ABSTRACT

Title of dissertation: DEEP INFERENCE ON MULTI-SENSOR DATA

Arthita Ghosh
Doctor of Philosophy, 2019

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Computer vision-based intelligent autonomous systems engage various types of sensors to perceive the world they navigate in. Vision systems perceive their environments through inferences on entities (structures, humans) and their attributes (pose, shape, materials) that are sensed using RGB and Near-InfraRed (NIR) cameras, LAser Detection And Ranging (LADAR), radar and so on. This leads to challenging and interesting problems in efficient data-capture, feature extraction, and attribute estimation, not only for RGB but various other sensors. In some cases we encounter very limited amounts of labeled training data. In certain other scenarios we have sufficient data but annotations are unavailable for supervised learning. This dissertation explores two approaches to learning under the conditions of minimal to no ground truth. The first approach applies projections on training data that make learning efficient by improving training dynamics. The first and second topics in this dissertation belong to this category. The second approach makes learning without ground-truth possible via knowledge transfer from a labeled source domain to an unlabeled target domain through projections to domain-invariant shared latent spaces. The third and fourth topics in this dissertation belong to this category.

For the first topic, we study the feasibility and efficacy of identifying shapes in LADAR data in several measurement modes. We present results on efficient parameter learning with less data (for both traditional machine learning as well as deep models) on LADAR images. We use a LADAR apparatus to obtain range information from a 3-D scene by emitting laser beams and collecting the reflected rays from target objects in the region of interest. The Agile Beam LADAR concept makes the measurement and interpretation process more efficient using a software-defined architecture that leverages computational imaging principles. Using these techniques, we show that object identification and scene understanding can be accurately performed in the LADAR measurement domain thereby rendering the efforts of pixel-based scene reconstruction superfluous.

Next, we explore the effectiveness of deep features extracted by Convolutional Neural Networks (CNNs) in the Discrete Cosine Transform (DCT) domain for various image classification tasks such as pedestrian and face detection, material identification and object recognition. We perform the DCT operation on the feature maps generated by convolutional layers in CNNs. We compare the performance of the same network with the same hyper-parameters with or without the DCT step. Our results indicate that a DCT operation incorporated into the network after the first convolution layer can have certain advantages such as convergence over fewer training epochs and sparser weight matrices that are more conducive to pruning and hashing techniques.

Next, we present an adversarial deep domain adaptation (ADA)-based approach for training deep neural networks that fit 3D meshes on humans in monocular RGB input images. Estimating a 3D mesh from a 2D image is helpful in harvesting complete 3D information about body pose and shape. However learning such an estimation task in a supervised way is challenging owing to the fact that ground truth 3D mesh parameters for real humans do not exist. We propose a framework based on domain adaptation for single-shot (no re-projection , no iterative refinement), end-to-end training with joint optimization on real and synthetic images on a shared common task. Through joint inference on real and synthetic data, the network extracts domain invariant features that are further used to estimate the 3D mesh parameters in a single shot with no supervision on real samples. While we compute regression loss on synthetic samples with ground truth mesh parameters, knowledge is transferred from synthetic to real data through ADA without direct ground truth for supervision.

Finally, we propose a partially supervised method for satellite image super-resolution by learning a unified representation of samples from different domains (captured by different sensors) in a shared latent space. The training samples are drawn from two datasets which we refer to as source and target domains. The source domain consists of fewer samples which are of higher resolution and contain very detailed and accurate annotations. In contrast, samples from the target domain are low-resolution and available ground truth is sparse. The pipeline consists of a feature extractor and a super-resolving module which are trained end-to-end. Using a deep feature extractor we jointly learn (on two datasets) a common embedding space for all samples. Partial supervision is available for the samples in the source domain which have high-resolution ground truth. Adversarial supervision is used to successfully super-resolve low-resolution RGB satellite imagery from target domain without direct paired supervision from high resolution counterparts.

DEEP INFERENCE ON MULTI-SENSOR DATA

by

Arthita Ghosh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Min Wu
Dr. Yaser Yacoob
Professor Vishal M. Patel
Professor Ramani Duraiswami

In loving memory of

Anil Kumar Mukherjee

and

Maya Mukherjee

Acknowledgments

First and foremost I would like to thank my advisor, Professor Rama Chellappa, for the opportunity and for his relentless support and guidance throughout this program. I had access to the wealth of his wisdom and foresight as well as the freedom to pursue my own ideas. I worked with an excellent research team and received all the computational resources necessary to support my research. His all-encompassing knowledge of computer vision combined with the unmatched efficacy of his mentorship over the years has helped create new research directions and blossom numerous careers. It is an absolute privilege to belong to his academic lineage.

I am grateful to Prof. Ramani Duraiswami, Prof. Min Wu, Prof. Vishal M. Patel and Dr. Yaser Yacoob for serving on my committee and for their insightful feedback on my dissertation.

Throughout my time in graduate school, I received valuable advice and mentorship from several individuals including Prof. Min Wu, Dr. Jun-Cheng Chen, Dr. Maya Kabkab, Dr. Kota Hara and Dr. Raviteja Vemulapalli, Dr. Xavier Gibert and Dr. Jie Ni. Major parts of my thesis are based on collaborative projects advised by Prof. Larry Davis, Dr. Yaser Yacoob and Prof. Vishal M. Patel. I am grateful for the support and feedback from my friends and fellow graduate students Dr. Rajeev Ranjan, Dr. Hongyu Zhu, Dr. Upal Mahbub, Dr. Pouya Samangouei, Dr. Sohil Shah, Hui Ding, Amit Kumar, Pirazh Khorramshahi, Ankan Bansal, Ilya Kavalerov, Joshua Gleason, Sunandita Patra, Sayantan Sarkar, Ananya Mukherjee and Rajdeep Talapatra. I am thankful for the administrative support I received during my program from Ms. Melanie Prange, Ms. Janice Perone, Ms. Arlene Schenk, Ms. Maria Hoo, Ms. Emily Irwin, Ms. Vivian Lu and the technical support from the UMIACS team.

This journey would not have been possible without the endless support and patience of my entire family. I am thankful to my grandparents for their unwavering belief in me and to Amit, for all the constructive criticism which has always pushed me to grow. I am grateful to my father for teaching me essential life skills and for his unconditional love and sacrifices. Most of all, I owe my gratitude to my mother for giving me my dreams and being the cornerstone of my career.

The work presented in this thesis was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center and the Office of the Director of National Intelligence (ODNI). The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Autonomous perception systems have become ubiquitous in our lives in recent years. These systems often use various types of sensors for vision-based perception of their environments. Perception is enabled through inferences on entities contained in the scene (structures, objects, humans) and their attributes (material, pose, shape). Among a wide variety of sensors used to capture the scene, some are active (*e.g.* LADAR, radar), while the others are passive (RGB camera, multi-spectral cameras aboard satellites). Active sensors such as LADAR are capable of capturing data using a host of different measurement modes which belong to different domains. Most state-of-the-art deep learning-based methods in computer vision are trained on large scale, fully annotated datasets mainly consisting of RGB (or binary) images. Training models for recognition, segmentation, classification, reconstruction and estimation have been studied particularly well in natural scenes captured in the RGB domain. However, applications on other types of data, such as remotely sensed imagery and range data, have been more limited. These alternate domains present novel challenges in efficient capture and feature extraction for various vision tasks. Absence of large, diverse, and annotated datasets (such as in remote sensing imagery) call for training strategies that require less annotations. Certain tasks, such as 3D pose and shape estimation require capturing ground truth using specialized sensors (such as MOCAP sensors, scanners etc.) which lead to benchmark datasets with very little diversity (in subjects, background). In the first part of this dissertation we study data projections to domains that lead to efficient learning (with less training data and less training time). The second part of the dissertation addresses cases where a domain has no annotated data available for training models for a particular task. In such scenarios, we utilize adversarial learning-based domain adaptation for knowledge transfer from a labeled source domain to effectively solve a task in an unlabeled target domain. The following sections outline the topics studied in this dissertation.

1.2 Computational LADAR Imaging

A LAser Detection And Ranging (LADAR) sensor obtains range information from a three dimensional scene by emitting laser beams and collecting the reflected rays from target objects in the region of interest. Most LADAR (Laser Radar, LIDAR) imaging systems use pixel-basis sampling, where each azimuth and elevation resolution element is uniquely sampled and recorded. In Chapter 2 we first discuss the design of an Agile Beam LADAR architecture that uses alternative sampling basis to make the measurement and interpretation process more efficient. This yields considerable energy savings and proves valuable when used on platforms with severe limitations on sensor size, weight and power. We apply multiple well known machine learning algorithms to train classifiers on simulated LADAR measurements in various alternative domains to identify scene objects. Using the results of these classifiers in randomized experiments of training sets of different sizes, we show that, the process of object identification and scene understanding can be performed accurately and efficiently in the LADAR measurement domain thereby rendering the efforts of pixel-based scene reconstruction superfluous.

1.3 Deep Feature Extraction in the DCT domain

As another case of projection-based learning efficiency, we study the effect of training deep networks on DCT coefficients. Convolutional Neural Networks (CNNs) are being widely used on 2D images for various types of computer vision tasks. The major difference between LADAR measurements and 2D images is data sparsity. In LADAR measurements points are spread out in 3D space and the object of interest is separated relatively easily from the background using depth. This is not the case in 2D images where the depth information is missing. In Chapter 3, we perform object identification in an alternate domain derived from 2D images and observe that learning is more efficient. We extract deep features from 2D images in the DCT domain for various computer vision tasks (such as pedestrian and face detection, material classification and object recognition). We apply the DCT operation on feature maps extracted by the convolutional layers in CNNs. We compare the performance of the same network, on the same datasets, with the same hyper-parameters, with or without the DCT step. Our results indicate that a DCT operation incorporated into the network after the first convolution has certain advantages such as convergence over fewer training epochs and sparser weight matrices that are more conducive to pruning or hashing-based model compression.

1.4 Single shot 3D mesh estimation via Adversarial Domain Adaptation

Existing datasets consisting of in-the-wild images of humans have limited availability of 3D groundtruth. While 3D keypoint ground truth is available to some extent, body shape data (obtainable using scanners) is very rare. We design an adversarial deep domain adaptation (ADA)-based approach for training deep neural networks that estimate 3D pose and shape of a human from a single image. We propose a novel deep architecture for 3D pose estimation and leverage the variation in pose, body shape and background in the synthetic datasets to train our network. Using ADA we adapt our network to real human images. We design a pipeline for joint 3D pose and shape estimation via mesh fitting. Estimating a mesh from a 2D input generates complete 3D information on body pose and shape. Learning mesh estimation in a supervised way is challenging owing to the fact that ground truth mesh parameters for real humans do not exist. Almost all existing works estimate these parameters by minimizing the re-projection error on keypoints. The re-projection step computes keypoints from the inferred mesh. Additionally, some researchers use self-supervision using segmentation mask, semantic body parts segmentation, motion from optical flow, etc. In contrast, we propose an ADA-based single shot, end-to-end training approach via joint optimization on real and synthetic images. Our method involves no re-projection and no iterative refinement. Through joint training on real and synthetic data, our network extracts domain invariant features that are further used to estimate the 3D mesh parameters in a single shot with no supervision on real samples. We compute the regression loss on synthetic samples with ground truth mesh parameters. Knowledge is transferred from synthetic to real data through ADA without direct keypoint-based supervision.

1.5 Multi-sensor satellite image super-resolution via adversarial domain adaptation

Satellite image super-resolution (in the absence of high resolution ground truth) can improve detection, segmentation, classification and 3D shape inference in low resolution imagery via sub-pixel information estimation. We propose a semi-supervised method for satellite image super-resolution by learning a unified representation of samples from different domains (captured by RGB sensors under very different conditions) in a shared latent space. The training samples are drawn from two datasets which we refer to as source and target domains. The source domain consists of fewer samples that are high resolution and contain very detailed and accurate annotations. In contrast, samples from the target domain are low-resolution and available ground truth is sparse. The super-resolving pipeline consists of deep convolutional neural networks that act as feature extractor and super-resolving module. The networks are trained end-to-end. Using a deep feature extractor, we jointly learn (on two datasets), a common embedding space for all samples. Partial supervision is available for the samples in the source domain using high-resolution ground truth. Adversarial supervision is used to successfully super-resolve low-resolution RGB satellite imagery from the target domain without direct paired supervision from high-resolution counterparts.

1.6 Contributions

In this dissertation, we address several well known computer vision problems in alternate domains. We present a projection-based efficient method for scene recovery from LADAR measurements and classifier training with less data. We demonstrate that deep feature extraction in the DCT domain improves model convergence time on RGB images. We study scenarios with an abundance of training data but no annotations for a specific task. We design modular pipelines for 3D body model estimation and satellite image super-resolution. We learn non-linear projections to a shared latent space via joint optimization on two domains. Using an ADA-based pipeline we enable knowledge transfer from a labeled source to solve the task in the unlabeled target domain.

1.7 Dissertation Organization

The rest of the dissertation is organized as follows. In chapter 2, we present a method for efficient data capture and interpretation in alternative measurement domains using LADARs. In chapter 3, we delve further into deep feature extraction in the DCT domain for several computer vision tasks. In chapter 4, we present a pipeline for 3D pose and shape estimation of humans in 2D input images. In chapter 5, we present a method for super-resolution of satellite imagery in the absence of direct supervision from high resolution data. Finally, in chapter 6 we conclude the dissertation with a brief summary and future directions.

Chapter 2: Computational LADAR Imaging

LADAR, also known as laser radar or LIDAR, is used in a variety of applications that require accurate, reliable, and fast three dimensional imaging of diverse objects at long standoff ranges in a wide range of environmental conditions [4]. Compared to radio frequency radar, LADAR has much higher cross-range resolution for a given aperture size and mostly maintains the image interpretability of visible wavelength electro-optics vision systems. LADAR has been and is increasingly a popular sensor for autonomous vehicles for both on-road and off-road use. Additionally, it has proven essential for mapping and geospatial imaging applications [5].

The primary drawbacks of LADAR systems with respect to passive imaging alternatives (e.g., a thermal infrared camera) are the size, weight, power and cost of the laser, scanning optics, and high-speed photoreceiver system. The laser power required for long range operation is especially significant. A laser with higher power output negatively impacts the size, weight, and power of the system while also emitting more power, which is undesirable.

For autonomous ground vehicles, the purpose of the perception sensor system is to generate a "world model", i.e., a sparse geometric understanding of the surrounding environment [6]. Recent research in compressed sensing presents a compelling opportunity to significantly reduce laser power while maintaining data acquisition performance at substantially the same level. At the same time, much of the measurements produced by a LADAR are not meaningful, i.e., measuring the scene with fixed scanning at a constant resolution or revisit rate means that considerable resources are wasted on irrelevant fields of view or unchanged objects. Ideally, an active sensor system such as LADAR would use laser emission and scanning sparingly and configure it in the way so that the information content per unit laser power emission is maximized. We refer to systems with this ability as “agile-beam” LADAR. They are able to control the illumination wavefront on-the-fly and under algorithmic control beyond the constraints of spot, line, or “flash” illumination. An agile-beam LADAR adjusts parameters like resolution and measurement basis to achieve some system performance objective in response to the environment, operating condition, or prior knowledge.

Sampling and post-detection processing algorithms are at the core of this concept; it is through the agile-beam optical hardware that the full potential of advanced sampling and computational imaging approaches are realized. Instances of such technologies include the Rice single-pixel camera [7] which leverages compressed sensing (CS) concepts for measurement efficiency and also supports cost effective hardware architectures. Some of the other compressive LADAR systems include [8], [9], [10]. In this chapter we present the demonstration of three key concepts for significantly improved performance: the piece-wise smooth recovery of three dimensional images sampled in non-pixel basis to reduce speckle noise, CS and robust recovery with 50% reduction in laser power, and measurement-domain object recognition. The approach that uses the principles of CS to reduce laser power by half and performs object recognition in a low-dimensional computational space rather than a high-dimensional pixel space, links CS to the system objective in robotics perception. We show that we can make more efficient use of laser power in the context of robotic perception to identify objects that would populate a world model. Since there is often no need for robotic perception data to be presented in a form that is interpretable by human vision, reconstruction to a pixel basis is relegated to an optional process. We demonstrate that the pixel-basis scanning used by substantially all LADAR systems today, including spot, line, and flash techniques, is an inefficient approach. In the applications we target, pixel-basis measurements are mostly, and now unjustifiably, constrained by isomorphism in traditional optical imaging systems and human interpretability.

An important feature of the Agile Beam LADAR concept is the incorporation of computational approach both in the measurement apparatus (implemented by analog optical operation of lenses and focal planes) as well as the subsequent interpretation process (performed on digital electronic computers). This software-based architecture combining the measurement and interpretation stages has led to opportunities for more efficient measurement and interpretation techniques than simply in the pixel basis rigidly defined in optical hardware.

We demonstrate the concept with a surrogate agile-beam system using commercially available liquid crystal spatial light modulators (SLM) in lieu of bona fide optical phased arrays (OPA). Recent progress in OPAs points the way toward chip-scale LADAR devices with significantly reduced size and weight [11]. Although some power reduction is inherently achieved by the elimination of mechanical beam scanning, laser power reduction in a chip scale system is vital to manage thermal loads without the system size and weight being dominated by cooling and thermal management components [12].

In Section 2.1 we describe the agile-beam architecture and the surrogate apparatus constructed to demonstrate the approach experimentally. In Section 2.2 we discuss the observation model and image recovery algorithm. In Section 2.3, we present the idea of LADAR measurement domain object recognition for sidestepping pixel-basis image reconstruction. In Section 2.4, we demonstrate experimental results for image recovery and recognition tasks in measurement domains. Finally, section 2.5 concludes the chapter with a brief summary and discussion.

2.1 Agile Beam LADAR Architecture

2.1.1 Apparatus

The prototype apparatus [1] constructed to support experiments is shown in Figure 2.1. The transmit beam entering the figure from the upper right hand corner is generated by a erbium-doped polarization-maintaining fiber laser at around 1550 nm center wavelength. The laser is pulse modulated with an approximately 2-ns FWHM pulse width. The linearly polarized emission is collimated using a multi-element lens, which is not shown in the figure. The beam reflects from a fixed folding mirror to laterally align the beam to the SLM at a small incident angle to reduce cross-talk and scattering between adjacent SLM elements.

The SLM is a rectangular array of 512 by 512 phase elements capable of at least 2π

[1] Built by Michael A. Powers at the U.S. Army Research Laboratory, 2800 Powder Mill Road, Adelphi, Maryland 20783,USA

phase shift in one polarization at 1550 nm. Computer-generated holograms (CGH) are generated and stored prior to data acquisition. The CGHs are phase-only and generated with the Gerchberg-Saxton algorithm [13]. Each far-field projection generates a column of the measurement matrices described in Section 2B.

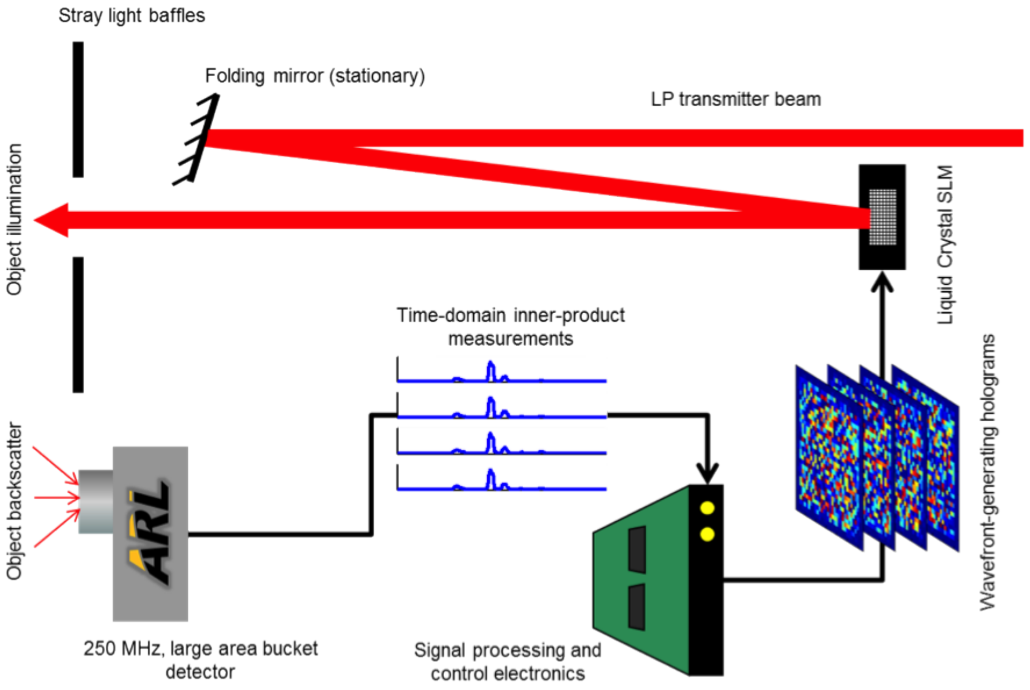

Stray light baffles
Folding mirror (stationary)
LP transmitter beam
Object illumination
Liquid Crystal SLM
Time-domain inner-product measurements
Wavefront-generating holograms
Object backscatter
ARL
250 MHz, large area bucket detector
Signal processing and control electronics

Figure 2.1: Block diagram of a prototype agile-beam LADAR architecture.

The beam, now with transverse phase modulation, reflects from the SLM and passes through baffles intended to block stray light. Only the +1 diffractive order is desired. Other orders, including the residual zeroth order, are blocked by the baffles. The desired diffractive order, containing substantially all of the diffraction, illuminates a distant target which is about 3 meters from the aperture.

The patterned illumination of the object in the far-field scatters diffusely, ideally as a Lambertian distribution. A photoreceiver with an instantaneous field of view that subtends the entire angle of illumination is located near the transmitter aperture. The backscatter emanating from the entire illuminated area is integrated and collected by the so-called ”bucket detector” photoreceiver [14]. The photocurrent, which is proportional to the inner product of the illumination pattern and spatially-distributed object reflectivity, feeds a 5 GSa/s 8-bit digitizer. A synchronization pulse from the laser is used to trigger the digitizer and maintain a constant t_0 time-of-flight measurement reference.

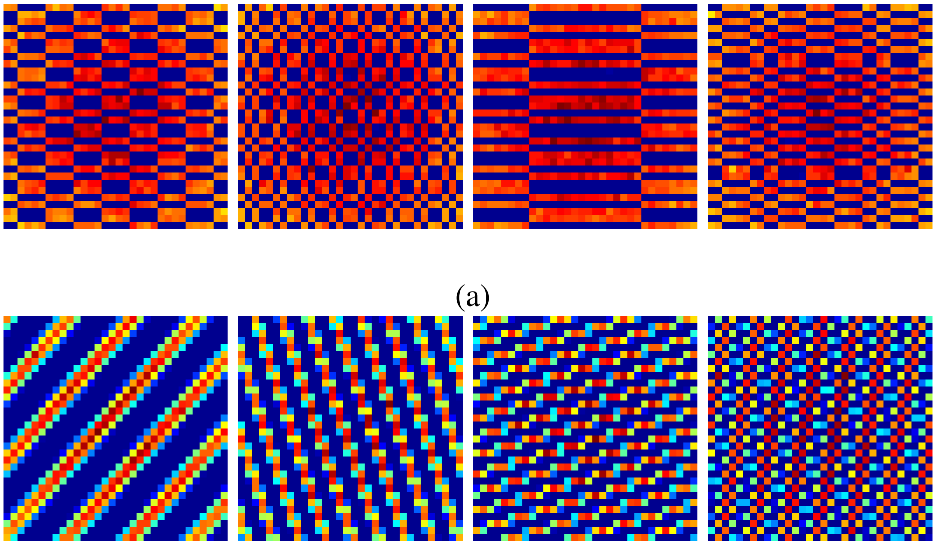
This apparatus serves as a surrogate for a true optical phased-array system in three ways. First, the liquid crystal SLM is a passive device that is reflective and must be illuminated with a beam at near-normal incidence. This significantly increases the size from the planar form factor of a phased array. Second, SLM devices are often, and will likely continue to be, much slower than the measurement rate requirements for imaging from or of moving platforms. We tolerate slow switching speeds, and therefore slow measurement rates, in a laboratory environment with static scenery. Finally, the SLM is a phase-only device. Although phase-only modulation is suitable, an optical phase array with amplitude and phase control would be superior.

2.1.2 Image Measurements

The apparatus described in Section 2B records the inner product of a measurement function and the object’s backscatter in the direction of the receiver. The pixel-basis imaging of most camera systems and most LADAR systems is also an inner product measurement where the measurement function is the identity matrix. The alternative basis functions we use in these experiments generalize this concept to other orthonormal measurement matrices, and require a structured multipoint, rather than a single point, illumination of the scene. Three dimensional images are enabled by time domain sampling of the backscatter, i.e., the inner product measurements resolved in time between the three dimensional scene and pulse-modulated illumination.

The raw data record is in the form of an M-by-T matrix, where M is the number of measured basis functions and T is the number of time-domain samples. Accompanying the measurements is an N-by-N measurement basis, where N is the number of pixels in the projected illumination. Simple inversion where the identity matrix is the measurement matrix and M = N would generate an N-by-T matrix to be reordered into an n-by-n-by-T voxel data cube. For the experiments described in this chapter, 32-by-32-by-301 image data cubes were generated.

Four measurement modes, each having a measurement matrix, were used in the demonstration described here. The first mode is raster scanning implemented by the identity matrix. This is equivalent to the sampling of many mechanically scanned LADAR systems or a serialization of “flash” LADAR measurements. The pattern generating CGHs in this case are Fresnel prism phase profiles. Each of the 1024 CGHs define one azimuth and elevation beam deflection angle in the 32-by-32 cross-range sampling grid. The second mode samples the image with the discrete cosine basis, whereby the far-field illumination patterns are reordered rows of the discrete cosine transform (DCT) matrix. Given that the measurements produced with this method are inner products of the measurement function, in this case the DCT basis, the mode is quite similar to capturing a JPEG-encoded image in the physical domain, rather than transformation after measurement which would occur in modern digital cameras. The third mode uses the Hadamard transform matrix and is used in a similar fashion to the DCT method. Finally, the fourth mode uses the Bernoulli matrix. A few projected patterns from far-field intensity simulations are shown in Figure 2.2. Each of the 1024 patterns in any sampling mode generates a 301 sample time domain waveform, recording the backscatter range profile for a given illumination pattern. Because the measurement matrices in these cases have full rank, inverting the measurement matrix transforms the measurements to the pixel basis. Reordering the result from a 1-by-N to an n-by-n is necessary for proper display as a two dimensional image in each range slice.



(a)

(b)

Figure 2.2: Example far-field illumination patterns for (a) Hadamard or (b) Discrete Cosine measurement bases.

2.2 Observation Model and Image Recovery

Assuming that the range of interest is known a priori, the measurement process can be written as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^N$, $N = n^2$ is the cross-range element, $\mathbf{A} \in M \times N$ with $M < N$ is the measurement matrix when CS used and $\mathbf{b}$ is the additive measurement noise. Furthermore, assuming that the range slice is a superposition of two different signals, (2.1) can be rewritten as[2]

$$\mathbf{y} = \mathbf{A}(\mathbf{x}_p + \mathbf{x}_t) + \mathbf{b} = \mathbf{A}\mathbf{x}_p + \mathbf{A}\mathbf{x}_t + \mathbf{b}, \tag{2.2}$$

where $\mathbf{x}_p$ and $\mathbf{x}_t$ are the piecewise smooth component and texture component of $\mathbf{x}$, respectively. We further assume that $\mathbf{x}_p$ is sparse in an $N \times N_p$ dictionary represented in matrix as $\mathbf{D}_p$, and similarly, $\mathbf{x}_t$ is sparse in an $N \times N_t$ dictionary represented in matrix form as $\mathbf{D}_t$ so that $\mathbf{x}_p = \mathbf{D}_p\boldsymbol{\alpha}_p$ and $\mathbf{x}_t = \mathbf{D}_t\boldsymbol{\alpha}_t$ for the piecewise smooth and the texture component, respectively. The sizes N_p and N_t are typically much larger than N. The texture dictionary $\mathbf{D}_t$ should contain atoms that are oscillatory in nature such as those found in the discrete cosine/sine transform and the Gabor transform. The dictionary $\mathbf{D}_p$ should be able to process images with geometric features such as edges. The matrix $\mathbf{D}_p$ should be some type of wavelet, shearlet, curvelet, or contourlet dictionary.

Using this decomposition, the LADAR measurement process at a given range slice

[2] See [15], [16] for theoretical details on decomposing an image into the sum of a piecewise smooth part and a textural part.

can be reformulated as

$$\begin{aligned}\mathbf{y} &= \mathbf{A}\mathbf{D}_p\boldsymbol{\alpha}_p + \mathbf{A}\mathbf{D}_t\boldsymbol{\alpha}_t + \mathbf{b}\\ &= \mathbf{B}_p\boldsymbol{\alpha}_p + \mathbf{B}_t\boldsymbol{\alpha}_t + \mathbf{b}, \end{aligned} \tag{2.3}$$

where $\mathbf{B}_p = \mathbf{A}\mathbf{D}_p$ and $\mathbf{B}_t = \mathbf{A}\mathbf{D}_t$. We propose to recover the LADAR image $\mathbf{x}$ by estimating the components $\mathbf{x}_p$ and $\mathbf{x}_t$ by solving the following problem:

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_p, \hat{\boldsymbol{\alpha}}_t = \arg\min_{\boldsymbol{\alpha}_p,\boldsymbol{\alpha}_t} \lambda\|\boldsymbol{\alpha}_p\|_1 + \lambda\|\boldsymbol{\alpha}_t\|_1 + \gamma TV(\mathbf{D}_p\boldsymbol{\alpha}_p)\\ + \frac{1}{2}\|\mathbf{y} - \mathbf{B}_p\boldsymbol{\alpha}_p - \mathbf{B}_t\boldsymbol{\alpha}_t\|_2^2, \end{aligned} \tag{2.4}$$

where γ and λ are two regularization parameters and TV is the total variation (i.e. sum of the absolute variations in the image). The two components are the corresponding representations of the two parts and can be obtained by $\hat{\mathbf{x}}_p = \mathbf{D}_p\hat{\boldsymbol{\alpha}}_p$ and $\hat{\mathbf{x}}_t = \mathbf{D}_t\hat{\boldsymbol{\alpha}}_t$. Once the two components of $\mathbf{x}$ are recovered, we obtain the final estimate as

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}_p + \hat{\mathbf{x}}_t. \tag{2.5}$$

This notion of separating a signal into different morphologies using sparse representations is often known as Morphological Component Analysis (MCA) [16], [17].

2.2.1 Speckle

In LADAR imaging, the multiplicative speckle noise model is often given by $\mathbf{z} = \mathbf{x} \cdot \mathbf{v}$, where $\mathbf{z}, \mathbf{x}$ and $\mathbf{v}$ are the noisy LADAR image, the noise free image and speckle considered as a random vector, respectively. Here, $\cdot$ represents the point wise multiplication. See [18] for an extensive investigation of the statistics and the origins of speckle.

