice from the 3D volume, a manual BV (green) and body (red) segmentation, and a 3D rendering (visualized in natural orientation relative to the HFU probe). The numbers below each 3D rendering indicate corresponding image size in voxels. The arrow in a) indicates an ambiguous boundary due to contact between the body and uterine wall. The arrows in b), c) and d) indicate motion artifacts because of irregular physiological movements of the anesthetized pregnant mice. The arrows in e) and f) indicate missing head boundaries due to either specular reflections or shadowing from overlaying tissues.

and locations of the body and BV due to various embryonic stages; (3) large variation in embryo posture and orientation; (4) the presence of missing or ambiguous boundaries (Fig. 2.1(a)(e)(f)) and motion artifacts (Fig. 2.1(b)(c)(d)); and (5) large variation in image size, from $150 \times 161 \times 81$ to $210 \times 281 \times 282$ voxels.

A nested graph cut (NGC) algorithm [34] was first developed to perform segmentation of the BV from the manually selected head portion of HFU of the mouse embryo. NGC relied on the nested structure of the BV, head, uterus and surrounding amniotic fluid and successfully overcame the missing head boundary problem (Fig. 2.1(e)(f), Challenge 4). This problem is caused by a loss of HFU signal due to either specular reflections or shadowing from overlaying tissues. Subsequent work focused on BV and body segmentation in whole-body images by extending the NGC algorithm to first detect and segment the interior of the uterus and then to detect and segment the BV and body regions [35]. Although this framework performed well on an initial set of 36 embryos [35], it did not generalize well to larger, unseen data sets because the framework was developed based on manually crafted assumptions and several parameters were hand-tuned on the smaller data set.

Given the success of Fully Convolutional Networks (FCN) for semantic segmentation tasks [42], we developed a deep-learning-based framework for BV segmentation [54] that outperformed the NGC-based framework in [35] by a large margin. Because the BV makes up a very small portion (<0.5%) of the whole volume, the algorithm in [54] first applied a volumetric CNN on a 3D sliding window over the entire volume to identify a 3D bounding box containing the whole BV, followed by a FCN to segment the detected bounding box into BV or background. Despite achieving high accuracy of 0.904 Dice Similarity Coefficient (DSC) for BV, this method was inefficient because it required hundreds of thousands of forward passes through a classification network in the first sliding-window-based localization step. The challenges for body segmentation are similar to those for BV segmentation, except that the extreme imbalance between foreground and background is somewhat alleviated (the body makes up 10.6% of the whole volume on average). Hence, the localization step is not necessary for body segmentation. Qiu et al. [55] first applied an FCN to segment each sliding window over the entire volume, and then determined the final body segmentation by merging results from all the sliding windows. However, this sliding-window-segmentation approach suffered from the same inefficiencies as the localization method in [54].

Here, we propose an efficient end-to-end auto-context refinement framework for joint BV and body segmentation from volumetric HFU images. The proposed approach is to: (1) generate a ROI from the original image through one-pass low-resolution segmentation and cropping in order to circumvent the class imbalance problem without the use of a sliding window; and (2) combine the low-resolution map with a cropped, fine-resolution image as an auto-context [76] input so that the fine-resolution segmentation network can utilize valuable global information and produce more accurate results. Specifically, a VNet (VNet I) [48] is first applied to a downsampled HFU 3D image to jointly segment the BV and body (Fig. 2.2). The resulting low-resolution body segmentation map is then up-sampled to the original resolution, and a bounding box containing the body is generated. Next, the original image and the coarse probability map for the up-sampled body in the bounding box are concatenated as localized auto-context and fed into another VNet (VNet II) to generate the final refined body segmentation map. A parallel process is applied to generate final refined BV segmentation using a third VNet (VNet III). Each VNet is initially trained separately and then fine-tuned in an end-to-end manner.

Compared with previous methods, this segmentation framework has the following advantages:

1. The class imbalance problem is mitigated by cascading the networks from low resolution of the whole image to fine resolution in localized regions without the need for a time-consuming sliding window.

2. An auto-context input is created by concatenating the initial blurred low-resolution segmentation map with the high-resolution image (Fig. 2.3). This auto-context input improves segmentation accuracy by providing a full-resolution refinement network with rich global context information.

3. The gradient of the refinement networks can flow end-to-end back to the low-resolution segmentation network by combining localization and auto-context modules in a differentiable pipeline which further improves segmentation accuracy.

Our proposed segmentation framework allows end-to-end training and efficient, real-time, one-pass inference while achieving comparable segmentation accuracy with the substantially more time consuming sliding-window-based approaches.

The morphology of the BV of a *En1* mutant mouse embryos is notably deformed compared to the wildtype phenotype. Moreover, a difference in spine curvature has been reported to exist between *En1* mutants and normal mouse embryos [86]. Because visually identifying mutants is time consuming, an automatic classification model for *En1* mutants using the BV and body segmentation maps would be advantageous. We therefore extended our BV and body segmentation work to include mutant classification using a volumetric VGG-based CNN approach [69] . The purpose of this work is not simply to classify embryos, but to better understand the underlying morphological changes associated with a mutant phenotype. Therefore, to understand the underlying physiological structures that influence the classification process, the method introduced in [68] is used to visualize the trained network by backpropagating the gradient of the prediction with respect to the input BV and body segmentation map.

In this thesis, we first present a real-time and accurate BV and body segmentation algorithm, which is built on our previous efforts [87] with a more thorough exposition of the methods, a more expansive discussion of the results, and a comparison with the prior NGC-based method [35] for the same data set. Then we use the BV and body segmentation to perform mutant classification together with a simple method for automatically rotating the segmentation maps so that the body and BV shapes will all follow the same canonical orientation. A standard 3D image orientation not only helps improve classification results, but also assists better visualization of the 3D volumes. Finally, we leverage gradient-backpropagation-based visualization of the data to understand what features the learnt classifier uses to make its decision. It is worth noting that preliminary mutant classification results based on the BV segmentation only were reported in [55].

2.2 Related Work

2.2.1 Segmentation

Segmentation is a critical component of any pipeline designed to aid in image-based analyses of mouse mutants. Registration-based analysis of magnetic resonance images has been used extensively for studying postnatal brain phenotypes [15, 51] and brain development [72]. This approach makes use of atlases of normal mouse brain anatomy that were derived from image registration and averaging in combination with manual segmentation by experts. Then, individual mouse brains of unknown phenotype are segmented automatically via registration to the atlas. Current pipelines designed for detecting and analyzing mutant mouse embryos have taken a similar approach [14] using *ex vivo* micro-CT [84] or optical projection tomography (OPT) [85] images. In contrast, HFU is uniquely suited to providing *in vivo* data on mouse embryonic development [3], but HFU embryo atlases have not been established. In the current study, we investigate deep learning approaches as an alternative to the more conventional, registration-based segmentation methods.

Deep learning has been widely employed in biomedical image analysis tasks [4, 37, 40, 64, 88, 93]. Milletari et al. [47] applied deep CNNs to localize and segment the midbrain in MRI and ultrasound images in a patch-wise manner with Hough voting. Although this method attempted to implicitly incorporate a shape prior through Hough voting, the patch-wise training strategy ignored the interdependent relationships between neighboring patches during training of the CNN classifiers. Long et al. [42] developed the influential FCN by replacing all the fully connected layers of traditional CNN-based classifiers with a transpose convolutional layer and then Ronneberger et al. [57] improved the FCN model by introducing symmetric skip connections between the encoder and decoder, leading to the widely known UNet model. Liu et al. [41] proposed to use 2D FCN with feature pyramid attention for automatic prostate zonal segmentation in 3D MRI images. Although this 2D-based FCN was shown to outperform UNet, it was still deficient in capturing inter-slice correlation information compared to 3D-based models. Milletari et al. [48] further adapted UNet to VNet for volumetric medical image segmentation and also introduced a Dice-based loss function.

Tu [76] first proposed the auto-context concept for high-level vision tasks, such as image segmentation. Specifically, the idea behind auto-context [76] is to iterate in order to approach the reference segmentation through a sequence of models, where the input and output of the previous model are concatenated to form the input for the next model such that the next model can make use of richer context information from the output of the previous model. It is possible to cascade two or more segmentation networks for the purpose of either localization or auto-context. Roth et al. [58] focused on abdominal CT image segmentation. They applied two cascaded 3D FCNs using the initial segmentation results to localize the foreground organs, which were then input into the second FCN. The initial segmentation was used only for localization and was not concatenated with the raw image as auto-context input to the second FCN. Tang et al. [74] cascaded four UNets and trained them in an end-to-end manner for skin lesion segmentation. However, this framework did not use the segmentation output of a previous UNet to reduce the spatial region to the next UNet, and was restricted to 2D binary segmentation. Chen et al. [11] cascaded two residual FCNs for volumetric MRI brain segmentation in order to use the first FCN's output as context information for the second FCN. This framework also did not use the initial segmentation for localization of desired structures to reduce spatial input into the second network. In contrast to these efforts, the cascading networks we propose not only serve to localize the ROI, but also function as an auto-context module for multi-class volumetric image segmentation. The Mask-RCNN [21] has been well-known for accurate object detection and instance segmentation. It is less appropriate in our application, because each 3D HFU image has only a single embryo (as opposed to imaging multiple embryos at once) and Mask-RCNN was designed to detect and segment multiple objects in an image.

2.2.2 Classification and Visualization

Increasingly powerful neural network architectures (e.g. AlexNet [33], VGGNet [69], ResNet [23], DenseNet [27] and SENet [26]) have led to successful breakthroughs in a variety of classification tasks. For example, Wang et al. [82] demonstrated that a VGG16 model can outperform a radiomics-based method for thyroid nodules classification in ultrasound images. Moreover, numerous works have focused on interpreting the decision making process of these neural networks. Simonyan et al. [68] proposed visualizing the image-specific class saliency map by backpropagating the gradient from the top-1 class prediction unit to the input image. Springenberg et al. [70] adapted [68] to only backpropagate the positive gradient to reduce noise in the saliency map. Zhou et al. [91] proposed using global average pooling to replace fully connected layers such that the predicted class score can be mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM can highlight the class-specific discriminative regions. Selvaraju et al. [63] generalized CAM to any CNN-based architecture by using gradient backpropogation and also combined guided backpropagation [70] for better visualization.

In biomedical applications, it is essential that classification results are accurate and interpretable. Wang et al. [81] embedded Grad-CAM [63] as an attention branch into a classification network (ResNet-152 model) for 14 thorax diseases diagnosed in chest X-ray images. The embedded Grad-CAM branch enabled the learned feature maps from the classification branch to be converted into an attention map that highlighted the locations of disease-specific regions under the supervision of image-level class labels.

Given a small data set, we adopted a shallow volumetric VGG-like network with 9 layers to perform the mutant classification. We developed an automatic procedure to rotate the BV and body segmentation maps to a canonical orientation. This greatly reduced the orientation variance among the training samples and between the training and testing samples, leading to significant improvement in the classification accuracy. Then, the technique in [68] was utilized to visualize the saliency map by backpropagating the gradient from the top-1 class prediction unit to the input. The pipeline was simple and fully automatic and proved to be highly effective for our classification and visualization tasks.

2.3 Data Set

The data set used for developing our segmentation framework consists of 231 embryonic mouse HFU volumes which were acquired *in utero* and *in vivo* from pregnant mice (10-14.5 days after mating) using a 5-element 40-MHz annular array [3, 31]. The dimensions of the HFU volumes varied from $150 \times 161 \times 81$ to $210 \times 281 \times 282$ voxels and the size of each voxel is $50 \times 50 \times 50$ μm. For each of the 231 volumes, manual BV and body segmentations were conducted by trained research assistants using commercial software (Amira, FEI, Hillsboro, Oregon, USA). It is worth noting that the BV could be segmented accurately because BV is a relatively small and dark region in the brain and can be segmented using the region growing function in Amira. Then, our trained research assistants would refine the BV boundaries. For BV, the labeling process took around 10 minutes for each image volume. Because the body is much larger than the BV and has more variations, the manual body segmentation was achieved by labeling every ≈ 10 2D slices in a volume and then using the label interpolation function in Amira to complete the 3D segmentation. Then, the interpolated slices were examined and slices with large errors were corrected. The number of 2D slices manually segmented in each 3D volume varied depending on the image quality and the number of slices of the image. The body labeling process took around 30 minutes for each 3D image. Any interpolation artifacts from this approach had minimal impact on the analyses. The data set containing 231 images was randomly split into 139 cases for training, 46 cases for validation and 46 cases for testing. The validation set was used to determine the stopping criterion during training. The data used in this study will be provided upon request.

Among the 231 data sets with manual BV and body segmentation, there were only 35 mutant images, which was not sufficient to train and test a mutant classification algorithm. In addition, we also had 336 HFU embryo data sets without manual segmentation, but have ground truth about the presence of the mutation. Hence, the developed segmentation algorithm was applied to these 336 unlabeled HFU images. After auto segmentation of the unlabeled data sets, we manually reviewed the results and selected 321 sets with visually satisfactory segmentation. This process resulted in a total of 552 (231 manual + 321 automatic) data sets with BV and body segmentation containing 102 mutant and 440 normal. Because 102 mutant images were a small data set, six-fold cross validation was employed to develop and evaluate the mutant classification algorithm.

