BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

PRIVACY-PRESERVING SMART-ROOM VISUAL ANALYTICS

by

JIAWEI CHEN

B.Eng., Harbin Institute of Technology, 2013
M.Eng., Duke University, 2015

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2019

Approved by

First Reader

Janusz Konrad, PhD
Professor of Electrical and Computer Engineering

Second Reader

Prakash Ishwar, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Third Reader

Brian Kulis, PhD
Associate Professor of Electrical and Computer Engineering
Associate Professor of Systems Engineering
Associate Professor of Computer Science

Fourth Reader

Kate Saenko, PhD
Associate Professor of Computer Science

The philosopher Tsang said, "I daily examine myself on three points:– whether, in transacting business for others, I may have been not faithful;– whether, in intercourse with friends, I may have been not sincere;– whether I may have not mastered and practiced the instructions of my teacher."

The Confucian Analects, translated by James Legge (1893)

Acknowledgments

I would first like to express my sincere appreciation and thanks to both my research advisors, Prof. Janusz Konrad and Prof. Prakash Ishwar for their invaluable support and guidance over the past 4 years. I was fortunate to receive their mentorship and guidance which ultimately shaped and refined my methodology towards research. I truly appreciate their patience and the enormous time and effort that they dedicated to every stage of my research. I could not have imagined having better advisors and mentors for my Ph.D study. Apart from my advisors, I also would like to thank my committee members, Prof. Brian Kulis and Prof. Kate Saenko for contributing time from their schedules and giving feedback and suggestions.

From the Information Data Science group, I would also like to acknowledge some current and former members: Jonathan Wu, Jinyuan Zhao, Christy Lin, Douglas Roeper, M. Ozan Tezcan, Hiroki Kawai, Yuting Chen, Ruidi Chen, Rui Chen, Michael Farag, Andrew Cutler. Thanks are also due to the (NSF) National Science Foundation (under Lighting Enabled Systems and Applications ERC Cooperative Agreement No. EEC-0912056) for financial support without which I would not have been able to develop my scientific discoveries.

A heartfelt thanks go to my wonderful parents and sister. They gave me warm love, endless moral support, encouragement and motivation to accomplish personal goals. In the end, a special thanks to my soul mate Xiaoyi Zhang, for all your love and support.

PRIVACY-PRESERVING SMART-ROOM VISUAL ANALYTICS

JIAWEI CHEN

Boston University, College of Engineering, 2019

Major Professors: Janusz Konrad, PhD
Professor of Electrical and Computer Engineering

Prakash Ishwar, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

ABSTRACT

The proliferation of sensors in living spaces in the last few years has led to the concept of a smart room of the future - an environment that allows intelligent interaction with its occupants, be it a living or conference room. Among the promised benefits of future smart rooms are improved energy efficiency, health benefits and increased productivity. To realize such benefits, accurate and reliable localization of occupants and recognition of their poses, activities, and facial expressions are crucial. Extensive research has been performed to date in these areas, primarily using video cameras. However, with increasing concerns about privacy, the use of standard video cameras seems ill-suited for smart spaces; alternative sensing modalities and visual analytics techniques, that preserve privacy, are urgently needed. Motivated by such demand, this thesis aims to develop image and video analysis methodologies that protect occupant's (visual) privacy while preserving utility for an inference task. We propose two distinct methodologies to accomplish this.

In the first one, we address privacy concerns by degrading the spatial resolution of

images/videos to the point where it no longer provides visual utility to eavesdroppers. We have conducted proof-of-concept studies for the problems of head pose estimation, indoor occupant localization, and human action recognition at extremely low resolutions (eLR) (lower than 16×16 pixels). For the problem of pose estimation, specifically head pose, from a single image at resolutions as low as 10×10 pixels or even 3×3 pixels, we developed an estimation algorithm using a classical data-driven approach. For occupant localization based on data from overhead-mounted single-pixel visible-light sensors, we developed both coarse- and fine-grained estimation algorithms using classical machine learning techniques. For action recognition from eLR visual data, motivated by the success of deep learning in computer vision, we developed multiple two-stream Convolutional Neural Networks (ConvNets) that fuse spatial and temporal information. In particular, we proposed a novel semi-coupled, filter-sharing network that leverages high-resolution videos to train an eLR ConvNet. We demonstrated that practically useful inference performance can be achieved at eLR.

While the use of eLR data can mitigate visual privacy concerns, it can also significantly limit utility compared to full-resolution data. Thus, in addition to developing inference methods for eLR data, we took advantage of recent advancements in representation learning to design an identity-invariant data representation that also permits synthesis of utility-equivalent realistic full-resolution data with a different identity. To this end, we proposed two novel models tailored for 2D images. We tested our models on a number of visual analytics tasks such as recognizing facial expressions, estimating head poses, or illumination condition. A thorough evaluation of the proposed approaches under various threat scenarios demonstrates that our approaches strike a balance between preservation of privacy and data utility. As additional benefits, our approach enables performing expression-and head-pose-preserving face morphing.

Contents

List of Tables

List of Figures

List of Abbreviations

CCR	.....	Correct Classification Rate
ConvNet	.....	Convolutional Neural Network
eLR	.....	Extremely Low Resolution
GAN	.....	Generative Adversarial Network
GLS	.....	Generator Label Score
HOG	.....	Histogram of Oriented Gradients
HR	.....	High Resolution
IRL-VGAN	.....	Invariant-Representation-Learning VGAN
IS	.....	Inception Score
MAE	.....	Mean Absolute Error
PCSRN	.....	Partially-Coupled Super-Resolution Network
PPRL-VGAN	.....	Privacy-Preserving Representation-Learning VGAN
QSVR	.....	Quantized Support Vector Regression
ROI	.....	Region of Interest
SVM	.....	Support Vector Machine
SVR	.....	Support Vector Regression
VAE	.....	Variational Auto-Encoder
VGAN	.....	Variational Generative Adversarial Network
VLC	.....	Visible Light Communication

Chapter 1

Introduction

1.1 Motivation

Smart Room Visual Analytics: The proliferation of sensors in living spaces in the last few years has led to the concept of a smart room of the future - an environment that allows intelligent interaction with its occupants, be it a living or conference room (see Fig. 1·1). Among the promised benefits of future smart rooms are improved energy efficiency, health benefits and increased productivity. For example, energy savings can result from lowering illumination in regions void of people, while health benefits can be realized by task-optimized lighting, e.g., reducing screen glare, and thus eye strain, when reading off a screen. As for productivity, localization of occupants may help maximize throughput rates in visible light communication (VLC) between fixed transceivers (ceiling, walls) and mobile devices (smartphones, tablets, laptops), also known as LiFi. Finally, hand gestures can be used to control various room conditions (e.g., temperature, light). To realize such benefits, it is crucial to accurately and reliably estimate useful information from the room such as the locations, pose, activities and facial expressions of the occupants. Recent advances in computer vision technologies have made possible the development of intelligent video-based monitoring systems that can automatically interpret visual data and capture the aforementioned information about occupants. But these new technologies also impose a threat to occupant's privacy as they are able to collect and index a huge amount of private information about each individual. With increasing concerns about

privacy, the use of standard computer vision techniques with no respect for user's privacy seems ill-suited for smart spaces; alternative visual analytics techniques and sensing modalities, that preserve privacy, are urgently needed.

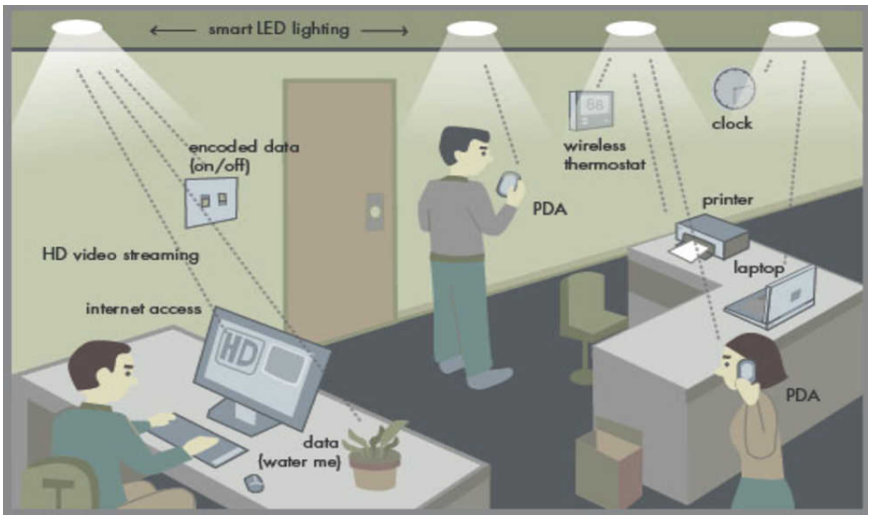


Figure 1·1: Vision of the smart room of the future

Preserving Visual Privacy: To date, efforts have been made to develop privacy-preserving visual recognition solutions. Classical cryptographic solutions were developed to locally encrypt data and protect it against unauthorized access (Erkin et al., 2009; Yonetani et al., 2017). However, the sensitive information could be uncovered if an adversary has the right key for decryption. In addition, encryption algorithms usually have large computational complexity. Alternatively, some works proposed to modify private content based on either image processing operations such as blurring (Butler et al., 2015), pixelation (Butler et al., 2015) and cartooning (Winkler et al., 2014), or privacy-preserving optics that can filter out sensitive information (Pittaluga and Koppal, 2015; Pittaluga and Koppal, 2016). However, simple filtering methods may fail to protect privacy information (e.g., identity) if a rival recognition algorithm is trained using images that have the same distortions as the test image (Newton et al., 2005; Padilla-López et al., 2015). Recently, a few learning-based

approaches (Hamm, 2017; Raval et al., 2017; Sokolic et al., 2017; Wu et al., 2018) were developed to learn privacy-preserving image transformations by optimizing the trade-off between preservation of visual privacy and data utility. By the term utility we refer to data intelligibility which represents the amount of useful information that can be extracted from the visual data. The definition of visual privacy requires a concrete application context. In the context of smart room, occupant's identity is the sensitive information that cannot be disclosed. Therefore, protecting visual privacy is equivalent to anonymization of visual data, e.g., modifying a person's visual appearance to make him/her look like a different person. One major drawback of learning-based approaches is that they have to be fine-tuned for each inference task. Another interesting recent work (Dai et al., 2015) showed that even at extremely low resolutions (eLR), e.g., 16×12 pixels, acceptable indoor human action recognition performance is attainable. This is inspiring because at such low resolutions, any privacy concern can be largely ignored. Further, low resolution cameras are inexpensive and have additional benefits such as low transmission cost that could contribute to the scalability of smart rooms. However, the previous work (Dai et al., 2015) handles only the action recognition task using a simple data-driven method (K-nearest neighbors). It is unclear if useful performance can be obtained for other inference tasks at eLR. It is also not clear if better task performance can be achieved using more sophisticated methods such as deep learning algorithms.

In this thesis, we first explore and attempt to address visual privacy concerns by conducting studies for various visual analysis tasks including head pose estimation (Chen et al., 2016), indoor occupant localization (Roeper et al., 2016) and human action recognition (Chen et al., 2017) at eLR using both classical machine learning algorithms and modern deep learning techniques.