This multiplicative noise model can be rewritten as a sum of the noise free image and a signal dependent additive noise as

$$\mathbf{z} = \mathbf{x} \cdot \mathbf{v} = \mathbf{x} + (\mathbf{v} - \mathbf{1})\mathbf{x}, \tag{2.6}$$

where $\mathbf{1} = [1, \cdots, 1]^T$. In this setting, when the underlying image contains no textures, the first and the second terms in (2.6) can be viewed as $\mathbf{x}_p$ and $\mathbf{x}_t$, respectively.

2.2.2 Optimization

Various methods can be used to obtain the solution of (2.4). In this chapter, we adapt an iterative shrinkage algorithm known as the Separable Surrogate Functionals (SSF) to solve the separation problem [19], [20]. Algorithm 1 summarizes the different steps of the SSF algorithm, where $\mathcal{S}_\lambda(\mathbf{x}) = \text{sign}(\mathbf{x})(|\mathbf{x}| - \lambda)_+$ is the element-wise soft-thresholding operator with threshold λ, $(a)_+$ denotes the function $\max(a, 0)$ and $\mathbf{H}$ denotes the undecimated Haar wavelet dictionary. Note that we have replaced the TV correction term by a redundant Haar wavelet-based shrinkage estimate as this seems to give the best results. This adjustment is applied only to the piecewise smooth component to control the ringing artifacts near the edges caused by the oscillations of the atoms in the dictionary $\mathbf{D}_p$. The same adjustment was used in [16] and the substitution was partially motivated by observing the connecting between TV and the Haar wavelet given in [21]. See [19], [20] for more details on the derivation of this algorithm.

Algorithm 1 The SSF iterative shrinkage algorithm to solve (2.4).

```
**procedure** SSF($\mathbf{y}$)
    $k \leftarrow 1$
    $\hat{\boldsymbol{\alpha}}_p^0 \leftarrow 0$
    $\hat{\boldsymbol{\alpha}}_t^0 \leftarrow 0$
    $\mathbf{r^0} \leftarrow \mathbf{y} - \mathbf{B}_p\boldsymbol{\alpha}_p^0 - \mathbf{B}_t\boldsymbol{\alpha}_t^0$
    **repeat**:
Update the estimates of $\boldsymbol{\alpha}_p, \boldsymbol{\alpha}_t$
        $\tilde{\boldsymbol{\alpha}}_p^k \leftarrow S_\lambda(\frac{1}{c}\mathbf{B}_p^T(\mathbf{r}^{k-1}) + \boldsymbol{\alpha}_p^{k-1})$
        $\boldsymbol{\alpha}_p^k \leftarrow \mathbf{D}_p^T\mathbf{H}S_\gamma(\mathbf{H}^T\mathbf{D}_p\tilde{\boldsymbol{\alpha}}_p^k)$
        $\boldsymbol{\alpha}_t^k \leftarrow S_\lambda(\frac{1}{c}\mathbf{B}_t^T(\mathbf{r}^{k-1}) + \boldsymbol{\alpha}_t^{k-1})$
Update the residual
        $\mathbf{r}^k \leftarrow \mathbf{y} - \mathbf{B}_p\boldsymbol{\alpha}_p^k - \mathbf{B}_t\boldsymbol{\alpha}_t^k$
    **until** : stopping criterion is satisfied
    **return** $\hat{\boldsymbol{\alpha}}_p = \boldsymbol{\alpha}_p^k, \hat{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_t^k$
**end procedure**
```

2.3 Measurement Domain Object Recognition

LADAR is very useful in robotic perception applications that require accurate three dimensional detection of objects in the environment. However in many of these cases, reconstruction of the 3D world is not necessary as long as interpretation of measurements in alternate domains can provide sufficient information. Thus avoiding pixel-basis reconstruction when non-pixel basis measurements are made can save us computational time and effort while providing accuracy in detection and identification. Incorporating prior knowledge about the environment can further increase the efficiency of the perception module in a robotic system or reduce its complexity.

Therefore in this chapter we also explore the idea of object recognition in alternate domains (rather than one in the pixel basis). The observations made suggest that it is possible to recognize the objects directly in the measurement domain with sufficient levels of accuracy and even attaining comparable performance with the original (pixel) domain.

2.4 Experimental Results

In this section, we present results of our method[3] using the prototype agile-beam LADAR architecture. In our experiments, we use a dictionary corresponding to the undecimated wavelet transform (Daubechies wavelet with 6 vanishing moments) for the piecewise smooth part and the local DCT dictionary for the texture part. We decrease the threshold value of $\lambda = \lambda^k$ during each iteration and stop the iterations when $\lambda^k = 2.1\epsilon$, where ϵ is the additive noise level [23].

[3] Recovery methods were originally proposed in [22]

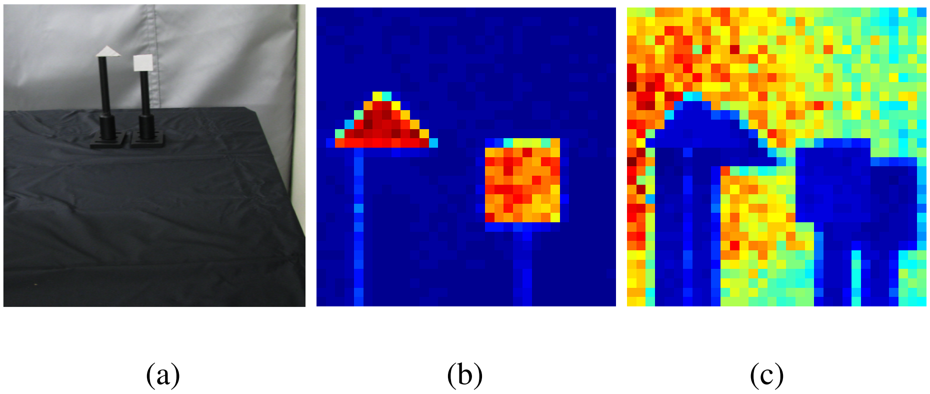


Figure 2.3: Measured data. (a) Objects used for experiments. (b) Reconstructed objects at depth 3 meters. (c) Background clutter.

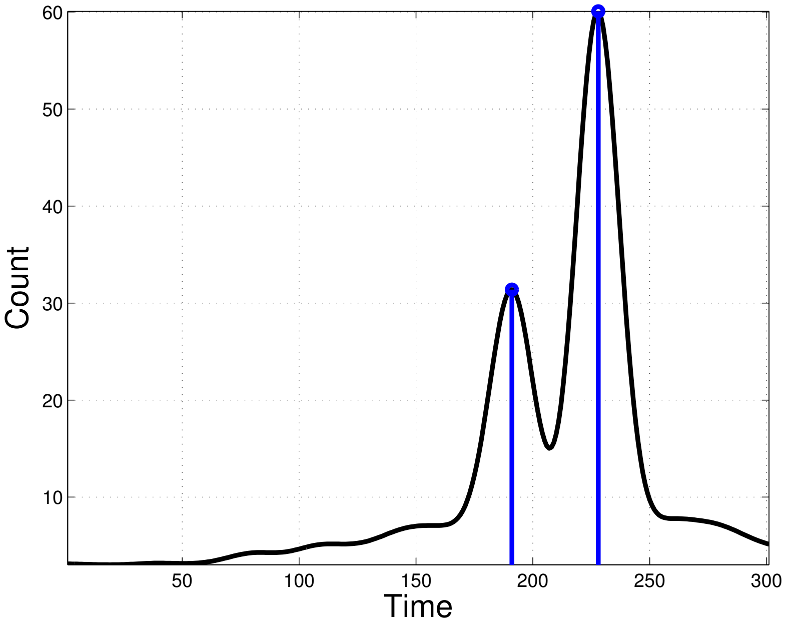


Figure 2.4: Range histogram. Peaks represent the object (at 191) and back wall (at 228) in left-to-right order.

Four different measurement modes (raster, Hadamard, DCT, Bernoulli) and a CS mode (under-determined Bernoulli) are used for collecting data. For CS, only 50% of random measurements were collected (i.e. $M = N/2$). Two shapes, a square and a triangle were cut from white cardboard and attached to a black post. They were placed about 3 meters away from the sensor within its field of view (See Figure 2.3 (a)). Peaks in the time histograms were used to identify the slice containing the objects. In the time histogram Figure 2.4, the first peak at 191 corresponds to the objects and the other peak at 228 corresponds to the background clutter. Figures 2.3(b) and (c) show the range slices corresponding to objects and clutter, respectively when raster scan is used for collecting data. One can clearly see the presence of speckle in these images.

2.4.1 Robust Recovery

In this section, we compare the reconstructions obtained by our method with that of the original system (simply by inverting the system matrix induced by the far-field illumination patterns) and using a standard sparsity promoting ℓ_1-minimization algorithm where the identity basis is used as the sparsifying transformation. In particular, the following problem is solved to obtain the cross-range element from (2.1)

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2. \tag{2.7}$$

We employed a highly efficient algorithm that is suited for large scale applications known as spectral projected gradient (spgl1) algorithm for solving the above problem [24]. Note that we do not reduce the number of measurements in this experiment (i.e. $M = N$).

Reconstruction results are shown in Figure 2.6. As can be seen from the first row of this figure, simply by inverting the system matrix, one obtains range slides that have high amount of speckle especially in the DCT and Hadamard modes. The sparsity promoting reconstructions (second row) are able to remove some noise but the best reconstructions are obtained by our method (last row). In this figure, we only show the piecewise smooth component as the texture component contains the speckle noise. This experiment clearly shows the significance of using structured representation for obtaining speckle free reconstructions directly from the LADAR measurements.

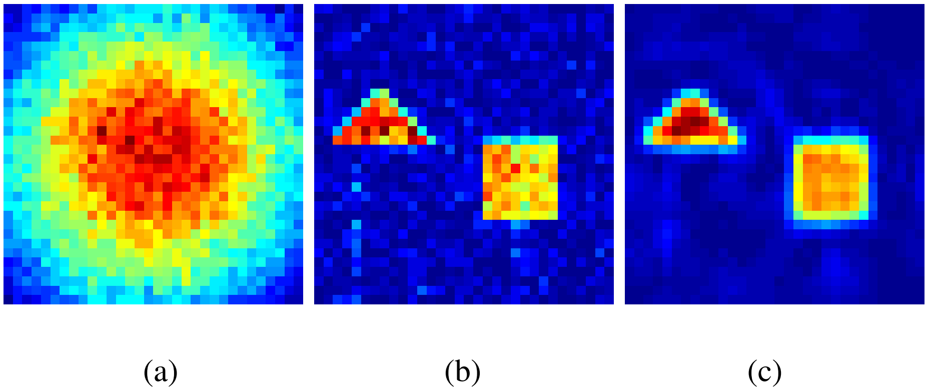
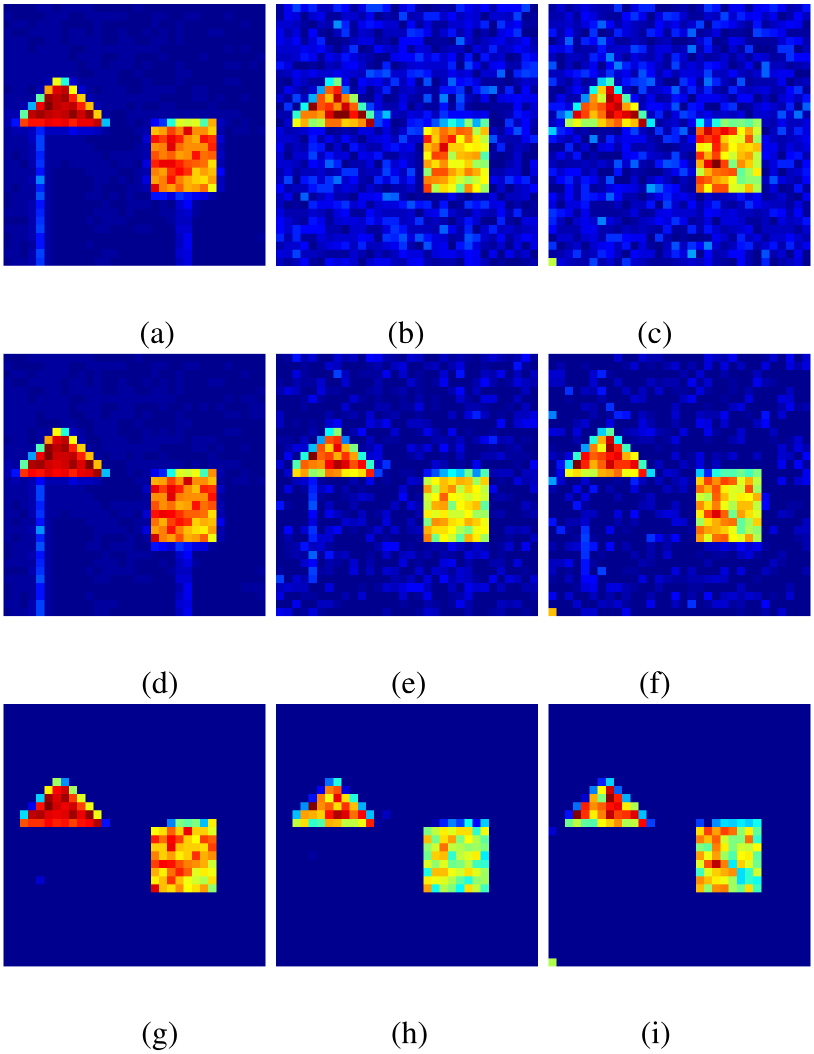


Figure 2.5: Reconstructions from compressive measurements. (a) Reconstruction by simply inverting the system matrix (e.g. ℓ_2 reconstruction). (b) Reconstruction by solving (2.7). (c) Reconstruction using our method.

2.4.2 CS Recovery

In the second set of experiments, we use the compressive data collected using a random Bernoulli $(p = 0.5)$ matrix in our agile-beam LADAR system. We retained only 50% of the measurements (i.e. M=N/2). We compare the reconstructed range slices using our method with the reconstruction obtained by solving (2.7) and by simply inverting (ℓ_2 reconstruction) the measurement matrix in Figure 2.5. As expected, the ℓ_2 reconstruction completely fails to properly reconstruct the range slice. The ℓ_1-minimization algorithm does reconstruct the range slice properly, however, it suffers from high amount of speckle. In contrast, our method is able to not only reconstruct the objects properly but it is also able to remove the speckle from the range slice.



(a) (b) (c)

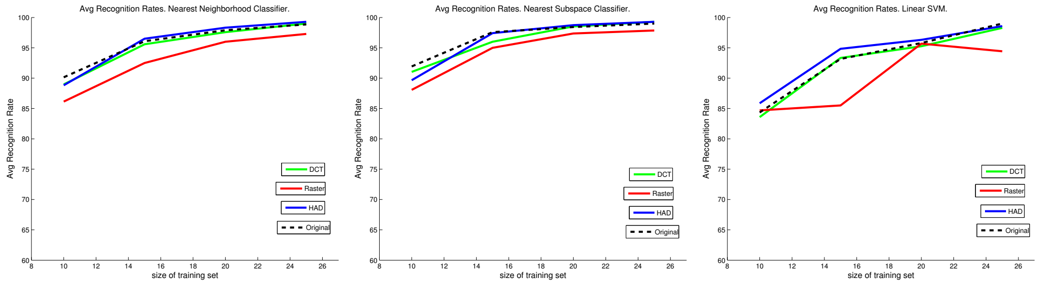
(d) (e) (f)

(g) (h) (i)

Figure 2.6: Object reconstruction. First, second and third columns correspond to raster scan, DCT and Hadamard mode reconstructions, respectively. Images in (a)-(c) show the reconstructions obtained by simply inverting the system matrix. Images in (d)-(f) show the reconstructions obtained by solving (2.7). Images in (g)-(i) show the reconstructions using our method.

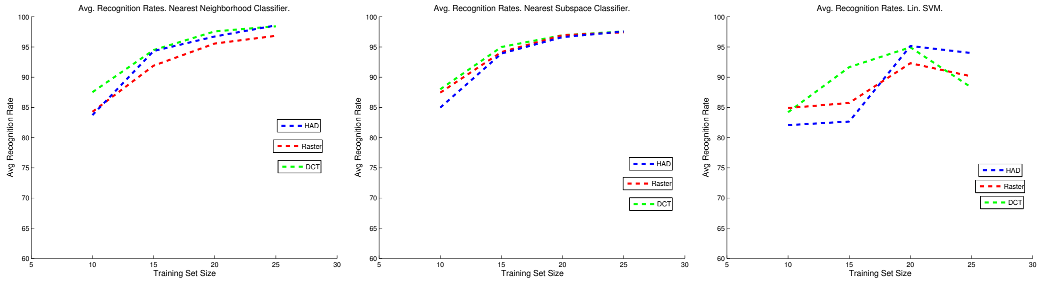
2.4.3 Object Recognition in Measurement Domain

In order to demonstrate the performance of object recognition in measurement domains, we simulate the LADAR measurements using the measured Point Spread Functions (PSF) from our agile beam LADAR prototype. These PSFs are convolved with binary images to produce data that simulates actual agile beam LADAR measurements under various modes. For the experiments, we use the Binary Alphadigits dataset. The Binary Alphadigits dataset contains binary digits of 0 through 9 and capital A through Z. Each digit is of size 20x16. There are 39 samples of each class. We use only 10 classes corresponding to digits 0 through 9. We first resize the digits to 32x32 and then convolve the digits with the measured psfs. Optimization problem is solved to reconstruct the image from the LADAR measurements. We compare the performance of three well known classifiers - nearest neighbor (NN), nearest subspace (NS) and linear SVM on this data. The effect of varying the training dataset size on the performance of the classifiers, is also studied. We randomly combine samples into training and test sets multiple times and obtain average accuracy rates for various ratios of training set vs. test set size. Moreover we compare the performance achieved in the measurement domains with those attained in the original domain of the binary numeric data. The plots in Figure 2.7 show the recognition performance in different domains. Each plot is for one the three classifiers we considered, namely, NN, NS and SVM. The different colors represent different domains in which the recognition task is carried out. The broken lines in black show the average performance achieved for corresponding training sets on the original binary numeric images in the pixel domain. Good performance is achieved in most cases. But accuracy in the raster scan mode is uniformly dominated by the other alternate modes like Hadamard and DCT. Moreover, the performance in some domains is nearly as good or in certain cases slightly better than on the original binary images. These results indicate that the idea of LADAR measurement domain object recognition can be quite effective in many cases thereby rendering the scene reconstruction stages superfluous for certain applications.



(a) (b) (c)

Figure 2.7: Performance comparison of different classifiers in the measurement domain and the original pixel domain. (a) Nearest neighbor. (b) Nearest subspace. (c) Linear SVM.



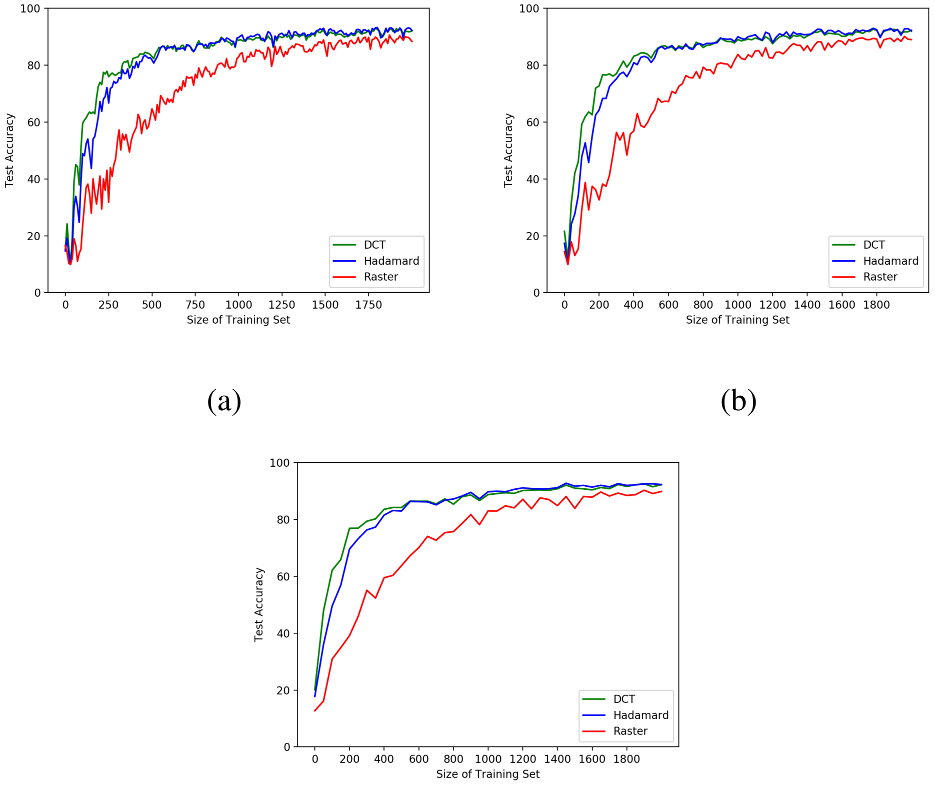
(a) (b) (c)

Figure 2.8: Performance comparison of different classifiers when the images are recovered by inverting the sensing matrix. (a) Nearest neighbor. (b) Nearest subspace. (c) Linear SVM.

We perform another set of experiments to study the effect of scene reconstruction on recognition accuracy. To simulate this we carry out inversion of the transformation(e.g. Hadamard, DCT) on binary images to the visual domain. The plots in Figure 2.8 show the performance of the three different classifiers on the inverted data. They indicate that inversion of transformation from the Hadamard and DCT domains result in equally good classification performance compared to classification in the raster domain in the case of NN and NS classifiers. However these values, again, are similar to the accuracy levels achieved by direct classification in the Hadamard and DCT domain which can be observed by comparing the corresponding(blue and green) curves in Figure 2.7 and Figure 2.8 for the particular type of classifier. For example, in the case of the small dataset with 250 training and 110 test samples the average time for NNclassification in the measurement domain was 0.5050 while that for the inverted domain was 0.5247 which showed a 3.9increase. In the case where we have more samples to invert, the inversion time will be higher. Therefore, the observations made from the above experiments suggest that LADAR data measurement in the DCT or Hadamard domains followed by classification in the same domain itself yields very good performance and cuts down the requirement of computational time and resources. In addition to the classification results based on non-deep features presented above, we conducted some experiments on the MNIST handwritten digits dataset (which has 10 classes). We used 2000 training and 1000 test samples. We convolved the MNIST images with the Hadamard, DCT and Raster point spread functions to simulate LIDAR measurements in these modes. We performed a random permutation of the training data and trained a simple convolutional neural network(convnet) for 1 epoch. The convnet has 2 convolutional layers first with 32 filters of size 5x5 and second with 64 filters of size 5x5. Each convolution is followed by a thresholding operation (Tanh()) and maxpooling (stride 2, size 2x2). This is followed by two fully connected layers. The architecture was borrowed from the torch7 demo codes for training a digit classifier. While training, we trained the model with set sizes that were incremented in steps of 10, 20, and 50 samples and tested performance after training on each of these sets. The plots in Figure 2.9 show the performance in DCT, Hadamard and Raster domains. The convolutional network-based experiments produced outputs similar in nature to non-deep classification methods. Classification performance in the DCT and Hadamard domains dominated performance in the Raster domain by a noticeable margin especially when the training set size is smaller.

2.5 Conclusion

In this chapter, we studied the problem of directly reconstructing two separate components of LADAR range slices from the LADAR measurements. The approach is based on an optimization formulation that can be solved in the form of thresholding iteration scheme. One of the important advantages of this approach is that the separate components are more amenable to classification tasks yet most importantly solving for these components individually provides an improved reconstruction fidelity. Part of the improvement comes from being able to obtain improved estimates of the salient and speckle elements. This is because these components are being estimated using separate dictionary that is best adapted to these features unlike previous formulation of this LADAR image recovery problem. Furthermore, we studied whether is it possible to directly recognize



(a)

(b)

(c)

Figure 2.9: Results obtained by a DCNN digit classifier on MNIST data in Hadamard, DCT and Raster measurement domains. Plots (a), (b) and (c) correspond to the same model trained on training sets with sizes increasing in steps of 10, 20 and 50 samples respectively

objects in the LADAR's measurement domain, as opposed to a reconstruction to the pixel domain. Empirical results obtained from experiments with three different classifiers show that indeed it is possible to recognize objects in the measurement domain without explicitly reconstructing them. Classification accuracy in the measurement domain is as good as the original and reconstructed domains. This speeds up the process of interpretation and perception of the three dimensional scene by avoiding reconstruction.

Chapter 3: Deep Feature Extraction in the DCT domain

Convolutional neural networks are recently being used for a wide variety of applications. They have shown remarkably good performance on various computer vision tasks such as image classification, object detection, face recognition and digits/character classification. However, most of these CNN-based models deal with millions of parameters depending on the depth of the network. Deeper neural networks, with a large number of layers and more nodes per layer, while providing good performance for image classification tasks, often require a large number of training epochs to converge and more storage space. Some recently proposed methods aim to optimize the convolutional layers which are responsible for the bulk of computational requirements in training CNNs. For example, a method to modify pretrained CNNs to obtain speed up on character recognition experiments is proposed in [25] by approximating the CNN filter banks using a low-rank basis of filters. A strategy for reduction of training and inference time is proposed in [26] by computing convolutions as point wise products in the Fourier domain and reusing the transformed feature maps.

In another recent work, there has been some focus on image classification tasks using fast learning, shallow convolutional networks [27]. They report state-of-the-art performance(or near state-of-the-art) on MNIST, SVHN and NORB-small datasets, for digit recognition or 3D shape recognition respectively. For applications where online learning or periodic updating of the model weights is required when fresh training data becomes available, shallow models that learn faster and need fewer training epochs are desirable.

Several recent papers also aim at compressing CNNs using various techniques to make them more space and energy efficient. For example, some of these works involve pruning the network by removing redundant connections and learned quantization of the weights so that multiple connections share the same weight. Trained quantization is performed as a separate step in the pipeline proposed by [28], in which clustering of weights and codebook learning are done separately after network training is completed. Although many such strategies have been proposed to compress CNNs, mostly as a separate stage after the training process, an elegant way to do this would be to integrate compression techniques within the training phase of the network so that the resulting weight matrices learned by training the network are automatically more concentrated with fewer higher value weights. Such actions can facilitate the process of pruning weights and compressing the network structure into smaller files that are easier to access.

In [29] the authors demonstrate, that using a two layer convolution network, where the layers were replaced by wavelet filters, yielded results for the Caltech dataset that are comparable to pretrained CNNs of the same depth. They use the Scattering Transform, which is composed of cascaded wavelet convolutions and modulus non-linearities, to replace the learning and pooling steps in the initial layers of the network. Using this technique in the first two layers they capture translation, rotation and scaling variability. While wavelets work in the space-frequency domain, DCT is another analogous image transform technique that operates in the frequency domain. When it comes to energy compaction techniques, the DCT is a classic and time-tested method which has been in wide use since the introduction of the JPEG standard in 1992 [30]. DCT operation on images help in compaction of information into just a few coefficients in the frequency domain and has been shown to retain essential perceptual information within those few coefficients. In addition, DCT can be computed very efficiently.

In this work, we study the effect of deep feature extraction using CNNs in the DCT domain by transforming the feature maps generated by the network. We test our model on different types of datasets and demonstrate it's effectiveness in solving different computer vision tasks by contrasting the performance with CNNs without the DCT operation where the features are extracted entirely in the visual domain. While no explicit quantization was performed during the training phase, we observe greater compaction in the trained weight matrices with a higher proportion of near zero weights which is promising for applications that may benefit by weight pruning, hashing or quantization techniques for compression of deep neural networks.

3.1 Related Work

Recent research in deep learning indicates that there is significant parameter redundancy in several deep learning models. Deep networks typically require large amounts of training data in addition to significant amount of computational time and resources. These requirements arise primarily owing to the fact that typical deep models learn a large number of parameters(in the millions) from the training data which are then used for predictions on the test data. This makes it quite challenging to implement online learning deep models. However recent studies suggest that some of these issues can be tackled using different approaches to compress deep networks or remove redundant parameters.

A recent work by [31] applies hashing techniques to compress neural networks. The resulting network known as HashedNets employs a low-cost hash function to group weights randomly into buckets so that all connections within a bucket share the same parameter value. This technique reduces storage requirements without deteriorating the generalization performance of the network. Another work [28] employs pruning, trained quantization and Huffman coding to achieve compression. An algorithm to automatically optimize the network architecture by selective removal of parameters at the training time is proposed in [32]. Another work in predicting the parameters in deep networks [33] illustrates the redundancy in parameters by using a small set of given weights to predict the remaining (95%) network weights. All these works show that there's a considerable benefit in integrating compressive techniques into deep networks.

The discrete cosine transform(DCT) has been applied to various computer vision tasks in the past. It has been particularly successful in image compression related applications and forms the core of the JPEG standard. The DCT operation in visual image data converts 2D spatial information to spectral information which is then quantized and manipulated for compression.

Besides image compression, DCT has also been applied to other problems such as image feature extraction in the DCT domain. For example, SIFT feature extraction in the DCT domain led to fewer SIFT keypoints and higher efficiency [34]. In many Content Based Image Retrieval(CBIR) applications, different types of DCT features have been used [35], [36], [37], [38]. A significant number of images available on the internet are represented in a compressed format using the JPEG standard. Image feature extraction on these compressed images can be made more efficient by performing the extraction directly in the compressed domain. The works mentioned above are motivated by this fact and develop low-level feature extraction techniques using DCT.

In the chapter on efficient image retrieval in the DCT domain by hypothesis testing [37], the authors use the KL-divergence based-ranking on empirical distributions of DCT coefficients of query and target images obtained by partial decoding of JPEG images. In another work on image retrieval in the DCT domain [35], the authors extract perceptual information in the compressed domain using the quantized histogram of statistical texture features in DCT blocks and show the robustness of this approach for image retrieval. Such examples of applications of DCT in image feature extraction are numerous and widespread [30].

3.2 The DCT-CNN Model

In our model, which we shall refer to as the DCT-CNN model, we add the following operation: a DCT of the feature maps generated by the Convolutional layer of the CNN. This step, performed after non-linear thresholding of the convolutional features, is followed by the usual stages of max pooling, and subsequent convolutional layers. We evaluated many ways of integrating DCT into the network and observed that the best results were achieved when the DCT operation was performed only once, following the very first convolution and threholding steps.

3.2.1 Motivation

The usual effect of DCT when applied to 8x8 blocks of image data is to concentrate most of the larger values in the block to the upper left-hand. However, in our model, we apply the DCT on feature maps generated by the first convolutional layer in the network. A visualization of the effect of this operation on feature maps is provided in Figure 3.1. The features maps generated by earlier convolutional layers in a network usually try to capture low level image features. A DCT operation performed on this map has a similar effect, visually, on the features maps as it would have on a normal image. The high intensity values are concentrated in the upper left hand corner. The features extracted from the image are thresholded and then converted to the DCT domain and thereafter follows the usual process of pooling and convolutions in the same domain. Thus subsequent feature extraction beyond the very first convolution layer, occurs in the frequency domain. It has been established in several previous studies that important perceptual information in the images are well preserved under the DCT operation. This observation serves as a motivation for our work to study the performance of deep feature extraction in the DCT domain for various image classification tasks. Another objective is to observe the effect of the transformation on the network weights and compare and contrast the network weight matrices obtained when feature extraction takes place under the transformation versus without it. The expectation is that the DCT operation within the network will result in sparser weight matrices trained over data which is more conducive to compression techniques such as hashing and quantization for efficient compression and storage of CNNs.