2.4 Methods

2.4.1 Segmentation Framework

An overview of our proposed end-to-end BV and body segmentation framework is shown in Fig. 2.2. The pipeline consists of an coarse segmentation stage and a segmentation refinement stage. The initial coarse segmentation produces low-resolution segmentation maps for BV and body, simultaneously. Next, the original data and the low-resolution label for each object are passed to a Loc-Con module (Fig. 2.3), which first generates a bounding box for the object using the centroid of the up-sampled predicted probability map. Then, for the corresponding object, the Loc-Con module concatenates the full-resolution original image and the initial up-sampled predicted probability map in the bounding box as auto-context input for the refinement network. Next, the refinement network generates a full resolution segmentation map within the bounding box. This segmentation is then projected back to the entire image volume through zero padding. We first train the initial coarse

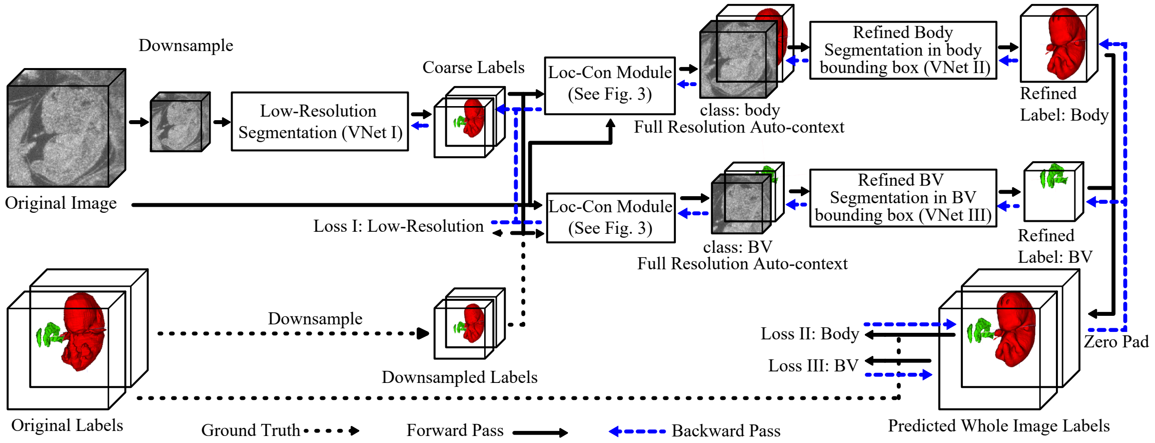
Downsample
Coarse Labels
Low-Resolution Segmentation (VNet I)
Loc-Con Module (See Fig. 3)
class: body
Full Resolution Auto-context
Refined Body Segmentation in body bounding box (VNet II)
Refined Label: Body
Original Image
Loss I: Low-Resolution
Loc-Con Module (See Fig. 3)
class: BV
Full Resolution Auto-context
Refined BV Segmentation in BV bounding box (VNet III)
Refined Label: BV
Downsample
Downsampled Labels
Loss II: Body
Loss III: BV
Zero Pad
Original Labels
Predicted Whole Image Labels
Ground Truth
Forward Pass
Backward Pass

Figure 2.2: The pipeline of the joint BV and body segmentation from 3D HFU images of mouse embryos.

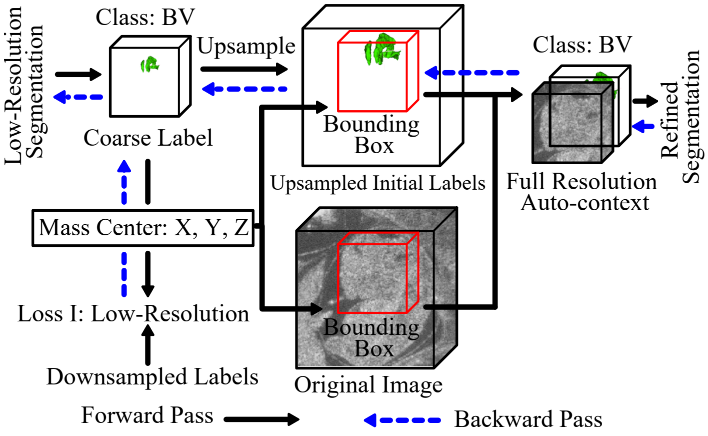

Figure 2.3: Diagram of localization-auto-context (Loc-Con) module for the BV. A similar configuration is used for the body. The gradient produced by the refinement loss can flow back to the low-resolution segmentation network (blue arrows).

segmentation network and then separately train the BV and body refinement networks using output from the trained initial segmentation network. Finally, we fine-tune all three networks (VNet I, VNet II, VNet III) in an end-to-end manner. These three networks follow the exact VNet structure [48] and only the first and last layers are changed based on how many channels the input has and how many classes are predicted.

Initial Segmentation on Low-Resolution Volumes

Because of memory constraints and large variations in image sizes, all the images were padded and downsampled (3rd order spline interpolation) to a low resolution volume of 160^3 voxels. A VNet [48] (VNet I) was trained to perform BV and body segmentation, simultaneously, at low resolution. The output of the VNet had 3 channels, representing the background, BV, and body. The Dice loss [48] for each class was summed and used as the training loss (loss I in Fig. 2.2).

Localization-auto-context Module

To better utilize the information obtained from the low-resolution segmentation result for each structure, the Loc-Con module was introduced to produce the localized auto-context input for refinement. For each foreground object (BV or body), the Loc-Con module steps are as follows (Fig. 2.3):

1. Up-sample (trilinear interpolation) the initial coarse segmentation map to original resolution.
2. Generate a fixed-pixel size bounding box (144^3 for BV and 224^3 for body) located at the corresponding centroid of the up-sampled predicted probability map for each class. Images were zero-padded when smaller than the bounding box.
3. Concatenate the original resolution image and the initial predicted probability map (after up-sampling) in the bounding box to create the auto-context input for the refinement network.

Going beyond previous works [11, 58, 74], our Loc-Con module served as a hard attention mechanism by leveraging the low-resolution, rough segmentation to crop an ROI (the bounding box) at the original resolution. It made use of a conventional auto-context strategy [76] by employing the initial predicted probability map, in conjunction with the original image, as an additional input channel. The initial coarse segmentation map obtained from the whole image at low resolution can provide global context information, which improved the results of the subsequent segmentation networks (VNet II and III). Because trilinear up-sampling was used, the gradient from the subsequent refinement networks (VNet II and III) was able to flow back to the initial segmentation network (VNet I), which made end-to-end fine-tuning feasible (Fig. 2.2).

Pre-training of Fine-Resolution Refinement Network

Two refinement networks (VNet II and III) were trained for body and BV segmentation, respectively. For each object, the fine-resolution raw image and the up-sampled initial segmentation probability map in the localized bounding box were concatenated and used as the input. The structure of the refinement network was exactly the same as the initial segmentation, except that it required 2 channels as input (original cropped image and corresponding initial segmentation probability map), and produced a single channel as output (indicating the probability of being part of the body or BV at each pixel). Using the detected centroid information on the object, the output was zero-padded back to original image size.

End-to-end Refinement on Fine-Resolution

During the pretraining of the refinement stage, the parameters of the initial segmentation network (VNet I) were frozen until the refinement network for each object (VNet II and III) converged. After that, all three networks were jointly optimized to minimize the sum of Dice losses measured on the fine-resolution image (loss II and III in Fig. 2.2). The gradient backpropagation path for this end-to-end refinement is indicated in Fig. 2.2.

2.4.2 Mutant Classification and Visualization Pipeline

Subtle structural differences in the BV and body shapes between mutant and normal mouse embryos have been reported [86]. Hence, it is possible to use the BV and body segmentation map (the output of our proposed segmentation algorithm in Sec. 2.4) to train a volumetric CNN for mutant vs. normal binary classification. Due to the small number of training images, each with varying orientation, it was important to first rotate all images into a canonical orientation in order to reduce the input variance prior to feeding it into a classifier. This approach is more efficient for discovering subtle structural differences between the normal and mutant images, and for improving classification performance.

As shown in Fig. 2.4, we made use of the structural characteristics and relative positions of the BV and body to rotate the BV and body segmentation map into a canonical space. Specifically, we rotated the shape images so that the first Principal Component (PC) of the BV shape was aligned with the X-axis. We then rotated the images so that the first PC of the body shape was in the X-Y plane. Then the centroid positions of the BV and body were used to flip the images to make sure they all have the same up-right orientation along the Y axis. Finally, we made use of the fact that the front BV is wider than the back BV to flip the images along the X axis such that all the images had the same orientation. With all the BV and body shapes in the same orientation, a $256 \times 192 \times 160$ bounding box was cropped at the centroid of the body to remove unrelated regions. This bounding box was then downsampled by 2 to reduce input size. Finally, a 9-layer volumetric VGG-like CNN (Fig. 2.5) was trained on the rotated, cropped, and downsampled segmentation map to perform mutant vs. normal binary classification. Note that the input segmentation map has only one channel with different values indicating the BV, body and background.

In order to understand the decisions made by the trained network, we visualized the trained network through the gradient of the prediction with respect to the input segmentation map [68]. 20% of the maximum gradient value was used as the threshold to obtain a binary saliency image. Guided gradient backpropagation [70] was also implemented, and similar visualization results were obtained. These saliency images served as the explanations of the classifier's decisions between mutant and normal mice.

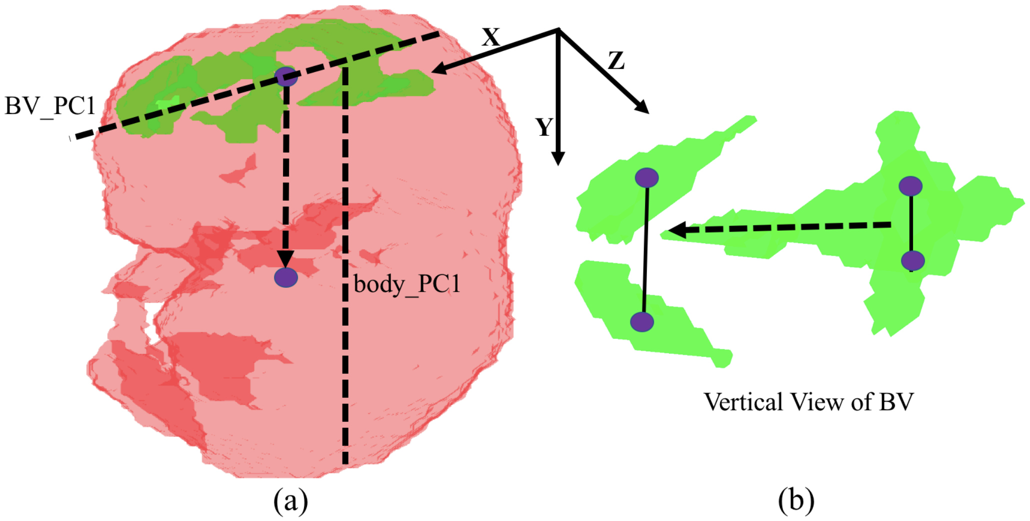

Figure 2.4: Procedure for rotating each BV and body segmentation map into a canonical orientation. The first Principle Component (PC) of BV and the first PC of body are used (indicated as dash line in (a)). The up-down direction (dash arrow in (a)) is determined by the comparative centroid positions of BV and body. The front-back direction (dash arrow in (b)) is determined by the structural characteristic of BV because the front BV region is wider than the back BV region (solid line in (b)).

2.4.3 Implementation Details

All the codes were written in Python 3.6.3. All neural network models were implemented in PyTorch 1.2 [53], with CentOS 7.4, CUDA 9.1 using 2 NVIDIA Tesla P40 graphic-processing-units (NVIDIA Corp., Santa Clara, CA, USA) with 2 x 24 GB of memory.

To compensate for the limited amount of training data for the segmentation networks, data augmentation was employed. Available volumes were randomly rotated from $-180°$ to $180°$ along each of the three axes, then randomly translated -30 to 30 voxels and, finally, randomly flipped. During the initial pretraining step (VNet I), the Dice loss (loss I) between the predicted segmentations and the downsampled manual labels were averaged across the three classes (background, body and BV). During the pre-training refinement stage (VNet II & III) and end-to-end refinement stage (VNet I, II and III), the Dice loss for body and BV (loss II and III) were used to train the networks. All networks were trained with the Adam optimizer [32] using a learning rate of 10^{-2} for the initial segmentation and pre-training refinement stages. A learning rate of 10^{-3} was used for the end-to-end refinement stage. The batchsize was 8 for the first stage segmentation; 2 for the second stage and refinement stage. The training data size was 139 and we applied a drop-last data-loader. Thus, for the first stage segmentation, 17 updates were performed per epoch; for the second stage and refinement segmentation, 69 updates were performed per epoch. An independent validation data set was used to determine the stop criterion. The first stage took 803 epochs; the second stage took 349 and 270 epoch for BV and body, respectively; and the refinement

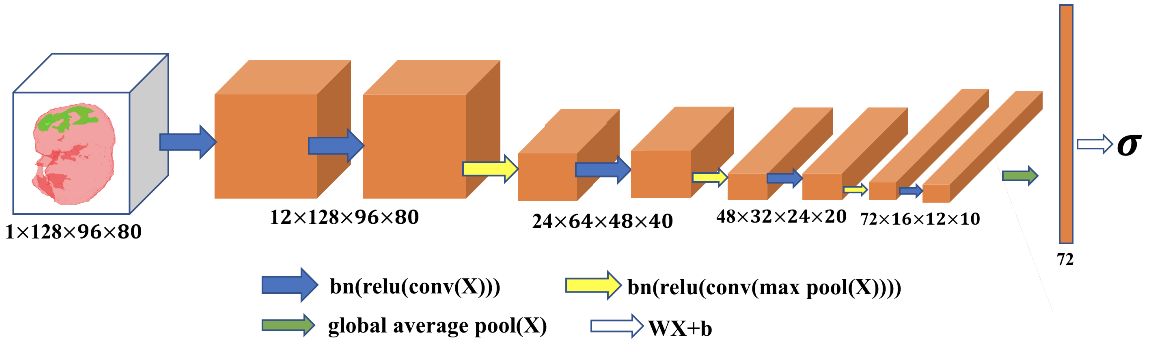

Figure 2.5: Pictorial representation of the mutant classification network. The numbers below each box indicate channel, depth, height and width. “bn”, “relu”, “conv”, “max pool”, “global average pool” and “WX+b” indicate batch normalization, rectified linear unit, convolution, max pooling, global average pooling and fully connected layer operations, respectively.

stage took 117 epochs. The models were chosen to minimize validation loss.