While the use of eLR data can mitigate visual privacy concerns, it can also degrade

utility compared to full-resolution data. In order to strike a balance between privacy preservation and data utility, we propose a new approach: seamless *replacement* of the private information in an image without significantly degrading its visual quality or data utility. Specifically, we develop two distinct models (Chen et al., 2018; Chen et al., 2019) that leverage variational generative adversarial networks (VGANs) to learn an identity-invariant representation of an image while enabling the synthesis of a utility-equivalent, realistic version of this image with a different identity. Both the resulting representation and the synthesized image are largely disentangled from the original identity information, and therefore can be made public without compromising user's privacy. The main contributions of this thesis are summarized in Fig. 1·2.

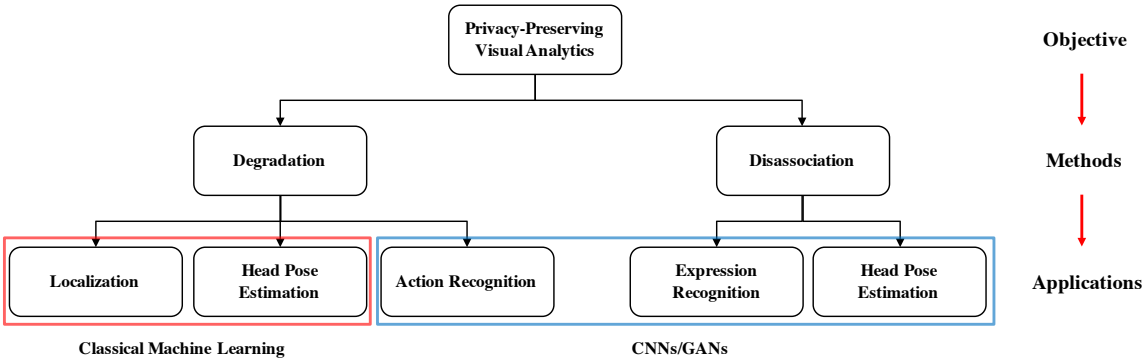


Figure 1·2: Summarization of contributions

1.2 Related Work

With pervasive cameras for surveillance and smart indoor spaces, privacy-preserving visual analytics has drawn increasing attention. There is a growing body of research on methods to perform various visual analysis tasks from data in a manner that does not disclose private information. According to how privacy is protected, the literature can be broadly classified into reversible and irreversible approaches (Badii et al., 2013).

Reversible methods are those which permit exact data recovery. However, they are also prone to exposing the original data to possible hacks. Scrambling and encryption (Dufaux and Ebrahimi, 2006; Dufaux and Ebrahimi, 2008; Sadeghi et al., 2009; Dufaux, 2011; Ziad et al., 2016; Wang et al., 2017; Kuroiwa et al., 2007; Martínez-Ponte et al., 2005; Gilad-Bachrach et al., 2016; Senior et al., 2005; Ye, 2010; Zeng and Lei, 2003) are two commonly used reversible methods to protect the privacy of data. Most scrambling methods use only permutation operations and can operate only in a specific domain, e.g., spatial domain (Senior et al., 2005; Ye, 2010), frequency domain (Dufaux and Ebrahimi, 2006; Dufaux and Ebrahimi, 2008; Zeng and Lei, 2003) or codestream domain (Kuroiwa et al., 2007; Martínez-Ponte et al., 2005). They are also vulnerable to chosen-plaintext attacks (Wirt, 2004; Tews et al., 2011). Image and video encryption algorithms are typically more secure than scrambling-based methods. (Erkin et al., 2009; Yonetani et al., 2017) developed cryptographic solutions to locally encrypt visual information using homomorphic encryption algorithms. However, encryption-based solutions have large computational complexity (Rivest et al., 1978). Furthermore, extracting information for an inference task from encrypted data is a challenging problem. Some recent works (Gilad-Bachrach et al., 2016; Wang et al., 2017) have proposed neural networks for encrypted-domain recognition and they perform reasonably well on simple image datasets, e.g., character recognition in MNIST (LeCun et al., 1998). However, it is unclear how they will perform in nuanced inference tasks such as pose and action recognition on real-world data.

Irreversible methods are those that do not allow exact data recovery. The most commonly encountered techniques in this category are based on image processing and filtering (Krinidis et al., 2014; Erturk, 2007; Park and Kautz, 2008; Chaaraoui et al., 2012; Zhang et al., 2012; Frome et al., 2009; Jalal et al., 2012). Some methods, such as (Zhang et al., 2012; Jalal et al., 2012), use only depth data from RGBD

cameras as a way to preserve privacy. However, depth data has been proved to be insecure for preserving a person's identity (Plagemann et al., 2010; Haque et al., 2016). In (Park and Kautz, 2008; Chaaraoui et al., 2012), privacy is protected by using only silhouettes of the detected foreground objects for an inference task. However, the performance of these approaches depends heavily on the accuracy of foreground detection. In (Newton et al., 2005; Gross et al., 2005; Gross et al., 2006; Bitouk et al., 2008), the focus is on developing face de-identification methods by altering faces in an image or a video to hide a person's identity. Alternatively, (Frome et al., 2009; Kitahara et al., 2004; Zhang et al., 2006; Neustaedter et al., 2006; Neustaedter and Greenberg, 2003; Boyle et al., 2000) use image filters to obscure sensitive regions like human faces, bodies or even background. (Winkler et al., 2014) proposed a customized camera that applies a cartoon-like effect based on mean shift filtering. (Pittaluga and Koppal, 2015; Pittaluga and Koppal, 2016) developed privacy-preserving optics to filter sensitive information from the incident light-field before sensor measurement are made, by averaging together a target face image with $k-1$ of its neighbors (according to some similarity metric). Nevertheless, it has been shown that simple filtering methods do not fool identity recognition algorithms if they are trained using images that have the same distortions as the test images (Newton et al., 2005; Padilla-López et al., 2015). In addition to designing hand-crafted filters, learning-based approaches were proposed to learn a data sanitization function that optimizes the utility-privacy trade-off (Hamm, 2017; Raval et al., 2017; Sokolic et al., 2017; Wu et al., 2018). In spite of achieving good empirical results, their methods have to be tuned for each inference task of interest (e.g., action recognition).

A recent line of research (Dai et al., 2015) explored visual recognition at extremely low-resolutions (eLR). At such extreme scenarios, the data gathered no longer provides any "visual utility" to eavesdroppers. One additional benefit of using eLR data

is that it has low data transmission and processing requirements. The reported results show it is possible to achieve reasonable recognition performance with eLR data. However, their work studies only a limited set of tasks using classical data-driven methods. Our work improves upon the previous work on eLR by developing more robust methodologies based on classical machine learning algorithms and modern deep learning techniques, and expanding the scope to cover various vision tasks including head pose estimation (Chen et al., 2016), indoor occupant localization (Roeper et al., 2016) and human action recognition (Chen et al., 2017).

Adversarial training has also been leveraged recently for privacy-preserving visual analytics tasks. In (Brkic et al., 2017), the focus is on full-body de-identification without an additional utility criterion such as accuracy of facial expression. Their methodology relies upon a segmentation algorithm to accurately extract the silhouette of a person to be de-identified. While (Raval et al., 2017; Wu et al., 2018) use adversarial networks to jointly optimize privacy and utility objectives, as mentioned previously, their methods have to be tuned for each usage scenario. Different from previous research that uses adversarial networks, this thesis develops two novel representation learning frameworks that explicitly learn an invariant image representation with the explicit goal of utility-preserving identity replacement in the synthesized output image which is required to look realistic. Both the generated image representation and the synthesized image retain the utility information of the original image, but eliminate the identity information. As a result, they can be safely released for processing without compromising user's privacy. We demonstrate that our approaches can be applied to various visual recognition tasks such as facial expression recognition (Chen et al., 2018), head pose estimation (Chen et al., 2019) and style classification (Chen et al., 2019).

1.3 Organization

The rest of this thesis is organized as follows. In Chapter 2, we introduce our approaches for the problems of head pose estimation, indoor occupant localization and action recognition at extremely low resolutions. In Chapter 3, we discuss two novel invariant representation learning models for privacy-preserving visual recognition. We demonstrate the effectiveness of our models on various visual analytics tasks. In Chapter 4, we summarize the conclusions and outline possible directions for future work.

Chapter 2

Methodologies Based on Extremely-Low-Resolution (eLR)

As mentioned in Chapter 1, using eLR data is a plausible approach to alleviate privacy concerns while achieving reasonable target-task performance. It has additional benefits such as low transmission cost and processing complexity. However, careful studies are demanded to find the limit to which we can reduce spatial resolution without significantly impacting performance of target tasks. More importantly, customized computer vision approaches are needed to maximize the utility of eLR data.

In this chapter, we first present a framework for estimating head pose orientation with a monocular RGB camera at 3 extremely low spatial resolutions. Next, we detail a system for occupant localization in an indoor setting that uses 6 single-pixel visible-light sensors and thus does not violate an individual's privacy, even with eavesdropping. Finally, we introduce multiple Convolutional Neural Networks (ConvNets) for privacy-preserving action recognition at eLR. Further, we propose a semi-coupled, filter-sharing network that leverages high-resolution (HR) videos during training in order to assist an eLR ConvNet

2.1 Head Pose Estimation from eLR Images

Automatic and robust algorithms for head pose estimation are important for effective intelligent interaction in smart spaces. In this section, we propose a classical nonlinear regression method based on widely used visual features for estimating human head

pose with a monocular RGB camera at 3 extremely low spatial resolutions: 10×10, 5×5 and 3×3 pixels. Specifically, appearance-based features are extracted across a variety of head pose images, and passed to a Support Vector Regressor (SVR) for training and, subsequently, testing.

2.1.1 Related Work

Over the last two decades, numerous publications have appeared in the computer vision literature on human head pose estimation. For example, Wang *et al.* (Wang et al., 2013), estimate pose by learning a random regression forest with 2D SIFT and 3D HoG features from RGB and depth images captured by a Kinect sensor. Saeed *et al.* (Anwar Saeed, 2015) first localize the face using the Viola-Jones face detector, then extract 2D HOG features from RGB and depth images, and finally infer the head pose by applying SVR to concatenated RGB and depth feature vectors. Fanelli *et al.* (Fanelli et al., 2013) jointly estimate the nose tip location and head orientation using a discriminative random regression forest with only depth appearance patches.

In this thesis, we are interested in fine-grained estimation of yaw, pitch and roll angles of the human head using *single extremely low resolution* RGB frame. Some related work in this domain can be found in (Ahn et al., 2015; Drouard et al., 2015; Gourier et al., 2007) and (Murphy-Chutorian and Trivedi, 2009). Gourier *et al.* (Gourier et al., 2007) apply a linear auto-associative neural network on normalized 23×30 facial regions to estimate yaw and pitch angles. However, both training and testing angles for the neural network range from $-90°$ to $90°$ with coarse steps of $15°$. Robertson and Reid (Robertson and Reid, 2006) use nearest-neighbor matching based on skin to non-skin distribution to estimate head pose in surveillance videos with heads as small as 20 pixels in height. Since ground truth is unavailable, the validation is based on subjective classification of observed samples into one of 8 discrete directions ($45°$ apart). Perhaps, the work most related to ours is that of Ahn

et al. (Ahn et al., 2015), who leverage a deep neural network to learn the mapping function between visual appearance and the yaw, pitch, and roll head pose angles. They report a 3-degree mean squared error for 32×32-pixel images. However, unlike the subject-independent cross validation used in this work, their evaluation protocol does not guarantee the same independence between training and testing data.