Effect of DCT on Backpropagation

In a standard network a weight $\omega_{k,j}$ connecting unit $k1, k2$ in layer $n-1$ to unit $j1, j2$ in layer n has gradient

$$\frac{\partial E}{\partial \omega_{k1,k2,j1,j2}} = \frac{\partial E}{\partial s^n_{j1,j2}} \frac{\partial s^n_{j1,j2}}{\partial \omega_{k1,k2,j1,j2}} \tag{3.1}$$

The term $\frac{\partial E}{\partial s^n_{j1,j2}}$ is obtained via chain rule. When we apply a two-dimensional discrete cosine transform to the output X^1 of the layer 1, we obtain

$$\mathbf{Y}^1 = \mathbf{D}\mathbf{S}^1\mathbf{D}^T$$

$$y^1_{i,j} = \mathbf{d}_{i,:}\mathbf{S}^1\mathbf{d}^T_{:,j}$$

where D is the DCT matrix. The new equation for computing gradient using chain rule now is

$$\frac{\partial E}{\partial \omega_{k1k2,j1,j2}} = \frac{\partial E}{\partial s^1_{j1,j2}} \frac{\partial s^1_{j1,j2}}{\partial \omega_{k1,k2,j1,j2}}$$

$$\frac{\partial E}{\partial s^1_{j1,j2}} = \frac{\partial E}{\partial \mathbf{Y}^1} \frac{\partial \mathbf{Y}^1}{\partial s^1_{j1,j2}}$$

$$\frac{\partial \mathbf{Y}^1}{\partial s^1_{j1,j2}} = \begin{bmatrix} d_{1j1}d_{1j2} & d_{1j1}d_{2j2} & \ldots & d_{1j1}d_{32j2} \\ \vdots & & \ddots & \vdots \\ d_{32j1}d_{1j2} & d_{32j1}d_{2j2} & \ldots & d_{32j1}d_{32j2} \end{bmatrix}$$

Figure 3.2 shows some of these filters.

3.2.2 Framework

The DCT-CNN structure

Figure 3.3 shows the incorporation of the DCT operation in a CNN. The usual architecture of a CNN consists of an input layer followed by convolutional, thresholding and pooling layers which are repeated multiple times followed by a classifier (e.g. Log-Softmax) on top. This network structure is augmented, as shown in Figure 3.3, with a DCT operation following the first convolutional layer that extracts several feature maps from the input RGB image and a thresholding operation. Thus the subsequent feature extraction steps are carried out in the DCT domain.

REDFT10 transform - type II DCT

The 2D DCT operation used on the feature maps is REDFT10. An REDFT10 transform (type-II DCT, sometimes called the DCT) is defined by:

$$Y_k = 2\sum_{j=0}^{n-1} X_j cos[\pi(j+1/2)k/n] \tag{3.2}$$

For two-dimensional DCT, which is a separable transform, the operation is first performed on the columns of the matrix followed by the rows. Each element of the transformed list is the inner (dot) product of the input list and a basis vector. The constant factors are chosen so that the basis vectors are orthogonal and normalized. The DCT can be written as the product of a vector (the input list) and the n x n orthogonal matrix whose rows are the basis vectors.

Implementation

We augment the network by introducing this step between the first convolutional and thresholding layer and the subsequent pooling stage. We take the output of the convolutional layer which is a three or four dimensional matrix depending on the batch size [BatchSize x NumFeatureMaps x Width x Height] followed by non-linear thresholding operation and perform DCT on this output cube layer by layer. Thereafter the transformed maps are passed onto the pooling layer for sub-sampling. Since we use small sized images(e.g. 32x32) we performed DCT on the entire image instead of any sliding windows. We studied the effect of this augmentation on fairly shallow networks. The networks had 3 convolutional layers. Max pooling [39] was seen to yield best results. Different types of activation functions(e.g. ReLU,Tanh) and the dropout technique were tested. Data augmentation was used in some cases to augment the small training sets. The experiments were done using CPUs. Other variations in the network structure, e.g. number of feature maps were also studied.

3.3 Experiments

3.3.1 Datasets

In order to establish the differences in the performance of CNNs versus CNNs with a DCT operation, we considered the following computer vision tasks: pedestrian detection, face detection, material categorization and object detection. We used the INRIA person dataset, the ETH Zurich Pedestrian datasets, the FDDB dataset, the Purdue ELab Face dataset, the Flickr Materials Dataset and Cifar-10 dataset.

The **INRIA** Person Dataset is a large set of 1805 cropped images(64x128) of upright people in various orientations standing against a wide variety of backgrounds. It was first introduced in [40].

The **ETH Zurich** pedestrian dataset was used in [41]. It was subsequently used for experiments in multi-person tracking [42] and moving obstacle detection [43]. This dataset offers several sequences of different types of scenes, with frames of size 640 x 480. We used the BAHNHOF and SUNNY DAY sequences for testing and training respectively. We used the updated annotations provided in [44] which contain bounding boxes for pedestrians down to a size of 48 pixels whereas in the original work bounding boxes of size 60 pixels or higher only were considered.

Face Detection Data Set and Benchmark(FDDB) is a dataset of face regions that was constructed to study unconstrained face detection problems [45]. The Purdue e-Lab face-dataset is another dataset [46] on which we tested our model.

The **Flickr Material Dataset** appeared in a recent work [47] and consists of high resolution images of ten different classes of materials, namely, fabric, foliage, glass, leather, metal, paper, plastic, stone, water, and wood with 100 images per category and has been used to study human material categorization.

Finally, the **CIFAR-10** [48] dataset is an object recognition database of ten different object categories, namely, airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Each class has 5000 32x32 color images for training and 1000 for testing.

3.3.2 Person Detection

A recent result on the pedestrian detection published in [49] uses a multistage, shallow deep network of two hidden layers and reports results on the Caltech, TUD Brussels and ETH Zurich datasets(trained on the INRIA dataset). For the experiments on the INRIA person dataset, we also deployed a fairly shallow CNN. Input is a 3x46x46 RGB image which passes through a convolution layer with 7x7 filters producing 32 feature maps. Followed by non-linear thresholding, DCT coefficients are computed on these feature maps. Max pooling with stride = 2 is used to reduce the size of feature maps. This is followed by two more convolution layers (with ReLU and a max pooling layer) which produce 64 and 128 channel feature maps respectively. Features maps are reshaped into vectors and passed through two fully connected layers with ReLU. LogSoftMax is used as a classifier on the output of the final layer.

This was followed by reshaping and classification using a two layer MLP (linear transform and thresholding) with a LogSoftMax function applied at the end.

We trained and tested these models on the INRIA dataset using the Torch7 demo code. The models were trained on 5000 images and tested on 1000 images. Here, we have highlighted the difference between adding a DCT operation on the feature maps generated by the first convolutional layer vs. training without it (Figure 3.4). It is evident from the plots that the model converges within fewer training epochs when DCT is performed on the convolution feature maps (Figure 3.4 : left). The maximum test accuracy for model without DCT (Figure 3.4: right) is **99.90%** whereas with DCT the highest accuracy is **99.49%**. There is a small drop of accuracy (0.4%) but convergence was much faster.

Experiments were carried out, with the same networks on the ETH Zurich dataset. The network was trained on the SUNNY DAY sequence which has 354 frames and 1900 each of pedestrian and background images(cropped). The BAHNHOF sequence which has 999 frames and 8467 bounding box annotations was used for testing. The maximum test accuracy achieved by the DCT-CNN network is around **96.3%**. After three training epochs the model reached 90.4% accuracy and in about 100 training epochs the network reached around 96% and almost converged, with only minute changes in accuracy beyond that. The same network without the DCT layer with exactly same values of hyper-parameters was trained and tested on the same dataset again. In this case, the maximum test accuracy achieved is **95.8%**. It crossed the 90% mark at 100 epochs, and convergence to around 95% accuracy was observed in about 300 training epochs.

3.3.3 Face Detection

The network described above was used on the FDDB dataset. The training data size was 5864 and the test dataset had 1822 samples. Negative samples were generated by random crops from the background. The highest test accuracy for the model without DCT is **96.81%** whereas with DCT this value is **98.10%** Thus in this case we observed a slight gain in test accuracy besides faster convergence. The plots in Figure 3.6 (left : Model with DCT right: Model without DCT) show the convergence with number of epochs.

Another set of experiments was carried out on the Purdue ELab face dataset. For this we again used the available demo code for Torch7 (train-face-detector). The training set size was 33,013 whereas the test set had 8254 samples. On the PurDue ELab Data, the DCT-CNN network achieved 95% test accuracy.

3.3.4 Material Identification

The Flickr Material Dataset(FMD) is particularly challenging. Images of objects belonging to one particular class vary widely. For example, as described in their website, ”The images of the satin ribbon, the crocheted nylon cap , the stuffed snail toy, and the flannel bedding look very different from each other". Results reported on this dataset upto IJCV'13 show that the highest accuracy at 60.6% was obtained with SVMs and using SIFT, color and micro-jet features. In our experiments, we used the masks to obtain a number of small 46x46 sized croppings from the object surface that we used for training. Moreover, we left out two images per class for generating text sample croppings in the same manner. We worked with a training data size of 19600 (1960 samples per class) and 400 test samples (40 per class). We tried different CNN structures, different thresholding techniques and the dropout technique. The CNN structure that gave the best performance is provided below. Table I summarizes the values of maximum test accuracy for a few different values of hyper-parameters on networks with the following no. of feature maps :

- $32- > 64- > 128- > 128$ :: Network 1
- $64- > 128- > 256- > 256$ :: Network 2

Point wise non-linearity introduced by means of Rectified Liner Units (ReLu) provided better results than TanH non-linearity. A network with higher no. of feature maps($64- > 128- > 256- > 256$) yielded 36.98% accuracy without the DCT operation. Dropout on the same network slightly diminished the performance(35.15%). The same network, with the DCT operation gave the highest accuracy of 38.3% and along with dropout, yielded 37.24%. So overall for the various network sizes and configurations we tested, feature extraction in the DCT domain proved to be helpful.

The overall performance of shallow CNNs on this dataset could not beat the SVM performance reported in [50]. In [50], the authors used many handcrafted features, split the data randomly into 50 training and 50 test images per class and contrasted the performance of various features and classifiers. The performance for one specific split was reported. In our experiments, we took 48 images per class for training and tested on two. This was done to maximize the amount of training data for the CNNs. It was observed that adding a DCT layer certainly seemed to help improve the top accuracy. Moreover convergence was faster. In general, the CNN training does well when the training data sets are larger. However, results indicate that feature extraction in the DCT domain can somewhat help the network performance even when there aren't as many training samples to capture all the variety in the dataset.

3.3.5 Object Recognition

On the Cifar-10 dataset the maximum accuracy achieved by the network in the DCT domain is 61.4% whereas the network in the visual domain yielded 68.5% accuracy. So in this case we observe a degradation in performance. However, what is worth noting in these experiments is that the CNN in the DCT domain converged a lot quicker. It started off with a higher accuracy (32.66%) and within 232 epochs it reached 61.1% accuracy. Comparatively, the CNN in visual domain started off with 8.94% accuracy and reached 61% accuracy in 734 epochs. It reached the maximum performance : 68.5% at 1712 epochs. Thus, although the maximum performance is slightly lower, the DCT-CNN is a lot faster in reaching around 60% accuracy.

Network	Learning Rate	Dropout	DCT	Performance
1	1e-3	No	Yes	**40.65%**
1	1e-5	No	Yes	39.6%
1	1e-4	No	Yes	38.2%
1	1e-3	No	No	36.1%
1	1e-4	No	No	35.2%
2	1e-3	No	Yes	38.3%
2	1e-3	Yes	Yes	37.24%
2	1e-3	No	No	36.98%
2	1e-3	Yes	No	35.15%

Table 3.1: Performance on FMD dataset

3.3.6 Effect on Trained Weight Matrices

The DCT operation within the CNN also led us to observe certain effects on the trained weight matrices. In Table 3.2, we report the number of weight parameters with absolute values less than $1e^{-5}$ learned in the convolution modules in models trained using a CNN and a DCT-CNN on various datasets. The number of training epochs after which this number is reported is different for CNNs and DCT-CNNs. This is because the latter reaches *maximum accuracy* more quickly. Also included is a plot [Figure 3.7 (left: DCT-

Dataset	DCT	No. of epochs	No. of weights: $abs(w) < 1e - 5$	Test Accuracy
INRIA	Yes	56	303	99.49%
INRIA	No	151	257	99.49%
FDDB	Yes	49	319	96.9%
FDDB	No	254	256	96.8%

Table 3.2: Weight Matrix Elements

CNN, right: CNN)] showing how the total number of weights($abs(w) < 1e^{-5}$) vary on FDDB dataset. DCT-CNNs achieve both greater test accuracy and more near-zero weights in a shorter period of time. In case of the trained networks on INRIA and FDDB datasets, when we round all the weights ($abs(w) < 1e^{-3}$) to zero, no significant change in performance is observed for both the DCT-CNN and the CNN models.

3.4 Discussion

This work was structured in a way so that it brings out the effects of applying DCT on the feature maps of a CNN. The accuracy plots of models with and without the DCT step have been contrasted with each other. In almost all the cases, with different variations of the network and various values of hyperparameters such as learning rate, momentum, weights decay rate etc. and on all these different types of datasets, we observed a higher test accuracy for the DCT-CNNs in the very initial stages of network training and found faster convergence of the network. During the training phase we retain the low energy terms to maintain uniform size of the feature maps. The DCT operation performed at later stages on the network, (after the application of the non-linear operations like pooling or thresholding) did not yield good results. We observed an advantage(in terms of faster convergence) when DCT was applied right after the first convolution step.

The complexity of applying a DCT-II on an image of size NxN is $O(N^2 \log(N))$. While it is a fact that more work is being done per epoch to perform DCT, by parallelizing the operation faster implementations can be achieved so that multiple feature maps can be transformed simultaneously. In applications where the model weights need frequent updating and retraining, such gains can add up to saving a lot of time over a period. For example, even without parallelizing, for the INRIA dataset, per sample training time on a 1.6 GHz Intel Xeon Processor was 7ms for DCT-CNN and 4ms for the CNN. However, in about 15 epochs, the DCT-CNN reached 99$\%$ training accuracy which is faster than the CNN which reached 99$\%$ in 129 epochs. therefore the DCT-CNN is five times faster in this case, even though it was not parallelized. This ratio can be further improved by performing DCT in parallel over multiple feature maps. The time improvement will be even more dramatic for networks where the number of output maps from the very first convolutional layer is large. In some of the cases the DCT operation was observed to improve the test performance slightly. Moreover the weight matrices of the trained models contain fewer of the larger weights and hence it is easier to prune some of the inactive connections. All these experiments were CPU based and carried out on relatively shallow CNNs. Such models with lesser computational complexity can be more readily adapted for online learning purposes.

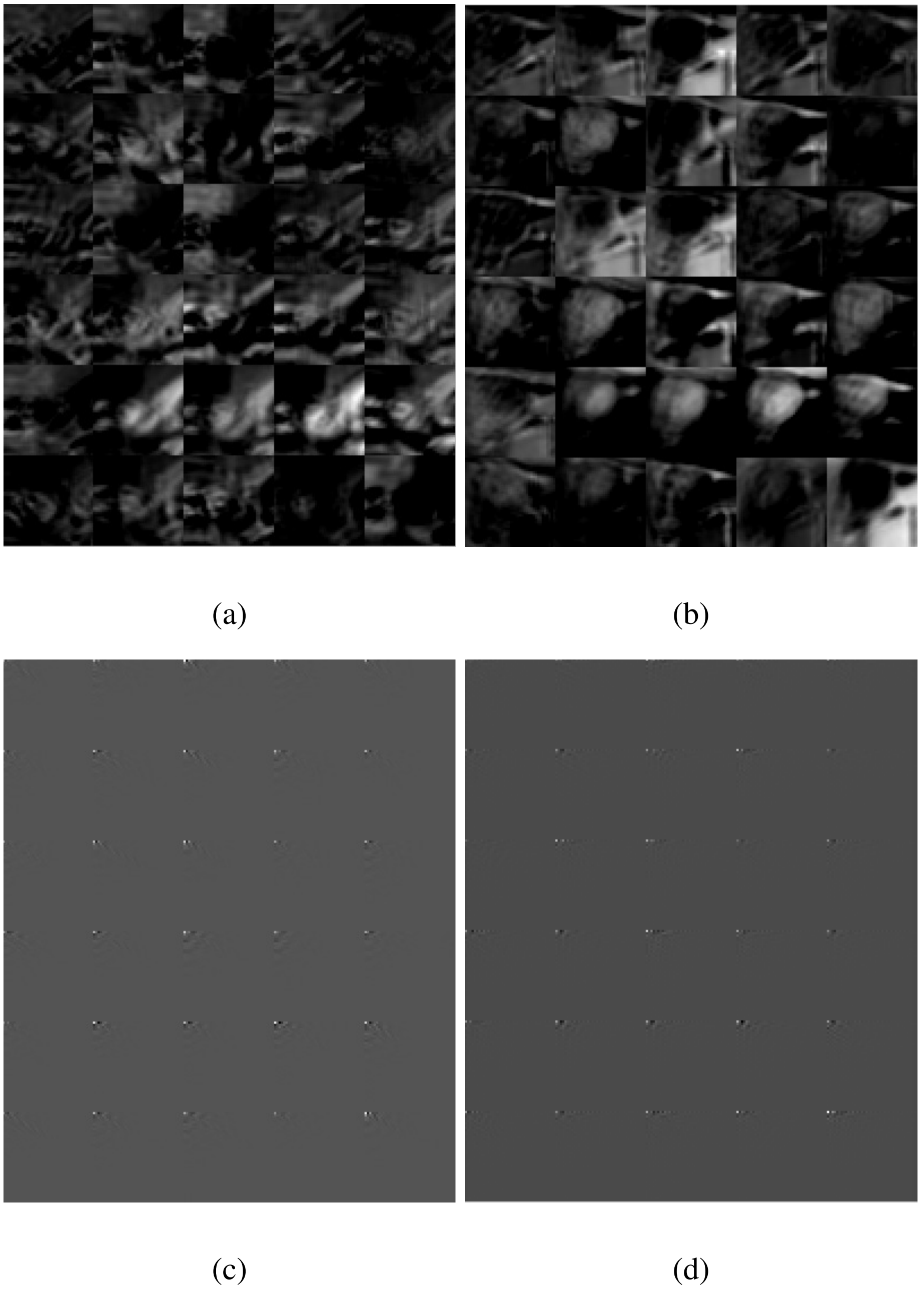


Figure 3.1: Feature maps before and after DCT. Figures 3.1(a) and 3.1(b) show several feature maps generated by the first convolution layer in the network during the training phase. Figures 3.1(c) and 3.1(d) show feature maps under DCT. All the feature maps were generated simultaneously in the same iteration, by the same layer in the same network. The white specs in the top left corner of each map correspond to low frequency DCT coefficients.

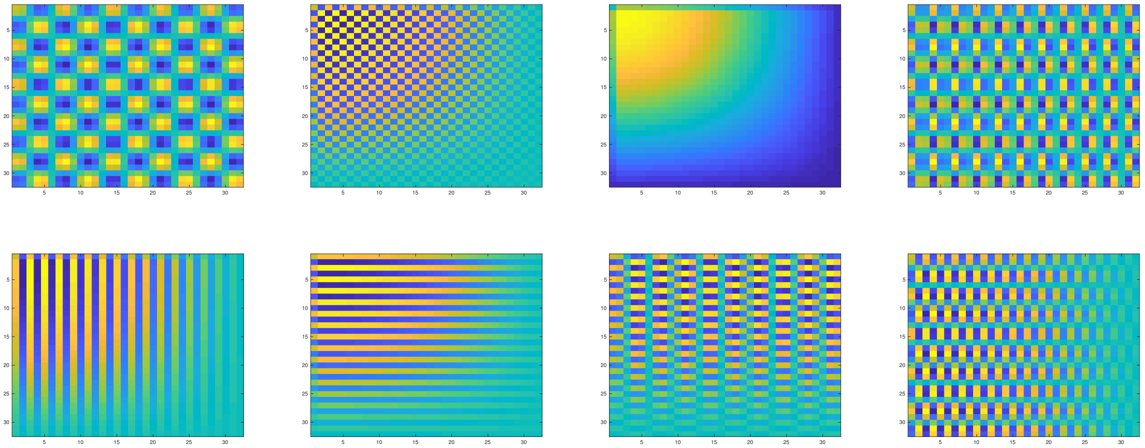


Figure 3.2: Visualization of some DCT filters

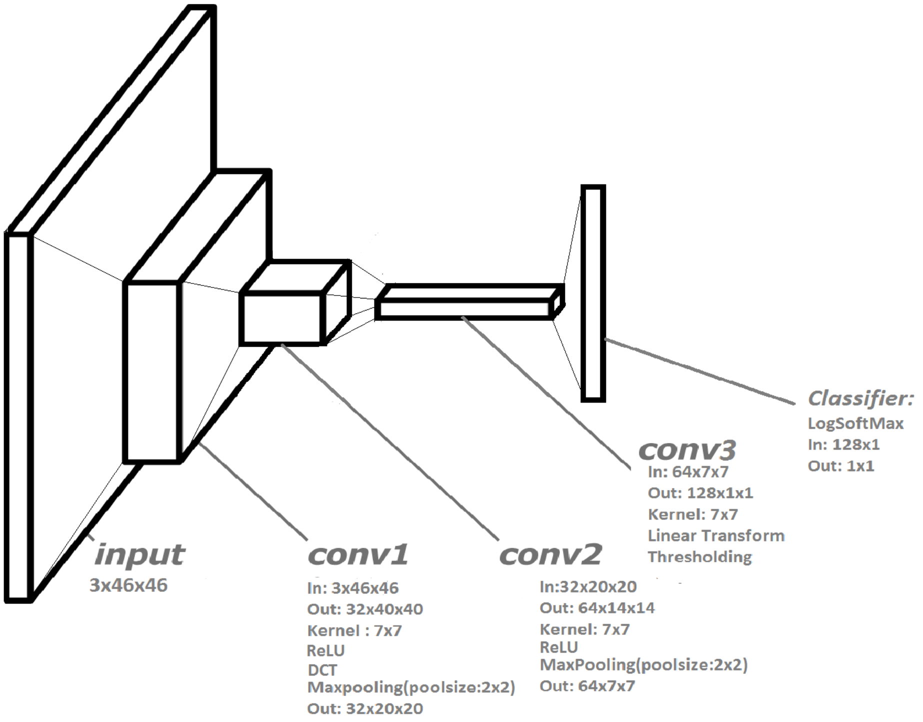


Figure 3.3: A shallow CNN structure with DCT

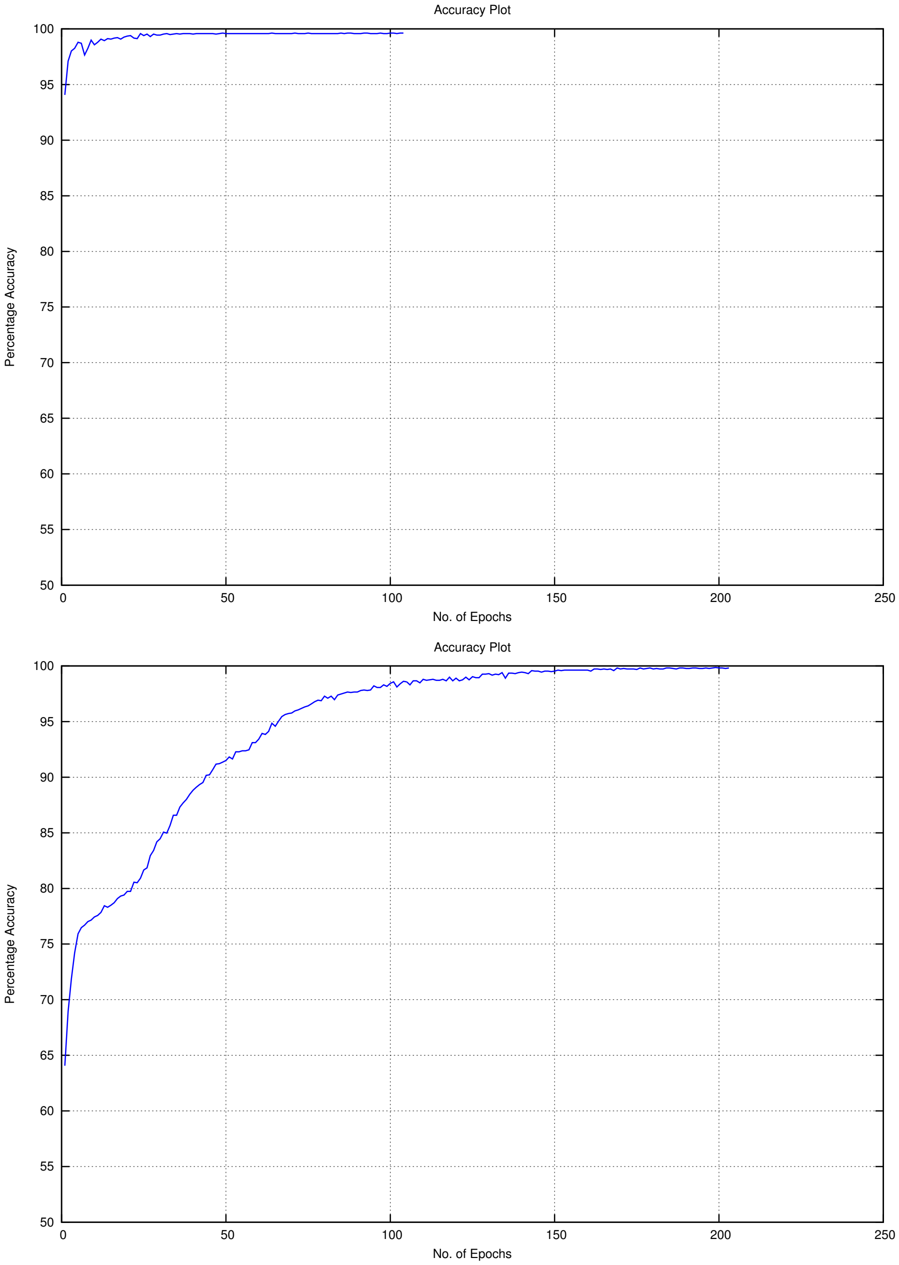


Figure 3.4: Test Accuracy Plots for Pedestrian Detection,top: DCTCNN, bottom: CNN, on the INRIA dataset

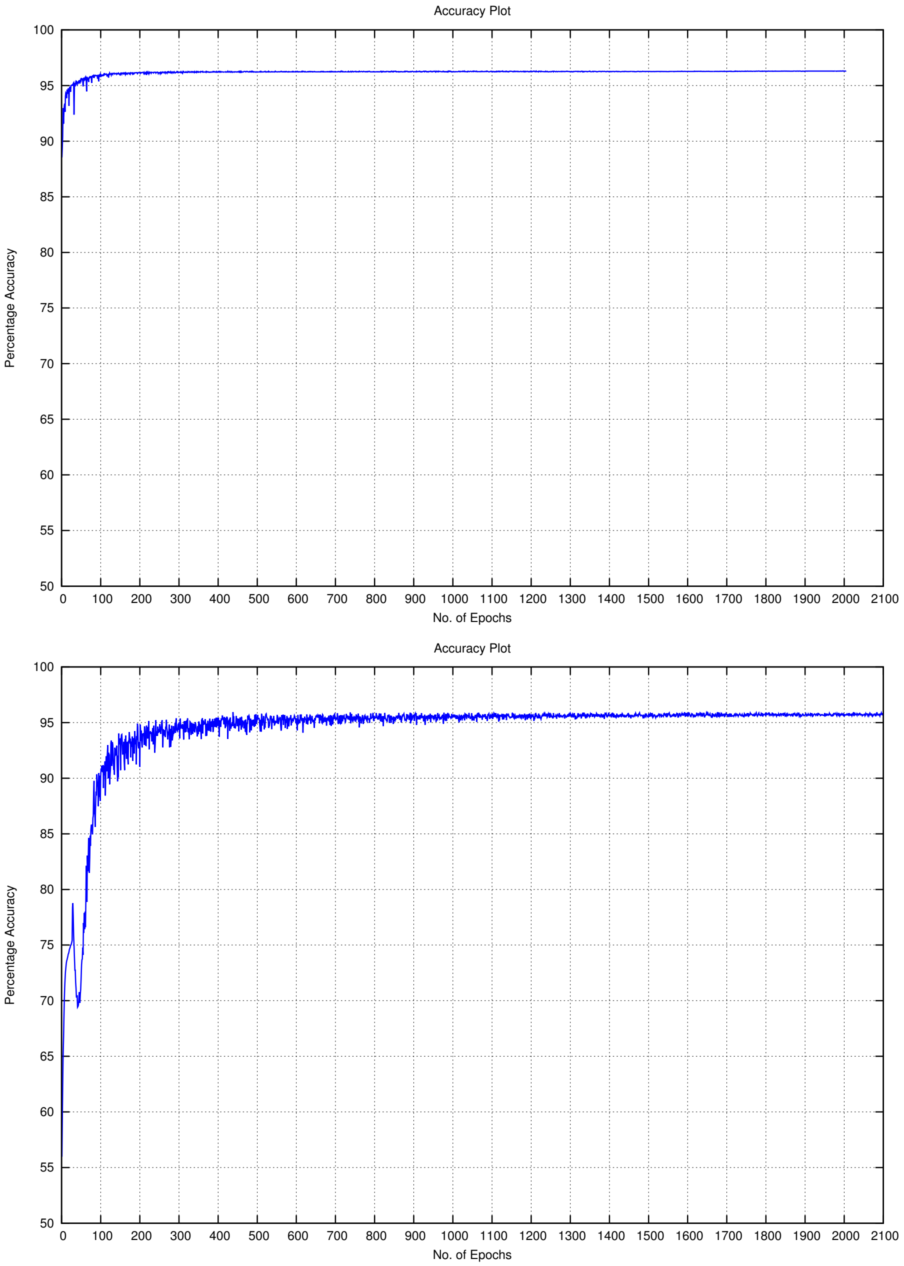


Figure 3.5: Test Accuracy Plots for Pedestrian Detection, top: DCTCNN, bottom: CNN, on the ETH dataset