In this work, ITK-SNAP [89] was used to visualize some initial and end-to-end segmentation results of our proposed framework.

In order to compensate for the imbalance between the amount of mutant and normal images (102 mutant and 440 normal), weighted cross entropy loss was used to train the classification networks with weights 3.5 and 1.0 for the mutant and normal classes, respectively. No data augmentation was used for the training of the classification network because all of the images were rotated into a canonical orientation before feeding into the network. The network was trained using the SGD optimizer with momentum 0.9 and weight decay 10^{-5}. The learning rate was set to 10^{-2} for the first 70 epochs and decreased to 5×10^{-3} for the remaining 30 epochs. Approximately, each epoch would have 57 updates.

2.5 Experimental Results and Discussion

2.5.1 Segmentation Results

In this work, we evaluated the performance of the proposed framework using the DSC score with standard deviation, which is widely employed to evaluate segmentation performance in biomedical imaging. Given reference segmentation G and predicted segmentation P, the DSC can be computed as: DSC $= \frac{2|G \cap P|}{|G|+|P|}$. As mentioned in Sec. 2.3, we used 139 image volumes for training, 46 image volumes for validation and 46 image volumes for testing. We used further data augmentation as described in Sec. 2.4.3 during the training. Note that because each voxel can be treated as a training sample, the relatively small number of image volumes for training was sufficient to generalize well to the test image volumes.

As shown in Table 2.1, the initial results of our proposed segmentation framework achieved a satisfactory average DSC score of 0.924 for the body and lower DSC score

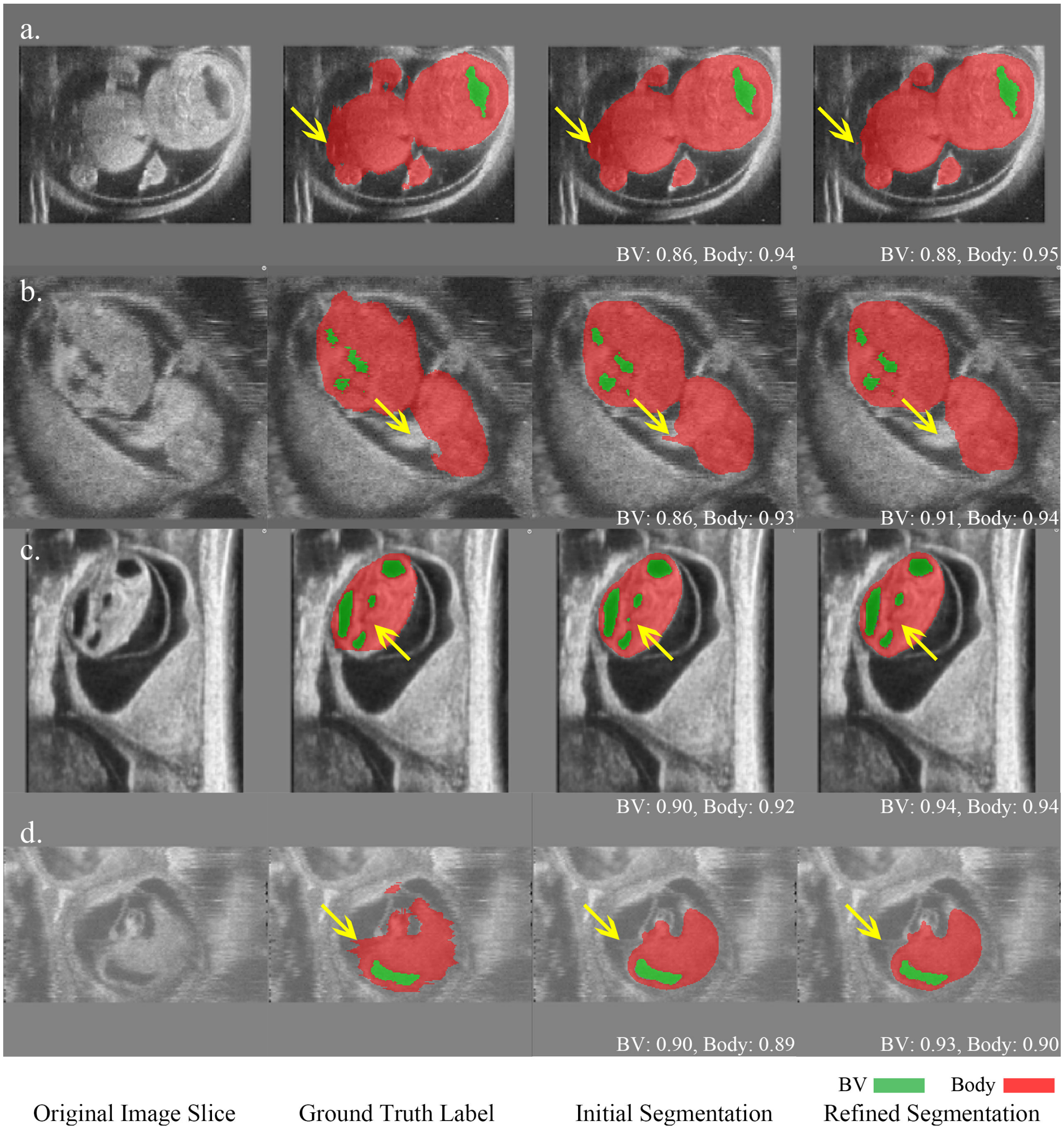

Figure 2.6: Comparison of initial coarse segmentation and refined segmentation for four HFU volumes. Green indicates BV, red indicates body and the numbers below the predicted segmentation correspond to DSC. In a), b) and c), yellow arrows indicate that the refinement improved the segmentation in terms of boundary and structure. d) represents an image with motion artifacts where the manual segmentation was noisy in the body background boundary while the refinement network produced a smooth boundary which was closer to the true physical structure. The refined BV segmentation in d) is also more accurate than the initial BV segmentation.

of 0.887 for the BV. This was expected because the BV was much smaller than the body. Hence, it was necessary to localize the ROI and refine the segmentation. The refinement using the raw image as well as the initial predicted probability map improved the average

Table 2.1: DSC with standard deviation and inference time per volume averaged over 46 test volumes

Results / Methods	BV DSC	Body DSC	Inference Time *
NGC-based Framework [35]	0.762±0.254	0.775±0.183	699.3 s
Sliding-window Benchmark	0.904±0.050 [54]	0.924±0.023 [55]	102.4 s
Coarse Segmentation	0.887± 0.055	0.924 ± 0.023	6 ms
Refinement Without Auto-Context Input	0.893±0.057	0.918±0.059	80 ms
Refinement With Auto-Context Input	0.898±0.052	0.927±0.026	90 ms
Refinement End-to-end	**0.899±0.056**	**0.934±0.015**	**90 ms**

* The average inference time was calculated on two NVIDIA Tesla P40 graphic cards. The sliding-window benchmark inference time was summed over separate BV [54] and body [55] segmentations.

DSC to 0.898 for the BV. In order to determine the efficacy of the auto-context approach, only the raw image cropped from the bounding box found in the localization step was fed to the refinement network. Compared to jointly using the initial segmentation and the raw image (auto-context input [76]), this refinement-without-auto-context approach yielded a lower DSC for the BV (between 0.893 and 0.898), which indicates context information is important to BV segmentation. In our application, jointly using the initial segmentation and the raw image (auto-context input) was similar to stacking two VNet together with the first one being frozen. In this way, the second VNet (the refinement network) had larger receptive field and was able to utilize more context information to improve the final segmentation results. Note that for the body segmentation, the gain from the refinement compared to the initial coarse segmentation was limited, because the body boundary was fairly smooth and did not suffer from a downsampled representation. Finally, end-to-end refinement improved BV DSC to 0.899 and body DSC to 0.934. As shown in Fig. 2.6, the initial segmentation produced reasonable BV and body segmentation results. After end-to-end refinement, the segmentation accuracy was improved along with better boundaries, and sometimes eliminated structural segmentation errors in the first stage. This implies that the fine-resolution refinement was more beneficial than suggested by the small improvement in the DSC metric.

Compared with other existing methods, our proposed segmentation framework outperforms the rule-based segmentation NGC-based framework (Tab. 2.1) [35] by a large margin. Moreover, the NGC-based framework was not robust with 8 failure cases in BV and 3

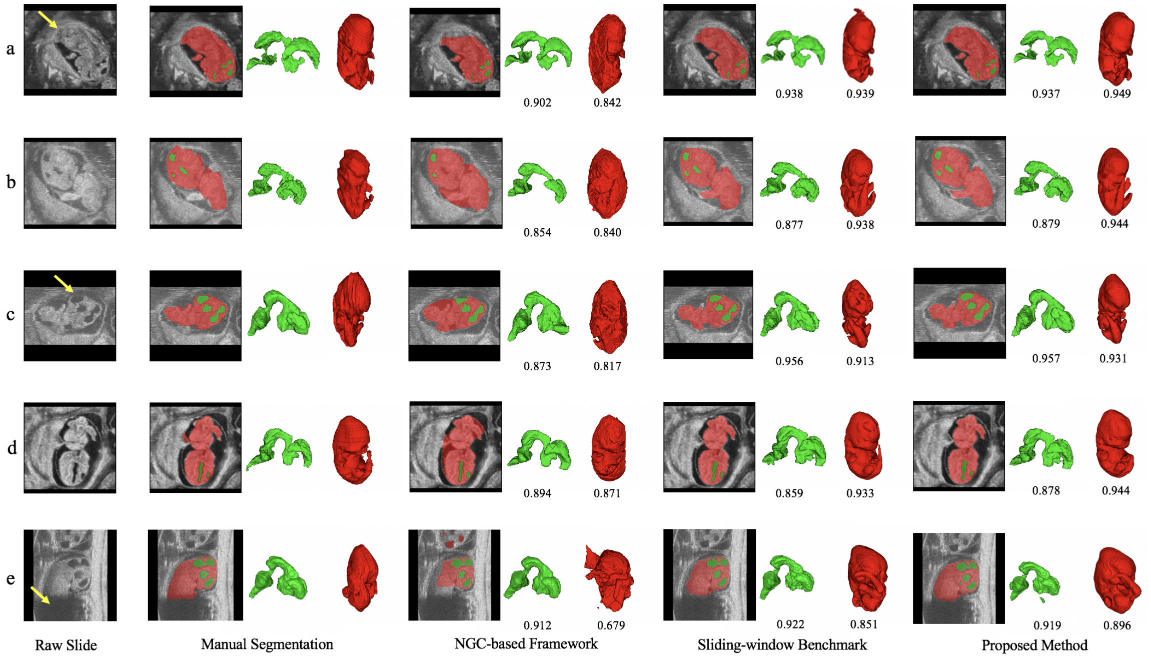

Figure 2.7: Comparison of qualitative segmentation results among different methods for five HFU images. Green indicates BV, red indicates body and the numbers below the predicted segmentation are corresponding DSC. Yellow arrow in a) indicates ambiguous boundary due to the deep touching of the body and uterine wall. Image b) has severe motion artifacts. Yellow arrow in c) indicates missing head boundary. Image d) has different contrast with image b) and c). Yellow arrow in e) indicates severe missing signal of body, which leads to unsatisfactory automatic body segmentation results across different methods.

in body (DSC < 0.6) while our proposed framework and the sliding-window-based methods [54,55] did not have any failure cases. Although the performance was comparable to the sliding-window-based methods, our proposed method achieved a 1000 fold inference time reduction from 102.36 to 0.09 seconds per volume, enabling real-time segmentation. For fair comparison, the networks from [54] and [55] were retrained using the same training set described here and evaluated on the same testing set. Therefore, the numbers reported here are slightly different from those reported in [54,55].