Compared to the above works, we study algorithms at much lower resolutions, in fact as low as 3×3 pixels.

2.1.2 Head Pose Estimation Algorithm

We propose to use a nonlinear regression method for estimating human head pose with a monocular RGB camera across 3 extremely low spatial resolutions: 10×10, 5×5 and 3×3 pixels (Fig. 2·1). Appearance-based features (HOG or gradients) are extracted across a variety of head pose images, and passed to SVR for training as well as testing.

Preprocessing

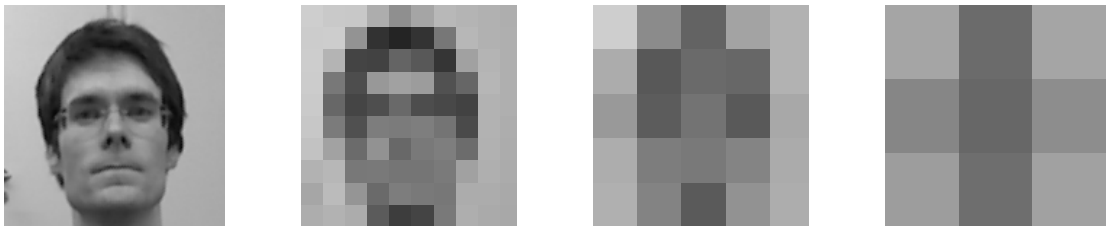
We first convert each RGB image to grayscale and then apply spatial mean-variance normalization. If $x_{i,j}$ denotes the grayscale value of a pixel at spatial location (i, j), we normalize each pixel in the image as follows:

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu}{\sigma}, \tag{2.1}$$

where μ and σ denote, respectively, the pixel mean value and empirical standard deviation computed from the whole image.

Feature Extraction

We evaluate two features in this work: HOG and a new gradient-based feature. HOG is commonly used in many applications and has been widely applied in head pose



Original 10×10 5×5 3×3

Figure 2·1: Visualization of a head pose at extremely low spatial resolutions. Pose becomes harder to distinguish at lower resolutions but privacy is preserved.

estimation as well (Anwar Saeed, 2015; Murphy-Chutorian and Trivedi, 2009; Wang et al., 2013). For HOG at 10×10 spatial resolution, we use a cell size of 2×2 pixels, and a block size of 2×2 cells. Each cell contains a histogram of 9 evenly spaced orientation bins from -180 to +180 degrees, and each block is spaced by a stride of one cell. For a 10×10 image, this results in a length 576 HOG feature vector. At lower resolutions (5×5 and 3×3), only HOG with a 1×1 pixel cell performs reasonably well. The block size and spacing are unchanged. The length of these HOG feature vectors are 576 and 144, respectively. Larger cells do not work well since not enough gradient information is available to meaningfully populate all the HOG bins.

Therefore, at 5×5 and 3×3 resolutions we introduce a new 4-dimensional pixel-wise feature descriptor defined as follows:

$$f_{i,j} = \left(\frac{\partial \hat{x}_{i,j}}{\partial x}, \frac{\partial \hat{x}_{i,j}}{\partial y}, \|\nabla \hat{x}_{i,j}\| \, , \theta_{i,j} \right) \tag{2.2}$$

where $\partial \hat{x}_{i,j}/\partial x$ and $\partial \hat{x}_{i,j}/\partial y$ are the first-order partial derivatives computed at pixel (i, j), and $\|\nabla \hat{x}_{i,j}\|$ and $\theta_{i,j}$ are, respectively, the gradient magnitude and orientation. In this way, each pixel is described by a 4-dimensional vector and the final feature descriptor of the entire image is an $R \times R \times 4$ - dimensional vector ($R = 10, 5, 3$).

Nonlinear Regression: SVR

Pose estimation can be formulated as a regression problem. In total, we estimate 3 regressors, one for each pose angle (yaw, pitch, roll). We use SVR, a supervised learning algorithm for nonlinear regression that is well-known for its generalization capability and resilience to over-fitting (Abe, 2005). Given a labeled training set $\{(\boldsymbol{f}_j, \theta_j), j = 1, \ldots, N\}$ of N (feature-vector, pose-angle) pairs, the SVR algorithm learns a parametric functional mapping from feature vectors to angle estimates of the form:

$$\hat{\theta}(\boldsymbol{f}) = \sum_{j=1}^{N} w_j \mathcal{K}(\boldsymbol{f}_j, \boldsymbol{f}) + b,$$

where $\boldsymbol{f}$ denotes the extracted feature vector of a test image, $\mathcal{K}(\cdot, \cdot)$ is a chosen positive definite symmetric kernel (e.g., polynomial, radial-basis, etc.), and $b, \boldsymbol{w} := (w_1, \ldots, w_N)^T$, are the parameters of the mapping which are learned from training data. Algorithms for SVR learn the parameters $b, \boldsymbol{w}$, as the solution to the following optimization problem:

$$\min_{b, \boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{j=1}^{N} \max(0, \left|\theta_j - \hat{\theta}(\boldsymbol{f}_j)\right| - \epsilon),$$

where C is a regularization parameter that controls the tradeoff between bias and variance (fidelity to data versus prior). Unlike least squares regression, SVR uses an ϵ-insensitive loss function of the approximation error: $\max(0, |\text{error}| - \epsilon)$, which ignores errors that are smaller than ϵ, is more robust to outliers, and produces solutions that are sparse in terms of the number of nonzero weights $w_1, \ldots, w_N$, that define the solution. The parameter ϵ controls the tradeoff between sparsity and accuracy with a larger ϵ favoring sparser solutions. To implement SVR, we used *lib-SVM* (Chang and Lin, 2011). In our experiments, we used a cubic polynomial kernel, and found the optimal SVR parameters, C and ϵ, through 4-fold cross-validation. For more details

about SVR, we refer the interested reader to (Gunn et al., 1998).

2.1.3 Experimental Results

Dataset

To assess the performance of our algorithm, we used the popular *Biwi Kinect Head Pose Dataset* (Fanelli et al., 2013). This dataset was produced using a Kinect sensor and contains over 15K images of 20 people (6 females and 14 males) continuously rotating their head within all three degrees of freedom: pitch, yaw and roll, across multiple environments and varying lighting conditions. The head orientation spans about $\pm 75°$ for yaw, $\pm 60°$ for pitch, and $\pm 50°$ for roll. For each frame, a depth image and the corresponding RGB image (640×480 pixels) are provided. The ground truth pose angles are provided as well; they were obtained using a user-specific 3D head template and the ICP algorithm (Zhang, 1994). Figure 2·2 shows the histograms of the ground truth values for yaw, pitch and roll. The median value and the average absolute deviation from the median value are, respectively, 6.1° and 23.9° for yaw, -13.8° and 18.8° for pitch, and -1.2° and 7.4° for roll.

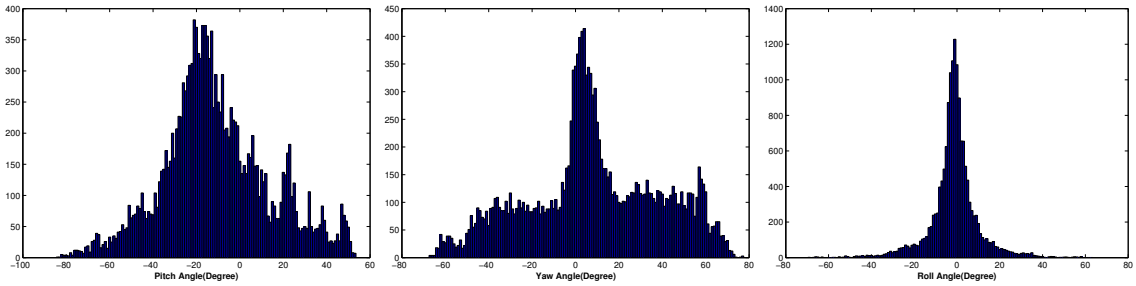


Figure 2·2: Histograms of ground-truth pitch, yaw and roll angles in the *Biwi Kinect Head Pose Dataset*.

In order to use this dataset in the context of low resolutions, we made a few modifications. First, we performed a crude localization of each person's head by manually extracting a fixed region of interest (ROI) for each person's set of full resolution images. In practice, this is equivalent to having a camera with optical settings such

that the subject's face occupies most of the camera's field of view (Fig. 2·1). These extracted ROIs were then decimated to extremely low resolution images (10×10, 5×5, and 3×3) using bi-cubic interpolation.

We performed 4-fold (leave-5-people-out) cross-validation in order to evaluate performance. First, the set of all head pose images from 20 people were split into 4 non-overlapping folds with each fold containing all the images of 5 out of the 20 people. Doing this ensures that not only is there no overlap of images across folds, but also no overlap of people across folds. Then, 3 out of the 4 folds of images are used for training the SVR algorithm and the remaining fold is used for testing. We cycled through all 4 choices of testing folds to get 4 sets of test errors. We repeated this entire process 3 times for different (random) initial splits of 20 people into 4 non-overlapping groups. This gave us 12 sets of test errors which we use to calculate all the mean absolute errors and their standard deviations. This is consistent with the cross-validation scheme performed in the original paper by Fanelli *et al.* (Fanelli et al., 2013).

Impact of spatial resolution

The impact of spatial resolution on the mean absolute error (MAE) of pose estimation was evaluated across 3 extremely low spatial resolutions. The results of our method, in comparison to state-of-the-art full-resolution methods, are summarized in Table 2.1. The algorithm proposed in (Wang et al., 2013) leverages full-resolution RGB images as well as depth maps. In (Anwar Saeed, 2015), the authors report two state-of-the-art MAEs, one when employing HOG features obtained from full resolution RGB images alone and another one when concatenated HOG features from both full resolution RGB and depth images are used. For each pose angle, the 'Median' estimate is the median value of ground truth across the entire dataset. The median is the best constant estimate (estimate with no test image) that minimizes the MAE.

Table 2.1: Mean and standard deviation of the absolute error for yaw, pitch, and roll for various methods.