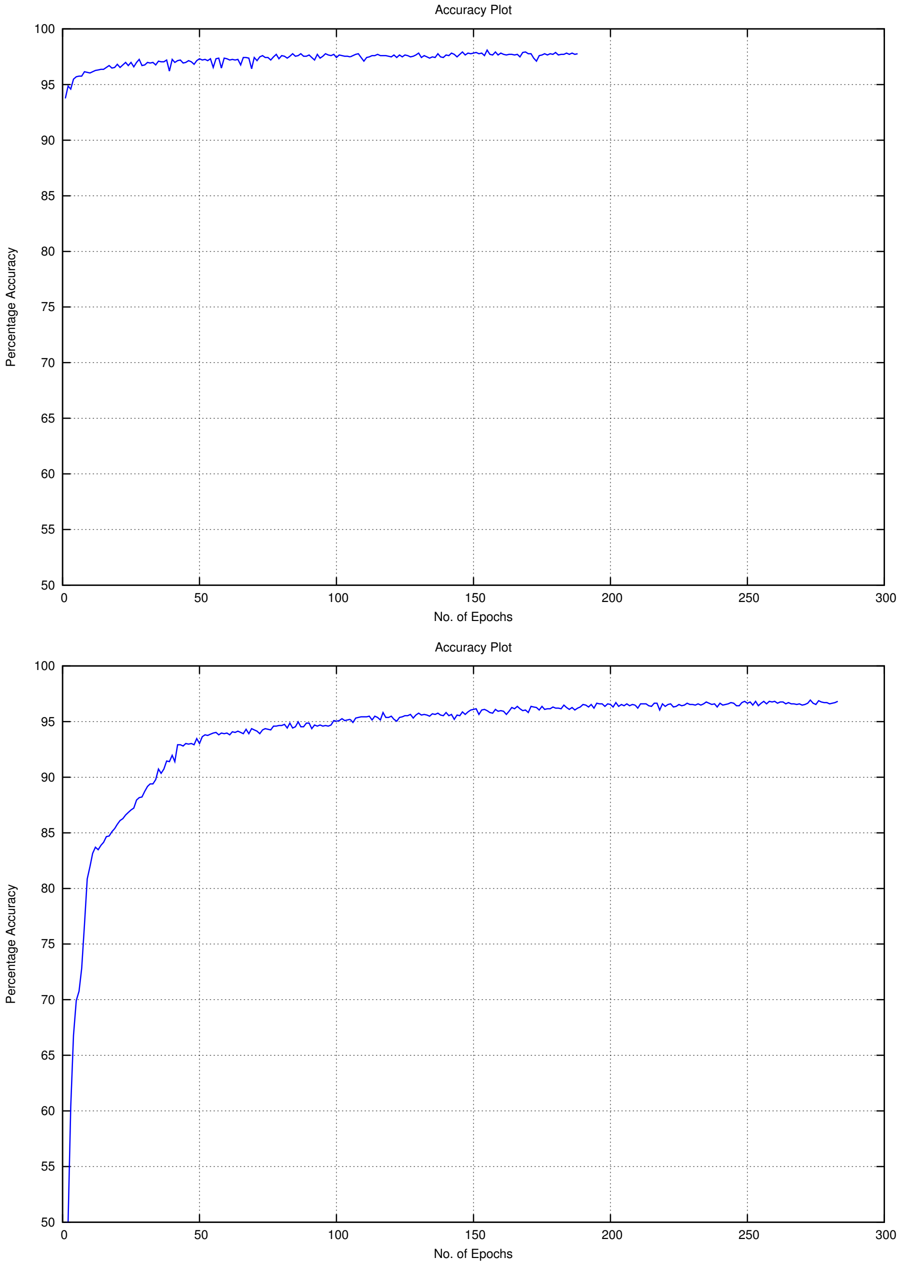


Figure 3.6: Test Accuracy Plots for Face Detection , top: DCTCNN, bottom: CNN, on the FDDB dataset

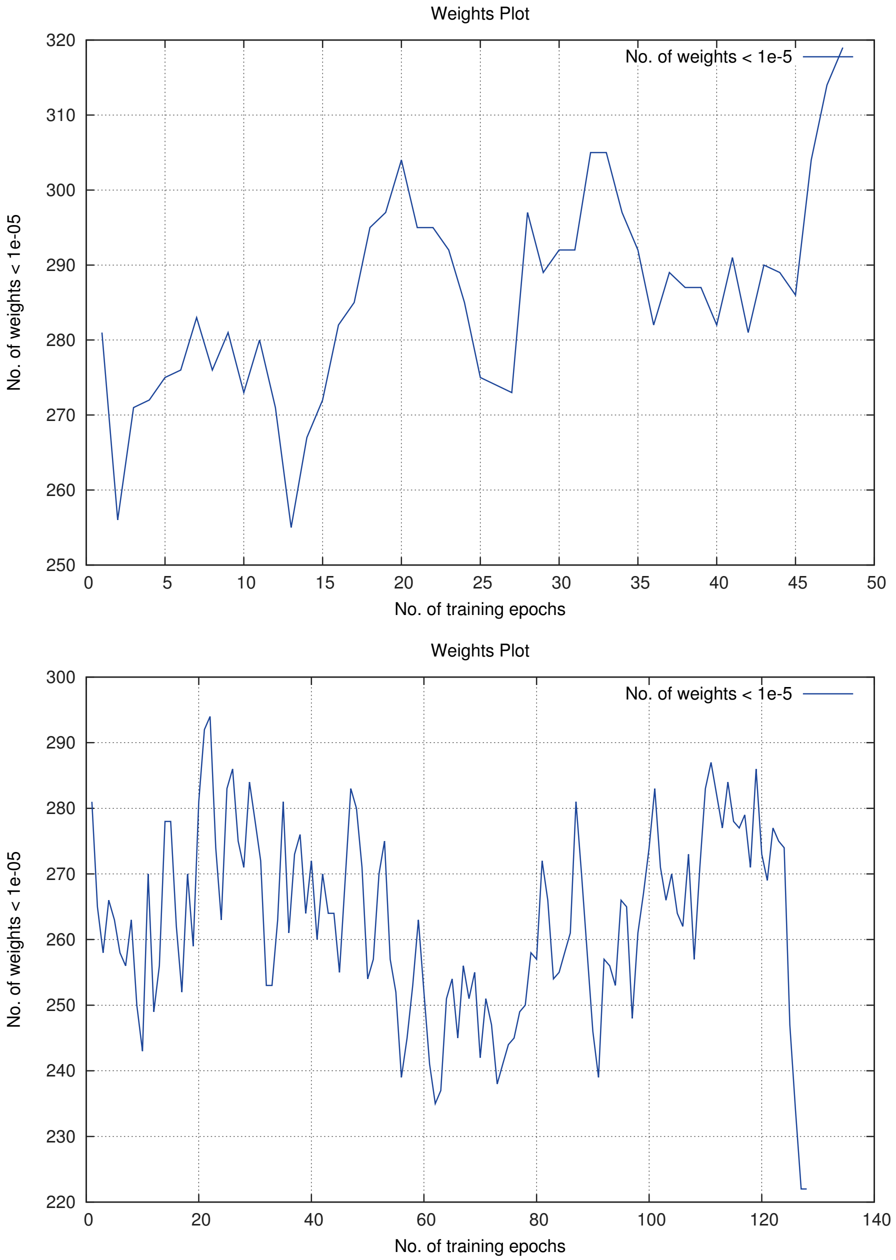


Figure 3.7: Plots showing variation in count of weights $abs(w) < 1e - 5$ for top: DCTCNN, bottom: CNN on the FDDB dataset

Chapter 4: Single shot 3D mesh estimation via Adversarial Domain Adaptation

4.1 Introduction

4.1.1 3D pose from 2D image

Pose estimation of humans from a single 2D image is a widely studied problem in computer vision. Numerous recent works have applied deep convolutional neural networks (DCNNs) with great degree of success for 2D pose estimation. Several of these methods solve for pose estimation in 2D by generating an 'attention map' which encodes joint locations in the XY plane as Gaussians on 2D heatmaps. Points in the XY plane are one-hot-encoded into the map [51–55]. Liu *e*t al. [56] show that neural networks are more effective at making inferences in the one-hot-pixel space than learning mappings from images to Cartesian coordinate space. Methods proposed in [57–66] estimate 3D skeletal pose from a single image by regressing on depth of joints. Methods proposed in [67, 68] use the classification approach to categorize 3D poses and some works [65, 69–71] use pixel or voxel maps for depth regression from 2D input. In order to infer 3D pose from a single 2D image the model must learn the geometry of the human skeleton. Estimating depth information from a single 2D image is an ill-posed problem owing to scale ambiguity. By assuming certain priors about the human skeleton (*eg.* average skeleton size computed from the training data) one can estimate depth of joints. 3D pose estimation from a monocular view has several real world applications. It avoids dependence on power intensive active sensors that estimate depth. It also avoids solving a homography from two calibrated cameras simultaneously capturing the same scene. As most images and videos that exist on the internet today are monocular, any method that infers 3D pose from monocular view will be effective on such data.

Training DCNNs end-to-end for 3D pose estimation requires 3D keypoint supervision. Human pose datasets that provide such ground-truth are captured in indoor locations or have very few subjects [66,72,73]. Combining real with virtual humans is an effective way to quickly scale up in terms of variability of body shape, appearance and background in these images. Synthetic data provides complete and exact ground-truth for 3D pose, depth, segmentation of body parts and optical flow (in case of video sequences). Varol *et al.* [74] provide a large scale synthetic dataset with virtual humans modeled using [75] against a wide variety of backgrounds. A 3D pose estimation model that leverages the variety of a synthetic dataset and is adapted to real human images can handle the task more effectively in 'in-the-wild' data.

4.1.2 3D mesh from 2D image

A 3D pose estimation technique infers joint locations in 3D from a 2D image. But it provides no information on body shape of the subject. One way to infer the body shape is by fitting a 3D mesh on the subject. Inferring a 3D mesh from a 2D image of a

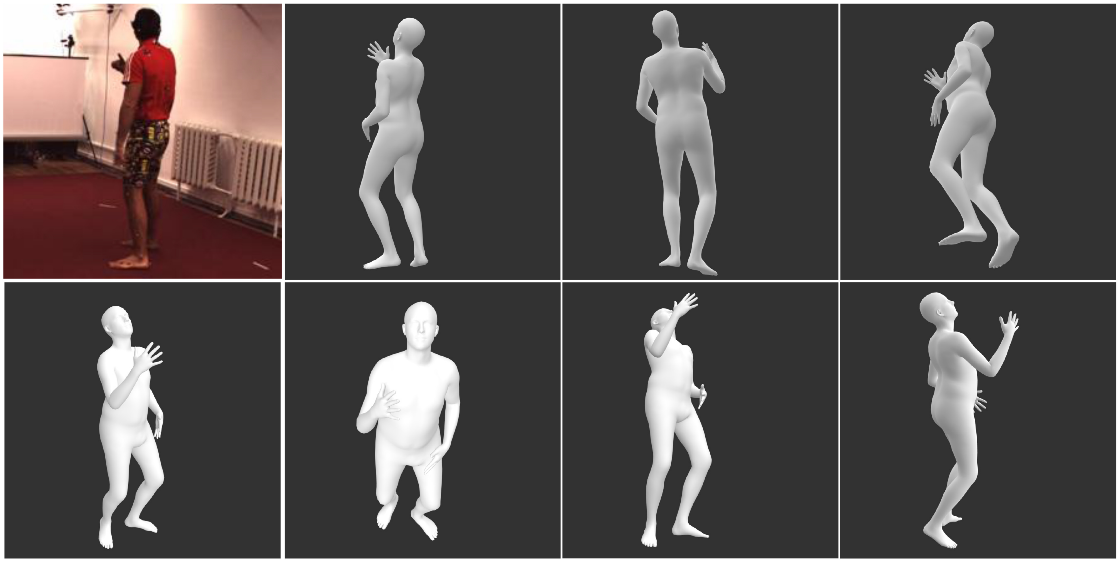


Figure 4.1: 3D pose and shape estimation and novel view synthesis from a single input image

human is an interesting problem with several applications including those in body mass index (BMI) estimation from image/video, video synthesis, mixed reality and virtual reality. Parametric 3D body models proposed in [75] represent the human body as a large number of vertices forming a 3D mesh. Inferring 3D mesh parameters from a single image can provide a complete estimate of body shape and pose in 3D. It can be useful in synthesizing novel views and animating the inferred 3D mesh model. Moreover it can extend currently available synthetic datasets (modeled based on standard MOCAP datasets) by adding mesh estimates for several new poses and activities. Datasets consisting of RGB images/videos of humans have manually annotated landmarks/ body joint locations in 2D [76, 77] or use marker-based motion capture systems to provide 3D location of landmarks on humans in indoor environments [66, 72]. State-of-the art methods for 2D pose estimation use various types of deep neural architectures [51, 52, 78, 79] trained on sufficiently large, manually annotated datasets to produce very accurate and reliable 2D landmark location estimates. Fitting 3D mesh on RGB images using 2D landmarks has been implemented in [80,81]. Bogo et al. [80] have shown that 2D landmarks themselves carry a significant amount of information about body shape parameters necessary for the mesh fitting task.

DCNN-based methods to estimate shape and skeletal-kinematic-tree parameters (rotation of body parts relative to their parents in the skeletal kinematic tree) using 3D ground-truth further improve the fitting accuracy as well as speed of inference. However datasets containing ground truth values of mesh parameters for real humans are hard to acquire. This is a challenge for direct end-to-end supervision. Some recently released datasets such as UP-3D [81] and SURREAL [74] try to address the absence of such datasets. UP-3D provides estimates of mesh parameters for images of real humans using their own approach of 2D landmark fitting using decision trees. SURREAL provides photo-realistic image sequences of 3D synthetic humans [75] generated based on CMU MOCAP data [82] along with the exact mesh parameters and 3D joint locations corresponding to those sequences.

The greatest obstacle to learning the 3D mesh parameters for real humans is the lack of ground truth data. This makes it challenging to directly train DCNNs for mesh fitting. Several techniques have been proposed in recent years to circumvent this issue [1,83–87]. All of these methods deploy some form of self-supervision and/or supervision via reprojection (inferred 3D mesh vertices to keypoints). Kanazawa *et al.* [1] use only body joint location information. Segmentation mask ground truth is used in [83–85]. Body part segmentation information is used by [86, 87].

4.1.3 Reprojection and use of Mosh'ed estimates

Directly optimizing on keypoint re-projection loss can yield results that do not obey the anthropometric constraints on body shape and pose. This has been reported in [1] and also observed in our experiments. In such cases a powerful discriminator that differentiates mesh parameters corresponding to valid human skeletal configurations from invalid ones is necessary for succesfully training the model. For results on Human3.6M, Kanazawa *et al.*use SMPL parameters estimated from the training images using MoSh [88]. They also use Mosh'ed estimates from two other real human datasets [82,89] for training their adversarial prior. In contrast, our approach to learning 3D mesh estimation does not utilize any Mosh'ed data on real humans. It solely learns from the ground truth of a synthetic dataset which is based on [82]. We demonstrate that despite this domain gap between real and synthetic images as well as poses, our learning approach works on real human subjects. Almost all DCNN-based methods for mesh fitting use reprojection loss to supervise the network. This makes learning of 3D mesh inference on any dataset directly dependent on keypoint annotations.

Our method uses only keypoints to align synthetic and real domains. The domain aligned features are used to infer mesh parameters using the second part of the network which is partially supervised using ground truth mesh parameters only from synthetic dataset. Additionally, adversarial networks are trained to assist the domain alignment process and guide the SMPL mesh estimator towards producing valid estimates on real samples.

4.1.4 Domain Adaptation

In this work, we present an alternative approach to 3D mesh fitting by directly learning from synthetic data. We design an end-to-end pipeline for domain adaptation of DCNNs between real and synthetic data. We achieve good fit on real data with minimal supervision. Domain adaption is a versatile tool for knowledge transfer that aims to achieve good performance in a target domain by training on samples in a source domain. Unsupervised domain adaptation is applicable to cases where labels in target domain are missing for a specific task. This makes it impossible to learn model parameters simply by restricting training to the target domain. Adaptation of a similar task learned in a source domain can be useful in such scenarios. Generative Adversarial Networks (GANs) have demonstrated a great deal of success in unsupervised domain adaptation of deep networks [90–94]. For the mesh parameter inference task, the source domain consists of synthetic human images with ground truth 3D mesh parameters. Images of real humans, for which this ground truth is non-existent, constitute the target domain. We train a stacked hourglass network jointly on a union of synthetic (source) and real human images (target). We optimize 3D pose loss for both domains. Additionally, we apply adversarial loss from a domain discriminator (synthetic vs. real) on features extracted by the hourglass network. The features used to infer 3D pose are simultaneously used to infer 3D mesh parameters using the second part of the pipeline. A second adversarial network discriminates between sets of generator predicted mesh and ground truth mesh from the synthetic dataset. The feature extractor (pose estimating hourglass network) and the mesh estimator together constitute the generator. The adversaries and the generator are trained together. A schematic diagram in Figure 4.4 illustrates these networks. Adversarial domain adaption is used for mitigating the domain shift between synthetic and real data and learning to infer 3D meshes on real images sans ground truth supervision.

4.1.5 Summary of results

We extend the design of a nested hourglass network for 3D pose estimation from a single image. We further solve the pose and shape *regression* task by applying *adversarial deep domain adaptation* on this proposed variant of hourglass network as well as another version from [62]. We produce results for 3D body mesh fits from 2D images of real humans, comparable to state of the art unsupervised approaches with minimal usage of annotations on real data. *We do not use any segmentation or body part information, camera parameters for re-projection loss in 2D or Mosh'ed [88] estimates of pose and shape on real data as ground truth.* Our method relies on the existence of a diverse synthetic dataset with complete availability of ground truth mesh parameters. It does not rely on any prior method to provide domain specific mesh estimates on real data. The networks are trained end-to-end, along with adversarial losses to guide the adaptation process, and produce plausible shapes and poses for the 3D mesh estimates. We study the impact of adversarial supervision from a domain discriminator and incorporate additional supervision on real data beyond pose loss. We exploit the modular design of our proposed pipeline and study different variants of architectures by training them using the same framework. The proposed approach is easily extendable to several other types of objects and their parametric models.

4.1.6 Organization

In section 4.2 we discuss the relevant prior work in the area of 3D pose and shape estimation from a monocular view. Section 4.3 presents an overview of the network architectures and training strategy used in the proposed approach. Subsection 4.3.1 details the proposed extension to hourglass architectures for 3D pose estimation via front and side view attention maps. Subsection 4.3.2 discusses the various components of the pipeline designed to infer a 3D mesh from a 2D input image. In section 4.4, we include experimental observations on 3D pose estimation performance (4.4.2) and 3D mesh fitting (4.4.3) and some additional ablation studies (4.4.3.1, 4.4.3.2, 4.4.3.3). Section 4.5 discusses the results presented in section 4.4. We summarize the results and conclude in section 4.6.

4.2 Related Work

Performance of deep feature-based models rely heavily upon well annotated training datasets. Large amounts of fully annotated real data are often unavailable for various machine learning tasks. Developing approaches to learn with minimal supervision is immensely helpful for the vast amounts of unlabeled/partially annotated data which exists on the internet today. On the other hand, synthetic datasets are easy to generate and come fully annotated with complete and accurate ground truth information. Methods that draw from the richly annotated, abundant and diverse synthetic domain and successfully adapt to real data can effectively reduce the dependence on ground truth for training DCNN models. A standard way to substitute synthetic for real data is to pre-train on synthetic datasets and thereafter fine-tune the networks on small amounts of real data. For certain tasks, such as estimating synthetic mesh parameters for real data, a complete lack of ground-truth in the target domain prohibits fine-tuning on real human images. ADA-based techniques can be applied in such scenarios to transfer knowledge acquired via supervision in synthetic domain to real data. Adaptation is achieved via joint-optimization-based construction of a shared latent space and adversarial supervision.

4.2.1 Synthetic Pose and Action Datasets

A recently released dataset of synthetic human action sequences is SURREAL [74]. SURREAL consists of photo-realistic image sequences of synthetic humans, generated using the Skinned Multi Person Linear model (**SMPL**) [75]. It offers a wide variety of pose, shape, view angle, illumination and background. Samples are labeled with exact ground truth 2D and 3D pose, depth, segmentation, body parts, optical flow, as well as *SMPL parameters used to generate the sequences*. Real human datasets often fall short in such diversity and completeness. Other partially/ fully synthetic human datasets are discussed in [81, 95]. Synthetic humans in [95] are based on a rag-doll model that are less realistic than SMPL. Video sequences in SURREAL dataset are generated using MOCAP data from real humans. This ensures that the animations closely mimic motion of real humans. Body shape parameters of SURREAL subjects are randomly sampled from a distribution with large variations. The frames have low resolution (240x320) and backgrounds are random static images.

4.2.2 2D and 3D Pose Estimation from Monocular View

Although a 3D mesh provides more complete information regarding shape and joint angles, pose estimation is more frequently performed as a keypoint localization task. Supervised training is easier in this setting since keypoint annotations for monocular images are easier to obtain from human annotators. Recent 2D pose estimation approaches [54, 55, 96–103] generate extremely accurate predictions on humans in the wild. Chu *et al.* [99] propose a *nested hourglass architecture* for 2D pose prediction. Several methods have also been designed to solve the problem of 3D pose estimation from monocular images/videos. [62, 63, 70, 104–113].

3D keypoint ground truth is harder to obtain for outdoor, in-the-wild images. Current approaches for 3D mesh fitting are based on re-projection to keypoints. Methods for accurate and reliable 3D estimation of joint/landmark locations are closely related to 3D mesh fitting. Any re-projection based training approach for mesh estimation essentially implements the task as a sub-step in pose estimation.

4.2.3 3D parametric body models

Statistical body models express the surface of complex shapes (such as the human body) in low-dimensional space via a small set of parameters. An example of a widely used statistical body model for humans is Skinned Multi-Person Model (SMPL) [75], which is a vertex-based model with parameters to control pose and body shape. These parameters are learned from a large number of aligned 3D meshes of people of various body shapes in different poses. The SMPL model is used to generate realistic skinned models that cover the entire space of human pose and shape variation. Loper *et al.*use a vertex-based skinning approach with 6890 vertices forming a 3D mesh modeling the human body. The pose of an SMPL body model is defined by a vector $\theta \in \mathbb{R}^{72}$ which is an axis angle representation of relative rotations of twenty three body joints with respect to their parent joints in the skeletal kinematic tree. In addition to this, three parameters of θ provide the root joint orientation. A second parameter vector $\beta \in \mathbb{R}^{10}$ controls the body shape. The shape of an SMPL body is expressed as a linear function in which β are the shape coefficients for ten orthonormal principal components of shape displacements. Together, these two parameters express complete information on pose and shape of the 3D mesh. In our work we use the SMPL body models to obtain 3D mesh fits on real humans. Other significant prior work on statictical body modeling include [114, 115].

4.2.4 Non-deep Body Mesh fitting Methods

Bogo *et al.* [80] present SMPLify as the first method for 3D pose and shape estimation of human body from a single image. They use a CNN-based 2D pose estimator [116] to estimate body joint locations from images and thereafter, fit the SMPL model to the estimated 2D joints. Following this, Lassner *et al.* [81] propose an extension to the SMPLify method. They obtain 3D body model fits for several 2D human pose datasets by using a larger number of body landmarks. They released an annotated database of human images and corresponding estimates of kinematic and shape parameters and joint locations (UP-3D). Lassner *et al.* use *Decision Forests* to predict pose and shape parameters from 2D image coordinates of 91 landmark predictions in 2D. The parameters are predicted independent of each other for speed.

4.2.5 DCNN based Body Mesh fitting Methods

Method	Key -points	Seg- mentation	Optical -Flow	Mosh'ed real data
[85]	✓	✓		
[87]	✓	✓		
[1]	✓			✓
[84]		✓		
[83]	✓	✓	✓	
[86]	✓	✓		
Our Method	✓			

Table 4.1: Summary of mesh fitting methods and ground truth information utilized

Table 4.1 lists recently proposed DCNN-based methods for estimating the SMPL parameters from input images and their ground truth utilization. Most of the approaches use segmentation and/or semantic body-part segmentation during training. Reprojection error on 2D keypoints is computed by [87]. Reprojection error on both 2D keypoints and segmentation mask is computed by [85] and [83] (along with optical flow in [83]). The repojection loss on segmentation mask is used in [86] along with supervision on keypoints and body-part segmentation. The method prposed in [1] doesn't use segmentation ground-truth. But it utilizes Mosh'ed mesh parameter estimates for real data from multiple datasets for either direct supervision or training an adversarial prior. The methods listed above use some form of re-projection loss to train their network. We propose a re-projection free approach wherein the network learns to predict mesh parameters from supervision on synthetic data in conjunction with domain adaptation between real and synthetic domains.

4.2.6 Domain Adaptation for Deep Learning

Rozantsev [117] *et al.* propose a two-stream DNN for domain adaptation in deep networks without weight sharing. However, in practice, weight sharing reduces the number of parameters to be learned and GPU memory requirement while training. Joint optimization of deep networks on inputs from multiple domains has been studied in several recent works [95, 118–123]. Genova *et al.* [124] use unsupervised regression for 3D face modelling. Zhou *et al.* [2] use ADA for 3D keypoint estimation. Shu *et al.* [125] decompose real faces into shape and texture information in an unsupervised fashion. Cross domain detection is demonstrated for objects [126] and faces [127]. Domain difference between real and synthetic data is addressed for mutli-task learning [128] , depth estimation [128] and semantic segmentation [129]. Bak *et al.* [130] use synthetic humans for unsupervised person reidentification on real data. Synthetic face data, in conjunction with unsupervised fine-tuning on real faces is used to infer scene lighting by [131]. Bousmalis *et al.* [132] perform pixel level DA between real and synthetic data using GANs. Synthetic data for semantic segmentation on real videos using ADA is proposed in [133, 134]. Domain adaptation where classes in source and target domains do not completely match has been studied by [135, 136]. Chen *et al.* [137] retrieve 3D shapes based on 2D sketches using an adversarial learning approach. Rhodin *et al.* [138] learn a 3D representation from multi-view data for 3D human pose estimation. Pumarola *et al.* [139] perform unsupervised image synthesis in arbitrary poses. Other related research efforts include ADA via joint distribution optimal transport [140], multi-source ADA via multi-way adversarial learning [141], zero shot domain adaptation for classification [142], unsupervised real-to-virtual scene adaptation [123, 143–147], unsupervised learning via joint training of related tasks [148], attention alignment [149], activation matching [150], and adapting stereo network trained on synthetic data for depth estimation on monocular real videos [151]. A majority of recent UDA/ADA techniques using GANs focus on classification, retrieval, embedding and categorization. Fewer works address the more challenging tasks such as detection or regression. Pose regression is a more nuanced task since the network is learning to produce estimates in a continuous output domain and variation in human pose and body shape is relatively higher than many other rigid objects.

4.3 Method

In subsection 4.3.1 we present a derivative of the nested hourglass architecture proposed by Chu *et al.*in [99]. The DCNN in [99] is trained for 2D pose estimation from monocular images. We extend this architecture for 3D pose estimation by splitting the task into front and side view pose estimation from only a front view input. Front view pose (FV) estimates X and Y coordinates of joints whereas the side view pose (SV) estimates Z (and Y) coordinate. Jointly, FV and SV map X, Y and Z coordinates into one-

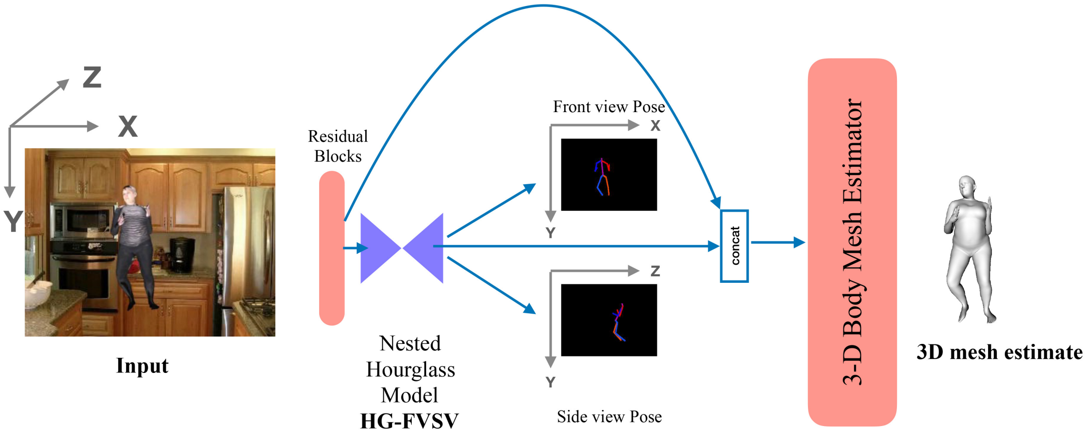


Figure 4.2: The proposed hourglass network designed for generating front view(FV)-side view (SV) attention maps and features for 3D mesh estimation.

hot-encoded pixel space. The network is trained using the mean square error (MSE) loss to detect body joints in 2D (FV) and implicitly learn the geometry of the human skeleton to infer pose in a different view (SV). We design three variants of the architecture to study their pose estimation performance on synthetic data.

Following this, in subsection 4.3.2 we discuss how we train the architecture proposed in subsection 4.3.1 (and another variant of hourglass network) as feature extractors for domain alignment of real and synthetic data. Based on features extracted from the feature extractor module, we train a SMPL mesh estimation network with adversarial losses and partial supervision to produce 3D mesh fits on input images of real humans. The two networks (feature extractor and SMPL mesh estimator) are trained together end-to-end. The output of the pipeline is a mesh representing the 3D pose and shape of the human.

4.3.1 3D pose estimation from a single image

The nested hourglass architecture trained by Chu *et al.* [99] is a highly accurate model for 2D pose estimation. Their network consists of 8 stacked hourglasses. Hourglass network for pose estimation was proposed in [152]. Nested hourglasses are hourglass modules with residual units hourglass units within them. In between the stacks are sets of convolutional layers that generate *Attention Maps*. The attention maps are subsequently refined by the following stacks. The output of the final stack is used to generate the predictions of joint locations. Further specifics on the nested hourglass design can be found in [99].

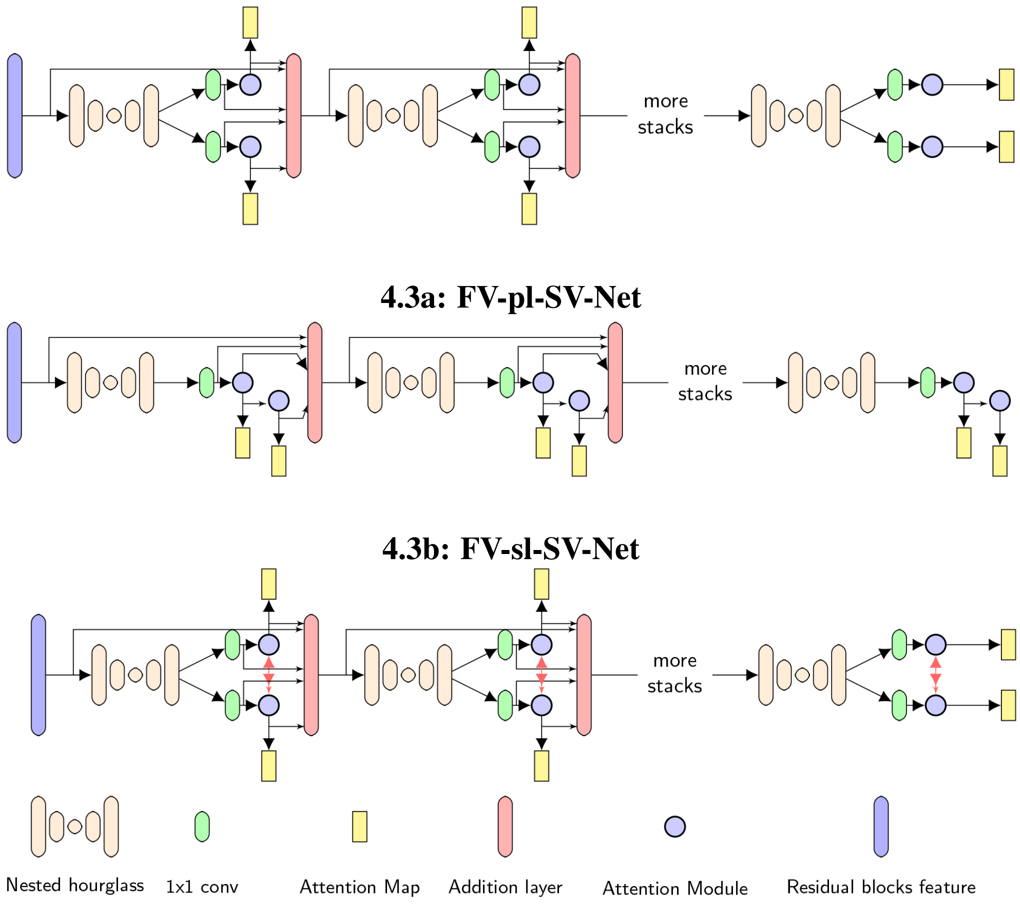
Modified Nested Hourglass We modify the stacked nested hourglass architecture to produce two attention maps instead of one. The first attention map predicts the locations of body joints in the front view (XY-plane) . The second predicts the locations of body joints in the side view (YZ-plane). The predictions are based on features generated by the hourglass modules from the input image in the XY-plane. The inputs to the network are unaltered. (The inputs to the network in [99] are image, scale of the person in the image and a center point on the body. Scale and body center information are used for cropping and transformation of label data.) The pathways predicting front and side view have identical design. The additional side view pose prediction is used to estimate the depth (Z coordinates) of body joints. Ground truth for supervising the network are attention maps with Gaussians centered around keypoint locations on 64x64 2D maps. The FV maps are generated from 2D keypoint ground truth. SV maps are generated from 3D keypoints that are scaled using 2D keypoints. The root joint in SV map is centered to coincide with the root joint in FV map.