Qualitatively, the proposed segmentation pipeline and sliding-window-based methods performed consistently well when challenged with ambiguous or missing boundaries, motion artifacts, and differing image contrast (Fig. 2.7). In contrast, the NGC-based framework performed worse and failed to produce correct segmentation when boundaries were missing or ambiguous. Our manual body segmentation was achieved by labeling every few images in a volume and then using a label interpolation function to complete the 3D segmentation. Although we determined that the interpolated manual segmentation was reliable for algorithm development and other down-stream analyses, this protocol can potentially lead to interpolation artifacts (e.g. Fig. 2.7(b)(c)). Advantageously, these artifacts were mitigated by our deep-learning-based framework with segmentation results closer to the true physical structure. For similar interpolated 2D slices across different 3D images, some were slightly under-segmented on the boundaries while others were slightly over-segmented. Moreover, there are still sufficient number of manually labeled slices which are accurate on the boundaries. When we trained our deep neural networks with over a hundred such 3D images, the network learned the true boundaries so that the interpolation artifacts were mitigated. This is similar to small random noise being added to the manual label, but the trained network can still generalize well [2].

Table 2.2: Confusion matrix of mutant classification results summed over validation samples with six-fold cross validation. The threshold value was set to 0.5 and the average accuracy is 0.969.

Predict True	Mutant	Normal
Mutant	96	6
Normal	11	429

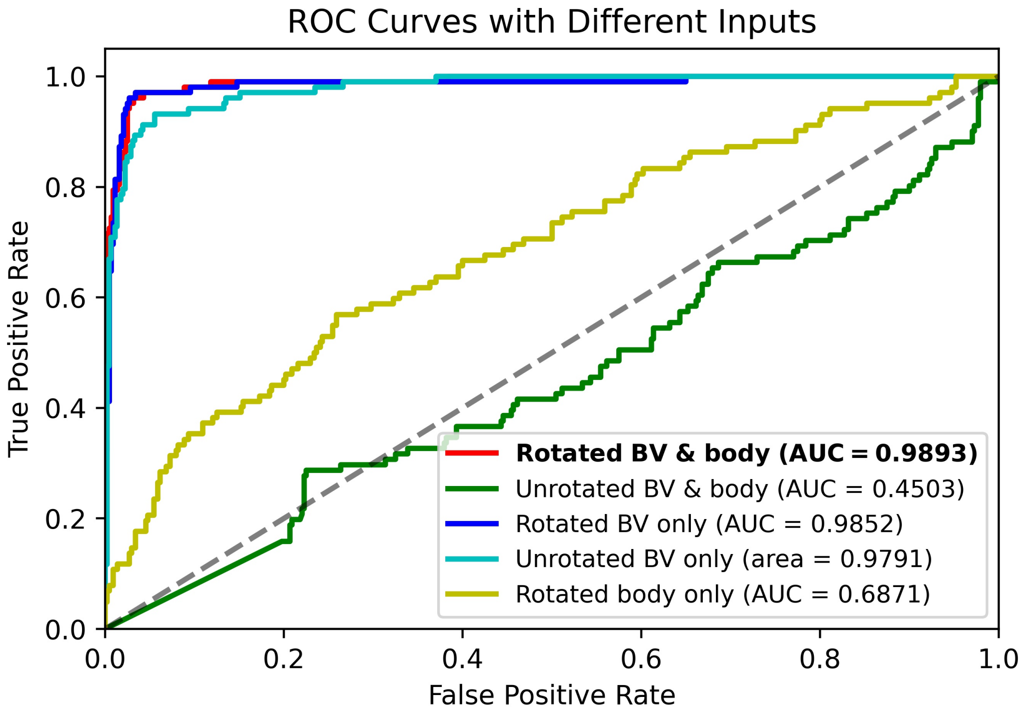

Figure 2.8: ROC curves and AUC scores of the mutant classification results with different input combinations (each obtained with six-fold cross validation).

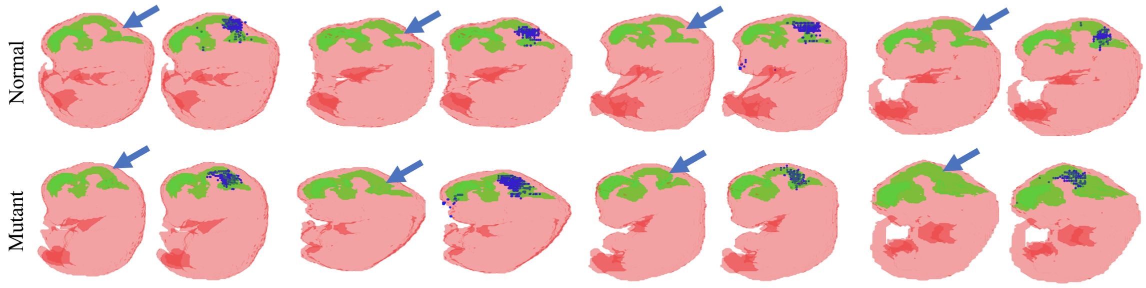

Figure 2.9: Saliency images of the trained mutant classification neural network. The first row is the normal mouse embryo BV (green) and body (red) segmentation while the second row is mutant. Two images are presented for each sample. The blue arrow in the first image indicates the known structural differences between *En1* mutant and normal BVs while the blue dots in the second image (salient points) indicate where the trained network focused when making the prediction.

2.5.2 Classification and Visualization Results

Using the rotated BV and body segmentation maps, a volumetric CNN (Fig. 2.5) was trained to perform mutant vs. normal binary classification. Due to the limited number of mutant images (102 mutant vs. 440 normal), we conducted a six-fold cross validation, where each fold had the same mutant vs. non-mutant ratio. In each run, one fold was used for validation while the other five folds were used for training. The average classification accuracy among validation samples are shown in Table 2.2. We also show the Receiver Operating Characteristic (ROC) curve in Fig. 2.8 (red curve), which was obtained by using different thresholds on the predicted probability for the mutant class. The average classification accuracy was 0.969, and the area under the ROC curve (AUC) was 0.9893.

In order to verify the utility of rotating the shape images into a canonical space, we also used unrotated BV and body segmentation maps as input to train the same classification network. Without pre-processing rotation, the BV and body were in a wide variety of orientations in the segmentation maps. Because of the limited amount of training data, additional segmentation maps were generated by using existing data but with random combinations of 90-, 180-, or 270-degree rotations around each axis or image flipping along each axis. Using this data, the network still failed to converge to good results (AUC of 0.4503, Fig. 2.8 green curve). The reason for the divergence of the network could be that the subtle structural differences between normal and mutant mouse embryos (Fig. 2.9 blue arrows) were overwhelmed by the large variations of embryo orientation. Hence, for our somewhat limited data set it was critical to perform the rotation into a canonical space such that the embryo orientations were aligned. A standard 3D image orientation also assists better visualization of the 3D volumes.

To investigate which shape information (BV, body or both) was important for final classification, only the rotated BV or body segmentation maps were used to train the same classifier. This approach achieved an AUC of 0.9852 for BV and 0.6871 for body (Fig. 2.8 blue and yellow ROC curves), respectively. It is worth noting that the AUC of 0.9852 obtained based on just the BV segmentation is similar to the AUC of 0.9893 obtained using BV and body segmentation maps. These AUC values indicate that the BV contributes the most to successful mutant classification. This is consistent with the saliency images (Fig. 2.9), where most salient points were located around the known structural differences between *En1* mutant and normal BVs. We then defined a tight bounding box around unrotated BV data and trained the same classifier with the same data augmentation as the above unrotated BV and body segmentation maps (Fig. 2.8 green curve). Using this approach, we achieved an AUC of 0.9791 (Fig. 2.8 cyan curve), which indicates that unrotated BV (with data augmentation) is sufficient to train an accurate mutant classification network. These results also explain why, even with sufficient data augmentation, unrotated BV and body data (Fig. 2.8 green curve) fails, because the large bounding box includes the BV and body. The large bounding box makes the BV too small relative to the input image size such that the subtle BV differences between *En1* mutant and normal embryos were easily overwhelmed by the variations of embryo orientation. Our rational for using the BV and body together to train a classifier is that a difference in spine curvature exists between *En1* mutant and normal mouse embryos [86]. Using the BV and body together, the visualization in (Fig. 2.9) would have the potential to highlight differences in spine curvature. Unfortunately, our trained classifier does not seem to make use of the spine curvature in its decision. The reason might be the spine curvature difference is not as consistent and conspicuous as the BV difference between mutant and normal mouse embryos.

More importantly, as shown in Fig. 2.9, we visualized the trained network through the gradient of the prediction with respect to the input segmentation map [68] and used 20% of the maximum gradient value as the threshold to obtain a binary saliency image. The visualization of the trained classifier (using rotated BV and body segmentation maps) demonstrated that the trained network focused on regions where *En1* mutation is known to cause the loss of brain tissue and thickening of the BV (Fig. 2.9 blue arrows). This BV region is the main ROI when performing manual segmentation for mutant vs. normal classification *En1*. If had not known a priori where to detect the difference in BV between normal and *En1* mutant mouse embryos beforehand, the visualization results of the trained network would have highlighted these relevant regions. This observation indicates that gradient backpropagation of trained, deep-learning classifiers has the potential to automatically detect unknown phenotypes associated with a known genetic mutation.

2.5.3 Limitations

Our study had a few limitations. First, the proposed two-stage segmentation framework only provided limited improvement over the first stage initial segmentation. If less-accurate automatic segmentation quality for some down-stream analyses is acceptable, the initial segmentation would be enough, which will further reduce the inference time by another factor of 15 (from 90 ms to 6 ms per image). Second, the reference manual segmentation for the 3D images was obtained by labeling every few frames and then using a label interpolation function, which inevitably introduced some interpolation artifacts. Finally, although a difference in spine curvature has been reported to exist between *En1* mutant and normal mouse embryos [86], our trained classification neural network did not seem to use this difference to perform the classification (Fig. 2.9). Further work is necessary to understand why our methods did not detect these differences.

2.6 Summary of Major Contributions

- For segmentation, an end-to-end two-stage framework was proposed for accurate and real-time segmentation of the BV and body in 3D, *in vivo* and *in utero* HFU images of mouse embryos. The initial coarse segmentation stage acted as an ROI localization module and provided global context information for the second-stage, fine-resolution refinement network. The results demonstrated the efficacy of this two-stage structure. The proposed method achieved high DSC scores of 0.899 for BV and 0.934 for body segmentation, comparable to the previous benchmark (i.e. sliding-window-based methods), and was approximately one thousand times faster in inference time.

- For mutant vs. normal classification, a deep-learning-based method was also developed using the BV and body segmentation maps. To overcome the limited data problem, a fully automatic method was developed to rotate the raw segmentation maps such that the BV and body shapes were in a canonical orientation, thus, removing uninformative input variations. Using this pre-processing approach, the model achieved a high average accuracy of 0.969 and AUC of 0.9893 over six cross validation folds.

- For network visualization, the trained classification model was shown to differentiate between mutant and normal mouse embryos by focusing on the BV region where the phenotype associated with the *En1* mutation typically manifests. The proposed pipeline has the potential to uncover unknown phenotypes manifested as shape changes associated with different gene mutations.

To sum up, our segmentation, mutant classification and network visualization algorithms may be applicable and invaluable in streamlining developmental biology studies.

Chapter 3

Camera Pose Estimation

3.1 Introduction

Estimating the 3D position and 3D orientation (6DoF pose) of an agent or an object with respect to a reference coordinate frame is a fundamental requirement in robotics applications. Visual localization offers several advantages compared to other modalities for deriving 6DoF poses: it is effective both indoors and outdoors, it requires no extra infrastructure, and it can be precise and accurate using only a single RGB image.

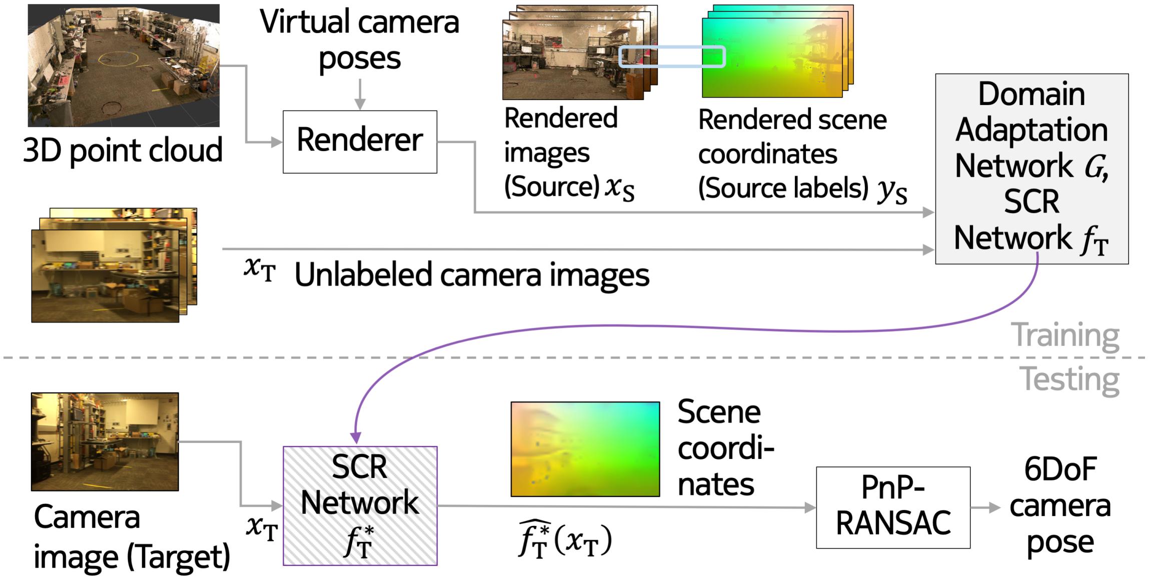

Figure 3.1: At training time, the SCR network is trained using unsupervised deep domain adaptation techniques, which bridge the domain gap between labeled synthetic images and real camera images of the scene. At test time, only the trained SCR network is kept, it regresses 3D scene coordinates for each pixel, from which the camera pose is calculated via PnP-RANSAC.