Method/Resolution	Pitch Error°	Yaw Error°	Roll Error°
SVR:HOG_{rgb} 10×10	12.9±17.2	9.9±12.4	6.9±9.8
SVR:$Grad_{rgb}$ 10×10	14.1±18.6	12.4±15.6	7.2±10.3
SVR:HOG_{rgb} 5×5	16.1±20.1	15.2±19.3	7.6±10.9
SVR:$Grad_{rgb}$ 5×5	13.7±17.6	11.2±14.4	7.7±10.9
SVR:HOG_{rgb} 3×3	18.7±23.1	22.8±28.9	7.6±11.6
SVR:$Grad_{rgb}$ 3×3	15.9±20.2	16.3±20.8	8.0±11.5
Median -	18.8±15.9	23.9±18.9	7.4±8.9
(Wang et al., 2013) HOG_d + $SIFT_{rgb}$ Full resolution	8.5±11.1	8.8±14.3	7.4±10.8
(Anwar Saeed, 2015) HOG_{rgb} Full resolution	5.7±6.1	4.9±5.1	4.8±5.9
(Anwar Saeed, 2015) HOG_{rgb} + HOG_d Full Resolution	5.0±5.8	3.9±4.2	4.3±4.6

The MAE consistently increases, with a decreasing spatial resolution. The performance of our algorithms at 10×10 resolution is significantly better than that of the median estimate. For yaw and roll angles, the performance of our SVR:HOG_{rgb} algorithm is close to that of the algorithm reported in (Wang et al., 2013) which is based on full-resolution RGB and depth images. Compared to the state-of-the-art algorithm (HOG_{rgb} + HOG_d) which uses RGB and depth images simultaneously (Anwar Saeed, 2015), our best results for HOG features at 10×10-pixel resolution are worse by about 5.5° averaged across pitch, yaw and roll angles. This is encouraging because it indicates that a reasonable quality head pose estimate can be obtained even with a 10×10 pixel monocular camera. When one considers the better of the two SVR methods at each spatial resolution, the performance drop from 10×10 pix-

els to 5×5 pixels is not significant: 12.9° to 13.7°, 9.9° to 11.2°, and 6.9° to 7.6°, for pitch, yaw, roll, respectively. At 3×3 resolution, however, the estimation performance breaks down: results are close to or slightly worse than those for the median estimate. This suggests there is very little pose information that can be learned from appearance at such a low resolution.

In terms of individual rotation angles, the MAE of the roll angle is the smallest. This is consistent with the reduced roll variation in this dataset – pitch and yaw vary much more (see histograms in Fig. 2·2). This is also consistent with the observation that the median estimate of the roll angle has a drastically lower MAE than the median estimates for the yaw and pitch angles. Additionally, the estimation performance of the yaw angle is better than that of the pitch angle at 10×10 and 5×5 resolutions. This is intuitive, as the appearance change in the vertical direction is less distinguishable than in the horizontal direction at low spatial resolutions except the very lowest resolution of 3×3 when very little data is available

Regarding privacy preservation performance, we used correct classification rate (CCR) in person identification to measure how much privacy is preserved (the lower, the better). In the BIWI dataset only 4 out of the 20 subjects have two videos recored in different room settings. Thus, we leveraged those 4 subjects' data and performed 2-fold cross-validation (each of the two videos of a subject becomes testing data once) for evaluation. We used HOG features to train support vector machines (SVM) for identification. Table. 2.2 summarizes the identification performance under the three eLR resolutions, the full resolution and a random guess performance. We observe that the identification CCRs are 33.7% at 5×5 resolution and 30.32% at 3×3 resolution, which are close to a random guess. The identification CCR at 10×10 resolution increases to 57.30%, but is still much lower than that from using the full resolution data (100%). These results verify that using eLR data can effectively preserve user's

Table 2.2: Person identification performance at different spatial resolutions.

Method/Resolution	Identification
SVM:HOG$_{rgb}$ Full resolution	100.00%
SVM:HOG$_{rgb}$ 10×10	57.30%
SVM:HOG$_{rgb}$ 5×5	33.70%
SVM:HOG$_{rgb}$ 3×3	30.32%
Random Guess	25.00%

privacy.

Fig. 2·3 shows how the ground truth and estimated yaw, pitch and roll angles change over time in a sample image sequence. Shown are the best estimates at all three extremely low resolutions that we studied: 10×10 resolution with HOG features as well as 5×5 and 3×3 resolutions with gradient-based features. Clearly, smoother and more accurate angular estimates are achieved at higher spatial resolutions, as should be expected due to the continuity of head movements in the sequence. Additionally, the error is more significant at large pose angles as compared to small angles (0° corresponds to facing forward straight towards the camera). For example, at the spatial resolution of 10×10 with HOG features, the MAE for pitch, yaw and roll angles within ±40° are, respectively, 10.7°, 8.5° and 6.5° across the whole dataset. These numbers increase to 32.4°, 13.7° and 34.4° for the angles outside ±40°. We believe this is likely due to large angular rotations becoming far less discriminative at extremely low resolutions.

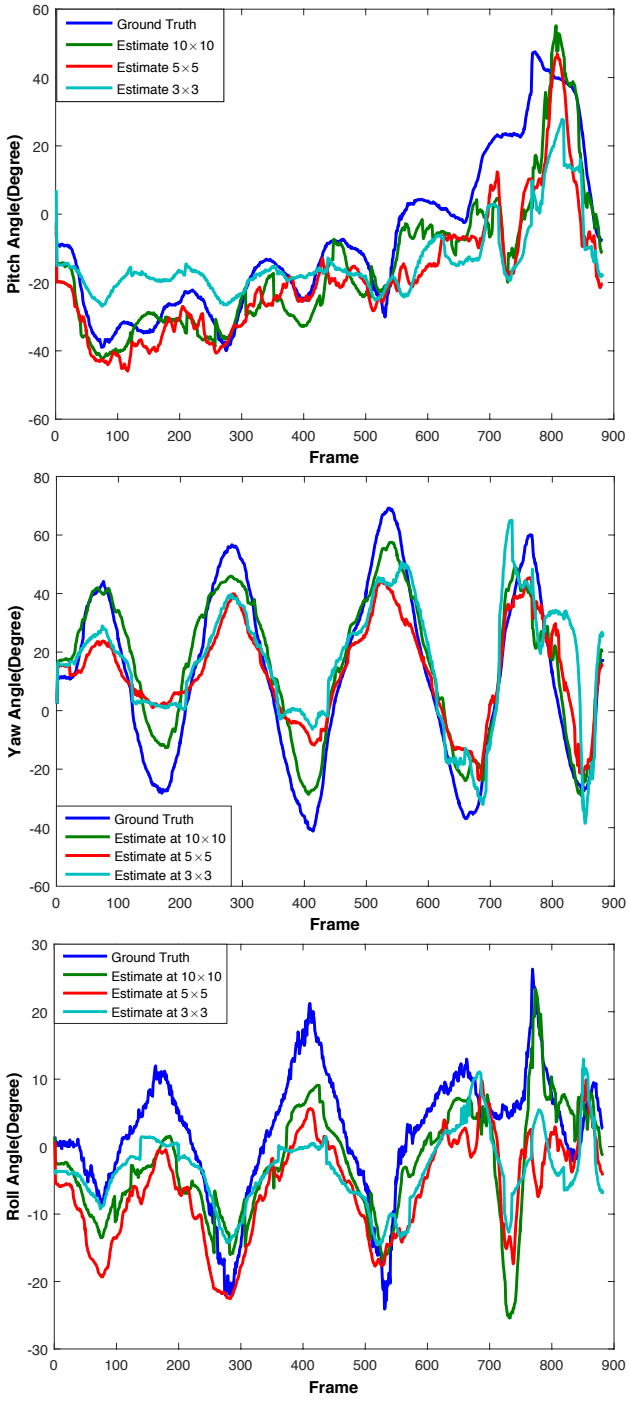


Figure 2·3: Estimates of pitch, yaw, and roll angles at spatial resolutions of 10×10 with HOG feature, and 5×5 and 3×3 with gradient-based feature against ground truth for image sequence #9.

2.1.4 Discussions

We studied the impact of resolution on the accuracy of human head pose estimation and found that a monocular camera with 10×10 resolution can provide estimates that have about twice the error of state-of-the-art methods at full resolution. Even at 5×5 resolution, reasonable results are still attainable (11.2° yaw error). To our knowledge, this is the first attempt to investigate human head pose estimation at such low resolutions. We believe a better face localization would enhance performance. Although we only used RGB information, using depth data could potentially solve many of the inherent problems present in RGB domain. Finally, using multiple frames jointly to exploit the continuity of pose changes, would likely help further. In future work, we are planning to pursue these directions.

2.2 Indoor Occupant Localization Using An Array of Single-Pixel Sensors

Indoor localization has long been of interest in surveillance, for example monitoring of seniors or children in home environments. However, our main motivation for this work is the recent proliferation of sensors in living spaces that has lead to the concept of a smart room.

Early localization systems have mainly focused on location accuracy and involved the use of custom hardware that is expensive to deploy in practice. Newer systems use video cameras and computer vision techniques. While acceptable in scenarios where no expectation of privacy exists (e.g., airports, shopping malls, classrooms), such methods are unlikely to be deployed in a smart home.

We seek to design a system that can strike a balance between localization accuracy and privacy preservation. While methods have been developed that degrade the output of a camera to preserve privacy, they are not immune to eavesdropping and still

require costly cameras and processors. As an alternative, we propose to use overhead-mounted single-pixel visible-light sensors to estimate an occupant's location. To fully explore the potential of the proposed system, we consider two scenarios: classification, for coarse-grained localization of an occupant within a rectangular cell of suitable size, and regression, for fine-grained localization to find continuous coordinates of an occupant.

2.2.1 Related Work

Our focus is on localization of people indoors using infrastructure-mounted sensors (e.g., overhead). We do not consider localization methods that leverage radio signals or cameras in mobile devices carried by individuals that are often used in location-based services, e.g., indoor navigation, advertising.

One can classify indoor localization systems into two categories: active systems and passive systems (Deak et al., 2012). Active systems require users to wear a physical electronic device, while passive systems do not. Some active systems, such as ActiveBadge (Want et al., 1992), Cricket (Priyantha et al., 2000), and Ubisense (Steggles and Gschwind, 2005), can provide accurate position estimates. However, the main drawback of these systems is the need to carry a sensor. Furthermore, improper placement of sensors can impact performance (Kunze and Lukowicz, 2014).

On the other hand, passive localization techniques have become more popular as they are low-cost and user-friendly. One type of passive systems uses standard WiFi infrastructure to infer location (Krumm and Horvitz, 2004; Moussa and Youssef, 2009; Youssef et al., 2007; Kosba et al., 2009). Localization performance, e.g., with a median error of 1.5 meters (Krumm and Horvitz, 2004), is attainable. The main shortcoming of this approach is that in real scenarios signal measurements are affected by multi-path, reflections, obstacles (individual not in line-of-sight), etc. (Krumm, 2009).

Another recently-proposed solution is a "smart floor", a floor with an embedded fiber sensor array (Feng et al., 2015). The sensor array generates a pressure distribution map that can be used for indoor target localization. However, this is difficult to install within existing buildings and is costly.

Vision-based localization approaches have been proposed focusing on transforming a simple environment into an intelligent one (Krumm et al., 2000). The Microsoft EasyLiving project (Brumitt et al., 2000) uses cameras installed in a room to localize occupants and trigger events based on their location. In a later work (Yu et al., 2006), the authors proposed a solution that fuses data from RGB and floor sensors to enhance localization accuracy.

A recent camera-based occupant localization framework has been developed to address residents' higher-level needs, like evocation of memory. In the usage scenario of a board game, the system can automatically document the events by taking photos of players at predefined time intervals without drawing the residents' attention away from the situation (Engelbrecht et al., 2015).