4.3.1.1 Variations in Network Design

We vary the number of stacks in the hourglass network and observe its effect on the accuracy of the predicted 3D pose. In the original 8-stack design by Chu *et al.* [99], the last four stacks generate ‘part attention maps’ (*i.e.*the network learns spatial correlations for each joint location estimate separately). In our experiments with the 4-stack network, only the last stack learns part based correlation. The 8-stack network, on the other hand, learns spatial correlations for individual joint locations over stacks 4 to 8. In order to observe the interdependence in inference of front view (FV) and side view (SV) poses, we train networks with three variations of pathways predicting front and side view pose. Figure 4.3 presents these different versions diagrammatically.

FV-pl-SV-Net In the first design 4.3a, the output of each hourglass module forks into two separate pathways (identical in design) which predict two attention maps (XY-plane and YZ-plane). We train a 8-stack version of this architecture.

FV-sl-SV-Net In the second design 4.3b, the hourglass module output passes through 1x1 convolution modules. Following this, a series of convolutions, together composing the attention module, generate the front view (XY-plane) attention map. This attention map is subsequently used to generate features from which the side view attention map is predicted. The YZ-plane pose inference is based on the features derived from the inferred



4.3a: FV-pl-SV-Net

4.3b: FV-sl-SV-Net

4.3c: FV-Ytied-SV-Net

Figure 4.3: Bottom row shows legends. The network has six residual blocks [3] to extract features from input image to feed into the first hourglass module. The pathways generating the Attention maps (Attention module and 1x1 convolution layers) are ordered differently in 4.3a (and 4.3c) vs 4.3b. The diagrams have been simplified to highlight the main points of difference between the the serial and parallel designs. The only difference between 4.3a and 4.3c is an additional loss term minimizing the difference in Y-coordinate predicted in XY and YZ attention maps.

XY-plane attention maps. This is thus the serial design with a direct dependence of side view pose estimation on the front view pose estimation.

FV-Ytied-SV-Net This design is identical to FV-pl-SV-Net with an additional regularizing term. Since the pose predictions are made in the XY-plane and YZ-plane, we introduce a third loss function between the two maps tying together the predicted Y coordinates in the two attention maps. We sum the attention maps along X and Z axes and minimize the difference (attention distribution along Y axis). This network, is shown in Figure 4.3c. We train 4 stack versions of FV-sl-SV-Net and FV-Ytied-SV-Net for comparison.

All networks output a list of pairs of front view and side view attention maps. Each pair comes from one hourglass stack in the network. Additionally we extract the concatenated feature maps from the last stack (right before the attention module) and output of the last residual block (right before the first hourglass module). In case of the FV-Ytied-SV-Net, we compute a sum along the columns of every row in the generated attention maps (FV and SV) from the last stack and return the difference. The feature maps are used to estimate a 3D mesh as discussed in subsection 4.3.2.

4.3.1.2 Loss Functions for 3D pose estimation

We use the mean square error (MSE) loss to train the network for 3D pose estimation. We provide intermediate supervision, *i.e.*loss is computed on attention map (both XY-plane and YZ-plane) output of every stack. The ground truth 3D joint location labels are converted to 2D Gaussian maps (in XY and YZ planes respectively) with means on the corresponding joint locations in those planes. Let H_{XY} and H_{YZ} be the K channel attention maps, in XY-plane and YZ-plane respectively, where K=number of joints.

$$L_{XY} = \sum_{k=1}^{k=K} \sum_{p \in H_{XY}} ||\hat{H}_{XY}(k,p) - H^*_{XY}(k,p)||_2^2 \tag{4.1}$$

$$L_{YZ} = \sum_{k=1}^{k=K} \sum_{p \in H_{YZ}} ||\hat{H}_{YZ}(k,p) - H^*_{YZ}(k,p)||_2^2 \tag{4.2}$$

In the network design shown in 4.3c, we introduce a third prior term in the loss function which minimizes the difference in predicted Y coordinate values for the XY-plane (front) and YZ-plane (side) attention maps. This loss acts as a regularizing term and ensures that predicted Y values are consistent across the two views.

$$\begin{gathered} \vec{Y_1} = \sum_X (\hat{H}_{XY}); \vec{Y_2} = \sum_Z (\hat{H}_{YZ}) \\ L_{Ydiff} = ||\vec{Y_1} - \vec{Y_2}||_2^2 \end{gathered} \tag{4.3}$$

4.3.2 3D Body Mesh Estimation

We propose a pipeline which is end-to-end trainable for pose and shape inference from a single input image. The design is broadly based on an idea that utilizes domain adaptation to learn inference of synthetic model parameters for real data. Modularity of the pipeline allows for defining different sub-tasks based on the problem and experimenting with different versions of deep architectures for the sub-tasks. Source domain data is generated from synthetic models that are developed based on an initial set of real data. Adaptation between synthetic and new real datasets is useful in extending the richness of synthetic models (in this case by learning new fits on new poses from real data) and lead to more detailed understanding of real data (complete 3D information from a single 2D input). Our method assumes the availability of synthetic imagery with corresponding

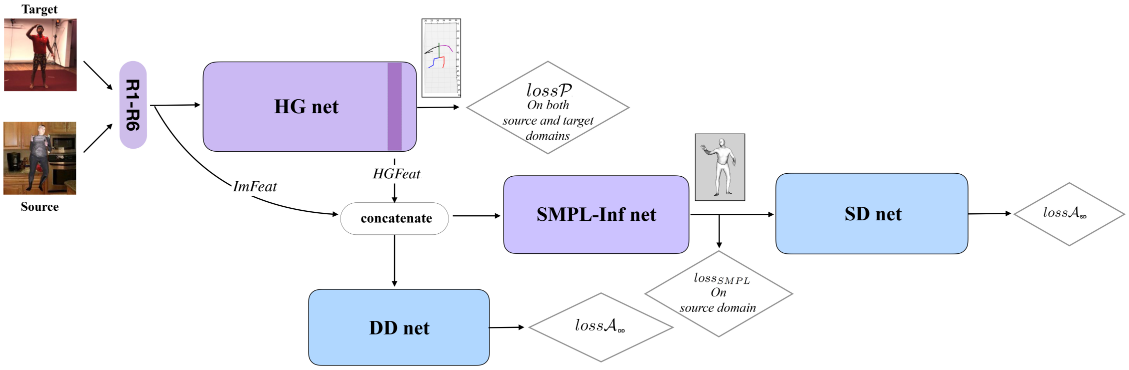


Figure 4.4: Deep domain adapting networks for 3D mesh inference from a single RGB image. HG net generates domain aligned features via joint supervision on real and synthetic samples on a 3D pose estimation task. The SMPL-Inf net generates 3D mesh estimates based on features from HG net. Domain discriminator DD net provides adversarial supervision on features generated by HG net. SMPL mesh discriminator (SD net) discriminates between actual SMPL meshes and the SMPL-Inf net predictions.

ground truth synthetic parameters for partial supervision of the pipeline. It does not rely on any estimates of synthetic parameters for real data (obtained via pre-existing methods) or any additional annotations other than the 2D and 3D annotations corresponding to 16 body joints. The scale of the person in the image is used as an input. The scale is computed from the ratio of skeleton size in 2D pixel space and in 3D.

4.3.2.1 SMPL Parameters

SMPL is a vertex-based mesh model with parameters to control pose and body shape. The SMPL model is used to generate realistic skinned models that cover the entire space of human pose and shape variation. The mesh is parameterized by a *pose vector* - θ of length 72 and a *shape vector*-β of length 10. The *pose vector* is the axis-angle representation of 23 joints with respect to their parents in the skeletal kinematic tree, and orientation of the root joint. The *shape vector* contains the linear coefficients defining shape of the body. A blend shape function and a pose-dependent blend shape function together define the displacements of mesh vertices (in rest pose) based on the shape parameters β. The θ vector is a set of quaternions expressing pose (as joint angles) based on which a rotation is applied to the mesh vertices by a blend skinning function. Further details can be found in [75]. *Our objective is to estimate θ and β from a single 2D image of a real human*. This task can be directly supervised in synthetic data generated using SMPL models in arbitrary body shapes and poses (poses of body models in the synthetic dataset are drawn from real humans).

4.3.2.2 Feature Extractor

We use a deep feature extractor to project samples from both source (synthetic) and target (real) domains onto a shared latent space. The feature maps are aligned by the feature extractor via joint optimization on a *shared common task*. We provide supervision on the shared common task using annotations that are available for both source and target domains. We use 3D keypoints for domain alignment. We experiment with two versions of the feature extractor network both of which are variants of stacked hourglass networks.

HG Net version I : **HG-FVSV** In section 4.3.1 we describe an extension to the nested hourglass architecture for 3D pose estimation via keypoint regression. We study serial and parallel variants of the architecture and a parallel one with an additional regularizing term. Based on our observations on the performance of these designs on synthetic samples, we choose a *two stack version of FV-Ytied-SV* as a feature extractor in our mesh estimation pipeline. The feature extractor (HG-FVSV), the mesh estimator and the adversarial network(s) are all trained simultaneously starting with random normal weights. The number of stacks is reduced to limit the model size so that a reasonable size of mini-batch can be fit on GPUs during training.

HG Net version II : **HG-3D** We begin with a DNN with trained weights for 3D pose estimation [62]. The hourglass network for 3D pose inference from 2D images is already trained on Humans 3.6M dataset. We refer to this as the HG-3D net.

For SMPL parameter inference, we use the features from the HG net. The input to the HG net is a 256x256 RGB image centered on the human. The input passes through a 7x7 convolution layer and 6 residual blocks (with two max-pooling layers). These layers together constitute the initial feature extraction block. The output of these layers are 256 channel 64x64 feature maps. We refer to these maps as *ImFeat*. *ImFeat* passes through 2 stacks of Hourglass modules which extract features useful for inferring 3D pose and progressively refine the outputs of previous hourglass modules. We take the output features of the last hourglass module. We refer to this as *HGFeat*. *HGFeat* is a set of 256x64x64 feature maps. Final 2D pose prediction and depth regression modules of the HG net utilize *HGFeat*. We concatenate *ImFeat* and *HGFeat* to obtain 512x64x64 feature maps (*Feat*).

4.3.2.3 Mesh Estimator

The mesh estimator part of the pipeline consists of two residual blocks and fully connected layers. It uses *Feat* (concatenation of ImFeat and HGFeat) as input to infer SMPL mesh parameters.

SMPL-Inf Net: We use two variants of the SMPL-Inf Net with two feature extractors (HG-3D and HG-FVSV). The first residual block (2 layer basic block with stride=2) [3] of the SMPL-Inf Net receives 512 channel 64x64 feature maps from the feature extractor. This is followed by another basic block (stride = 1). The first version of the mesh estimator (SMPL-Inf Net) is trained end-to-end with HG-3D. The output of the second basic block is average pooled and passes through two fully connected layers that return estimates of θ, β. The second version is trained with HG-FVSV. In order to reduce the overall memory bandwidth we reduce the size of the fully connected layers. For this we use a point-wise convolution layer (with batch-normalization and ReLU non-linearity) at the output of the second basic block to reduce dimensionality. This substantially reduces the number of parameters of the fully-connected layers and the memory foot-print of the models. This in turn enables larger mini-batches during training. We will refer to this second version of the mesh estimator module as the **SMPL-Inf1x1** net.

The hourglass stacks in HG-FVSV are nested and therefore have more parameters than the basic hourglasses in HG-3D net. HG-3D net is initialized with 3D pose estimation weights for Humans3.6M. HG-FVSV net is initialized with random normal weights. Each feature extractor is thereafter jointly trained with the SMPL-Inf net on mixed batches of real and synthetic samples. HG-FVSV has more hidden layers (than HG-3D) which contributes to higher model capacity. On the other hand the SMPL-Inf net trained with HG-3D has higher model capacity than the SMPL-Inf1x1 net (which uses point-wise convolution to reduce feature dimensionality and corresponding sizes of fully connected layers). Therefore the two versions of the pipeline : HG-3D + SMPL-Inf and HG-FVSV+SMPL-Inf1x1 distribute model capacity differently among the two tasks of domain alignment and subsequent mesh estimation.

4.3.2.4 Partial Supervision

We use mini-batches comprising of real as well as synthetic samples for training the HG net and SMPL-Inf net end-to-end. For real data, we compute 3D pose loss from 3D ground truth joint locations. For synthetic data, we compute 3D pose loss as well as regression loss (mean square error) on SMPL parameter predictions. In addition to these, we apply adversarial losses from two discriminator networks - namely 1) Domain Discriminator and 2) SMPL Discriminator. HG net receives supervision on both real and synthetic samples. In contrast, the SMPL-Inf Net is *partially supervised* only on the synthetic samples in each mini-batch. Adversarial losses are computed on all samples. The SMPL parameter regression loss is:

$$loss_{SMPL_{\varphi1,\varphi2}} = \lambda_{smpl} * ||SMPLInf_{\varphi_2}(Feat_{syn_{\varphi_1}}) - targetSMPL_{syn})||_2 \quad (4.4)$$

4.3.2.5 Shared common task

The HG net simultaneously optimizes both source and target domain losses via a shared common task. We align the features of real and synthetic domains, via this shared task, such that features extracted for similar pose are similar for both domains. We use ground truth 3D joint locations from Humans 3.6M and SURREAL datasets to train the HG net. The network learns to infer 3D pose for both datasets simultaneously. We utilize the extracted features simultaneously for 3D pose estimation and to infer SMPL parameters using the SMPL-Inf net. The parameters of the SMPL-Inf net are optimized based on ground truth labels for synthetic data and adversarial loss on outputs from real samples. Inference is based on the features produced by the HG net which is jointly optimized on both domains.

$$\hat{Pose}_{real_{\varphi1}}, Feat_{real_{\varphi1}} = HG_{\varphi_1}(Im_{real}) \quad (4.5)$$

$$\hat{Pose}_{syn_{\varphi_1}}, Feat_{syn_{\varphi_1}} = HG_{\varphi_1}(Im_{syn}) \quad (4.6)$$

$$
\begin{aligned}
loss\mathcal{P}_{\varphi_1} = \lambda_{pose} * (\alpha * \mathfrak{L}_{3Dpose}(\hat{Pose}_{syn_{\varphi_1}}, target3D_{syn}) \\
+ \beta * \mathfrak{L}_{3Dpose}(\hat{Pose}_{real\varphi_1}, target3D_{real}))
\end{aligned} \quad (4.7)
$$

$\mathfrak{L}_{3Dpose}$ is implemented in the same way as [62]. α,β can be used to assign different weights to real and target domain samples.

4.3.2.6 Adversarial Learning

We use two discriminator networks : 1) Domain Discriminator and 2) SMPL Discriminator. The Domain Discriminator (DD) net uses the same input as the SMPL-Inf net (*i.e.*concatenation of *ImFeat* and *HGFeat*). It has the same structure as the SMPL-Inf net barring the last fully connected layer that predicts a class (0 for synthetic, 1 for real) (instead of regressing joint angles and shape parameters as is done by the SMPL-Inf net). The SMPL Discriminator (SD) net consists of fully connected layers which discriminate between ground truth SMPL parameters from the synthetic dataset vs. SMPL parameter estimates for real data from the SMPL-Inf net generator network. Discrimination is performed in the vector space (72D + 10D=82D) of SMPL parameters (pose in axis-angle representation of 23 joints, 10 shape parameters). Adversarial losses for the HG net and SMPL-Inf net are computed as:

$$
\begin{aligned}
loss\mathcal{A}_{\varphi_1,\varphi_2} = \lambda_{gan1} * \mathbf{E}_{Feat\sim HG_{\varphi_1}(Im)}[DD(Feat_{real})^2 \\
+ DD(Feat_{syn})^2] \\
+ \lambda_{gan2} * \mathbf{E}_{\theta\beta\sim SMPLInf_{\varphi_2}(Feat)}[\\
SD(\theta\beta_{real})^2 + SD(\theta\beta_{syn})^2]
\end{aligned} \quad (4.8)
$$

The losses computed for the discriminator networks:

$$loss\mathcal{DD}_{\varphi3} = \mathbf{E}_{Feat \sim HG_{\varphi_1}}[(1 - DD_{\varphi3}(Feat_{real}))^2 + DD_{\varphi3}(Feat_{syn})^2] \quad (4.9)$$

$$loss\mathcal{SD}_{\varphi_4} = \mathbf{E}_{\theta\beta \sim SMPLInf_{\varphi2}(Feat_{real})}[(1 - SD_{\varphi_4}(\theta\beta_{real}))^2] + \mathbf{E}_{\theta\beta \sim p_{synGT}}[SD_{\varphi_4}(\theta^* \beta^*)^2] \quad (4.10)$$

4.3.2.7 Weighted Losses

It is important to assign proper weights to the losses during the training process. For real data, there is no ground truth SMPL parameter for computing regression loss. Assigning higher weight to the 3D pose estimation task on real data is critical for ensuring that the HG net produces pose features that are discriminative enough to correctly regress SMPL pose parameters from. 3D pose estimation is a shared task between real and synthetic domains, which directs the HG net towards producing similar features for similar poses, irrespective of the domain of the input.

The HG net and SMPL-Inf net optimization is defined as:

$$\underset{\varphi_1,\varphi_2}{\text{argmin}} \quad loss\mathcal{P}_{\varphi_1} + loss\mathcal{A}_{\varphi_1,\varphi_2} + loss_{SMPL_{\varphi_1,\varphi_2}} \quad (4.11)$$

Domain Discriminator and SMPL Discriminator networks are optimized as:

$$\underset{\varphi_3}{\text{argmin}} \quad loss\mathcal{DD}_{\varphi_3} \quad (4.12)$$

$$\underset{\varphi_4}{\text{argmin}} \quad loss\mathcal{SD}_{\varphi_4} \quad (4.13)$$

4.4 Experimental Results

4.4.1 Datasets

SURREAL consists of more than 6.5 million frames. Joint locations in 2D and 3D, depth maps, optical flow and segmentation masks are provided. Frames in this dataset are of resolution 240x320. From this we crop square tiles centered on the synthetic human. The resulting input is of lower resolution. It is then resized to a 256x256 input. There is wide diversity in background and in some cases the poor contrast between foreground and background is challenging. The training set consists of roughly 3M frames. Our models are trained on still monocular images randomly picked from these sequences. The dataset provides 3D locations of 23 joints. Following several pose prediction models such as [51, 52, 62], we only predict locations of 16 body joints. Sequences are generated based on MoSh'ed [88] joint estimates from CMU Mocap dataset [82]. Significant variations in body shape are present among subjects since shapes are randomly assigned.

Humans3.6M consists of real video sequences with seven subjects, thirteen actions, two sub-actions per category, each shot from four different viewing angles [72]. Under **Protocol 1**, we train on five subjects S1, S5, S6, S7, S8 and all four cameras. We validate the trained network on all action sequences performed by subjects S9 and S11, all 4 cameras. We use the 14 LSP joints - ankle, knee, hip, wrist, elbow, shoulder, neck and head joint predictions for evaluation.

MPI-Inf-3DHP While Humans3.6M consists of images captured only in an indoor location, MPI-Inf-3DHP contains two test subjects' video sequences captured outdoors. We adapt our networks by jointly training on synthetic data and MPI-3D-Inf [66] training images. We use subjects S1-S7 for training, and same cameras used in [65]. On MPI-Inf-3DHP we study the effect of including DD net in training. We also predict 3D meshes on subjects TS5, TS6 captured in outdoor locations.

4.4.2 Supervised training for 3D pose estimation

We train the three modified versions of stacked nested hourglass architecture [Figure 4.3]. We compare the performance of these three variants of HG-FVSV. Visual results of 3D pose estimates on synthetic data are presented in Figure 4.5. The training and validation curves are included for comparison in Figure 4.6. The front view pose estimation performance is very close for all networks in training and validation. Additional depth is useful for the more challenging task of inferring side view pose from the input image. In both front view and side view prediction, the 8Stack-FV-pl-SV network outperforms the 4-stack networks. However, a larger gap is observed between training and validation curves in case of the 8Stack-FV-pl-SV network. The 4Stack-FV-Ytied-SV Net has the closest training and validation performance - suggesting that the additional regularizing loss term improves generalization capability. The 4Stack-FV-Ytied-SV Net outperforms 4Stack-FV-sl-SV in front view pose estimation and matches with the 8Stack-FV-pl-SV Net. The results on the test set for 8 stack FV-pl-SV Net are tabulated in Table 4.2. We use 2D PCKh@0.5 scores in XY and YZ planes respectively for evaluation.

4.4.3 Adversarial Domain Adaptation for 3D mesh estimation

We begin the training with initial weights for the HG-3D net trained on Humans3.6M 3D pose estimation task [62]. We use the initialization proposed in [153] for the SMPL-Inf net, HG-FVSV and the two discriminator networks. The generators (the HG net and SMPL-Inf net) and adversaries (SD net and DD net) are trained together. Initial loss from the HG-3D for the 3D pose estimation on the Humans3.6M dataset is a much smaller compared to other losses. The SMPL parameter regression loss on SURREAL dataset is relatively much higher. Not assigning a high enough weight to pose loss results in the feature extractor network quickly forgetting 3D pose estimation while trying to learn the mesh fitting task. To avoid the scenario of *catastrophic forgetting*, we assign a higher weight to $loss\mathcal{P}_{\varphi_1}$ on target domain (real images). We train the HG-3D net, which already computes features for 3D pose estimation, to produce features that are good for SMPL parameter regression, without forgetting how to estimate pose. HG-FVSV is trained starting from a random initialization [153]. Initial pose loss value for the HG-FVSV is high.

Since we aim to use pose-based feature alignment for SMPL parameter regres-

Set	Front Pose (XY-plane)	Side Pose (YZ-plane)
Run 0	98.52	84.05
Run 1	98.49	83.97
Run 2	98.40	83.86

Table 4.2: 8-stack FV-pl-SV Net : PCKh@0.5 test score on SURREAL test set.

sion, the HG net must perform well on both source (synthetic) and target (real images). The hyper-parameters for weighing different losses are set by trial so that losses in both domains are roughly equal. Network parameters are optimized using the RMSProp algorithm [154]. RMSProp uses a per-parameter adaptive learning rate which is based on the gradient of the loss value with respect to that parameter. It is essential for the parameters of the feature extractor to reach a stable point with good domain alignment (stable and comparable performance on the shared task for both datasets). The weights (λs) assigned to the different losses define the shape of loss surface. The models are implemented in PyTorch [155] and trained using a multi-GPU framework with a batchsize of 24. Figure 4.8 shows some visual results of ADA-based mesh fitting on the Humans3.6M dataset. Quantitative evaluation of the predictions can not be directly obtained since there exists no ground truth SMPL parameters for real humans. The quaternion + shape parameter estimation error (MSE) on SURREAL training samples at the stopping point was roughly 0.15. Following other works, we evaluate our predictions by regressing 3D keypoints from mesh vertices and computing the MPJPE error on these keypoints. Table 4.3 provides a comparison against the latest methods. Our model, Tung *et al.* [83] and Kanazawa *et al.* [1] predict 3D mesh, whereas Zhou *et al.* [2] predict only keypoints. Kanazawa *et al.* [1] use re-projection loss for supervision and train their adversarial prior based on Mosh'ed data on MoCAP sequence of multiple datasets including Humans3.6M and CMU [82]. Tung *et al.* [83] use synthetic data generated based on Humans 3.6M dataset. In contrast, we use synthetic data based solely on CMU MoCAP sequence. Generating synthetic data based on more pose datasets such as Humans3.6M can further improve the performance our model. We adapt our networks to produce results on

MPI-Inf-3DHP. We use training data from seven subjects S1-S7. Weights are initialized from models trained on Humans3.6M and SURREAL, and the networks are jointly optimized on MPI-Inf-3DHP and SURREAL images. Visual results are shown in Figures 4.8 and 4.11

4.4.3.1 Role of Domain Discriminator

In the presence of a shared common task to enforce constraint on the HG net across both domains, it is debatable if the domain discriminator DD net is useful in training. We conduct two training experiments to test this hypothesis. In the first experiment, we start with the HG, SMPL-Inf and SD net weights previously trained on SURREAL+Humans3.6M data upto a certain epoch. The DD net is initialized from scratch. The set of networks is trained together on a mix of MPI-Inf-3DHP and SURREAL. In the second experiment, we start with an identical setting, except with *zero weight on adversarial domain discriminator (DD net) loss*. The training curves are plotted together in Figure 4.9. We observe that including the DD net in training is helpful in the initial stages of training. Although the DD net initially boosts learning of the HG net in the new domain, eventually the two domains become indistinguishable. The common task is essential in ensuring that features from the HG net are aligned for similar poses across real and synthetic domains. Pose-based discriminative features are useful for mesh parameter regression and can help avoid a possible scenario of mode collapse in the target domain. It may be useful to include the DD net initially in the training process and subsequently getting rid of it once the HG net has reached a jointly optimal point for both domains.

This can reduce memory requirement in training. We do not use the DD net while training the HG-FVSV+SMPL-Inf1x1 pipeline to be able to fit mini-batches (of the same size as HG-3D+SMPL-Inf) on GPUs.

4.4.3.2 Supervision via Re-projection

We compare the initial result of ADA-based fitting to those from a network (with the same architecture), trained further to minimize the re-projection loss on 3D keypoints. Row 1 in Table 4.4 provides the performance of the network before supervised training with re-projection loss is performed.

The SMPL-Inf net predicts the SMPL parameters, which is used to compute the vertices on the SMPL body. We regress the 2D keypoints from the predicted SMPL vertices. We compute the mean square error (MSE) in re-projection using ground truth 2D joints. The re-projection loss is evaluated only on real samples. Table 4.4 provides the quantitative performance for two types of training. In our experiments, supervision with re-projection loss increased the MPJPE error. Moreover the network starts producing abnormally contorted mesh predictions.

4.4.3.3 Effect of Multi-view self-supervision

In the Humans 3.6M dataset, every sequence is recorded by four cameras at different viewpoints. 3D joint angles are same for the same pose, irrespective of the view. We utilize this fact to enforce an additional constraint on the network *during training time*. We compute the L_2 loss on the SMPL joint orientation (*except root joint*) and shape predictions for pairs of views of the same scene. The SMPL-Inf net now jointly optimizes in both the domains (real and synthetic). In case of multi-view training, in every mini-batch, we do a pass on a single view of synthetic data and two alternate views of real data. We compute the 3D pose regression loss and SMPL parameter regression loss in source domain (synthetic). On target domain (real images), we evaluate the 3D pose regression loss on the *primary view* and the L_2 distance between inferred SMPL parameter vectors (Kinematic parameters for all but the root joint) between the *primary and alternate views*.

$$\begin{aligned} loss_{mv_{\varphi 1,\varphi 2}} = \lambda_{mv} \cdot ||SMPLInf_{\varphi_2}(Feat_{realV_{1_{\varphi_1}}}) \\ - SMPLInf_{\varphi_2}(Feat_{realV_{2_{\varphi_1}}})||_2 \\ \text{where} \lambda_{mv}[\text{root joint}] = \mathbf{0} \end{aligned} \quad (4.14)$$

Since the number of real samples loaded per mini-batch is twice now (pairs), we save some GPU memory by not using the domain discriminator DD net. In Table 4.5, we summarize the performance of the HG net on the pose estimation task. We observe that multi-view supervision helps with the 3D pose estimation task on target domain by slightly improving the MPJPE score. However, *in the absence of DD net*, this quickly leads to *mode collapse*. All outputs on real data start getting mapped to the mean SMPL pose. The SMPL-Inf net is solely supervised on synthetic data. Minimizing the cross-view difference in mesh parameter prediction leads the network to map all real inputs to the mean SMPL pose. Thus multi-view supervision deteriorates the mesh parameter prediction performance despite improving 3D pose estimation score.

4.4.3.4 Low-level vs hourglass features

In our mesh estimation pipeline, we extract features from the HG net at two points - first from the last residual block before the hourglass stacks (*ImFeat*) and second from the output features of the last hourglass block (*HGFeat*). Each of these yields 256 channel 64x64 feature maps which are concatenated and used as input to the SMPL-Inf net. This combines low-level image features from the earlier layers with high level pose features from the last hourglass. Deeper layers of hourglass stacks are likely to yield more domain invariant features due to joint optimization on the pose task.

In this subsection we experiment by removing one of these sets of features and using the other as the only input to the SMPL-Inf Net. In each case the input has 256 channels (instead of 512). The SMPL-Inf net has identical designs in both setups and weights are randomly initialized. The HG-net is initialized with weights learned via joint-optimization in prior experiments. Figure 4.10 shows the training curves over first two epochs of training. The plot indicates that the SMPL-Inf net learns faster with the domain aligned features from the hourglass module than low level features from early layers (ImFeat). As highlighted in some examples in Figure 4.11, the mesh prediction performance is not entirely dependent or correlated with pose prediction. Rather, these two tasks are learned together as a multi-tasking problem. The main role of 3D pose supervision is to produce domain invariant features. This enables mesh parameter inference on real data without direct/indirect supervision from any Mosh'ed estimates.