Prior solutions for pose estimation from RGB images can be split into two categories: those that use hand-crafted features and those based on machine learning. The methods based on hand-crafted features incorporate extensive prior knowledge about the problem, and often achieve better accuracy today. However, learned methods can offer higher speed, robustness to occlusions, and access to continuous intermediate representations, and are therefore gaining popularity. Furthermore, learning-based methods such as CNN can offer potentially more compact representations of spaces in their weights compared to Simultaneous Localization and Mapping (SLAM) maps, and have been shown to outperform many hand-crafted methods in textureless areas [80].

The major drawback of learning-based methods in the past has been that they require supervised training on large image sets with known camera poses, and need to be re-trained for every new scene or in case any changes in the target scene. In general, a labeled training set needs to be generated for each new environment, making data collection tedious and hindering the scalability and practical applicability of these techniques. A typical way to generate a training set is to track the camera with an external localization system while it moves through the environment, implying high overhead. Infrared tracking systems using fixed infrastructure cameras and active beacons (e.g., WorldViz or Vicon) can capture sub-cm and sub-degree accurate pose labels for the moving camera images. However, the cost and difficulty of using these localization systems is not trivial and is even infeasible in some places. Another method of obtaining labeled training images is to use on-board RGB-D SLAM as in 7Scenes [66] and ScanNet [13].

In this thesis, we take a representative method that solves the scene coordinate regression problem, and show that it can be trained with synthetically generated images at a fraction of the cost compared to acquiring real pose labels, and it still achieves the same median error. As an alternative to training with expensive labeled camera images, we propose a novel pipeline Domain Adaptation of Networks for Camera pose Estimation (DANCE), shown in Fig. 3.1, which relies on a lower-cost combination of unlabeled camera images and labeled synthetic images.

Our key contributions are (i) providing a domain adaptation methodology to train a neural network for the task of SCR using only labels generated in simulation (via rendering). This enables (ii) the training of camera pose estimation at significantly lower overhead cost compared to fully supervised learning-based solutions. Specifically, we render a large number of images with known (arbitrary) poses and scene coordinates from a laser scan of the space. The appearance of these images is far from real photos, and to bridge this domain gap, (iii) we employ domain adaptation at the image level. These techniques allow the training of the SCR network with domain adapted labeled rendered images only.

3.2 Related Work

Vision-based localization approaches can be categorized into retrieval-based and regression-based families, both requiring a large number of images that cover the whole scene.

Retrieval-based methods typically extract global descriptors from keyframes [12] [79] and/or local descriptors from feature points [71] [49] [50], and build a database of scene descriptors. Then, descriptors extracted from the query image or sequence of images are matched to the closest entries in the database and assigned a location, which is finally validated by geometric constraints. Feature point-based methods are more robust, but tend to be slower than keyframe-based methods. While we focus on indoor scenarios, our problem is highly related to large-scale visual localization methods [61] [28] [59] [43] which are predominantly based on features and sparse 3D maps, but have recently begun incorporating learning-based components as well [60]. Prior knowledge about the coarse location from GPS [61] [90] [43], radio signals [29] [20], a LiDAR map [16] or other means can significantly reduce the search space and can make these methods applicable even at city scale [43].

Regression-based methods are both robust and fast, and therefore offer a promising new direction. However, at the time of writing, they are less accurate, limited in scale, and expensive to train. One family of regression approaches use learned models [66] [78] [6] [10] to perform SCR and then input point samples from the intermediate scene coordinate map to a PnP-RANSAC pose estimator. More recent examples of this group of methods are DSAC, DSAC++, and DSAC* [8], which add differentiable approximations for all steps of the pipeline, including SCR, PnP, and RANSAC and achieve state-of-the-art pose estimation performance. Another family of regression approaches including PoseNet [30], PoseLSTM [80], and RelocNet [5] directly return the 6DoF pose from a single image and can be trained end to end. In addition to localization of a single image, VLocNet++ [56] implements learning-based odometry and adds semantics, and for the first time exceeds the accuracy of feature-based methods. Sattler et al. [62] analysed why direct pose regression methods generally fall behind feature-based methods and concluded that CNNs rather learn to retrieve similar images instead of learning a 3D map of the space. Further research is urged for, and our training technique makes that a lot easier than before.

An often cited drawback of learning-based methods is that they are trained for a particular scene and are difficult to adapt to other environments. New techniques have attempted to address this drawback using a variety of methods. The authors of [10] and [9] show how to adapt a random forest to a new place at runtime. RelocNet [5] performs regression of relative poses and thus avoids the need to retrain for every scene, while ESAC [7] breaks a scene into smaller parts and trains a network for each before using an ensemble network to decide which subnetwork to use, thus allowing SCR to be performed for larger areas. Unfortunately, all regression models require supervised training on image datasets with ground truth pose labels or scene coordinates, which are very expensive to acquire. Similarly in feature-based methods, the need to support the innumerable variations in scenes resulting from lighting changes, weather conditions, etc., and the cost of collecting data across these conditions can be prohibitively expensive. These challenges can be avoided by synthetically generating rather than collecting data [67] [73] [75] [45].

Because modeling every aspect of the real world in a rendering pipeline is infeasible, synthetic images inevitably differ from those captured with a real camera – this difference in appearance is referred to as the *domain gap*. The authors of [65] demonstrate that feature-based retrieval using the representations learned by a PoseNet trained on purely synthetic images is highly effective for synthetic queries, but fails when used on real images.

Researchers have taken a variety of approaches to attempt to reduce this domain gap. [65] transforms real features to look more similar to synthetic features using an autoencoder. Other recent works on 6DoF object pose estimation [1] [39] [83] apply domain randomization, i.e., generating synthetic training data with randomized rendering parameters in order to robustify the trained networks. Several domain adaptation works [19] [77] apply feature-level alignment for image classification; other recent works [25] [38] combine the CycleGAN-based image-level alignment, and adversarial feature-level alignment for image segmentation. CyCADA [25] performs both the image-level and feature-level adaptation in an end-to-end manner while BDL [38] decouples them. Because both image segmentation and our SCR task require dense predictions, it is imperative for the adapted synthetic images to preserve both the semantic content and the geometric structure. We propose to employ the contrastive unpaired translation (CUT) model [52] for image-level domain adaptation to train the SCR network.

3.3 Method

In order to successfully bridge the domain gap, we must transform rendered images such that they appear to come from the same distribution as the real camera images while preserving both the semantic content and the geometric structure to enable effective camera pose estimation. We utilize a generative adversarial network (GAN) based framework (the CUT model [52]) for the image-to-image translation step. Specifically, using the rendered images X_S along with a set of unordered, unlabeled photos X_T from the same scene, a mapping network $G_{S \to T}$ is trained using the CUT framework to map from the source domain of rendered images X_S to the target domain of real photos X_T without changing the geometric and semantic content in X_S. In this way, the corresponding rendered scene coordinate labels Y_S of X_S can be reused for $\hat{X}_T$, where $\hat{X}_T = G^*_{S \to T}(X_S)$ (* denotes converged models after training). Finally, an SCR network f_T is trained with ($\hat{X}_T$, Y_S) for use in the target domain X_T. Note that direct training on domain X_T is not possible because the target labels Y_T are not available and real photos X_T do not have any corresponding or pairwise relationship with synthetic images X_S. At inference time, we apply the target network f_T^* in the target domain X_T (real photos) and feed the predicted scene coordinates $f_T^*(X_T)$ to a traditional PnP-RANSAC [17] to compute the final pose estimates. This whole process is illustrated in Figure 3.1.

One could ask why not just train on (X_S, Y_S) and apply the converged model in the target domain X_T directly. As shown in Table 3.1 (a) (blind transfer), this approach leads to severely degraded performance. In order to bridge the domain gap between the rendered images X_S and photos from the real world X_T, we train the SCR network on the domain adapted labeled rendered images ($\hat{X}_T$, Y_S).

3.3.1 Mathematical Description

We consider an unsupervised domain adaptation problem, where we are provided distributions for source data X_S, source labels Y_S and target data X_T, but no target labels. The ultimate goal is to learn a model f_T that can accurately predict the label on the target distribution X_T. In our problem, X_S indicates rendered (source domain) images, Y_S indicates rendered (source domain) scene coordinate labels, X_T indicates real (target domain) camera images, and f_T indicates the SCR network trained using DANCE to perform well in the domain of real camera images (target domain). The rendered scene coordinates Y_S encode the (X, Y, Z) coordinates in the model of the world (point cloud) which correspond to each pixel (U, V) in the images X_S. The objective of the trained SCR network f_T is to predict these $((U, V), (X, Y, Z))$ correspondences for images in the target domain. The training procedure is illustrated in Fig. 3.2 and described below.

Domain Adaptation

We use a combination of preprocessing and image-level domain adaptation techniques to map the source images into the target domain. First, simple histogram matching brings the color distribution of the rendered images X_S closer to that of the target domain X_T, because our laser scanner's automatic white balance setting leads to a mismatch with the query camera. Hereafter, X_S indicates source rendered images after histogram matching.

Next, an image-level domain adaptation network $G_{S\to T}$ is trained to translate the rendered images (source) X_S into the domain of real photos (target) X_T so that they can fool an adversarial discriminator network D_T (see Fig. 3.2). The following GAN objective is employed:

$$\begin{aligned}
&\mathcal{L}_{GAN}(X_S, X_T; G_{S\to T}, D_T) \\
=&\mathbb{E}_{x_t \sim X_T}\left[\log D_T(x_t)\right] \\
+&\mathbb{E}_{x_s \sim X_S}\left[\log(1 - D_T(G_{S\to T}(x_s)))\right]
\end{aligned} \tag{3.1}$$

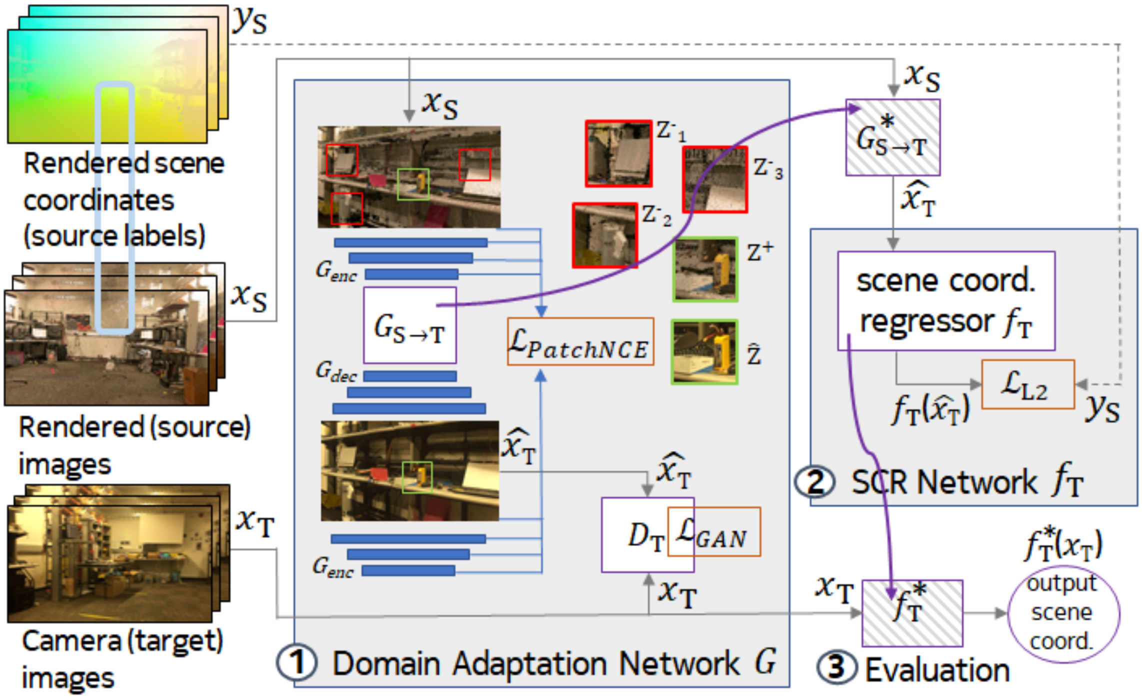

Figure 3.2: The training pipeline for the SCR network with unsupervised deep domain adaptation.

The mapping network $G_{S\to T}$ tries to minimize the loss function while the discriminator D_T tries to maximize it. This optimization procedure ensures that the learned mapping $G_{S\to T}$ is able to translate source images to convincing target images. Note that there is no corresponding or pairwise relationship between source data X_S and target data X_T.