While most of the camera-based localization systems violate an individual's privacy, recently there has been work to address this issue in the context of other vision tasks. Dai *et al.* (Dai et al., 2015) have studied trade-offs between action recognition performance and the number of cameras and their resolution (spatial and temporal) in a smart room environment. They reported that 5 single-pixel cameras can achieve reasonable action recognition performance. Jia and Radke (Jia and Radke, 2014) explored privacy-preserving tracking and coarse pose estimation using a network of ceiling-mounted time-of-flight (TOF) sensors. However, their framework requires a dense deployment of TOF sensors (with spacing of less than 0.25m) to achieve acceptable localization performance. Compared to their work, our method requires fewer and simpler sensors for similar localization performance.

2.2.2 System Setup

We use six TCS3472 color sensors from AMS AG in a configuration depicted in Fig. 2·4. All sensors are mounted overhead on an aluminum scaffolding (Fig. 2·5) and pointed downward. The floor area being monitored has dimensions 2.37m×2.72m. The TCS3472 color sensor outputs 16-bit intensity measurements for red, green, blue (filtered), and white (unfiltered) light. The sensors also have a configurable gain and integration time, which were set to 60× and 100.8 ms, respectively. Each sensor's lens limits its field of view to about 36°. All sensors are networked via a Raspberry-Pi multiplexer connected to a host computer.

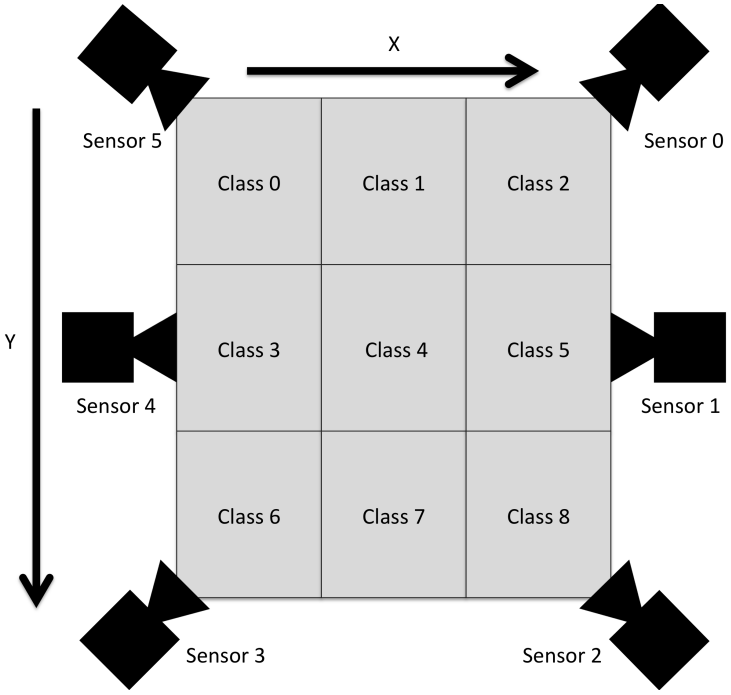


Figure 2·4: Schematic representation of the physical testbed.

In order to train our localization algorithms and evaluate our system's performance, we captured ground-truth data using OptiTrack (Point, 2011), a commercial motion capture system. The system uses 12 infrared cameras with infrared light sources to track reflective markers with very high precision (about 1mm). In our

experiments, the markers were attached to a helmet worn by each subject during data capture. The color sensors and OptiTrack system were configured to enable synchronization of the recorded data in time. Synchronization was achieved using a light impulse that produced an easily-identifiable feature in the signals. To see this pulse in the OptiTrack system, one of its cameras was configured to capture visible light.

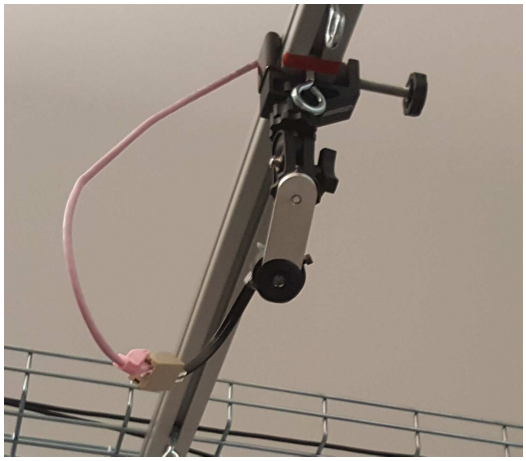


Figure 2·5: One of six TCS3472 single-pixel visible-light sensors.

2.2.3 Localization Algorithms

We develop two purely data-driven (learning-based) localization algorithms. The first is a coarse localization algorithm that classifies a subject's location as belonging to one of the 9 cells arranged in a 3×3 grid (Fig. 2·4) using a support vector machine (SVM) classifier. The algorithm provides, at each time instant, an estimate of the cell in which the subject is supposedly located. The second localization algorithm uses support vector regression (SVR) (Smola and Vapnik, 1997) to provide, at each time instant, fine-grained real-valued estimates $\widehat{x}, \widehat{y}$ of the true x and y positions of the subject. The x and y positions are estimated separately (one regressor per estimate) and in a memoryless manner, i.e., without the use of tracking algorithms. At each

time instant, the algorithm provides two real values indicating the location on the floor.

Preprocessing and Feature Extraction

Each sensor measures R, G, B (filtered) and W (unfiltered) light components in its field of view. In order to eliminate bias due to the color of clothing that subjects may wear, we calculate luminance from R, G, B readings at each time instant. Let $I_{j,t}$ denote the computed luminance for sensor number j at time t. We normalize the luminance to the range [-1,1] as follows:

$$\widehat{I}_{j,t} = 2 \times \frac{I_{j,t} - I_{j,min}}{I_{j,max} - I_{j,min}} - 1$$

where $I_{j,max}$ and $I_{j,min}$ are the maximum and minimum values of luminance for sensor j over time that are both estimated from the dataset.

After normalization, the luminance values from all 6 sensors at time t are concatenated to form a six-dimensional feature vector $\mathbf{f}_t = (\widehat{I}_{0,t}, ..., \widehat{I}_{5,t})^\top$ that is used in our SVM and SVR algorithms below.

Coarse-grained Localization via Classification

In order to obtain coarse-grained position estimates, we treat localization as a classification problem with 9 classes corresponding to each of the 9 rectangular cells in Fig. 2·4. We use cell positions obtained from the OptiTrack measurements as the ground-truth labels to train a "one-versus-one" 9-class kernel SVM classifier (Hsu and Lin, 2002) based on the 6-dimensional feature vectors $\boldsymbol{f}_t$. We use the radial basis function as the kernel. A "one-versus-one" multi-class SVM classifier is based on training $\binom{9}{2}$ binary classifiers, one for every pair of distinct classes. The label of a test sample is determined as the class which "wins" most against all other classes in one-versus-one comparisons. The SVM classifier is well-known for its generalization

capability and resilience to over-fitting (Abe, 2005).

Fine-grained Localization via Regression

We formulate fine-grained localization as a regression problem and train 2 regressors separately, one for each dimension, x and y. The machine-learning algorithm we use is kernel SVR, a variation of kernel SVM for regression. In regression, the label becomes a real coordinate value l instead of a class index. Similarly to kernel SVM, the kernel SVR algorithm learns a parametric function mapping from feature vectors $\boldsymbol{f}$ to coordinate estimates $\widehat{l}$ (either $\widehat{x}$ or $\widehat{y}$) of the form:

$$\widehat{l}(\boldsymbol{f}) = \sum_{i=1}^{N} w_i \mathcal{K}(\boldsymbol{f}_i, \boldsymbol{f}) + b,$$

where N is the number of labeled training samples and $\mathcal{K}(\cdot, \cdot)$ is a chosen positive definite symmetric kernel (we use the radial basis function). The details of how SVR learns the parameters b and $\boldsymbol{w} = [w_1, ..., w_N]^T$ can be found in section 2.1.2

To implement SVM and SVR, we used *LIBSVM* (Chang and Lin, 2011). In our experiments, we used radial basis function with parameter σ as a kernel in both algorithms and performed grid search to find the optimal parameters.

2.2.4 Experimental Results

Dataset

In order to test the robustness of our methodology on real data, we collected a dataset of locations of 4 people. Each person took four separate walks of about 90 seconds within a 2.37m $\times$ 2.72m floor area in the field of view of the sensors. The data collected from each walk consists of measurements from each of the six color sensors and the location of the person's head as provided by the OptiTrack system (Fig. 2·6 and 2·7). Subjects were instructed to perform a random walk and were encouraged

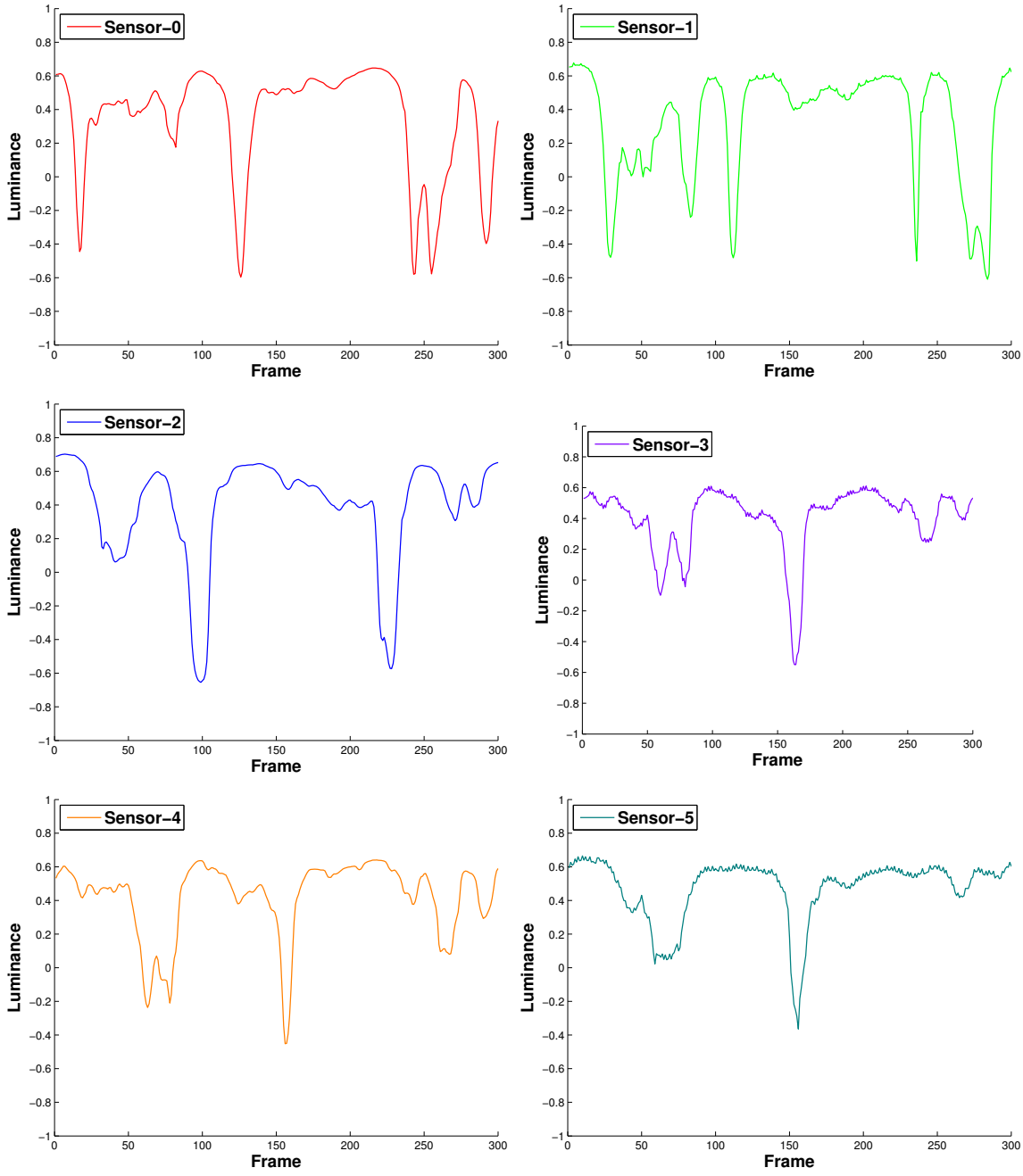


Figure 2·6: Example of a walk for one subject: luminance evolving over time for each of the 6 sensors.

to vary the walking speed.