4.5 Discussion

From our experimental observations, ADA-based training for 3D mesh fitting is found to be quite stable. Validations performed at various stages on training *always yield anthropometrically plausible predictions*. Avoiding minimization of reprojection error makes training faster by avoiding the re-projection step and simultaneously keeps the objective function from converging into local minimas which produce impossible body shapes. More epochs of training lead to lower training error in mesh parameter estimation among SURREAL samples and greater degree of pose accuracy during validation. Even in the case of certain extreme poses (*e.g.* on the ground, large degree of occlusion of body parts), the joint angle predictions for occluded body parts are in reasonable ranges. Shape prediction is somewhat affected by clothing and view angles. SURREAL covers a wide variety of body shape. The synthetic models in SURREAL are rendered with skin tight clothing that reflect body shape exactly. In comparison, there is much less variation in body shape among subjects in the target domain (real humans). In real human datasets, subjects wear loose clothing, which hang from their body, especially in certain bent poses. *Loose clothing leads to less accurate shape predictions* in some cases. We observe a bias towards the heads of predicted SMPL bodies being oriented upwards even though the subject is facing downwards. This can be attributed to the fact that we use only one head keypoint from the ground truth joints and no additional face key points. Modifying the HG net to predict additional face keypoints can provide the SMPL-Inf net additional supporting information for better inference of head joint. Another source of error stems from a bias among synthetic samples towards bent knee joints. Poses in synthetic samples are based on Moshe'd [88] data from real humans which may be a source of this bias. Although mesh parameter prediction is correlated with 3D pose estimation, and derived from the same feature maps computed by the HG net, it is not entirely dependent upon 3D pose prediction accuracy. This is evident from several visualizations presented in Figure 4.11 alongside some 3D pose predictions by the HG net. In some cases the 3D mesh estimate provides a more natural looking pose than the skeleton estimated by the HG net. Certain domain specific poses in Humans3.6M are missing in SURREAL owing to the fact that SURREAL draws from a different pose dataset. This domain gap contributes negatively to the adaptation process. There is a fairly wide gap in MPJPE error of 3D pose ouputs from the HG net and 3D joints derived from mesh predictions of the SMPL-Inf net. This gap can be closed by enriching synthetic samples with a wider variety of poses. Since synthetic samples are the sole source of supervision on the SMPL-Inf net, this variety is imperative for it's performance. Locations of certain body joints (*e.g.* wrist, ankle) are more prone to mispredictions, even in 2D pose estimation tasks [51,52,76,78]. There are some cases of left and right confusion and sometimes there are inaccuracies in depth estimation of certain body joints in the predicted SMPL mesh. However natural looking poses are maintained despite these estimation errors. In case of occlusions due to viewpoint, sometimes the network hallucinates a hidden limb. These cases are depicted through some examples in Figure 4.11. Even though both Humans3.6M and MPI-Inf-3DHP training samples are set in indoor locations, we obtain reasonable estimates on subjects TS5 and TS6 of MPI-Inf-3DHP test set where samples were collected outdoors. Although we use 3D pose for domain alignment, mesh predictions are derived from intermediate features and not directly dependent on 3D pose supervision. Moving past key point re-projection loss with the help of ADA is a promising direction since a) It avoids implausible shape and pose predictions b) It removes direct dependence on keypoint ground truth for supervision.

4.6 Conclusion

We propose a bottom-up 3D mesh fitting method from a single image by learning directly from synthetic data. We design a modular pipeline based on adversarial domain adaptation. We experiment with different variants of deep architectures as modules in the proposed pipeline. We obtain reasonable single shot fits of 3D body meshes on images of real humans with partial supervision. Domain alignment is achieved via joint optimization on synthetic and real data using a shared common task. The training strategy is single stage and end-to-end. We replace fine tuning with knowledge transfer via ADA. Good quality single shot predictions may be used as an initialization for better/faster fit via iterative regression techniques. We propose a no-frills training strategy utilizing pose-based aligned feature extraction in real and synthetic domain followed by SMPL parameter inference from the aligned features. There is no dependence on MOCAP data, Mosh'ed estimates, segmentation, semantic body-part segmentation or temporal information. The method uses minimal ground truth annotations on real data. It does not rely on prior methods to produce domain specific mesh estimates on real data for training. The model outputs skeletal joint positions in 3D in conjunction with a 3D mesh. Performance may be enhanced by introducing synthetic samples corresponding to target domain specific poses missing in the synthetic (source) data and getting rid of biases introduced by some poses present in large number in source domain. A good variety of appearance, shape and a wide range of pose in the source domain is essential. This ADA-based training technique can be extended to learning other synthetic model parameter prediction for real data via other shared common tasks for domain alignment(*e.g.* 2D keypoint estimation).

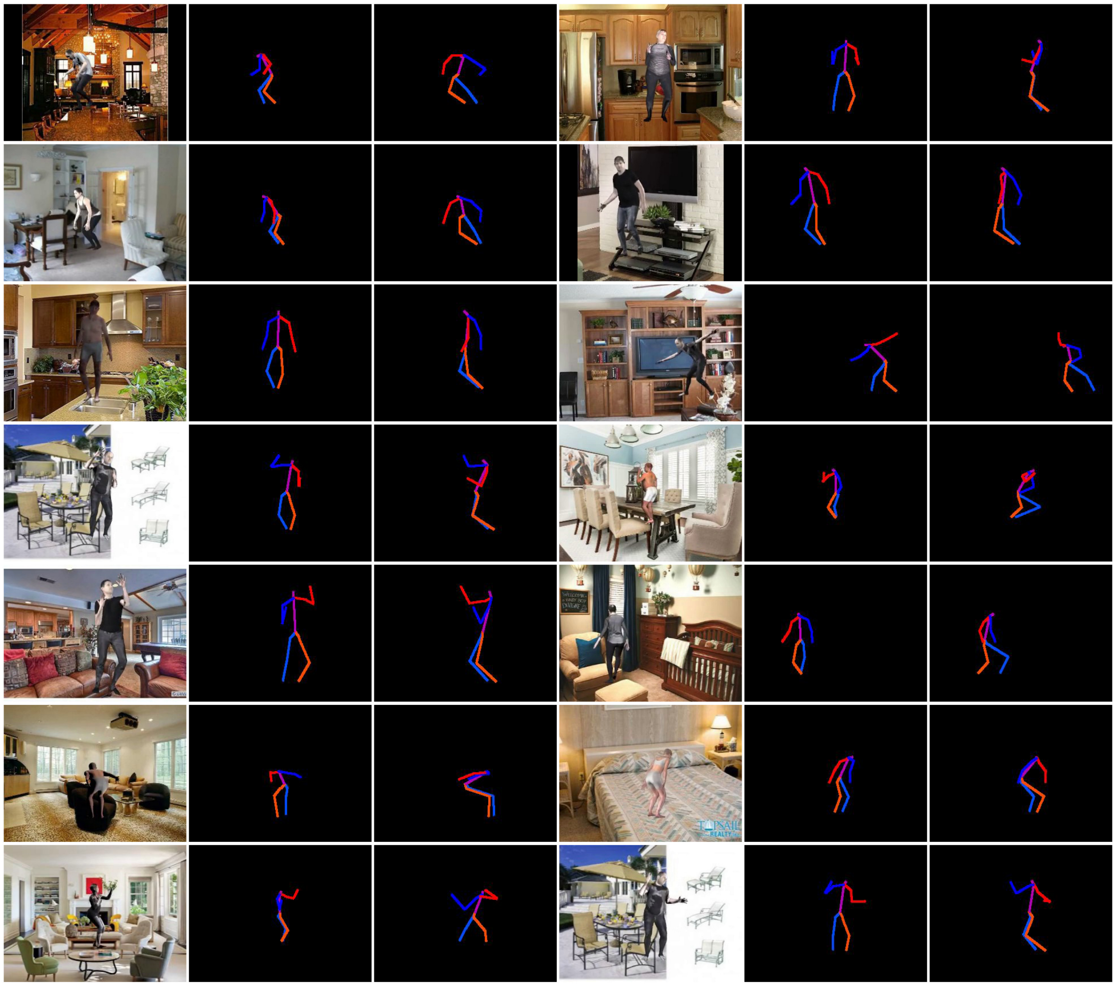


Figure 4.5: Visual results of 3D pose prediction on SURREAL synthetic data : 8 stack FV-pl-SV Net. Red/orange color is used for the left half of the body and blue for the right half. First and fourth columns show input images, second and fifth show front view pose estimate, third and sixth show side view pose estimate.

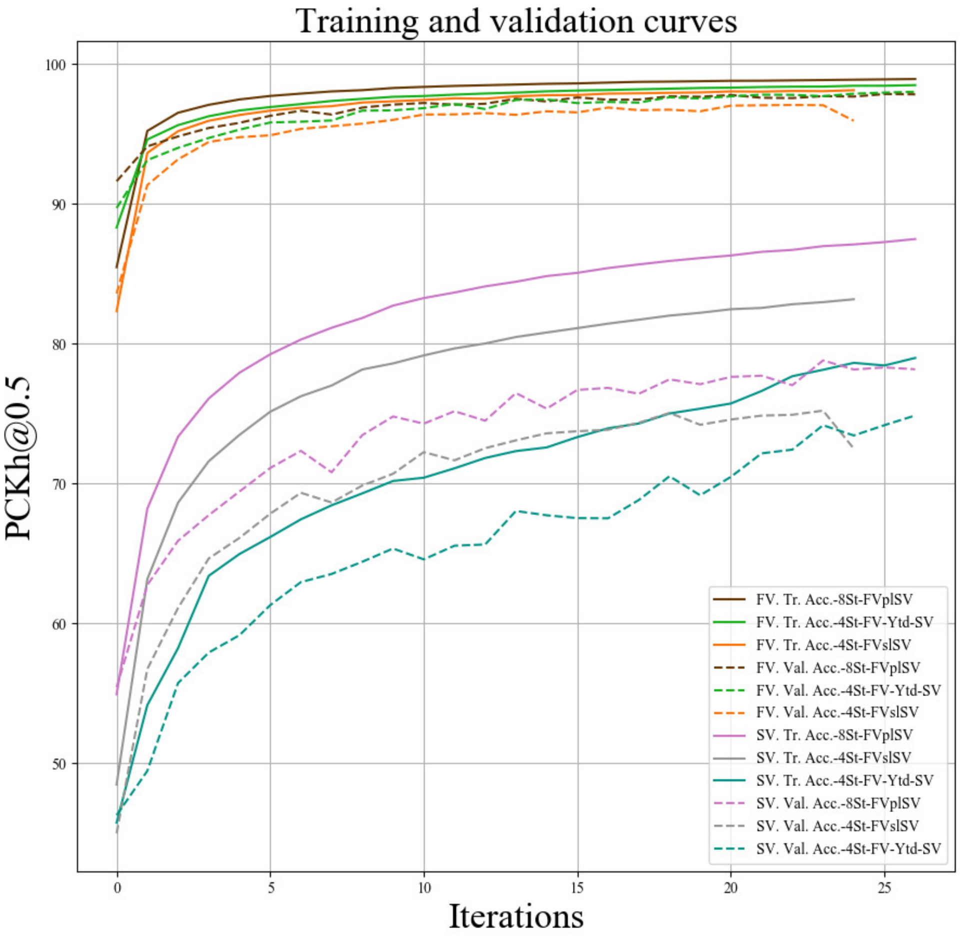


Figure 4.6: 3D Pose Prediction : Training and validation curves for 8-stack FV-pl-SV, 4-stack FV-sl-SV and 4-stack FV-Ytied-SV. More stacks greatly improve size view pose prediction performance albeit front view pose prediction performance is nearly same. Parallel design with additional regularizing term (FV-Ytied-SV) shows smaller gap between training and validation curves.

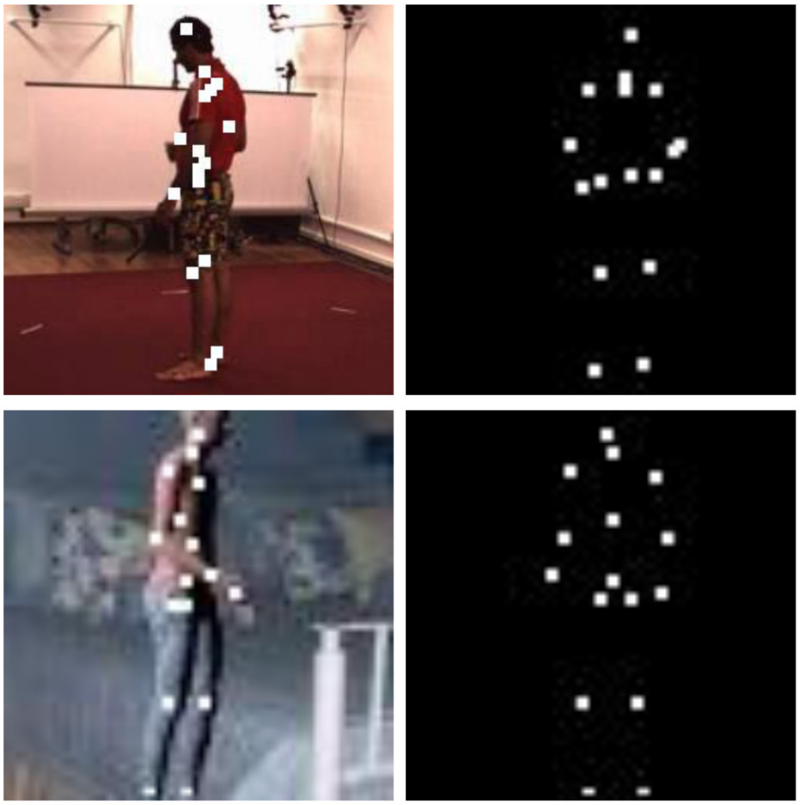


Figure 4.7: Example ground-truth (XY plane, YZ plane) from target (real) and source (synthetic domains) used for pose supervision in HG-FVSV. FV keypoints have been overlaid on input image. SV keypoints are generated at the same scale with co-incident root joint with FV. This amounts to a 90 degree rotation about Y-axis, no scaling or translation. For HG-3D ground truth pose in XY plane is used alonside a normalized value of depth for each joint, centered at the root joint.

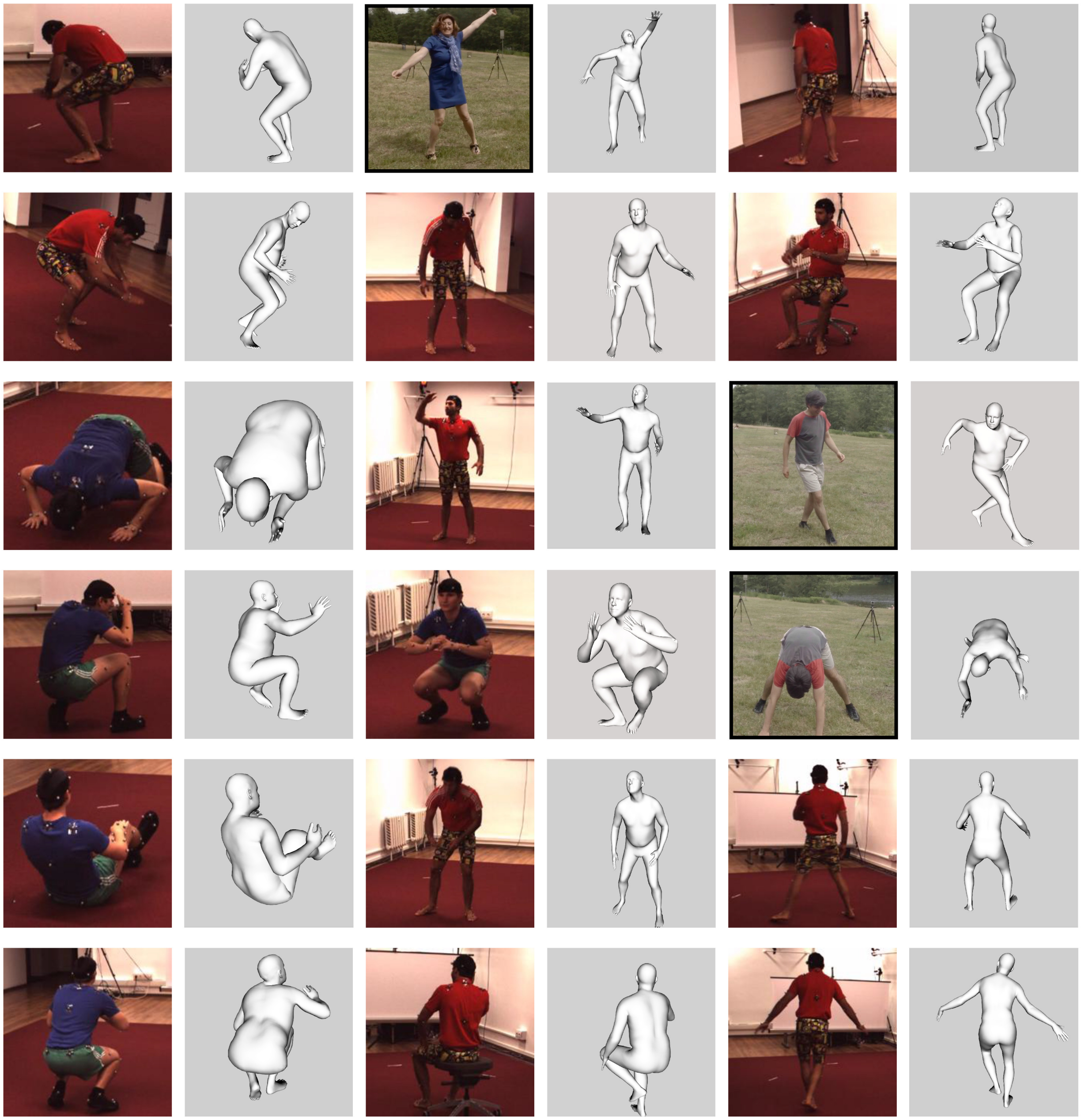


Figure 4.8: Single shot ADA-based 3D mesh fitting results. Visual examples are shown on subjects 9 and 11 of Humans3.6M test set and In-the-Wild images of subjects 5 and 6 of MPI-Inf-3DHP test set.

Method	**Technique**	**MPJPE**
Zhou [62]	Supervised on Humans3.6M	80.98 mm
HG-3D	Jointly optimized on SURREAL+Humans3.6M	85.84mm
Kanazawa [1]*	Re-projection error 3D keypoints and adversarial prior using Mosh'ed Humans3.6M data	67.45 mm
Tung [83]*	Re-projection error on 2D keypoints, segmentation and optical flow fine-tuning parameters at test time	98.4 mm
Tung [83]*	Pre-trained on synthetic data	125.6 mm
Zhou [2]	Unsupervised	113.44 mm
HG-3D+SMPL-Inf Net *	ADA, Partial Supervision with synthetic data	118.49 mm
HG-FVSV+SMPL-Inf Net1x1*	ADA, Partial Supervision with synthetic data	115.87 mm

Table 4.3: Quantitative Evaluation of our models with unsupervised training and other recent methods. Our model* and Kanazawa* [1] are evauated on Humans3.6M validation set, Protocol 1. Zhou [2] predict keypoints only and provide errors on subjects 6,7 of Humans3.6M training set. ** stands for models predicting 3D mesh.*

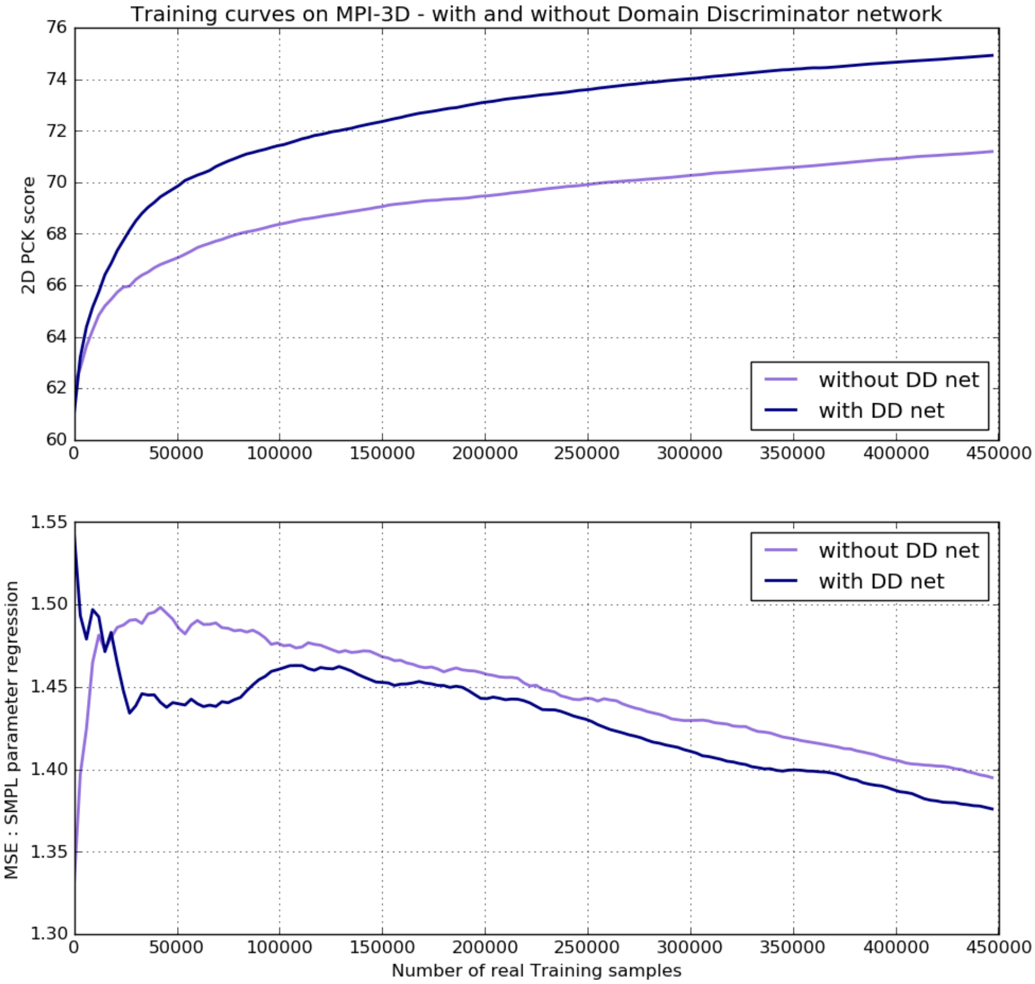


Figure 4.9: Training curves on MPI-Inf-3DHP, showing the effect of including the DD net in training. Initial weights for the HG net, SMPL-Inf net and SD net are drawn from networks trained on Humans3.6M+SURREAL datasets. Including the DD net in training results in higher accuracy in shared common task (Pose estimation) and SMPL parameter regression on SURREAL dataset

Training	**2D PCK**	**MPJPE**	**MPJPE**
	HG net	HG net	SMPL-Inf
Initial unsup. model	89.65	80.62	132.69
2D Re-projection loss	89.33	84.87	148.64

Table 4.4: Comparison of initial model with ADA-based training and after supervision with re-projection loss. MPJPE (mm) and 2D PCK(%) scores are computed on Humans3.6M Validations set, Protocol 1

Training	**2D PCK(%)**	**MPJPE**(mm)
	HG net	HG net
Initial unsup. model	89.65	80.62
Multiview Supervision	89.91	79.04

Table 4.5: Comparison of initial model with ADA-based training and after multi-view supervision on Humans3.6M Validations set, Protocol 1. We achieve marginal improvement in 3D pose estimation score by the HG net. But the performance of the SMPL-Inf net drastically drops as outputs for all real samples start getting mapped to the mean SMPL pose.

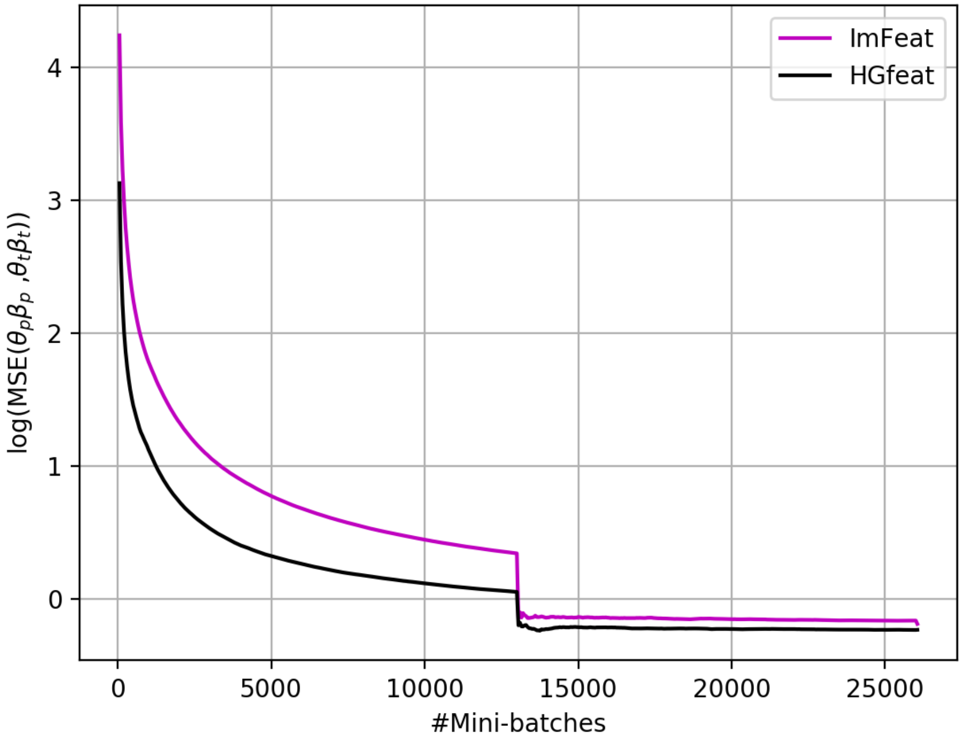


Figure 4.10: Training based on ImFeat vs training based on HGFeat. HG net is initilized with weights learned via prior joint optimization in 3D pose estimation on synthetic and real samples. The SMPL-Inf net learns mesh inference faster based on hourglass features than low-level features.

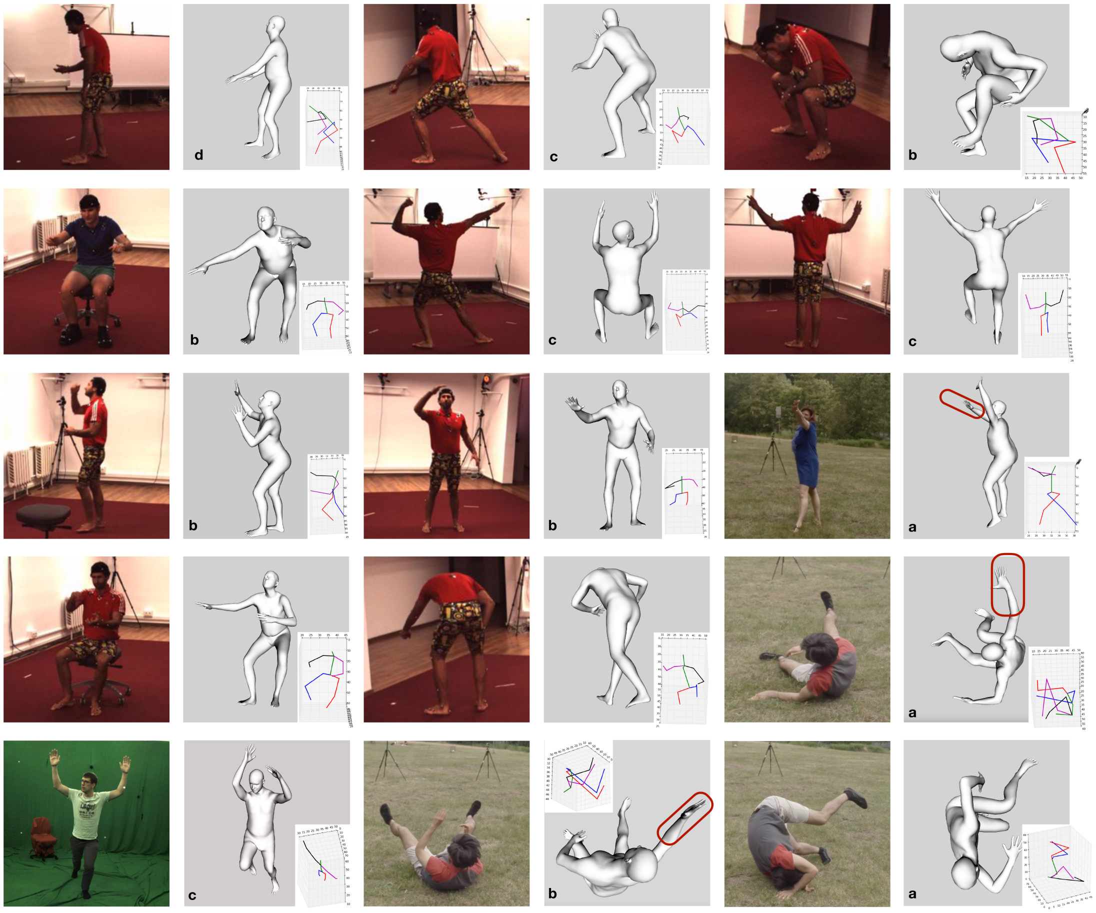


Figure 4.11: Examples of ADA-based mesh fitting with (**a**) predictions corresponding to occluded body parts (**b**) misaligned wrist predictions (**c**) inaccurate depth estimation (**d**) left and right ankle/wrist confusion. Mesh parameters are deduced from intermediate feature representation that are simultaneously used to predict the 3D pose. Corresponding 3D pose predictions by the HG net are included. Angle of elevation for 3D skeletons have been changed to depict depth prediction of joints more clearly.

Chapter 5: Multi-sensor satellite image super-resolution via adversarial domain adaptation

5.1 Introduction

Emergence of the commercial space industry and drones have led to an abundance of satellite/aerial imagery. The level of details in these images varies with the distance the data is captured from and other factors such as sensor properties. The resolution of the captured images is inversely proportional to the distance at which they are captured. In overhead images, the ground sampling distance (**GSD**) is widely used to express the resolution. GSD stands for the distance between image pixel centers when measured on the ground. A higher GSD corresponds to lower resolution. The technique to improve the resolution of images by estimating sub-pixel information is known as super-resolution. Estimation can be performed as an explicitly defined function of the neighboring pixels, or may be learned implicitly using data-driven parametric models. Deep convolutional neural networks (**DCNN**s) have recently been deployed in super-resolution of various types of images (*eg.* faces, natural images) with great degree of success. DCNNs traditionally extract features for computer vision tasks from large fully annotated datasets. The data augmentation step during training of DCNNs for specific tasks typically includes random scaling of inputs in a small range to make the models more robust to variation in scale. Yet deep networks find it challenging to adapt to larger variations in scale (such as 10x lower resolution). Performance often drops sharply when models trained on data at a particular resolution is deployed at a very different resolution. This is commonly encountered in several types of vision tasks such as detection, classification and segmentation. While training DCNNs, it is essential to avoid over-fitting in order to build models that are more generalizable. Over-fitting is frequently encountered with small training sets. Pooling data from different sources is an effective way of increasing the size of training set. However data from different sources often have very different properties. In the case of remote sensing imagery, properties such as *GSD, atmospheric condition, illumination, vegetation and distribution of man-made structures* can vary between datasets and contribute towards *'domain gap'* between sources. The domain gap is a hurdle in pooling data from multiple sources during training. In the absence of ground-truth (for model fine-tuning in the target domain), the domain gap also presents a substantial challenge in the deployment phase. We present a method to address domain gap between two datasets by optimizing a deep feature extractor jointly on samples drawn from both domains. The proposed approach super-resolves images from a low resolution dataset (without any high-resolution ground-truth) using partial supervision from a smaller set of high resolution samples from a different dataset. We train a DCNN-based feature extractor to produce a joint embedding of samples from two domains which is further used to transfer properties of one dataset (resolution and pixel level annotation) to another. Details of the feature extractor architecture are described in section 5.3.2. Furthermore, we append a super-resolving module (described in section 5.3.3) to the feature extractor and train the entire pipeline end-to-end for domain alignment via segmentation and subsequent super-resolution using the domain aligned features.