Because scene coordinate estimation from the input image is a dense prediction task, it is important that the mapping $G_{S\to T}(X_S)$ preserves the structure and content of the original image X_S. However, a traditional GAN model (with objective 3.1 only) can not ensure this consistency requirement. To enforce consistency, the following multi-layer patch-wise noise-contrastive estimation (PatchNCE) loss is adopted [52]:

$$\begin{gathered}\mathcal{L}_{PatchNCE}(X_S; G_{S\to T}, H) = \\ \mathbb{E}_{x_s\sim X_S}\sum_{l=1}^{L}\sum_{s=1}^{S} l(\hat{z}_l^{(s)}, z_l^{+(s)}, z_l^{-(S\backslash s)})\end{gathered} \tag{3.2}$$

A high-level understanding of the $\mathcal{L}_{PatchNCE}$ loss [52] is that the patches at the same location before and after the mapping network should be more similar than patches at other spatial locations in the same image. Here, $\hat{z}_l^{(s)}$ is the feature vector at location s from the l-th feature layer for the translated image $\hat{x}_T = G_{S\to T}(x_S)$. $z_l^{+(s)}$ is the feature vector at the same location s from the l-th layer for the input image x_S, and $z_l^{-(S\backslash s)}$ are the feature vectors at locations other than s from the l-th layer for the input image x_S.

The mapping network $G_{S \to T}$ can be decomposed into an encoder G_{enc} followed by a decoder G_{dec}. The feature vectors $\hat{z}_l^{(s)}$, $z_l^{+(s)}$ and $z_l^{-(S \backslash s)}$ are extracted from the l-th layer of G_{enc} and then passed through a small 2-layer multi-layer perceptron (MLP) network H_l, producing 256-dim final features. L (=5) is the number of layers chosen to extract features and S (=256) is the number of locations sampled in each layer. Each layer and spatial location of these extracted features represents a patch of the input image, with deeper layers corresponding to larger patches. The loss function

$$\begin{aligned} l(\hat{z}_l^{(s)}, z_l^{+(s)}, z_l^{-(S \backslash s)}) = \\ -\log \left[\frac{exp(\hat{z}_l^{(s)} \cdot z_l^{+(s)} / \tau)}{exp(\hat{z}_l^{(s)} \cdot z_l^{+(s)} / \tau) + \sum_{n=1}^{len(S \backslash s)} exp(\hat{z}_l^{(s)} \cdot z_l^{-(n)} / \tau)} \right] \end{aligned} \tag{3.3}$$

encourages the current query $\hat{z}_l^{(s)}$ to be closer to the positive example $z_l^{+(s)}$ but different from negatives $z_l^{-(S \backslash s)}$. τ (=0.07) is a temperature parameter to scale the similarity between two examples. For more details, please refer to [52].

The aggregate loss used to train the image level adaptation network $G_{S \to T}$ can be summarized as follows:

$$\begin{aligned} & \mathcal{L}_{CUT}(X_S, X_T; G_{S \to T}, D_T, H) \\ & = \lambda_{GAN} \mathcal{L}_{GAN}(X_S, X_T; G_{S \to T}, D_T) \\ & + \lambda_S \mathcal{L}_{PatchNCE}(X_S; G_{S \to T}, H) \\ & + \lambda_T \mathcal{L}_{PatchNCE}(X_T; G_{S \to T}, H) \end{aligned} \tag{3.4}$$

The third term $\mathcal{L}_{PatchNCE}(X_T; G_{S \to T}, H)$ is an identity loss for network regularization.

After training the domain adaptation network, we only keep the trained model $G^*_{S \to T}$ as a fixed transformation function from source images X_S (rendered images) to target images X_T. Hereafter, $\hat{X}_T = G^*_{S \to T}(X_S)$ indicates domain adapted images.

Target SCR Network Training

Finally, the paired training data ($\hat{X}_T$, Y_S) are used to train a target SCR model f_T with an L2 loss:

$$\mathcal{L}_{L2}(\hat{X}_T, Y_S; f_T) = \mathbb{E}_{(\hat{x}_t, y_s) \sim (\hat{X}_T, Y_S)} \left\| y_s - f_T(\hat{x}_t) \right\|_2 \tag{3.5}$$

After training the target model f_T with the above loss, the trained model f_T^* can be used to predict scene coordinates for target camera images X_T at testing time. Finally, the predicted coordinates $f_T^*(X_T)$ are passed to PnP-RANSAC to compute final pose estimates.

3.3.2 Implementation Details

Network Architectures

The SCR network f_T is a fully convolutional network consisting of a feature encoder followed by a regression head. Specifically, the feature encoder is a ResNet18 [22] after removing the last 2 layers (1000-d fc and average pool) and setting the last 2 stride-2 convolutional layers (conv4_1 and conv5_1) to stride 1. The regression head has 3 convolutional layers to transform the features from the encoder to the 3-channel scene coordinate predictions. For the image-level adaptation, we follow the network architectures of CUT [52] with a ResNet-based generator of 9 residual blocks ($G_{S \to T}$) and a PatchGAN discriminator (D_T).

Training

When training the CUT-based domain adaptation network $G_{S \to T}$, the image is randomly cropped to 320 × 320 pixels and the network is trained for 6 epochs with learning rate 2.0 $\times$ 10^{-3}, batch size 10 and Adam optimizer. The weights in equation 3.4 are set to $\lambda_{GAN} = 1$, $\lambda_S = 1$ and $\lambda_Y = 1$. In order to compute the multi-layer PatchNCE loss ($\mathcal{L}_{PatchNCE}$), features are extracted from 5 layers ($L = 5$), which correspond to RGB pixels, the first and second downsampling convolution, and the first and the fifth residual block. These layers correspond to receptive fields of sizes (i.e. patch sizes) 1×1, 9×9, 15×15, 35×35, and 99×99. For features of each layer, 256 ($S = 256$) random spatial locations are sampled, and a 2-layer MLP H_l is employed to extract 256-dim final features. To train the final SCR network f_T, the image is also randomly cropped to 320 × 320 pixels and the network is trained using the Adam optimizer with learning rate 1.0×10^{-4}, weight decay 1.0×10^{-5}, and batch size 48. Because our DANCE pipeline can generate innumerate training data, we do not perform other data augmentation methods besides random cropping. All the hyper-parameters and the stopping criterion are selected based on the experimental results of an independent validation dataset.

3.4 Dataset Generation

We evaluate the pose estimation performance of several techniques within our laboratory space (5.8 meters wide, 14.3 meters long, 3.0 meters high). We capture or generate all necessary datasets for training, validation, and testing to ensure a fair comparison among the techniques. The data used in this study will be provided upon request.

3.4.1 Laser Scan

While there is nothing specific to point clouds about DANCE, for simplicity we use a Leica BLK360 laser scanner to capture a color point cloud of our robotics lab space. To reduce occlusions, we merge 16 scans into a single point cloud of 118M points, each storing location (X, Y, Z) and color (R, G, B) information.

3.4.2 Synthetic Images

Synthetic images with corresponding scene coordinate labels are generated by placing a virtual camera with known random pose in the space and projecting the point cloud onto the virtual image plane using intrinsic parameters measured from the device camera. We render 100k synthetic images with corresponding scene coordinate labels from virtual camera poses drawn from a similar distribution as the device images (described below). These images and labels can be considered a representative sample set from the source distribution X_S and Y_S. The lab is equipped with a WorldViz infrared tracking system that serves as the reference coordinate frame and provides poses for evaluation purposes (this is the expensive step that DANCE seeks to circumvent). The transformation between WorldViz and the point cloud(s) is established by recording the coordinates of fixed WorldViz markers with respect to each frame and aligning them via the iterative closest point algorithm.

3.4.3 Camera Images

We sample the target distribution X_T (capture real photos) by moving an iPhone 6 in the space. The phone is mounted horizontally on a wheeled cart which is pushed manually through the space in order to mimic a robot with a fixed RGB camera. The ground truth pose of the camera is determined by tracking multiple WorldViz markers placed on the cart and by establishing the transformation between the cart and camera frames. Camera images are extracted from videos captured over four trajectories and downsampled to 640×360 pixels. We dedicate two trajectories (28411 images) for training the baseline networks for comparison with DANCE, one trajectory (1637 images) for validation, and one trajectory (2104 images) for testing purposes. The WorldViz ground truth pose labels are collected for all the photos, but these labels are not needed for the training of the DANCE pipeline. These pose labels are only used for training the fully-supervised baselines and for evaluation.

3.4.4 Domain Gap

Although we do not need to know where the domain gap stems from in order to bridge it, we hypothesize that in our case it stems from the simplicity of the rendering method. An image rendered from a point cloud has inevitable holes due to occlusions and splatting artefacts, so the resulting synthetic image largely differs from a photo of the scene.

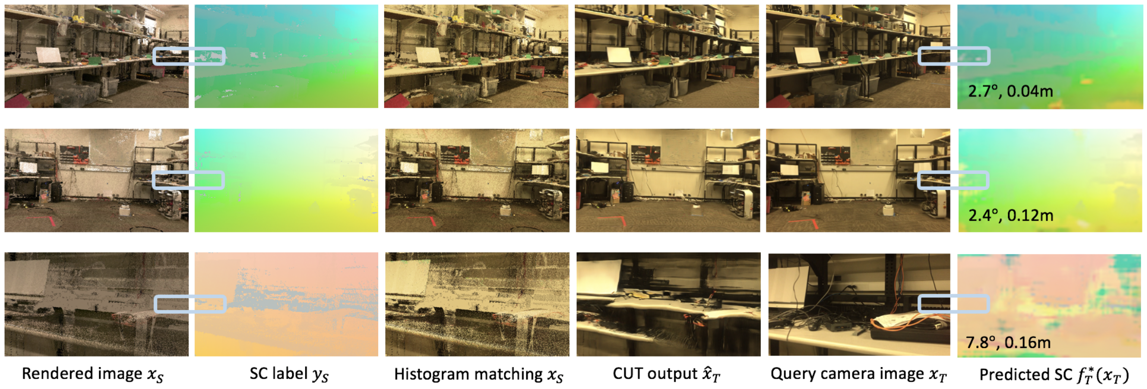

Figure 3.3: Three samples are shown with some intermediate results. The numbers on the predicted scene coordinates (SC) indicate estimated pose errors. The last row is an unsatisfactory case because the query image has small field of view and the corresponding location is poorly covered by the point cloud. For illustrative purposes, the rendered (source) images X_S and camera (target) images X_T are shown in the same poses. In practice, X_S and X_T do not have any pairwise relationship.

3.5 Experimental Results

We evaluate the predicted pose error of each technique with respect to the measured ground truth (WorldViz) pose. All the results are reported on the 2104 test images.

Comparing training strategies We compare side by side multiple training strategies of the same SCR network, including the original, expensive fully supervised labels, and we show that our proposed strategy can achieve comparable median performance. As summarized in Table 3.1, we first quantify an intuitive lower bound (a) and upper bound (e) of the $DANCE$ pipeline where the SCR networks are trained with different strategies. When the SCR network is trained on the 100k synthetic images with corresponding scene coordinate ground truth (without any domain adaptation, (X_S, Y_S)) and tested with real camera images X_T, it fails (Table 3.1(a)). This lower bound (Blind Transfer) performance indicates that the domain gap between the rendered and camera images is significant. When the SCR network is trained on the 28411 real images with scene coordinate labels (rendered from the point cloud using ground truth poses only for evaluation), 2.9°, 0.17m median error is obtained (Table 3.1(e)). This indicates that if there is no domain gap between the training and testing images, our proposed SCR and PnP-RANSAC pipeline can achieve good performance.

When we trained the SCR network within the proposed unsupervised deep domain adaptation framework (100k domain adapted labeled synthetic images ($G^*_{S\rightarrow T}(X_S)$, Y_S)), $DANCE$ (Table 3.1(d)) outperforms the lower bound method (no domain adaptation, Table 3.1(a)) by a large margin and is on par with the upper bound method (full supervision)

Method	Median error	95%-tile error	Requires
PoseNet [30]	$4.2°, 0.28$m	$16.9°, 0.69$m	RP
UcoSLAM [50]	$2.7°, 0.08$m	9.7% invalid	R+
(a): Blind Transfer	$110°, 7.57$m	$175°, 16.5$m	SSC
(b): (a) + Hist. match.	$14.1°, 1.09$m	$154°, 15.3$m	SSC,R
(c): (b) + CycleGAN	$4.2°, 0.22$m	$96.5°, 6.58$m	SSC,R
(d): (b) + CUT (**DANCE**)	$3.0°, 0.14\,m$	$25.7°, 0.95$m	SSC,R
(e): Fully Supervised [8]	$2.9°, 0.17$m	$11.1°, 0.47$m	RSC

Table 3.1: Comparison of existing pose estimation methods and variants of our DANCE proposal. Requirements: R Real images (lowest cost); R+ Real image sequence (low); SSC Synthetic images with Scene Coordinates (low); RP Real images with Poses (high); or RSC Scene Coordinates (highest). The synthetic labeled images are rendered from a color point cloud of the same scene. Compared to PoseNet and its fully supervised counterpart (e), DANCE achieves similar performance with significantly lower data acquisition cost. Compared to UcoSLAM which has 9.7% invalid pose estimates, DANCE has better tail performance and does not require the real images to be in sequence. (e) is a componentwise training variant of DSAC [8]. All the baseline methods are retrained using our training data.

in terms of median error (Table 3.1(e)). We do not claim 0.14m median error of DANCE is better than 0.17m of the upper bound method due to the lack of statistical test (Table 3.1 (d) vs (e)). This indicates that the domain gap necessitates the application of domain adaptation techniques, and that our proposed training pipeline is effective at narrowing the domain gap between the rendered images X_S and real camera images X_T. It is worth noting that the fully supervised upper bound method (Table 3.1(e)) is a componentwise training variant of DSAC [8] (a SCR based framework) where the SCR network and PnP-RANSAC were trained in an end-to-end manner. DSAC [8] showed that end-to-end training can only provide marginal performance gain compared with componentwise training, which indicates the upper bound method is a strong baseline for comparison.