We made sure that all walks have the same number of samples by randomly trimming excess samples from the start and end of each walk. We performed the

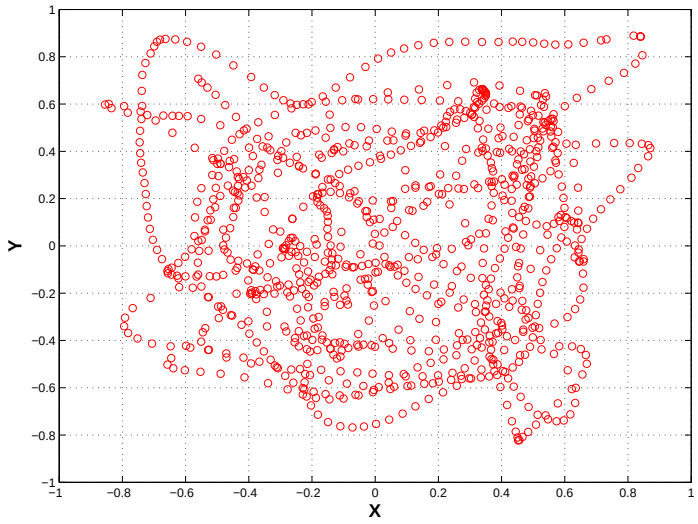


Figure 2·7: Example of a walk for one subject: corresponding locations recorded by the OptiTrack system and normalized to the range [-1,1].

trimming on both the sensor data and OptiTrack data. After trimming, each walk had 971 samples. In both algorithms, we used identical sizes for the training and testing sets as explained in the next subsection.

Scenarios

In our experiments, we made sure that samples from the same walk *are not included* in both the training and testing sets at the same time because consecutive feature vectors in a given walk are likely to be almost identical (e.g., subject slows down or pauses). Had we not done so, very similar feature vectors $\boldsymbol{f}_t$ could have appeared in both the training and testing sets and biased the results.

We evaluated performance with two primary usage scenarios in mind. The first usage scenario considers a public setting, such as a conference room, where the system cannot be trained on all subjects (new subjects, never seen by the system, may enter). The second usage scenario considers a private setting, such as a home, where the system is being used primarily by the same set of people and thus may be trained on

all users.

In the public scenario, we evaluated performance using *leave-one-person-out* cross-validation where samples from all 12 walks of three out of four people form the training set and samples from all 4 walks of the fourth person are used as the testing set. Creating data splits in this way ensures that there is no overlap of walks or people between the training and testing sets. Thus, in each of the four data splits, the number of samples used for training is equal to $N = 971 \times 12 = 11,652$ and the number of samples used for testing is equal to $N_{\text{test}} = 971 \times 4 = 3,884$. We repeat this procedure four times to cover all possible combinations of people left out and report classification and regression performance metrics averaged across all $4N_{\text{test}} = 15,536$ test samples from all four splits. It is important to note that this scenario could be considered the most challenging as the person in the testing set does not appear in the training set.

In the private scenario, we evaluated performance using *leave-one-walk-out* cross-validation. For each person, we take three of his/her walks and put them into the training set. Then, we take the remaining walks and use them for testing. This process is repeated four times with different combinations of walks from each person. The sizes of training and testing sets used in the private scenario therefore exactly match those in the public scenario.

Finally, in order to test how sensitive the performance is to the number of sensors, we repeated all the above experiments using only the 4 corner sensors (sensors 0, 2, 3, and 5 in Fig. 2·4) and compared them to the results obtained using all 6 sensors.

Classification Results

First, ground-truth classes are computed from the ground-truth locations provided by the OptiTrack system by quantizing the x and y ground-truth locations to one of the 9 cells depicted in Fig. 2·4. Then, for each test sample we compared the class estimate

produced by the SVM classifier with the ground-truth class label and calculated the overall correct classification rate (CCR) which is the percentage of the test samples that are correctly classified. Results for both public and and private scenarios, and for all 6 and 4 corner sensors are shown in Table 2.3.

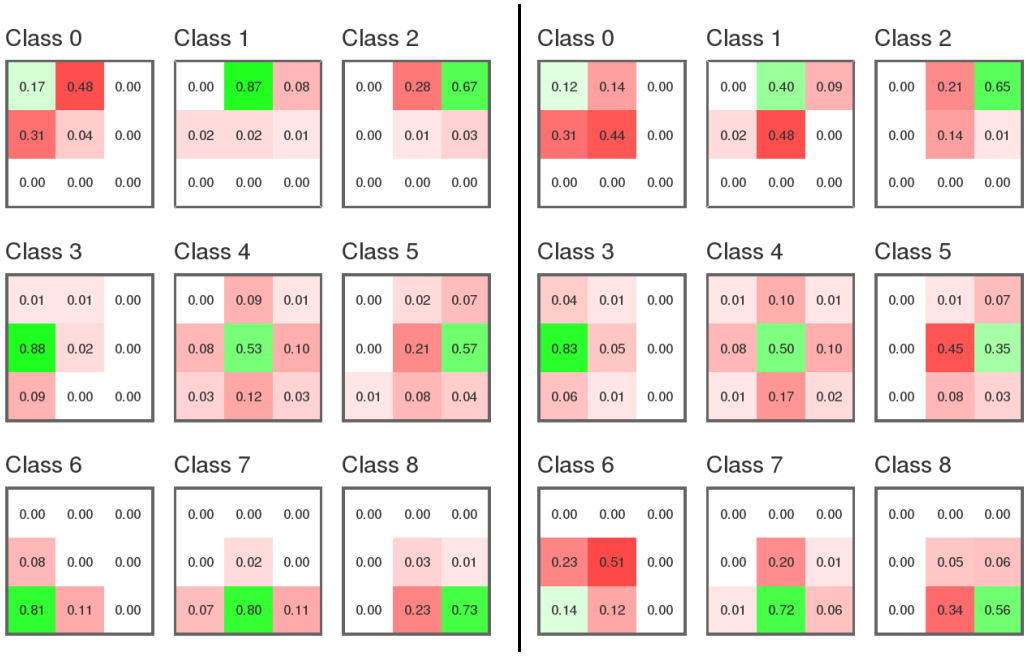


Figure 2·8: Confusion views for classification using SVM (left) and for quantized SVR (right) of 3×3 class estimates. The color intensity (shade) of each cell is inversely proportional to the recognition (green) or confusion (red) rate (the darker the color, the higher the rate).

Table 2.3: CCR for classification using either all 6 sensors or 4 corner sensors in public and private scenarios.

	6 Sensors		4 Sensors	
	Private	Public	Private	Public
SVM CCR	71.86%	66.68%	63.64%	56.06%
QSVR CCR	51.76%	48.46%	46.39%	44.14%

For comparison, Table 2.3 also includes quantized regression results ("QSVR CCR"). Basically, we quantized the (x, y) positions estimated using the SVR algorithm (Section 2.2.4) to the 3×3 grid and then calculated the CCR.

A detailed performance comparison of SVM and QSVR on a per-cell basis is shown in Fig. 2·8 where 9 confusion matrices for both SVM and QSVR are shown in a spatially-consistent fashion with the physical testbed layout (Fig. 2·4). Each 3×3 grid shows the correct and incorrect classifications for its respective class. The green cell in each grid is the correct classification and red cells are the incorrect classifications. For example, in the class 4 grid for SVM, 53% of the samples labeled as class 4 were predicted correctly, 9% of the class 4 samples were incorrectly classified as class 1, 1% as class 2, 8% as class 3, etc.

We note that both algorithms incorrectly predict class 0: CCR of 17% for SVM and 12% for QSVR. This significantly reduces the overall CCR and is likely due to having fewer samples recorded in this area. Samples from class 0 make up about 3% of the overall data set while the next smallest class 2 makes up about 6% of the data set.

Regression Results

In the regression case, we measure the performance for each coordinate separately using mean absolute error (MAE) and mean squared error (MSE) between the estimates $(\widehat{x}, \widehat{y})$ and the ground-truth measurements (x, y) over $4N_{\text{test}}$ samples. We also compute the mean and the standard deviation of the Euclidean distance between the estimated and ground-truth locations, and the associated $\pm 1\sigma$ confidence: $\pm$Std. Dev.$/\sqrt{4N_{\text{test}}}$ around the mean estimate. All these performance measures are shown in Table 2.4.

Fig. 2·9 shows location estimates $\widehat{x}$ and $\widehat{y}$ in normalized coordinates against ground-truth locations x and y for one sample walk. While there is a sizable discrepancy in positions at many time instants, the overall trends are quite similar. Furthermore, the estimates are more accurate when the subject significantly changes position which is to be expected as the recorded data are closely related to occlusions

Table 2.4: MAE and MSE for location estimates via regression, and mean/standard deviation of the distance between estimated and ground-truth locations, all in [m].

	6 Sensors		4 Sensors	
SVR	Private	Public	Private	Public
MAE x	0.1826	0.1972	0.2045	0.2138
MSE x	0.0609	0.0672	0.0791	0.0837
MAE y	0.2195	0.2459	0.2503	0.2843
MSE y	0.0860	0.1165	0.1087	0.1441
Mean distance	0.3144	0.3500	0.3555	0.3939
$\pm 1\sigma$	$\pm$	$\pm$	$\pm$	$\pm$
confidence	0.0018	0.0020	0.0020	0.0022
Std. Dev.	0.2192	0.2473	0.2478	0.2697

of the background by the subject's body (with larger movements, potentially more varied backgrounds are covered).