Generative adversarial networks (GANs) provide an alternative approach to training DCNNs which is particularly effective in the absence of direct supervision. Besides fully supervised DCNN models [156–162], adversarial supervision using GANs has been recently utilized in image super-resolution by [163–167]. De-convolution and interpolation (bi-linear, bi-cubic) are two commonly used methods for up-scaling. De-convolution (implemented as dilated convolution [168]) is known to produce grid like artifacts. Interpolation, on the other hand, does not require any ground-truth for parameter learning. However, previous works have demonstrated greater success using de-convolution-based upscaling with learned parameters. In addition to super-ressolution, GANs have been succesfully deployed for adversarial domain adaptation and domain generalization in the context of various computer vision problems in [118–123, 129].

In the context of aerial and remote sensing imagery, Bosch *et al.* [169] use a cycle-GAN for sensor adaptation to improve segmentation performance on aerial imagery. Various supervised methods for satellite video or image super-resolution have been proposed and surveyed in [170–173]. Other recently proposed DCNN-based super-resolution techniques for aerial imagery include a supervised model from Wang *et al.* [174] and a DCNN-based method with adversarial supervision from Bosch *et al.* [175] to super-resolve aerial imagery. In [175], Bosch *et al.* use adversarial loss as well as mean absolute error computed on high resolution ground-truth and a feature matching loss which minimizes the mean absolute distance between features computed on pairs of super-resolved images and high resolution target image. Paired supervision is unavailable in the case of a dataset which lacks low and high resolution input and target pairs. Low resolution satellite images present greater challenges in extracting detailed features for tasks such as segmentation, detection and capturing exact 3D shape for modeling. Super-resolution helps with generating sub-pixel information and thus improves performance in several cases [176, 177]. Improved resolution of satellite imagery helps with better understanding of road maps, population density, distribution and shape of urban structures, urban planning and mapping.

Contributions:

- In this work we implement a DCNN-based pipeline with adversarial supervision for *8x super-resolution of 50 cm GSD satellite images in the absence of high resolution ground-truth*.

- We combine a larger, low resolution dataset (URBAN3D [178] with building foot print ground truth) with a smaller, high resolution dataset (ISPRS Potsdam [179] 2D semantic labeling dataset with semantic segmentation labels for 6 semantic classes). We use adversarial deep domain adaptation for unsupervised super-resolution of URBAN3D data using a small, high resolution, richly annotated source domain (ISPRS Potsdam).

- Samples from both datasets are jointly mapped to a shared embedding space using a feature extractor trained on a supervised segmentation task. We use different versions of Stacked-UNets [180, 181] for feature extraction.

- Super-resolution is achieved through sub-pixel convolution [156, 182]. We show that the proposed approach produces sharper images with better separability of image regions. We combine various loss functions for partial supervision and joint optimization of the deep networks. We present experimental results comparing super-resolution achieved using different variants of our pipeline and bi-cubic interpolation in section 5.4.

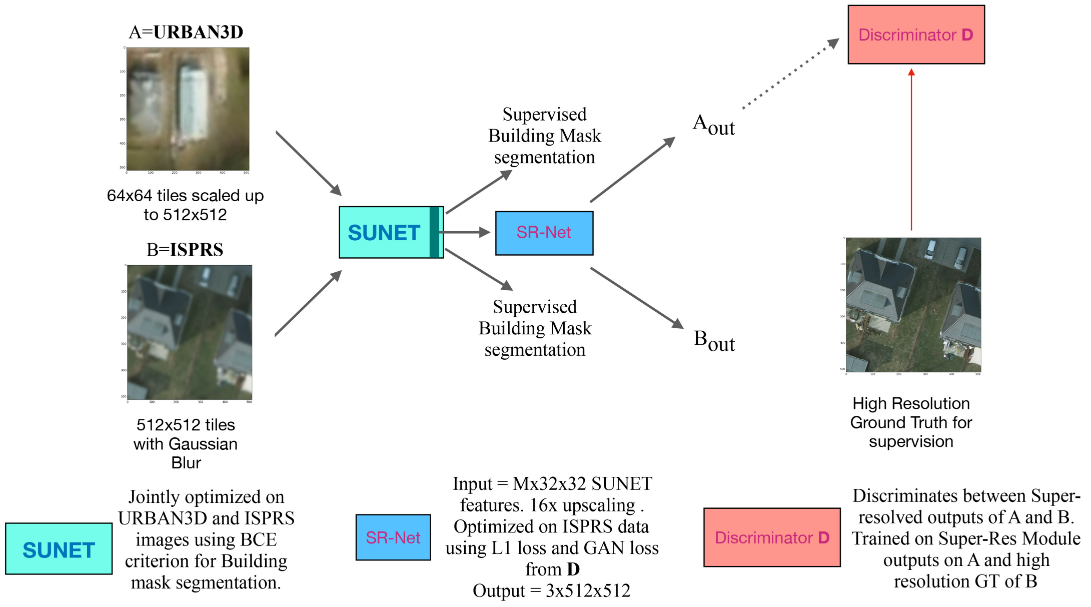

A=**URBAN3D**
64x64 tiles scaled up to 512x512
B=**ISPRS**
512x512 tiles with Gaussian Blur
SUNET
Supervised Building Mask segmentation
SR-Net
Supervised Building Mask segmentation
A_{out}
B_{out}
Discriminator **D**
High Resolution Ground Truth for supervision
SUNET
Jointly optimized on URBAN3D and ISPRS images using BCE criterion for Building mask segmentation.
SR-Net
Input = Mx32x32 SUNET features. 16x upscaling . Optimized on ISPRS data using L1 loss and GAN loss from **D**
Output = 3x512x512
Discriminator **D**
Discriminates between Super-resolved outputs of A and B. Trained on Super-Res Module outputs on A and high resolution GT of B

Figure 5.1: Overview of the Super-Resolution Pipeline. Three deep networks are trained together on mixed mini-batches of samples from source (ISPRS Potsdam) and target (URBAN-3D Jacksonville, FL) datasets. A stacked U-Net (SUNET), supervised using building footprint segmentation loss, extracts domain-aligned features in a shared latent fetaure space. The embedding from the feature extractor is used by SR-Net to produce super-resolved RGB outputs. An adversarial network is trained to discriminate between super-resolved images from SR-Net and real high-resolution images from dataset B

5.2 Datasets

We use satellite images from two publicly available datasets in our experiments. We jointly optimize the DCNNs on 512x512 training samples from both sets.

Dataset A: URBAN3D [178] is meant for training models for building footprint extraction. Samples consist of ortho-rectified RGB images at **50 cm GSD**. Labels are provided for building footprints (and a background class). Images are collected via passive imaging by commercial satellites, and processed using the standard Vricon production pipeline using 50 cm DigitalGlobe satellite imagery. URBAN3D images have been collected in Jacksonville, Florida over an area of $150km^2$ span roughly 48.9K buildings. Images are 2048x2048 pixels.

Dataset B: ISPRS [179] Potsdam 2D Semantic labeling contest samples are RGB orthophotos at **5cm GSD**. The dataset contains 38 patches of same size (6000x6000 images). Labels consist of 6 semantic classes namely building, cars, trees, low vegetation, roads (Impervious surfaces), clutter (background). The number of buildings spanned by the high-resolution airborne imagery data in **B** are about a few 100s.

For training the networks, we randomly sample 64x64 tiles from images in A and scale them to 512x512 using bi-cubic interpolation (**BCI**). We randomly crop 512x512 tiles from samples in B and introduce Gaussian blur. The pre-processing steps are used to produce input tiles roughly at the same scale. Standard data augmentation tricks such as random cropping, flipping and rotations are used in training. **A** *is our target domain*. **B** *is our source domain*.

5.3 Method

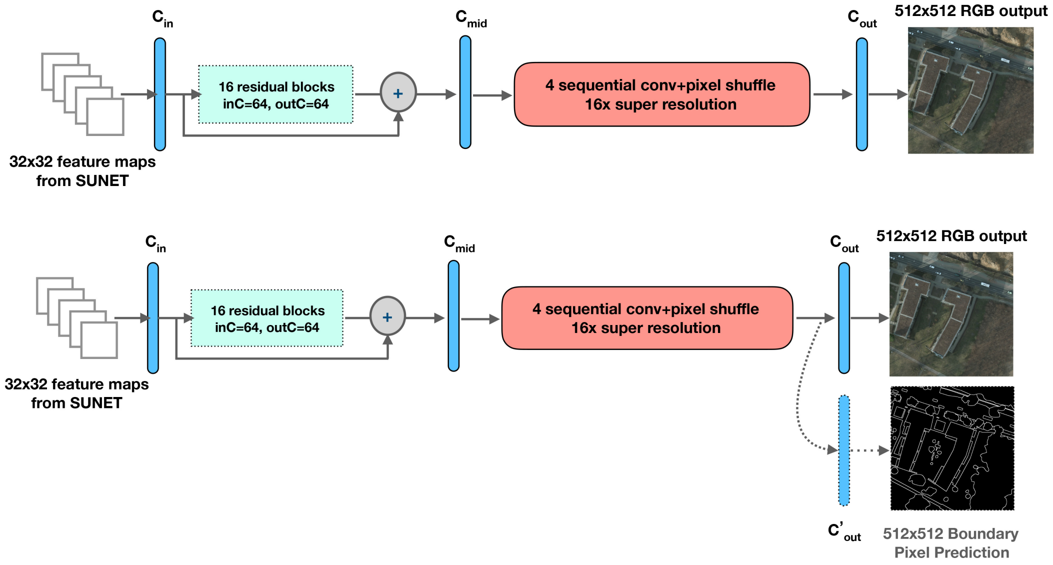

C_in
16 residual blocks inC=64, outC=64
+
C_mid
4 sequential conv+pixel shuffle 16x super resolution
C_out
512x512 RGB output
32x32 feature maps from SUNET
C_in
16 residual blocks inC=64, outC=64
+
C_mid
4 sequential conv+pixel shuffle 16x super resolution
C_out
512x512 RGB output
32x32 feature maps from SUNET
C'_out
512x512 Boundary Pixel Prediction

Figure 5.2: Super-Resolving Module. Feature maps from SUNET feature extractor passes through an input 2D convolution layer C_{in}that reduces the number of feature maps to 64. This is followed by a block of 16 sequential residual blocks, another convolution C_{mid}, and the pixel shuffling block with 4 sequential sub-pixel convolutions, each of which doubles the resolution of the input map. From 32x32 input feature maps, the SR-net produces 3x512x512 super-resolved outputs.

5.3.1 Domain Alignment and Feature Extraction

Domain adaptation between samples from A and B is enabled through a DCNN which is jointly optimized on both domains using a supervised ‘shared-common-task’. This optimization leads to a shared latent space where samples from A and B are jointly mapped. The common set of annotations, in URBAN3D and ISPRS Potsdam samples are derived from the building footprint segmentation labels. Similar to super-resolution, segmentation is a dense prediction task which is effectively solved by fully convolutional deep networks. Using building annotations we supervise a **Stacked U-Net (SUNET)** model [180] on mixed mini-batches with samples drawn simultaneously from both source and target domains. Input to SUNET is a 3x512x512 RGB image $\mathcal{I}$. Outputs of the feature extractor are domain-aligned feature maps (nFx32x32) and a segmentation map $\mathcal{P}$. Every pixel in $\mathcal{P}$ gives a probability score for it belonging to a building. Binary Cross Entropy loss (BCE) is computed on this output using the ground truth building annotations.

$$FeatMaps, \mathcal{P} = SUNET(\mathcal{I})$$

$$\mathcal{L}_{\mathcal{BCE}} = -\sum_{x \in \mathcal{P}} \mathcal{P}(x)log(\mathcal{P}(x)) + (1 - \mathcal{P}(x))log(1 - \mathcal{P}(x)) \quad (5.1)$$

5.3.2 Statcked U-net Architecture

Shah *et al.*, [180] design several versions on SUNETs with different sizes and varying levels of segmentation accuracy on natural image datasets ([183, 184]). Ghosh *et al.* [181] adapt SUNETs for land cover classification on the Deepglobe dataset [185]. In this work, we experiment with two variants of the SUNET model namely SUNET-64 (shallower) and SUNET-7128 (deeper) . SUNETs are composed of 4 blocks of stacked U-Net modules. Each U-Net in a SUNET has 2 levels of encoding and decoding respectively (1 of each in the last block). Reduction in feature map size in the encoding layers is achieved via strided convolutions. Dilated convolutions in decoding layers restore feature maps to the same size as fed into the U-Net module. Skip connections between encoding and decoding layers help with merging local information from higher resolution feature maps with global features from lower resolution feature maps. Additionally there are skip connections around each U-Net module to mitigate the problem of vanishing gradients. More details can be found in [180]. **SUNET-64** and **SUNET-7128** are trained as feature extractors on 3x512x512 RGB inputs. The final block of a SUNET-64 produces 1024 feature maps of size 32x32. SUNET-7128 produces 2304 channels of size 32x32. The ratio of reduction in feature map size (from 512 to 32) is defined as output stride (OS). The OS in our case is 16 (16x reduction in width/height due to two strided convolutions and two average pooling operations). The original architecture was designed for multi-class segmentation. In our case, there are 2 classes (building vs. background). Output activations pass through a Sigmoid layer to generate the probability score maps.

5.3.3 Super-Resolution with partial supervision and Adversarial Losses

The nFx32x32 dimensional domain aligned feature maps produced by the feature extractor (nF=1024 for **SUNET-64**, nF=2304 for **SUNET-7128**) are used to generate segmentation maps for building footprints. The same features are subsequently used as input to the super-resolving part of the DCNN-based pipeline shown in Figure 5.2. We experiment with three variants of the deep architecture for the super-resolving module (SR-Net) : **SR-ResNet, SR-ResNetB and SR-NetB**. Figure 5.2 shows the structure of SR-ResNetB. SR-ResNet is a similar architecture without the C$'_{out}$ layer and no additional output channel for boundary pixel prediction. In SR-NetB, we remove the sixteen residual blocks ([3]) between C$_{in}$ and C$_{mid}$ in SR-ResNetB. The super-resolving component of the SR-Nets are the four cascaded sub-pixel convolution blocks, each preceded by a 2D convolution layer which outputs r^2CxHxW feature maps. The sub-pixel convolution layer converts this to CxrHxrW feature maps. For each layer we set r=2. Four layers of sub-pixel convolution thus results in 16x upsampling of the 32x32 input feature maps. A final convolution layer C$_{out}$ produces a 3 channel 512x512 super-resolved output image. Overall, we achieve 8x super-resolution from input tiles of 64x64 from target samples in **A**. 8x super-resolution is equivalent to creating 63x3 new bytes of information over 8x8 neighborhoods for every pixel in the image. Equation 5.2 expresses the super-resolution and boundary pixel prediction process from domain aligned features and the mean absolute error loss computed on the samples in the source domain (ISPRS).

$$\mathcal{I}_{SR}, \mathcal{B}_{bound} = SRResNetB(FeatMaps)$$

$$\mathcal{L}_{\mathcal{MAE}} = \frac{1}{N_B} * \sum_{\mathcal{I} \in B} ||\mathcal{I}_{HR} - \mathcal{I}_{SR}||_1 \quad (5.2)$$

5.3.4 Adversarial Loss

The feature extractor (SUNET) and the super-resolving module (SR-Net) are trained end-to-end. Loss functions computed on ground truth used for partial supervision include a Binary Cross Entropy Loss (BCE) applied to the output of SUNET and Mean Absolute Error (MAE), also referred to as L1 loss applied to the outputs of SR-Net on samples from dataset **B**. Besides these, we apply an adversarial loss based on the outputs of the discriminator network D on the outputs of SR-Net on samples from both **A** and **B**. The discriminator network (**D**) is a sequential CNN with 2D convolutional layers (each followed by Leaky ReLU activation and batch normalization) followed by two fully connected layers and a sigmoid activation that predicts the probability of the input being a real high-resolution image. D is trained to predict 1 for high-resolution samples from **B** and 0 for super-resolved outputs of the generator on samples from **A**.

5.3.5 Loss Functions

We use a LSGAN [186] for training our pipeline. The discriminator predicts a tensor of 0s and 1s (0s correspond to super-resolved generator outputs and 1s correspond to high-resolution tiles from dataset **B**). LSGAN training is implemented in two phases that alternate for every mini-batch. In the first phase, the parameters of the discriminator network are frozen and its prediction on generator outputs is used to compute the LSGAN loss $\mathcal{L}_{adv}$. The target tensor in this phase consists of all 0s. In the second phase, the parameters of the generator networks (SUNET+SR-Net) are fixed and the weights of the discriminator are updated. In this phase the discriminator is provided with a mixed batch of high resolution ground truth samples from dataset **B** and generator outputs on samples from dataset **A**. The mean square error (MSE) loss in equation 5.3 is computed based on the predictions.

$$\begin{aligned}\mathcal{L}_{adv} = \mathbf{E}_{\mathcal{I}_{SR} \sim SR-Net(FeatMaps_A)}[||D(\mathcal{I}_{SR})||_2] \\ + \mathbf{E}_{\mathcal{I}_{HR} \sim B}[||D(\mathcal{I}_{HR}) - \mathbf{1}||_2] \\ \mathcal{L}_{gan} = \mathbf{E}_{\mathcal{I}_{SR} \sim SR-Net(FeatMaps)}[||D(\mathcal{I}_{SR}) - \mathbf{1}||_2] \\ Loss_G = \lambda_{recon} * \mathcal{L}_{\mathcal{MAE}} + \lambda_{seg} * \mathcal{L}_{\mathcal{BCE}} + \lambda_{gan} * \mathcal{L}_{gan}\end{aligned} \tag{5.3}$$

5.3.6 Boundary pixel supervision

Additional partial supervision - on a boundary pixel segmentation task - is provided to the DCNNs in two of our experiments. We compute the MSE loss on $\mathcal{B}_{bound}$ (predicted by the SR-ResnetB or the SR-NetB) on samples from **B** using boundary pixel ground-truth provided for **B**. The total loss in this case becomes

$$\mathcal{L}_{bound} = ||\mathcal{B}_{bound} - \mathcal{B}^{*}_{bound}||_1$$

$$Loss_G = \lambda_{recon} * \mathcal{L}_{\mathcal{MAE}} + \lambda_{seg} * \mathcal{L}_{\mathcal{BCE}} \\ + \lambda_{gan} * \mathcal{L}_{gan} + \lambda_{bound} * \mathcal{L}_{bound} \quad (5.4)$$

5.4 Results and Discussion

We compare the output of our super-resolution pipeline with images up-scaled using BCI. We experiment with three variants of the pipeline. In the first version we train **SUNET-7128 and SR-ResNet** end-to-end. In the second version of the pipeline we train **SUNET-64 and SR-ResNetB** (which is similar to the SR-ResNet with an additional supervision on boundary pixel prediction). In the third version, we use **SUNET-64 with SR-NetB** which is a lighter version of the SR-ResNetB without the residual layers. The sizes of networks (in terms of number of convolutional parameters) are provided in Table 5.1. We extract contours from the super-resolved outputs computed by our pipeline on the validation set and compare the average binary entropy (building vs. background) of areas under the contours. Figure 5.3 shows the resolution improvement obtained using our pipeline.

SUNET-7128	SUNET-64	SR-ResNet	SR-ResNetB	SR-NetB
36.7M	5.8M	13.8M	7.1M	5.9M

Table 5.1: Parameter count for different network variants

5.4.1 Weight Initialization

We use pre-trained weights from Ghosh *et al.* [181] on the deepglobe land cover classification dataset [185] to initialize the SUNET-7128 feature extractor. In our experiments with SUNET-64, which has fewer parameters, we start from Imagenet pre-trained weights. Super-resolving module SR-ResNet is initialized with random normal weights ([**?**]). Additional experiments with SR-ResNetB and SR-NetB are partially initialized with previously learned weights from the SR-ResNet.

5.4.2 Building Segmentation Results

Figure 5.4 shows segmentation outputs on low resolution tiles vs. those on super-resolved tiles. In several cases, buildings that are partially visible and heavily occluded by dense foliage are missed by the SUNET-64 segmentation model in the input upscaled by BCI but get detected in the super-resolved images. The F-score obtained by the model on the test set is 0.62. We compute the F-score over all the pixels as $F = \frac{2TP}{2TP+FN+FP}$ where TP (true positive) consists of all the pixels predicted as building which are marked as building in the ground-truth. FP (false positive) consists of all the pixels predicted as building which are marked as background in the ground-truth. FN (false negative) consists of all the pixels predicted as background which are marked as building in the ground-truth. This evaluation strategy is slightly different from and not directly comparable to the URBANMapper 3D challenge where true positive is the count of all predicted buildings with an intersection-over-union (IoU) ratio greater than 0.45 with the ground truth. That evaluation metric ignores the exact precision with which pixels are classified as long as the IoU threshold is above 0.45. Moreover participants in that challenge use DSM and DTM (height information) as input to their segmentation models, whereas we perform segmentation based only on the RGB data.

5.4.3 Visual comparison of outputs

Figure 5.5 compares the super-resolved outputs on LR images from dataset A obtained via BCI (Row 1) with different version of our pipeline. Row 2 presents outputs from SUNET-7128 trained jointly with SR-ResNet. Row 3 presents outputs from SUNET-64 trained with SR-ResNetB with additional supervision on boundary pixels for samples in dataset B. Row 4 presents outputs from SUNET-64 trained with SR-NetB (with additional supervision on boundary pixels). Adding boundary pixel supervision to SR-Nets make the building edges sharper in the outputs. The residual blocks in SR-ResNetB are essential in extracting features useful for sub-pixel information estimation. The outputs of SR-ResNetB (Row 3) thus appear sharper than those of SR-NetB (Row 4). Finetuning other supervised deep models for comparison was infeasible in the absence of 8x higher resolution ground-truth. However we do not have 8x higher resolution ground available for fine-tuning on the URBAN3D sample. Further analysis of the quality of super-resolution is performed by extracting contours (enclosing image regions with similar intensity).

5.4.4 Binary Entropy of Contour areas

Figure 5.6 shows visual examples of contours extracted on tiles from samples in validation set of target domain (**A**). Contours are curves joining points along boundary of image regions having uniform intensity/color. Mask obtained from a contour cover the entire area under the contour. Plots shown in Figures **??**, **??**, **??** and **??** analyze the property of the areas under extracted contours in terms of binary entropy. We study the property of areas under contours obtained from images super-resolved using our pipeline vs. upscaling using BCI. The red curves (red/orange/magenta) show the binary entropy of images upscaled using BCI. The blue curves (blue,sky-blue,cyan) show the corresponding contour area entropies in images super-resolved using our method. Binary entropy of a contour area is obtained from corresponding the building footprint ground truth and estimates how uniform the pixels are (buildings or background). Average binary entropy is computed over all the contours. In plots **??**, **??**, **??**, we consider contours at or above different sizes (number of pixels) in the range 30 to 500. The plots **??**, **??**, **??** illustrate average entropies at different overlap levels using SUNET-7128+SRResNet, SUNET-64+SRResNetB and SUNET-64+SR-NetB pipelines respectively. We also split the contours into groups that have at least 30%, 40% and 50% overlap with buildings. Plot **??** shows average over all contours (any degree of overlap with buildings) at or above different sizes. From our observations, super-resolved tiles from SUNET-64+SRResNetB produce the most uniform (low-entropy) contours - which indicates better separability between building and background regions of the image.

5.5 Conclusion

In this work we present a DCNN-based pipeline for satellite image super-resolution. We use adversarial domain adaptation in the absence of supervision from high resolution ground-truth. We achieve 8x up-scaling of images (originally captured at 50cm GSD) using a second high resolution dataset (captured at 5cm GSD). A jointly supervised feature extractor extracts domain aligned features for samples in both datasets which are super-resolved using sub-pixel convolutions. The super-resolving module of the pipeline that uses sub-pixel convolution is partially supervised only on the samples from the high resolution dataset. Owing to difference in latitudinal position of the areas of interest, the source (high-resolution) and target (low resolution) datasets exhibit substantial domain gap, not only in resolution, but also in illumination (sun angle) and vegetation (type of foliage in the area). Despite these differences, the feature extractor successfully aligns the source and target domains based solely on building segmentation labels. The super-resolved images produced by our method are sharper than images up-scaled using bi-cubic interpolation. We study three variants of the pipeline with different versions (different sizes) of the feature extracting and super-resolving networks. We train them end-to-end and compare results visually. We introduce partial boundary pixel supervision in training to improve edge sharpness. Furthermore we extract image contours on super-resolved images and compute entropy of the areas enclosed by the contours. Our results indicate different image regions become more separable as a result of super-resolution. We train a building segmentation model and test its performance. Small buildings heavily occluded by trees are segmented more accurately in super-resolved images.

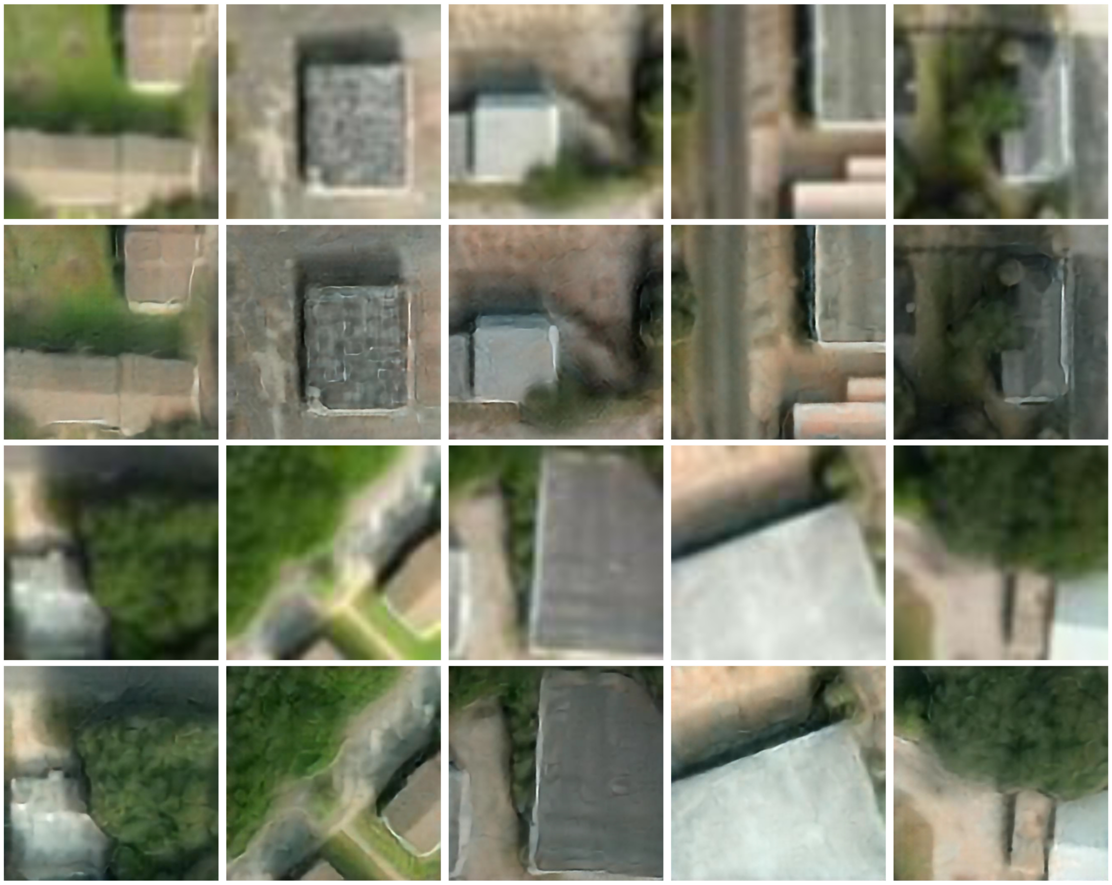


Figure 5.3: Super-Resolution via Adversarial Domain Adaptation. Visual examples of resolution improvement obtained using adversarial domain adaptation via our approach. First and third row show upscaled (bi-cubic interpolation) image tiles from URBAN 3D dataset. Second and fourth rows show super-resolved outputs.

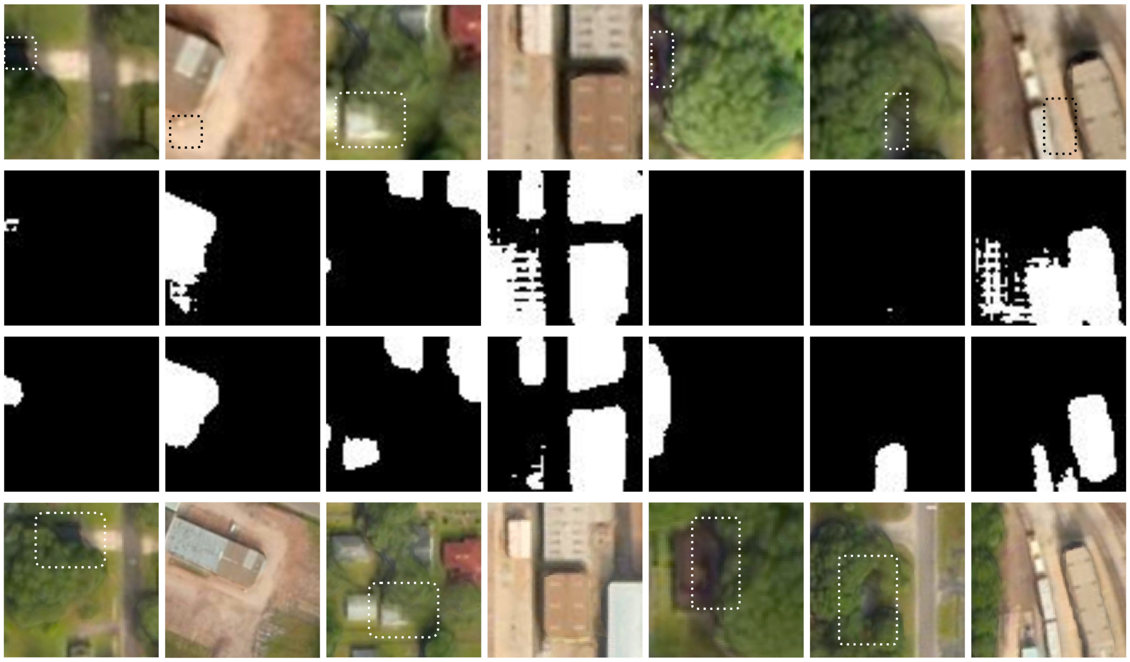


Figure 5.4: Comparison of segmentation on inputs up-scaled using BCI vs those super-resolved using our method. Row 1 shows the blurry input upscaled using BCI. Row 2 shows the corresponding segmentation output (building and background) by a SUNET-64 model trained on BCI upscaled images. Row 3 shows output of SUNET-64 model trained on super-resolved images. Row 4 contains zoomed out views of the input areas that provide context. The segmentation model trained on super-resolved inputs was able to correctly classify the bulding pixels in cases of heavy occlusion.