Compared with other existing methods, $DANCE$ achieves lower median errors than the fully supervised PoseNet [30] and comparable median errors to a fully supervised SCR based method [8] (Table 3.1(e)) with much lower deployment overhead. Specifically, PoseNet requires camera images with ground truth poses for training (28411 real images with ground truth poses in our experiments) while our pipeline only requires rendered images with ground truth scene coordinates (100k (X_S, Y_S)) and unlabeled camera images (28411 real images X_T without labels). The unlabeled camera images are only used to provide target domain information to train the mapping network $G_{S\rightarrow T}$ in our pipeline. Though PoseNet is not the state-of-the-art learning based method, the results demonstrate that our unsupervised pipeline is able to achieve better localization performance than a fully supervised approach. Furthermore, using DANCE to train an SCR network achieves comparable median performance to training the same SCR network using the harder to acquire fully supervised labels (Table 3.1(e)) thereby demonstrating that the DANCE pipeline can bridge the domain gap sufficiently to allow comparable performance to fully supervised methods.

In summary, as shown in Table 3.1 and Figure 3.4, the DANCE-trained SCR network can achieve performance comparable to fully supervised PoseNet, but without the need for tedious real pose labels.

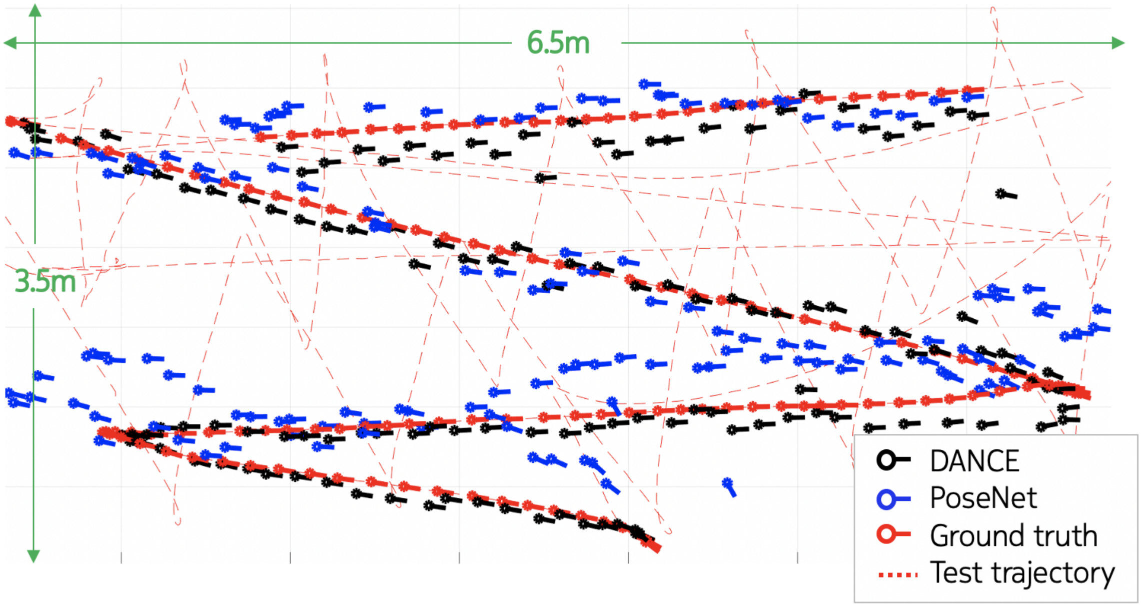

Figure 3.4: Top-down view of the room. The full test trajectory is shown in the background (dash red line), and pose estimates (solid arrows) on a subset of the trajectory are shown in the foreground for qualitative comparison. The solid arrows indicate the camera location and orientation. The statistics in Table 3.1 are w.r.t. the full test trajectory.

Comparison with feature-based relocalization We also compare the relocalization performance of our method with a state-of-the-art, feature-based SLAM system called UcoSLAM [50] (Table 3.1). This is a recent variant of the popular ORB-SLAM2 [49] with support for fiducial landmarks, map loading/saving, and a highly speed-optimized version of the DBoW2 [18] bag-of-words image matcher. We build the SLAM map with the same real image sequences we use in the training of our proposed method (two sequences with 28411 camera images). At test time, we enforce relocalization for each frame of the test sequence. We acknowledge this is not the ideal use case for a SLAM algorithm, but it makes a fair comparison with a single-frame localization method possible. While DANCEperforms slightly worse than UcoSLAM in terms of median errors, it is much more robust as it returns a valid pose for every frame while UcoSLAM sometimes fails to get valid results (25.7°, 0.95m 95%-tile error vs 205 frames out of 2104 testing frames are invalid, Table 3.1). It is also worth noting that SLAM requires a whole image sequence to build a 3D map while our proposed pipeline only requires unordered images to provide target domain information for training $G_{S\to T}$.

Other domain adaptation methods To investigate the efficacy of each domain adaptation component in our proposed pipeline (DANCE), we compare various architectural options in Table 3.1. The lower bound (a) error is 110°, 7.57m when training the SCR network on the synthetic images and testing on the real images. We then perform histogram matching (b) from rendered images to camera images and train the SCR network on these transformed images. Histogram matching improves the median error to 14.1°, 1.09m. Next, the image level adaptation network $G_{S\to T}$ is trained with different GAN frameworks (CycleGAN [92] vs CUT [52]) to map the rendered images after histogram matching to camera images (Table 3.1 (c) vs (d)). Our DANCE pipeline adopts the CUT framework to train the adaptation network $G_{S\to T}$, which was shown to have better unpaired image-to-image translation power than CycleGAN. By training the SCR network on these domain adapted labeled rendered images ($G^{*}_{S\to T}(X_S)$, Y_S), DANCECUT (Table 3.1(d)) outperforms DANCE-CycleGAN (Table 3.1(c)) by a significant margin. This indicates that a better unpaired image-to-image translation GAN model can further improve our DANCE pipeline.

Other 3D representations Besides a color point cloud (the 3D representation in DANCE pipeline) captured from a laser scan, we also tested our method in case the 3D representation is a SfM model of the space. While sparse SfM point clouds can be used for feature-based localization, generating scene coordinates requires a dense model. We performed dense reconstruction from our training images using Colmap [1], but the quality of the resulting 3D representation was poor with significant distortion and missing areas. We concluded that an SfM pipeline is not necessarily suitable for building the 3D representation of the scene in order to generate the labeled rendered images. In the future, better domain adaptation methods might be able to bridge such even larger domain gap. It is an interesting question what is the minimally required quality of a reconstruction for our domain adaptation technique to work, we leave this analysis for future work.

Other coordinate regression networks Note that our primary goal was to simplify the training process of pose (or scene coordinate) regression networks in general, in order to make this family of methods more accessible, and chose the fully supervised PoseNet as one of well-known baselines for comparison. We have shown to achieve results similar

[1] `https://colmap.github.io/`

to these fully supervised methods but only at a fraction of the cost. Since its original publication, several methods have improved on PoseNet, and we anticipate that swapping to a more powerful SCR method may improve performance. This is indeed possible in our general training framework and we see this as a key strength of our framework. There is no assumption on the pose estimation network (PoseNet or other), there is no assumption on the input 3D representation (point cloud), there is no strict assumption on the domain adaptation method used (we tested a CycleGAN like pipeline as well as CUT). Furthermore, while our general training framework could be applied to other, newer scene coordinate regression methods, we also expect that with better laser scanners, let alone better domain adaptation methods in the future, the accuracy could be even further improved.

3.6 Summary of Major Contributions

- We have shown that it is possible to train a neural network to perform the task of scene coordinate regression for monocular camera pose estimation on real images using only synthetic labeled images and a pool of unordered unlabeled photos. Our proposal achieves performance comparable with existing fully supervised techniques but with significantly lower overhead cost. These existing techniques require photos with ground truth camera pose labels, which are typically obtained using cumbersome motion capture systems that track markers mounted on the camera. For each room or environment, the motion capture system would need to be deployed to generate a new set of labeled training photos.

- In contrast to existing approaches, deploying DANCE in a new room is simpler, requiring (unlabeled) images along with a dense 3D representation of the room to generate synthetic labeled images. Our dense representation was captured with the push of a button using a tripod-mounted Leica BLK360 scanner. Alternatively, one could potentially use even cheaper capture systems such as recent iOS devices which come equipped with LiDAR. In general, we believe our DANCE pipeline will continue to benefit from both the rapid development of 3D capture techniques and more powerful unpaired image-to-image translation models.

- Furthermore, there is no reason why this pipeline cannot be applied in tasks beyond pose estimation, and modalities beyond images. Any task for which there is an abundance of unlabeled samples, and for which the construction of a crude simulation is easier than the direct gathering of labeled data should be amenable to this technique.

Chapter 4

Summary of the Thesis

In this thesis, we explore the deep learning application to two 3D vision tasks: (1) Volumetric HFU mouse embryo image analysis and (2) Camera pose estimation. In Chapter 1, we first introduce these two 3D vision problems and outline the structure of the thesis. Then in Chapter 2 and 3, we discuss and demonstrate the successful application of deep learning techniques to these two 3D vision tasks. Though we have summarized each project separately, there are still some common points which should be summarized together here:

- FCN architecture [42] is effective and efficient at performing dense predictions, which allows only one pass of the network to produce classification or regression results for every pixel of the input images. The use of FCN enables accurate and real-time BV and body segmentation as well as camera pose estimation.

- Deep learning methods are much more robust than traditional computer vision techniques. For the task of volumetric medical image analysis, the proposed deep learning method was shown to be much more accurate and robust than NGC-based framework [35]. For the task of camera pose estimation, the proposed deep learning pipeline was demonstrated to be more robust than UcoSLAM [50] in terms of 95%-tile error.

- Training data collection is the common concern when using deep learning models. For each of the 231 HFU volumes, manual BV and body segmentations were conducted by trained research assistants using commercial software (Amira, FEI, Hillsboro, Oregon, USA). This data collection procedure is time-consuming and tedious with around 10 minutes for each manual BV segmentation and around 30 minutes for each manual body segmentation. Later in order to ease the burden of collecting training data for camera pose estimation, we used a commercial laser scanner to capture a color point cloud, from which a large number of labeled synthetic images were generated to train the SCR network. However, the synthetic images and the real camera images have a domain gap, which will lead to severe model degradation during testing phase. Hence, a domain adaptation network was used to transform the synthetic images into real-looking images for bridging the domain gap.

- The use of prior or domain knowledge can make the deep learning models more effective and efficient. For BV and body segmentation, knowing that there is only one BV (or one body) in a single HFU volume, we can use the mass center of the BV (or the body) to perform ROI localization instead of using time-consuming sliding windows. Additionally, knowing the *En1* mutation phenotype mainly manifests itself in the BV region and some parts of the body, we used the BV and body segmentation maps to perform mutant classification achieving very high accuracy. In comparison, we failed to train a mutant classification network by using the original images, because the uninformative variations of the original images overwhelmed the subtle structural changes in the BV region. For camera pose estimation, DSAC [8] was shown to be more accurate and robust than PoseNet [30], because DSAC tried to make use of the prior knowledge that camera images are captured by projecting 3D scene object points onto the image planes. This is also why we built our proposed camera pose estimation pipeline based on DSAC.

- There is a current trend that deep learning techniques will become more and more dominant in different 3D vision tasks. Though deep learning techniques were only successfully applied to two 3D vision tasks in this thesis, we believe that deep learning could be more easily and effectively applied to other 3D vision tasks in the future. We also hope that this thesis can inspire more people to explore and advance other 3D vision tasks with the help of deep learning.

Bibliography

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving Rubik's cube with a robot hand, 2019.

[2] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

[3] Orlando Aristizábal, Jonathan Mamou, Jeffrey A Ketterling, and Daniel H Turnbull. High-throughput, high-frequency 3-d ultrasound for in utero analysis of embryonic mouse brain development. *Ultrasound in medicine & biology*, 39(12):2321–2332, 2013.

[4] Orlando Aristizabal, Daniel H Turnbull, Jeffrey A Ketterling, Yao Wang, Ziming Qiu, Tongda Xu, Hannah Goldman, and Jonathan Mamou. Scanner independent deep learning-based segmentation framework applied to mouse embryos. In *2020 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE, 2020.

[5] Vassileios Balntas, Shuda Li, and Victor Prisacariu. RelocNet: Continuous metric learning relocalisation using neural nets. In *European Conference on Computer Vision (ECCV)*, September 2018.

[6] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] E. Brachmann and C. Rother. Expert sample consensus applied to camera relocalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[8] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *CoRR*, 2020.

[9] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, V. A. Prisacariu, L. D. Stefano, and P. H. S. Torr. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10), Oct 2020.

[10] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. D. Stefano, and P. H. S. Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[11] Hao Chen, Qi Dou, Lequan Yu, and Pheng-Ann Heng. Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation. *arXiv preprint arXiv:1608.05895*, 2016.

[12] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal on Robotics Research*, 30(9), August 2011.