2.2.5 Discussions

Both algorithms perform better in the private scenario than in the public one, which is as expected. For the classification algorithm, we see from Table 2.3 that the CCR for 6 sensors in the public case is lower by 5.18% than the one for the private case. Similarly, for 4 sensors the difference is 7.58%. A similar observation can be made for the quantized regression algorithm (QSVR) although the differences are smaller. In the regression case (Table 2.4), both mean absolute and mean squared errors increase in the public scenario compared to the private one. Examining the differences in the mean distance between the private and public scenarios, we see that there is a consistent decrease in performance for both 6- and 4-sensor cases (Table 2.4). Between the private and public cases with 6 sensors, there is an increase of 0.0356m (11%) in mean distance between locations. This observation can also be made in the 4 sensor case, where there is a similar increase of 0.0384m (11%). These performance drops are not unexpected, since in the public case the classifier/regressor has not been trained

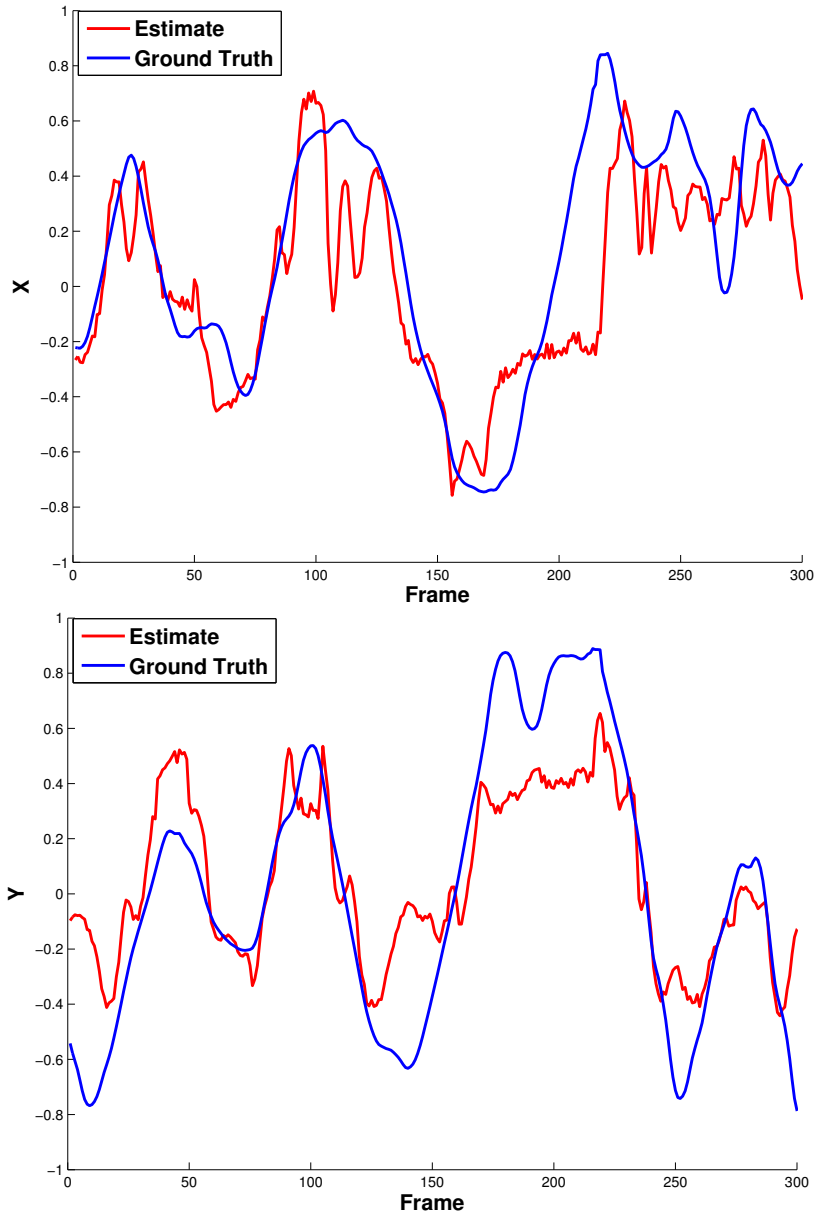


Figure 2·9: Estimates of locations shown against ground-truth locations for the sample walk from Fig. 2·7.

on one of the subjects while in the private case it has been trained on all subjects.

We also note a significant drop in performance when the number of sensors is reduced form 6 to 4. Again, this is not unexpected since with 6 sensors, data from additional viewpoints is available. For both the public and private scenarios, the mean dis-

tance between the estimated and ground-truth locations increases by 0.0411m (13%) and 0.0439m (12%), respectively.

We have demonstrated that on average the system is capable of localizing a single individual to within 35cm (14in) of the true position on a floor area of 2.37m×2.72m (see Table. 2.4). This is a useful result for visible light communication. With this precision and also knowing the subjects orientation (Chen et al., 2016) it would be possible to identify ceiling LED transmitters that are within line-of-sight from a VLC-compatible hand-held device, such as a smart-phone. This is a significant step towards making VLC a reality for mobile devices in a home environment.

2.3 Human Action Recognition from eLR Videos

Human action and gesture recognition have received significant attention in computer vision and signal processing communities (Simonyan and Zisserman, 2014b; Wang and Schmid, 2013; Xia et al., 2012). Recently, various ConvNet-based models have been applied in this context and achieved substantial performance gains over traditional methods that are based on hand-crafted features (Krizhevsky et al., 2012; Sharif Razavian et al., 2014). Further improvements in the performance have been realized by using a two-stream ConvNet architecture (Simonyan and Zisserman, 2014a) in which a spatial network concentrates on learning appearance features from RGB images while a temporal network takes optical flow snippets as input to learn dynamics. The final decision is made by averaging outputs of the two networks. More recent work (Feichtenhofer et al., 2016; Lin et al., 2015; Park et al., 2016) suggests fusion of spatial and temporal cues at an earlier stage so the appearance features are registered with motion features before the final decision. Results indicate that this approach improves action recognition performance.

As promising as these recent ConvNet-based models are, they typically rely upon

data at about 200×200-pixel resolution that is likely to reveal an individual's identity. In this section, we present multiple ConvNets to perform reliable action recognition at extremely low resolutions (thus protecting privacy). In particular, we adapt an existing end-to-end two-stream fusion ConvNet to eLR action recognition. Furthermore, we propose a semi-coupled two-stream fusion ConvNet that leverages HR (32×32-pixel resolution in our study) videos during training in order to help the eLR (12×16-pixel resolution in our study) ConvNet obtain enhanced discriminative power by sharing filters between eLR and HR ConvNets.

2.3.1 Related Work

ConvNets have been recently applied to action recognition and quickly yielded state-of-the-art performance. In the quest for further gains, a key question is how to properly incorporate appearance and motion information in a ConvNet architecture. In (Ji et al., 2013; Karpathy et al., 2014; Tran et al., 2015), various 3D ConvNets were proposed to learn spatio-temporal features by stacking consecutive RGB frames in the input. In (Simonyan and Zisserman, 2014a), a novel two-stream ConvNet architecture was proposed which learns two separate networks: one dedicated to spatial RGB information, and another dedicated to temporal optical flow information. The softmax outputs of these two networks are later combined together to provide a final "joint" decision. Following this pivotal work, many works have extended the two-stream architecture such that only a single, combined network is trained. In (Lin et al., 2015), bilinear fusion was proposed in which the last convolutional layers of both networks are combined using an outer-product and pooling. Similarly, in (Park et al., 2016) multiplicative fusion was proposed, and in (Feichtenhofer et al., 2016) 3D convolutional fusion was introduced (incorporating an additional temporal dimension). However, all these methods were applied to standard-resolution video, and have not, to the best of our knowledge, been applied in the eLR context.

There have been few works that have addressed eLR in the context of visual recognition. In (Wang et al., 2016), very low resolution networks were investigated in the context of eLR *image* recognition. The authors proposed to incorporate HR images in training to augment the learning process of the network through filter sharing (PCSRN). In (Dai et al., 2015), eLR action recognition was first explored using l_1 nearest-neighbor classifiers to discriminate between action sequences. More recently, egocentric eLR activity recognition was explored in (Ryoo et al., 2016). The authors introduced inverse super resolution (ISR) to learn an optimal set of image transformations during training that generate multiple eLR videos from a single HR video. Then, they trained a classifier based on features extracted from all generated eLR videos. The per-frame features include histogram of pixel intensities, histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), histogram of optical flow (HOF) (Chaudhry et al., 2009) and ConvNet features. To capture temporal changes, they used the Pooled Time Series (POT) feature representation (Ryoo et al., 2015) which is based on time-series analysis. This classifier was finally used in testing. However, in keeping with recent research trends our aim is to develop an end-to-end, ConvNet-only solution that avoids hand-crafted features and, therefore, minimizes human intervention. We benchmark our proposed methodologies against last two works, and show consistent recognition improvement.

2.3.2 Action Recognition Algorithms

We propose two improvements to the two-stream architecture in the context of eLR. First, we explore methods to fuse the spatial and temporal networks, which allows subsequent layers to amplify and leverage joint spatial and temporal features. Second, we propose using semi-coupled networks which leverage HR information in training to learn transferable features between HR and eLR frames, resembling domain adaptation, in both the spatial and temporal streams.

Fusion of two-stream networks

Multiple works have extended two-stream ConvNets by combining the spatial and temporal cues such that only a single, combined network is trained (Feichtenhofer et al., 2016; Lin et al., 2015; Park et al., 2016). In this section, we discuss three fusion methods that we explore and implement in the context of eLR.

In general, fusion is applied between a spatial ConvNet and a temporal ConvNet. A fusion function f: $f(\boldsymbol{x}_s^n, \boldsymbol{x}_t^n) \rightarrow \boldsymbol{y}^n$ fuses spatial features at the output of the n-th layer $\boldsymbol{x}_s^n \in \mathbb{R}^{H_s^n \times W_s^n \times D_s^n}$ and temporal features at the output of the n-th layer $\boldsymbol{x}_t^n \in \mathbb{R}^{H_t^n \times W_t^n \times D_t^n}$ to produce the output features $\boldsymbol{y}^n \in \mathbb{R}^{H_o^n \times W_o^n \times D_o^n}$, where H, W, and D represent the height, width and the number of channels respectively. For simplicity, we assume $H_o = H_s = H_t, W_o = W_s = W_t$, and $D_s = D_t$ (D_o is defined below). We discuss the fusion function for three possible operators:

Sum Fusion: Perhaps the simplest fusion strategy is to compute the summation of two feature maps at the same pixel location (i, j) and the same channel d:

$$\boldsymbol{y}^{n,sum}(i,j,d) = \boldsymbol{x}_s^n(i,j,d) + \boldsymbol{x}_t^n(i,j,d) \tag{2.3}$$

where $1 \leq i \leq H_o$, $1 \leq j \leq W_o$, $1 \leq d \leq D_o$ ($D_o = D_s = D_t$) and $\boldsymbol{x}_s^n$, $\boldsymbol{x}_t^n$, $\boldsymbol{y}^n \in \mathbb{R}^{H_o \times W_o \times D_o}$. The underlying assumption of summation fusion is that the spatial and temporal feature maps from the same channel will share similar contexts.