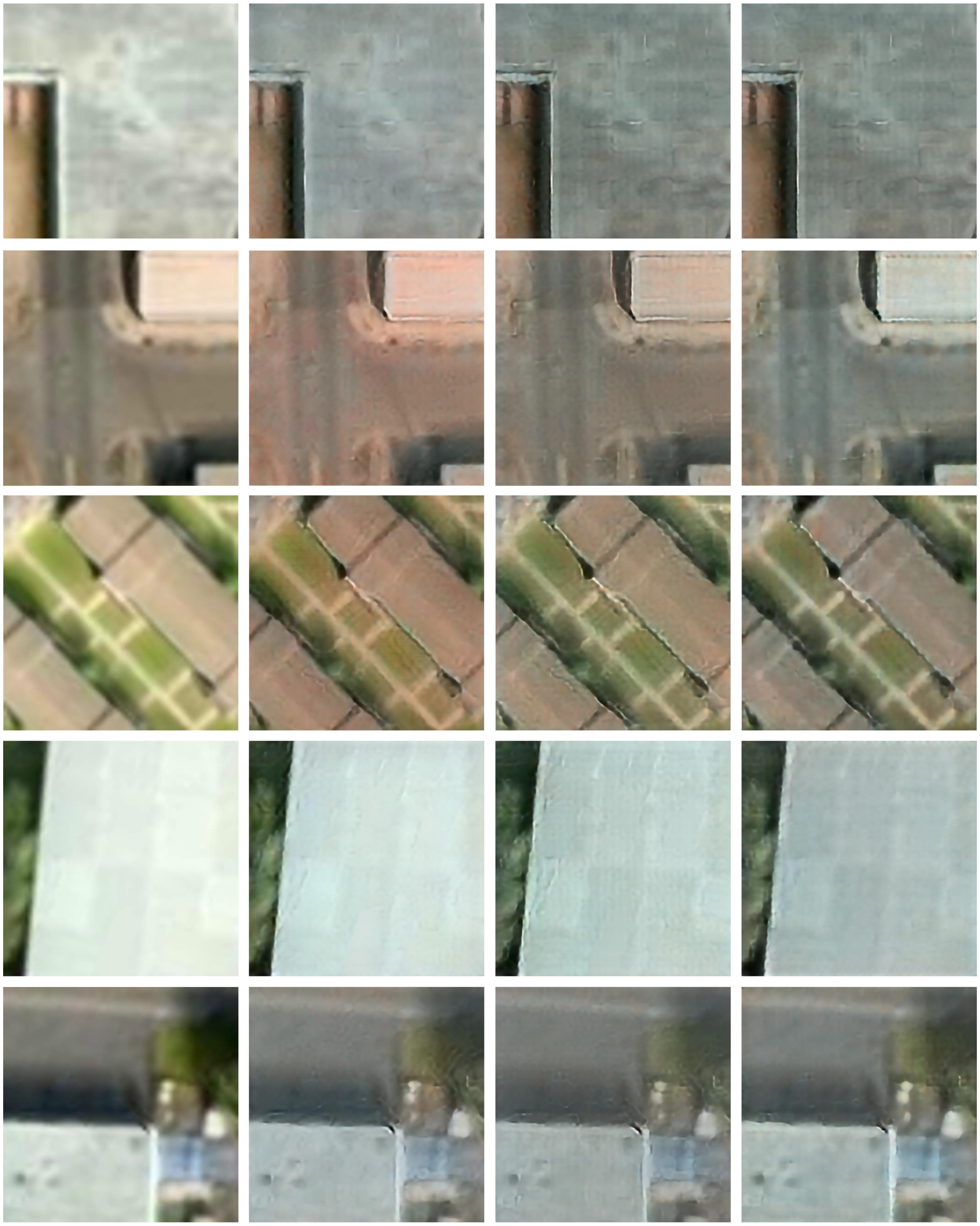


Figure 5.5: Visualization of super-resolved outputs using BCI vs different versions of pipeline comprised of different combinations of feature extractors (SUNET-7128, SUNET-64) and different SR-Nets (SRResnet,SRResnetB and SRNetB). Col 1: BCI, Col 2: SUNET-7128+SR-ResNet, Col 3: SUNET-64+SR-ResNetB, Col 4: SUNET-64+SR-NetB

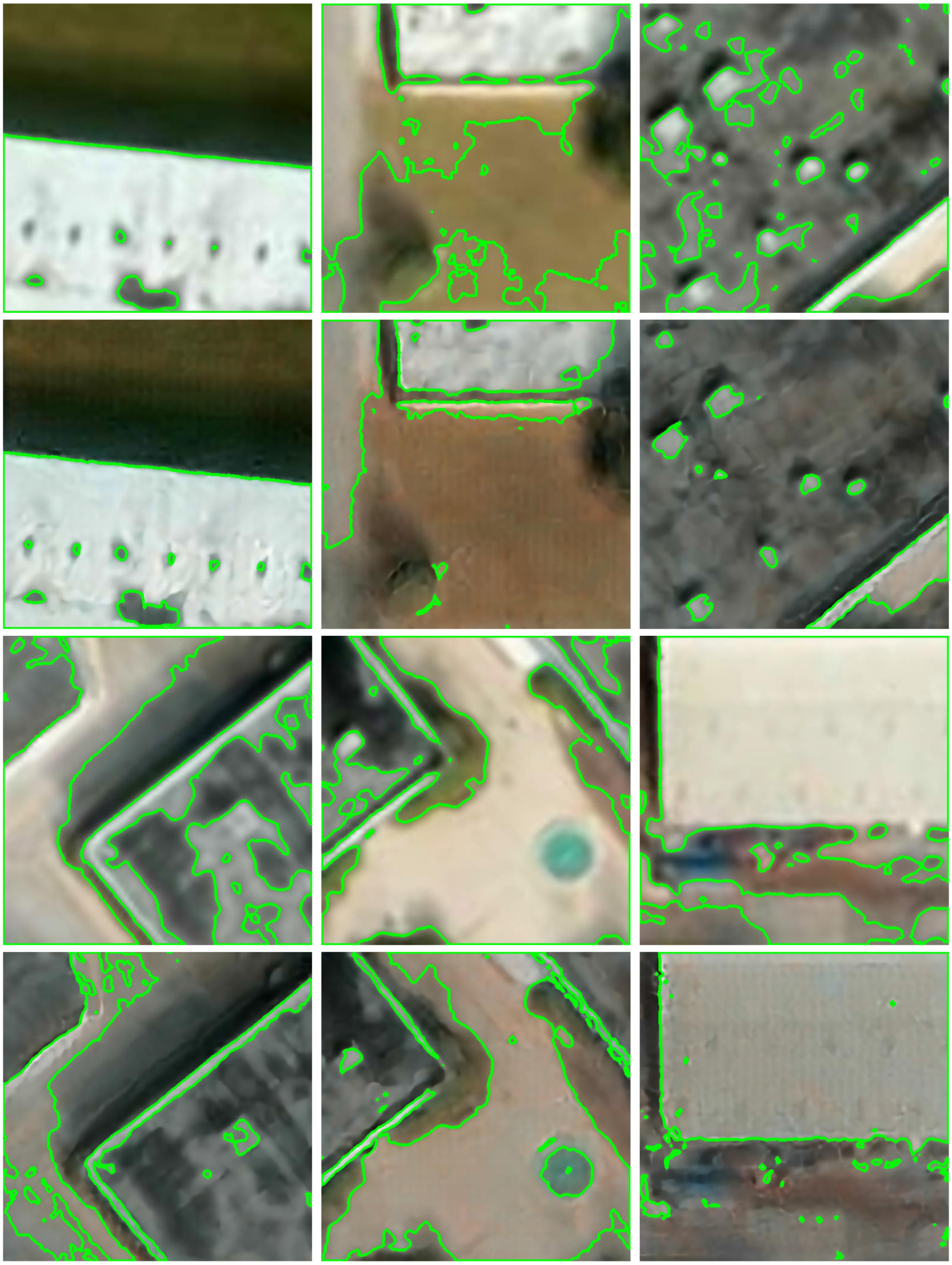
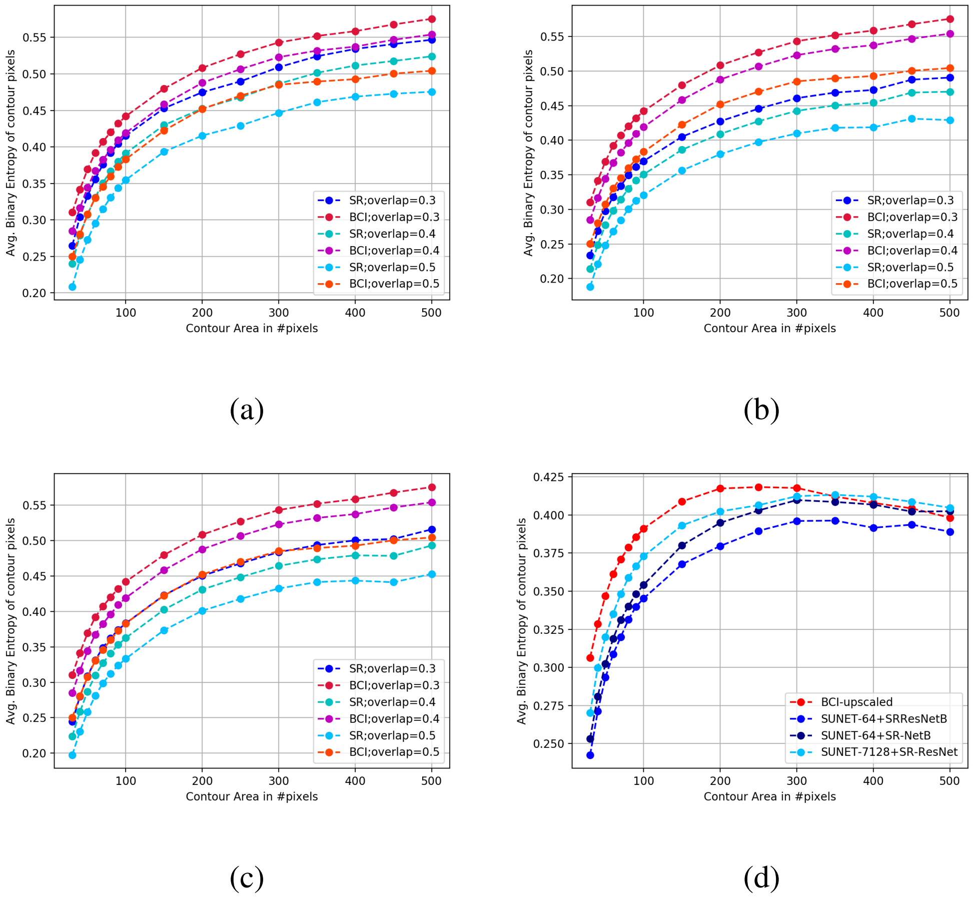


Figure 5.6: Contours extracted from samples in dataset A super-resolved using BCI (top row) vs. those super-resolved with our approach (bottom row). Super-resolved images show improved separability between structures and ground surface and better detection of tiny structures on roofs.



(a)

(b)

(c)

(d)

Figure 5.7: We extract contours on super-resolved images (and also images up-scaled via bi-cubic interpolation (BCI)). Contours encircle image regions of similar intensity. We plot the average binary entropy (building vs. background pixel) of contour areas. Lower entropy corresponds with more homogeneous image regions. Smaller contour areas are more likely to be more homogeneous than larger ones. We compute the average entropy over contour areas at greater than certain size thresholds. The x-axis of the plots show the thresholds. We plot the entropies of contours at varying levels of overlap (greater than certain thresholds) with building footprints. Figures 7 (a) (b) and (c) plot average entropies at different overlap levels using SUNET-7128+SRResNet, SUNET-64+SRResNetB and SUNET-64+SR-NetB respectively. Figure 7(d) plots the average entropies of contour areas at all degrees of overlap for the three versions of our pipeline.

Chapter 6: Conclusion and directions for future research

6.1 Summary of Work

In Chapter 2, we present Agile Beam LADAR architecture, which is capable of algorithmically controlling the illumination wave-front, to capture measurements in alternate sampling bases, using computational imaging principles for robust image recovery, speckle noise removal and performing object recognition directly in the measurement domain without explicit reconstruction. The proposed architecture is more efficient in terms of power consumption and speeds up the process of interpretation and perception of the three dimensional scene by doing away with the necessity of the reconstruction. Our experiments also indicate that learning is more efficient in these alternate domains with various machine learning models achieving higher test accuracy with fewer training samples.

In Chapter 3, we learn deep features in the DCT domain for various image classification tasks such as pedestrian and face detection, material recognition and object recognition. We compare the training convergence, test performance and learned weight matrices of networks with and without the DCT step. We observe that including a DCT operation within the CNN feature extractor improves training dynamics of CNNs which leads to convergence over fewer epochs on various types of computer vision tasks. Trained weight matrices are sparser in the DCT domain.

Chapter 4 focuses on the design and experimentation of an end-to-end trainable pipeline for 3D body pose and shape estimation of a human from a single input image. Using our approach, pose and shape are jointly inferred via fitting a synthetic 3D mesh. The task of mesh parameter estimation is learned from synthetic data. Synthetic and real images are projected to a shared latent space via joint optimization on a 3D pose estimation task. Adversarial domain adaptation enables knowledge transfer from synthetic to real domain. The proposed method uses minimal ground truth annotations on real data. It does not rely on prior methods to produce domain specific mesh estimates on real data for training. The modular design enables experimentation with different network architectures and objective functions.

Chapter 5 is a second application of ADA using a similar pipeline structure, albeit in a different domain and a completely different problem. Satellite images captured in different parts of the world vary greatly due to different climate, vegetation and population density. For certain regions we have large quantities of high resolution imagery and complete 3D information generated using LIDAR data. For many other regions satellite imagery is captured in a single pass. Resolution of images falls with higher GSD. It is much more challenging to recover small and occluded buildings and structures under such conditions. We develop an ADA-based partially supervised pipeline for super-resolution of low quality, high GSD satellite images using a small set of high resolution, well annotated satellite data from a different region. In this case the domains (high and low GSD images) are aligned via joint optimization on a building segmentation task. The aligned features are used to super-resolve low resolution images without direct supervision.

6.2 Directions for Future Research

In Chapters 2, 3, we study the effect of orthonormal projections on training dynamics of machine learning models. We study this effect on different domains, different tasks and machine learning models. Our results indicate that projections to certain non-pixel bases improve training dynamics (quicker convergence). It is not a necessity for neural networks or other machine learning models to learn to solve computer vision tasks on inputs provided in pixel or voxel space. Related subsequent research that support the fact that smaller, faster, more efficient models can be build upon DCT coefficients have been presented in several recent works such as [187–189]. Hoffer *et al.* [190] use Hadamard projection as a fixed classifier to remove redundant parameters from the final classification layer of neural nets without loss in performance. These compressive projections complement the performance of machine learning and deep models and effectively enhance training convergence with less data and with less training time. Albeit to a lesser degree, direct supervision is still necessary for training models using this approach. In Chapters 4 and 5, we address the second scenario when paired supervision is unavailable.

In Chapter 4, we propose an adversarial domain adaptation-based approach for learning to predict 3D pose and shape of humans in images. It is a re-projection free method of estimating 3D model parameters which is inherently independent of key-point annotations provided the source and target domains are aligned during training using a shared common task. This method can be easily extended to parametric 3D model fitting on faces and other objects.

In Chapter 5, we extend the ADA idea for domain adaptation between RSI datasets captured at very different resolutions. Instead of aligning real and synthetic data, we align data from two different sensors operating under very different conditions. This demonstrates the versatility of ADA-based knowledge transfer between domains. Deep networks are almost always fine-tuned on specific datasets for optimal performance. We demonstrate the advantage of joint optimization on data from multiple sources in transfer learning. Besides transferring resolution, such an approach can be useful in transferring detailed segmentation labels and other forms of dense annotations.

The idea of ADA-based training of deep networks has been explored in the context of computer vision problems of very different natures in Chapters 4 and 5. Chapter 4 focuses on learning synthetic parameters prediction (that are non-existent for real data) which enable expression of complex manifold (of the human body) in a low-dimensional space. Chapter 5, in contrast, performs a dense prediction task (super-resolution) based on another shared dense prediction task (building segmentation) without direct supervision. This supports evidence that modular pipelines that follow an underlying structure enabling domain alignment and partial supervision can be systematically built for a host of different problems. ADA also promotes sharing, merging and growth of information scattered across multiple sources (different datasets). Successful training of vision-based intelligent systems at present hinges upon large, uniform and fully annotated datasets which are difficult to collect. Merging of datasets increases diversity and aids in building more robust models. Besides applying this technique to other problems and datasets another challenging future research direction is to merge more than two sources of data. An equally important question to be answered is how to intelligently share weights for domain alignment to have the best possible performance on target domain(s).

Bibliography

[1] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Xingyi Zhou, Arjun Karpur, Chuang Gan, Linjie Luo, and Qixing Huang. Unsupervised domain adaptation for 3d keypoint estimation via view consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153, 2018.

[3] 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.

[4] B. Schwarz. Lidar: Mapping the world in 3d. *Nature Photonics*, 4(7):429–430, July 2010.

[5] Robert L. Fischer, Brian G. Kennedy, Mitchell Jones, Jeffrey Walker, Darian Muresan, Gregory Baxter, Mark Flood, Brian Follmer, Xiuhong Sun, William Chen, and Jeffrey G. Ruby. Development, integration, testing, and evaluation of the u.s. army buckeye system to the navair arrow uav. In *Proc. SPIE*, volume 6963, pages 696318–696318–11, 2008.

[6] Robert Michael S. Dean. Common world model for unmanned systems. In *Proc. SPIE*, volume 8741, pages 87410O–87410O–9, 2013.

[7] M.F. Duarte, M.A. Davenport, D. Takhar, J.N. Laska, Ting Sun, K.F. Kelly, and R.G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, March 2008.

[8] G. A. Howland, P. B. Dixon, and J. C. Howell. Photon-counting compressive sensing laser radar for 3d imaging. *Appl. Opt.*, 50(31):5917–5920, Nov 2011.

[9] A. Kirmani, A. Colao, F. Wong, and V. K Goyal. Exploiting sparsity in time-of-flight range acquisition using a single time-resolved sensor. *Optics Express*, 19(22):21485–21507, Oct 2011.

[10] A. Colao, A. Kirmani, G. A. Howland, J. Howell, and V. K Goyal. Compressive depth map acquisition using a single photon-counting detector: Parametric signal processing meets sparsity. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 96–102, 2012.

[11] J. Sun, E. Timurdogan, A. Yaacobi, E. S. Hosseini, and M. R. Watts. Large-scale nanophotonic phased array. *Nature Photonics*, 493(7431):195–199, Jan 2013.

[12] J. C. Hulme, J. K. Doylend, M. J. R. Heck, J. D. Peters, M. L. Davenport, J. T. Bovington, L. A. Coldren, and J. E. Bowers. Fully integrated hybrid silicon two dimensional beam scanner. *Opt. Express*, 23(5):5861–5874, Mar 2015.

[13] R. W. Gerchberg and W. Owen Saxton. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[14] B. L. Stann, J. F. Dammann, J. A. Enke, P.-S. Jian, M. M. Giza, W. B. Lawler, and M. A. Powers. Brassboard development of a mems-scanned ladar sensor for small ground robots. In *Proc. SPIE*, volume 8037, pages 80371G–80371G–8, 2011.

[15] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. American Mathematical Society, Boston, MA, USA, 2001.

[16] J.-L. Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, Oct 2005.

[17] V.M. Patel, G.R. Easley, R. Chellappa, and N.M. Nasrabadi. Separated component-based restoration of speckled sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1019–1029, Feb 2014.

[18] J. W. Goodman. Some fundamental properties of speckle. *J. Opt. Soc. Am.*, 66(11):1145–1150, Nov 1976.

[19] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57:1413–1541, 2004.

[20] M. Zibulevsky and M. Elad. L1-l2 optimization in signal and image processing. *IEEE Signal Processing Magazine*, 27(3):76–88, May 2010.

[21] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM Journal on Numerical Analysis*, 42(2):686–713, feb 2004.

[22] Vishal M Patel and Michael A Powers. Structured representation-based robust agile-beam ladar imaging. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3285–3289. IEEE, 2015.

[23] Jérôme Bobin, Jean-Luc Starck, Jalal Fadili, Yassir Moudden, and David L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Transactions on Image Processing*, 16(11):2675–2681, 2007.

[24] E. van den Berg and M. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2009.

[25] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[26] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. *arXiv preprint arXiv:1312.5851*, 2013.

[27] Mark D McDonnell and Tony Vladusich. Enhanced image classification with a fast-learning shallow convolutional neural network. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–7. IEEE, 2015.

[28] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[29] Edouard Oyallon, Stéphane Mallat, and Laurent Sifre. Generic deep networks with wavelet scattering. *arXiv preprint arXiv:1312.5940*, 2013.

[30] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[31] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pages 2285–2294, 2015.

[32] Minyoung Kim and Luca Rigazio. Deep clustered convolutional kernels. In *Feature Extraction: Modern Questions and Challenges*, pages 160–172, 2015.

[33] Misha Denil, Babak Shakibi, Laurent Dinh, Nando de Freitas, et al. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pages 2148–2156, 2013.

[34] Zhen Wu, Zhe Xu, Rui Nian Zhang, and Shao Mei Li. Sift feature extraction algorithm for image in dct domain. In *Applied Mechanics and Materials*, volume 347, pages 2963–2967. Trans Tech Publ, 2013.

[35] Baharum Bin Baharudin, Kifayat Ullah, et al. Efficient image retrieval based on quantized histogram texture features in dct domain. In *Frontiers of Information Technology (FIT), 2011*, pages 89–94. IEEE, 2011.

[36] Wenyin Zhang, Zhenli Nie, and Zhenbing Zeng. Image retrieval based on salient points from dct domain. *Lecture notes in computer science*, 3789:386, 2005.

[37] Daan He, Zhenmei Gu, and Nick Cercone. Efficient image retrieval in dct domain by hypothesis testing. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 225–228. IEEE, 2009.

[38] Guocan Feng and Jianmin Jiang. Jpeg image retrieval based on features from dct domain. *Image and Video Retrieval*, pages 93–134, 2002.

[39] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78, 1988.

[40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[41] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[42] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[43] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. Moving obstacle detection in highly dynamic scenes. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 56–63. IEEE, 2009.

[44] Christian Alexander Wojek. *Monocular visual scene understanding from mobile platforms*. PhD thesis, Technische Universität, 2010.

[45] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[46] E. Culurciello C. Farabet. Face detector e-lab. `https://github.com/e-lab/torch7-demos/tree/master/train-face-detector-elab`.

[47] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009.

[48] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[49] Xingyu Zeng, Wanli Ouyang, and Xiaogang Wang. Multi-stage contextual deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 121–128, 2013.

[50] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *International journal of computer vision*, 103(3):348–371, 2013.

[51] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. *arXiv preprint arXiv:1702.07432*, 1(2), 2017.

[52] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[53] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016.

[54] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[55] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[56] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.

[57] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.

[58] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.

[59] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017.

[60] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[61] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[62] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision*, 2017.

[63] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

[64] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G. Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[65] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.

[66] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 International Conference on*, pages 506–516. IEEE, 2017.

[67] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017.

[68] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in neural information processing systems*, pages 3108–3116, 2016.

[69] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1477–1485. IEEE, 2019.

[70] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[71] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[72] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[73] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

[74] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 4627–4635. IEEE, 2017.

[75] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[76] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[77] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *CoRR*, abs/1405.0312, 2014.

[78] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[79] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *arXiv preprint arXiv:1802.00434*, 2018.

[80] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[81] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 3, 2017.

[82] CMU MoCap. The data used in this project was obtained from mocap. cs. cmu. edu. the database was created with funding from nsf eia-0196217. *City*, 2003.

[83] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5236–5246, 2017.

[84] J Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. In *BMVC*, volume 3, page 6, 2017.

[85] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018.

[86] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. *arXiv preprint arXiv:1804.04875*, 2018.

[87] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. *arXiv preprint arXiv:1808.05942*, 2018.

[88] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):220, 2014.

[89] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1455, 2015.

[90] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[91] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[92] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alyosha Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation, 2018.

[93] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[94] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[95] César Roberto De Souza, Adrien Gaidon, Yohann Cabon, and Antonio Manuel López Peña. Procedural generation of videos to train deep action recognition networks. In *CVPR*, pages 2594–2604, 2017.

[96] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S. Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[97] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[98] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[99] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[100] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[101] Buyu Liu and Vittorio Ferrari. Active learning for human pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[102] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[103] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L. Yuille. Joint multi-person pose estimation and semantic part segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[104] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[105] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[106] Hao Jiang and Kristen Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[107] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[108] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. volume 36, 2017.

[109] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[110] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[111] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[112] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[113] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3d human pose estimation by predicting depth on joints. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[114] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *European Conf. on Computer Vision (ECCV)*, LNCS 7577, Part IV, pages 242–255. Springer-Verlag, October 2012.

[115] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[116] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[117] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):801–814, 2019.

[118] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[119] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[120] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5495–5504, 2018.

[121] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[122] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[123] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[124] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[125] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. *arXiv preprint arXiv:1806.06503*, 2018.

[126] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[127] Muhammad Abdullah Jamal, Haoxiang Li, and Boqing Gong. Deep face detector adaptation without negative transfer or catastrophic forgetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[128] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[129] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[130] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. *arXiv preprint arXiv:1804.10094*, 2018.

[131] Renjiao Yi, Chenyang Zhu, Ping Tan, and Stephen Lin. Faces as lighting probes via unsupervised deep highlight extraction. *arXiv preprint arXiv:1803.06340*, 2018.

[132] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[133] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[134] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.

[135] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. *arXiv preprint arXiv:1804.10427*, 2018.

[136] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. *arXiv preprint arXiv:1808.04205*, 2018.

[137] Jiaxin Chen and Yi Fang. Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval. *arXiv preprint arXiv:1807.01806*, 2018.

[138] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. *arXiv preprint arXiv:1804.01110*, 2018.

[139] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[140] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. *arXiv preprint arXiv:1803.10081*, 2018.

[141] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[142] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *European Conference on Computer Vision*, pages 793–810. Springer, 2018.

[143] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. *arXiv preprint arXiv:1804.05827*, 2018.

[144] Luona Yang, Xiaodan Liang, Tairui Wang, and Eric Xing. Real-to-virtual domain unification for end-to-end autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 530–545, 2018.

[145] Amir Atapour-Abarghouei and Toby P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[146] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[147] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[148] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, pages 38–55. Springer, 2018.

[149] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. *arXiv preprint arXiv:1801.10068*, 2018.

[150] Haoshuo Huang, Qixing Huang, and Philipp Krähenbühl. Domain transfer through deep activation matching. In *European Conference on Computer Vision*, pages 611–626. Springer, 2018.

[151] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018.

[152] Alejandro Newell, Kaiyu Yang, and Jia Deng. *Stacked Hourglass Networks for Human Pose Estimation*, pages 483–499. Springer International Publishing, Cham, 2016.

[153] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[154] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural networks for machine learning, Coursera lecture 6e*, 2012.

[155] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[156] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[157] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[158] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[159] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.

[160] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[161] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[162] Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5439–5448, 2017.

[163] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[164] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks. *arXiv preprint arXiv:1809.00219*, 2018.

[165] Daniele Ravì, Agnieszka Barbara Szczotka, Dzhoshkun Ismail Shakir, Stephen P Pereira, and Tom Vercauteren. Adversarial training with cycle consistency for unsupervised super-resolution in endomicroscopy. 2018.

[166] Y Yuan et al. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. *methods*, 30:32, 2018.

[167] Guimin Lin, Qingxiang Wu, Liang Chen, Lida Qiu, Xuan Wang, Tianjian Liu, and Xiyao Chen. Deep unsupervised learning for image super-resolution with generative adversarial network. *Signal Processing: Image Communication*, 68:88–100, 2018.

[168] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[169] Marc Bosch, Gordon Christie, and Chris Gifford. Sensor adaptation for improved semantic segmentation of overhead imagery. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 648–656. IEEE, 2019.

[170] Yimin Luo, Liguo Zhou, Shu Wang, and Zhongyuan Wang. Video satellite imagery super resolution via convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2398–2402, 2017.

[171] Kui Jiang, Zhongyuan Wang, Peng Yi, and Junjun Jiang. A progressively enhanced network for video satellite imagery superresolution. *IEEE Signal Processing Letters*, 25(11):1630–1634, 2018.

[172] Hatem Magdy Keshk and Xu-Cheng Yin. Satellite super-resolution images depending on deep learning methods: A comparative study. In *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–7. IEEE, 2017.

[173] Zhi-Zhong Wang, Qing-Jun Zhang, and Xiao-Lei Han. Satellite remote sensing image super resolution based on markov random fields. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 7256–7259. IEEE, 2016.

[174] Tingwei Wang, Wenjian Sun, Hairong Qi, and Peng Ren. Aerial image super resolution via wavelet multiscale convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(5):769–773, 2018.

[175] Marc Bosch, Christopher M Gifford, and Pedro A Rodriguez. Super-resolution for overhead imagery using densenets and adversarial learning. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1414–1422. IEEE, 2018.

[176] Liujuan Cao, Rongrong Ji, Cheng Wang, and Jonathan Li. Towards domain adaptive vehicle detection in satellite image by supervised super-resolution transfer. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[177] Liujuan Cao, Cheng Wang, and Jonathan Li. Vehicle detection from highway satellite images via transfer learning. *Information sciences*, 366:177–187, 2016.

[178] Hirsh R Goldberg, Sean Wang, Gordon A Christie, and Myron Z Brown. Urban 3d challenge: building footprint detection using orthorectified imagery and digital surface models from commercial satellites. In *Geospatial Informatics, Motion*

Imagery, and Network Analytics VIII, volume 10645, page 1064503. International Society for Optics and Photonics, 2018.

[179] Franz Rottensteiner, Gunho Sohn, Jaewook Jung, Markus Gerke, Caroline Baillard, Sebastien Benitez, and Uwe Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 1(3):293–298, 2012.

[180] Sohil Shah, Pallabi Ghosh, Larry S Davis, and Tom Goldstein. Stacked u-nets: a no-frills approach to natural image segmentation. *arXiv preprint arXiv:1804.10343*, 2018.

[181] Arthita Ghosh, Max Ehrlich, Sohil Shah, Larry Davis, and Rama Chellappa. Stacked u-nets for ground material segmentation in remote sensing imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 257–261, 2018.

[182] Andrew Aitken, Christian Ledig, Lucas Theis, Jose Caballero, Zehan Wang, and Wenzhe Shi. Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *arXiv preprint arXiv:1707.02937*, 2017.

[183] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[184] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[185] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raska. Deepglobe 2018: A challenge to parse the earth through satellite images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 172–17209. IEEE, 2018.

[186] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[187] Matej Ulicny and Rozenn Dahyot. On using cnn with dct based image data. In *Proceedings of the 19th Irish Machine Vision and Image Processing conference IMVIP*, 2017.

[188] Dan Fu and Gabriel Guimaraes. Using compression to speed up image classification in artificial neural networks. 2016.

[189] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In *Advances in Neural Information Processing Systems*, pages 3933–3944, 2018.

[190] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. In *International Conference on Learning Representations*, 2018.