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] Mary E Dickinson, Ann M Flenniken, Xiao Ji, Lydia Teboul, Michael D Wong, Jacqueline K White, Terrence F Meehan, Wolfgang J Weninger, Henrik Westerberg, Hibret Adissu, et al. High-throughput discovery of novel developmental phenotypes. *Nature*, 537(7621):508–514, 2016.

[15] AE Dorr, Jason P Lerch, Shoshana Spring, N Kabani, and R Mark Henkelman. High resolution three-dimensional brain atlas using an average magnetic resonance image of 40 adult c57bl/6j mice. *Neuroimage*, 42(1):60–69, 2008.

[16] Renaud Dubé, Andrei Cramariuc, Daniel Dugas, Hannes Sommer, Marcin Dymczyk, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMap: Segment-based mapping and localization using data-driven descriptors. *The International Journal of Robotics Research*, 39(2-3), 2020.

[17] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[18] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), Oct 2012.

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *32nd International Conference on Machine Learning (ICML)*, 2015.

[20] Zakieh S. Hashemifar, Charuvahan Adhivarahan, Anand Balakrishnan, and Karthik Dantu. Augmenting visual SLAM with Wi-Fi sensing for indoor applications. *Autonomous Robots*, 43(8), Dec 2019.

[21] 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.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] R Mark Henkelman. Systems biology through mouse imaging centers: experience and new directions. *Annual review of biomedical engineering*, 12:143–166, 2010.

[25] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *35th International Conference on Machine Learning (ICML)*, 2018.

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[28] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture, 2020.

[29] T. Ishihara, K. M. Kitani, C. Asakawa, and M. Hirose. Deep radio-visual localization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018.

[30] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[31] Jeffrey A Ketterling, Orlando Aristizabal, Daniel H Turnbull, and Frederic L Lizzi. Design and fabrication of a 40-mhz annular array transducer. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 52(4):672–681, 2005.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[34] Jen-wei Kuo, Jonathan Mamou, Orlando Aristizábal, Xuan Zhao, Jeffrey A Ketterling, and Yao Wang. Nested graph cut for automatic segmentation of high-frequency ultrasound images of the mouse embryo. *IEEE transactions on medical imaging*, 35(2):427–441, 2015.

[35] Jen-wei Kuo, Ziming Qiu, Orlando Aristizabal, Jonathan Mamou, Daniel H Turnbull, Jeffrey Ketterling, and Yao Wang. Automatic body localization and brain ventricle segmentation in 3d high frequency ultrasound images of mouse embryos. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 635–639. IEEE, 2018.

[36] Jen-wei Kuo, Yao Wang, Orlando Aristizabal, Daniel H Turnbull, Jeffrey Ketterling, and Jonathan Mamou. Automatic mouse embryo brain ventricle segmentation, gestation stage estimation, and mutant detection from 3d 40-mhz ultrasound data. In *2015 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE, 2015.

[37] Ming Li, Chengjia Wang, Heye Zhang, and Guang Yang. Mv-ran: Multiview recurrent aggregation network for echocardiographic sequences segmentation and full cardiac cycle analysis. *Computers in biology and medicine*, 120:103728, 2020.

[38] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] Zhigang Li, Yinlin Hu, Mathieu Salzmann, and Xiangyang Ji. Robust RGB-based 6-DoF pose estimation without real pose annotations, 2020.

[40] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[41] Yongkai Liu, Guang Yang, Sohrab Afshari Mirak, Melina Hosseiny, Afshin Azadikhah, Xinran Zhong, Robert E Reiter, Yeejin Lee, Steven S Raman, and Kyunghyun Sung. Automatic prostate zonal segmentation using fully convolutional network with feature pyramid attention. *IEEE Access*, 7:163626–163632, 2019.

[42] 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.

[43] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *International Journal of Robotics Research*, 39(9), 2020.

[44] MA Martínez-Martínez, Jesus Pacheco-Torres, Victor Borrell, and Santiago Canals. Phenotyping the central nervous system of the embryonic mouse by magnetic resonance microscopy. *NeuroImage*, 97:95–106, 2014.

[45] Tomohiro Mashita, Alexander Plopski, Akira Kudo, Tobias Höllerer, Kiyoshi Kiyokawa, and Haruo Takemura. Simulation based camera localization under a variable lighting environment. 2016.

[46] Terry M Mayhew and John M Lucocq. From gross anatomy to the nanomorphome: stereological tools provide a paradigm for advancing research in quantitative morphomics. *Journal of anatomy*, 226(4):309–321, 2015.

[47] Fausto Milletari, Seyed-Ahmad Ahmadi, Christine Kroll, Annika Plate, Verena Rozanski, Juliana Maiostre, Johannes Levin, Olaf Dietrich, Birgit Ertl-Wagner, Kai Bötzel, et al. Hough-cnn: deep learning for segmentation of deep brain regions in mri and ultrasound. *Computer Vision and Image Understanding*, 164:92–102, 2017.

[48] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[49] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), Oct 2017.

[50] Rafael Muñoz-Salinas and R. Medina-Carnicer. Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recognition*, 101, 2020.

[51] Brian J Nieman, Matthijs C van Eede, Shoshana Spring, Jun Dazai, R Mark Henkelman, and Jason P Lerch. Mri to assess neurological function. *Current protocols in mouse biology*, 8(2):e44, 2018.

[52] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision (ECCV)*, pages 319–345. Springer, 2020.

[53] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[54] Ziming Qiu, Jack Langerman, Nitin Nair, Orlando Aristizabal, Jonathan Mamou, Daniel H Turnbull, Jeffrey Ketterling, and Yao Wang. Deep bv: A fully automated system for brain ventricle localization and segmentation in 3d ultrasound images of embryonic mice. In *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6. IEEE, 2018.

[55] Ziming Qiu, Nitin Nair, Jack Langerman, Orlando Aristizabal, Jonathan Mamou, Daniel H Turnbull, Jeffrey A Ketterling, and Yao Wang. Automatic mouse embryo brain ventricle & body segmentation and mutant classification from ultrasound data using deep learning. In *2019 IEEE International Ultrasonics Symposium (IUS)*, pages 12–15. IEEE, 2019.

[56] N. Radwan, A. Valada, and W. Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4), Oct 2018.

[57] 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.

[58] Holger R Roth, Hirohisa Oda, Xiangrong Zhou, Natsuki Shimizu, Ying Yang, Yuichiro Hayashi, Masahiro Oda, Michitaka Fujiwara, Kazunari Misawa, and Kensaku Mori. An application of cascaded 3d fully convolutional networks for medical image segmentation. *Computerized Medical Imaging and Graphics*, 66:90–99, 2018.

[59] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[60] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021.

[61] T. Sattler, B. Leibe, and L. Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9), Sep. 2017.

[62] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixé. Understanding the limitations of cnn-based absolute camera pose regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[63] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[64] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[65] Sota Shoman, Tomohiro Mashita, Alexander Plopski, Photchara Ratsamee, Yuki Uranishi, and Haruo Takemura. REST: Real-to-synthetic transform for illumination invariant camera localization, 2018.

[66] 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 *IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[67] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. SIFT-realistic rendering. In *International Conference on 3D Vision (3DV)*, June 2013.

[68] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[70] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[71] J. Straub, S. Hilsenbeck, G. Schroth, R. Huitl, A. Möller, and E. Steinbach. Fast relocalization for visual odometry using binary features. In *IEEE International Conference on Image Processing (ICIP)*, 2013.

[72] Kamila U Szulc, Jason P Lerch, Brian J Nieman, Benjamin B Bartelle, Miriam Friedel, Giselle A Suero-Abreu, Charles Watson, Alexandra L Joyner, and Daniel H Turnbull. 4d memri atlas of neonatal fvb/n mouse brain development. *Neuroimage*, 118:49–62, 2015.

[73] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4), 2021.

[74] Yujiao Tang, Feng Yang, Shaofeng Yuan, et al. A multi-stage framework with context information fusion structure for skin lesion segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1407–1410. IEEE, 2019.

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

[76] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[77] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[78] J. Valentin, M. Nießner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[79] O. Vysotska and C. Stachniss. Effective visual place recognition using multi-sequence maps. *IEEE Robotics and Automation Letters*, 4(2), April 2019.

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

[81] Hongyu Wang, Haozhe Jia, Le Lu, and Yong Xia. Thorax-net: an attention regularized deep neural network for classification of thoracic diseases on chest radiography. *IEEE journal of biomedical and health informatics*, 24(2):475–485, 2019.

[82] Yongfeng Wang, Wenwen Yue, Xiaolong Li, Shuyu Liu, Lehang Guo, Huixiong Xu, Heye Zhang, and Guang Yang. Comparison study of radiomics and deep learning-based methods for thyroid nodules classification using ultrasound images. *Ieee Access*, 8:52010–52017, 2020.

[83] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se(3)-TrackNet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, 10/2020 2020.

[84] Michael D Wong, Yoshiro Maezawa, Jason P Lerch, and R Mark Henkelman. Automated pipeline for anatomical phenotyping of mouse embryos using micro-ct. *Development*, 141(12):2533–2541, 2014.

[85] Michael D Wong, Matthijs C van Eede, Shoshana Spring, Stefan Jevtic, Julia C Boughner, Jason P Lerch, and R Mark Henkelman. 4d atlas of the mouse embryo for precise morphological staging. *Development*, 142(20):3583–3591, 2015.

[86] Wolfgang Wurst, Anna B Auerbach, and Alexandra L Joyner. Multiple developmental defects in engrailed-1 mutant mice: an early mid-hindbrain deletion and patterning defects in forelimbs and sternum. *Development*, 120(7):2065–2075, 1994.

[87] Tongda Xu, Ziming Qiu, William Das, Chuiyu Wang, Jack Langerman, Nitin Nair, Orlando Aristizábal, Jonathan Mamou, Daniel H Turnbull, Jeffrey A Ketterling, et al. Deep mouse: An end-to-end auto-context refinement framework for brain ventricle & body segmentation in embryonic mice ultrasound volumes. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 122–126. IEEE, 2020.

[88] Guang Yang, Jun Chen, Zhifan Gao, Shuo Li, Hao Ni, Elsa Angelini, Tom Wong, Raad Mohiaddin, Eva Nyktari, Ricardo Wage, et al. Simultaneous left atrium anatomy and scar segmentations via deep learning in multiview information with attention. *Future Generation Computer Systems*, 107:215–228, 2020.

[89] Paul A Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[90] B. Zeisl, T. Sattler, and M. Pollefeys. Camera pose voting for large-scale image-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015.

[91] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[92] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[93] Wei Zhu, Haofu Liao, Wenbin Li, Weijian Li, and Jiebo Luo. Alleviating the incompatibility between cross entropy loss and episode training for few-shot skin disease classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 330–339. Springer, 2020.

Publication List

Ziming Qiu, Tongda Xu, Jack Langerman, William Das, Chuiyu Wang, Nitin Nair, Orlando Aristizábal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey A. Ketterling and Yao Wang, *“A Deep Learning Approach for Segmentation, Classification, and Visualization of 3-D High-Frequency Ultrasound Images of Mouse Embryos”,* in IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 68, pp. 2460-2471, Mar. 2021, doi: 10.1109/TUFFC.2021.3068156.

Jack Langerman*, **Ziming Qiu***, Gábor Sörös, Dávid Sebők, Yao Wang, Howard Huang, *”Domain Adaptation of Networks for Camera Pose Estimation: Learning Camera Pose Estimation Without Pose Labels,”* arXiv preprint arXiv:2111.14741 (2021), * equal contribution.

Tongda Xu*, **Ziming Qiu***, William Das, Chuiyu Wang, Jack Langerman, Nitin Nair, Orlando Aristizabal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey A. Ketterling, Yao Wang, *“Deep Mouse: An End-to-end Auto-context Refinement Framework for Brain Ventricle Body Segmentation in Embryonic Mice Ultrasound Volumes,”* in 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI 2020), * equal contribution.

Ziming Qiu, Nitin Nair, Jack Langerman, Orlando Aristizable, Jonathan Mamou, Daniel H. Turnbull, Jeffrey A. Ketterling, Yao Wang, *“Automatic Mouse Embryo Brain Ventricle Body Segmentation and Mutant Classification from Ultrasound Data Using Deep Learning,”* in IEEE International Ultrasonics Symposium (IUS), 2019.

Ziming Qiu, Jack Langerman, Nitin Nair, Orlando Aristizabal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey Ketterling, Yao Wang, *“Deep BV: A Fully Automated System for Brain Ventricle Localization and Segmentation in 3D Ultrasound Images of Embryonic Mice,”* IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2018, accepted, arXiv preview link.

Jen-wei Kuo, **Ziming Qiu**, Orlando Aristizbal, Jonathan Mamou, Daniel H. Turnbull, Jeffrey Ketterling, and Yao Wang, *”Automatic Body Localization and Brain Ventricle Segmentation in 3D High Frequency Ultrasound Images of Mouse Embryos,”* 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 635-639.

Orlando Aristizabal, Daniel H. Turnbull, Jeffrey A. Ketterling, Yao Wang, **Ziming Qiu**, Tongda Xu, Hannah Goldman, and Jonathan Mamou. *”Scanner Independent Deep*

Learning-Based Segmentation Framework Applied to Mouse Embryos," In 2020 IEEE International Ultrasonics Symposium (IUS), pp. 1-4. IEEE, 2020.