Concat Fusion: The second fusion method we consider is a concatenation of two feature maps (in an interleaved fashion) at the same spatial location (i, j) across channel d:

$$\boldsymbol{y}^{n,cat}(i,j,2d) = \boldsymbol{x}_s^n(i,j,d), \tag{2.4}$$

$$\boldsymbol{y}^{n,cat}(i,j,2d+1) = \boldsymbol{x}_t^n(i,j,d) \tag{2.5}$$

where $\boldsymbol{y}^{n,cat} \in \mathbb{R}^{H_o \times W_o \times D_o}$, $D_o = D_s + D_t$. Unlike the summation fusion, the concate-

nation fusion does not actually blend the feature maps together.

Conv Fusion: The third fusion operator we explore is convolutional fusion. First, $\boldsymbol{x}_s^n$ and $\boldsymbol{x}_t^n$ are concatenated as shown in (2.4, 2.5). Then, the stacked up feature map is convolved with a bank of filters $\mathcal{F} \in \mathbb{R}^{1\times1\times D_o\times D_o'}$ as follows:

$$\boldsymbol{y}^{n,conv} = \boldsymbol{y}^{n,cat} * \mathcal{F} + b, \tag{2.6}$$

where $b \in \mathbb{R}^{D_o'}$ is a bias term. The filters have dimensions $1 \times 1 \times D_o$, $D_o = D_s + D_t$ and are used to learn weighted combinations of feature maps $\boldsymbol{x}_s^n, \boldsymbol{x}_t^n$ at a shared pixel location. For our experiments, we have set the number of filters to $D_o' = 0.5 \times D_o$.

Note, that regardless of the chosen fusion operator, the network will select filters throughout the entire network so as to minimize loss, and optimize recognition performance. Also, we would like to point out that other fusion operators, such as max, multiplication, and bilinear fusion (Lin et al., 2015), are possible, but have been shown in (Feichtenhofer et al., 2016) to perform slightly worse than the operators we've discussed. Finally, it is worth noting that the type of fusion operation and the layer in which it occurs have a significant impact on the number of parameters. The number of parameters can be quite small if fusion across networks occurs in early layers. For example, convolutional fusion requires additional parameters since introducing a convolutional layer requires more filters. Regarding where to fuse the two networks, we adopt the convention used in (Feichtenhofer et al., 2016) to fuse the two networks after their last convolutional layer (see Fig.2·10). We later report the results of fusion after the last convolutional layer (Conv3) and the first fully-connected layer (Fc4) and contrast their classification performance.

Semi-coupled networks

Applying recognition directly to eLR video is not robust as visual features tend to carry little information (Wang et al., 2016). However, it is possible to augment ConvNet training with an auxiliary, HR version of the eLR video, but only use an eLR video in testing. In this context, we propose to use semi-coupled networks which

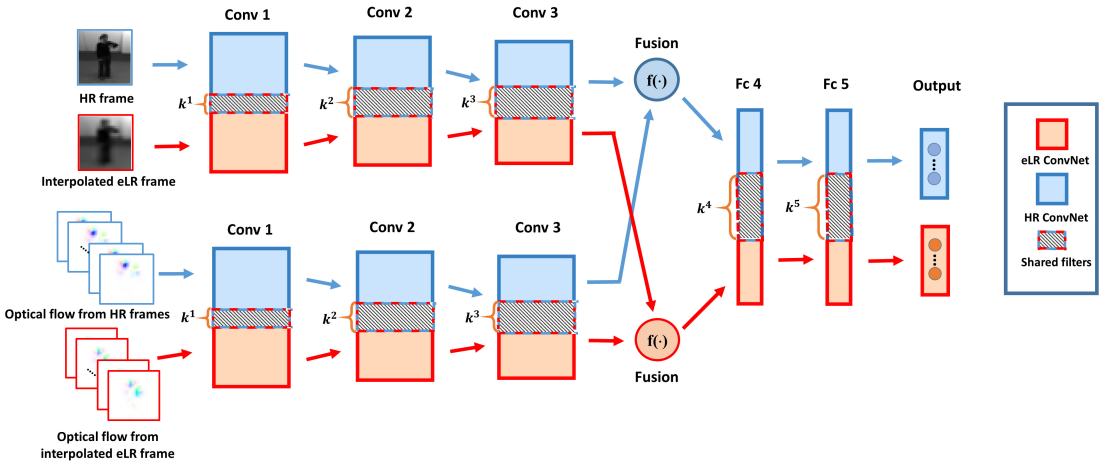


Figure 2·10: Visualization of the proposed semi-coupled networks of two fused two-stream ConvNets for video recognition. We feed HR RGB and optical flow frames (32×32 pixels) to the HR ConvNet (colored in blue). We feed eLR RGB (16×12 interpolated to 32×32 pixels) and optical flow frames (computed from the interpolated 32×32 pixel RGB frames) to the eLR ConvNet (colored in red). In training, the two ConvNets share k^n ($n = 1, ..., 5$) filters (gray shaded) between corresponding convolutional and fully-connected layers. Note that the deeper the layer, the more filters are being shared. In testing, we decouple the two ConvNets and only use the eLR network (the red network which includes the shared filters).

share filters between eLR and HR fused two-stream ConvNets. The eLR two-stream ConvNet takes an eLR RGB frame and its corresponding eLR optical flow frames as input. As we will discuss later, each RGB frame corresponds to multiple optical flow frames. The eLR RGB frames are interpolated to 32×32 pixels from their original 16×12 resolution. The eLR optical flow is computed from the interpolated

32×32 eLR RGB frames. The HR two-stream ConvNet simply takes HR RGB and its corresponding HR optical flow frames of size 32×32 as input. In layer number n of the network ($n = 1, \ldots, 5$), the eLR and HR two-stream ConvNets share k^n filters. During training, we leverage both eLR and HR information, and update the filter weights of both networks in tandem. During testing, we decouple these two networks and only use the eLR network which includes shared filters. This entire process is illustrated in Fig. 2·10.

The motivation for sharing filters is two-fold: first, sharing resembles domain adaptation, aiming to learn transferable features from the source domain (eLR images) to the target domain (HR images); second, sharing can be viewed as a form of data augmentation with respect to the original dataset, as the shared filters will see both low and high resolution images (doubling the number of training inputs). However, it is important to note that in practice, as shown in (Lui et al., 2009), the mapping between eLR and HR feature space is difficult to learn. As a result, the feature space mapping between resolutions may not fully overlap or correspond properly to one-another after learning. To address this, we intentionally leave a number of filters ($D^n - k^n$) unshared in layer n, for each n. These unshared filters will learn domain-specific (resolution specific) features, while the shared filters learn the non-linear transformations between spaces. To implement this filter sharing paradigm, we alternate between updating the eLR and HR two-stream ConvNets during training. Let $\boldsymbol{\theta}_{\text{eLR}}$ and $\boldsymbol{\theta}_{\text{HR}}$ denote the filter weights of the eLR and the HR two-stream ConvNets. These two filters are composed of three types of weights: $\boldsymbol{\theta}_{\text{shared}}$, the weights that are shared between both the eLR and HR networks, and $\boldsymbol{\theta}_{\text{eLR}^*}$, $\boldsymbol{\theta}_{\text{HR}^*}$, the weights that belong to only the eLR or the HR network, respectively. With these weights, we

update both networks as follows:

$$\boldsymbol{\theta}^{m}_{\text{eLR}} = \begin{bmatrix} \boldsymbol{\theta}^{m-1}_{\text{eLR}^*} + \mu \dfrac{\partial \boldsymbol{L}^{m-1}_{\text{eLR}}}{\partial \boldsymbol{\theta}^{m-1}_{\text{eLR}^*}} \\ \\ \boldsymbol{\theta}^{2m-2}_{\text{shared}} + \mu \dfrac{\partial \boldsymbol{L}^{m-1}_{\text{eLR}}}{\partial \boldsymbol{\theta}^{2m-2}_{\text{shared}}} \end{bmatrix} \tag{2.7}$$

$$\boldsymbol{\theta}^{m}_{\text{HR}} = \begin{bmatrix} \boldsymbol{\theta}^{m-1}_{\text{HR}^*} + \mu \dfrac{\partial \boldsymbol{L}^{m-1}_{\text{HR}}}{\partial \boldsymbol{\theta}^{m-1}_{\text{HR}^*}} \\ \\ \boldsymbol{\theta}^{2m-1}_{\text{shared}} + \mu \dfrac{\partial \boldsymbol{L}^{m-1}_{\text{HR}}}{\partial \boldsymbol{\theta}^{2m-1}_{\text{shared}}} \end{bmatrix} \tag{2.8}$$

where μ is the learning rate, m is the training iteration, and $\boldsymbol{L}_{\text{eLR}}$ and $\boldsymbol{L}_{\text{HR}}$ are, respectively, the loss functions of each network. In each training iteration, the shared weights are updated in *both* the eLR and the HR ConvNet, i.e., they are updated twice in each iteration. Specifically, in each training iteration m, we have

$$\boldsymbol{\theta}^{2m-1}_{\text{shared}} = \boldsymbol{\theta}^{2m-2}_{\text{shared}} + \mu \frac{\partial \boldsymbol{L}^{m-1}_{\text{eLR}}}{\partial \boldsymbol{\theta}^{2m-2}_{\text{shared}}} \tag{2.9}$$

$$\boldsymbol{\theta}^{2m-2}_{\text{shared}} = \boldsymbol{\theta}^{2m-3}_{\text{shared}} + \mu \frac{\partial \boldsymbol{L}^{m-2}_{\text{HR}}}{\partial \boldsymbol{\theta}^{2m-3}_{\text{shared}}} \tag{2.10}$$

However, the resolution-specific unshared weights are only updated once: either in the eLR ConvNet training update or in the HR ConvNet training update. Therefore, the shared weights are updated twice as often as the unshared weights.

Our approach has been inspired by Partially-Coupled Super-Resolution Networks (PCSRN) (Wang et al., 2016) where it was shown that leveraging HR images in training of such networks can help discover discriminative features in eLR images that would otherwise have been overlooked during image classification. PCSRN is a super-resolution network that pre-trains network weights using filter sharing. This pre-training is intended to minimize the MSE between the output image and the target HR image via super-resolution. In our approach, we differ from this work

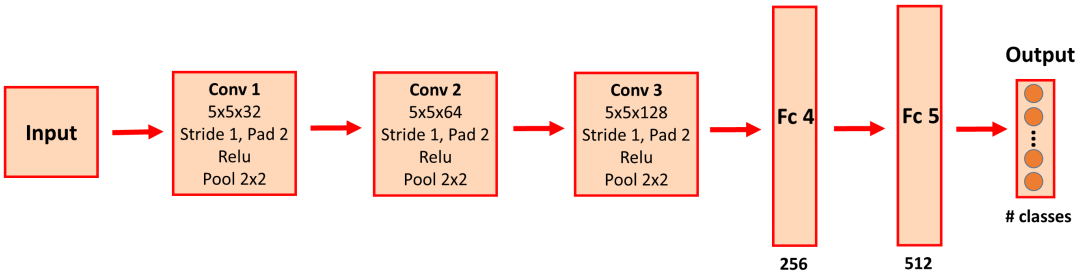


Figure 2·11: Basic ConvNet used in our model. The spatial and temporal streams have the same architecture except that the input dimension is larger in the temporal stream (the input to the temporal stream is a stacked optical flow). In our two-stream fusion ConvNets, two base ConvNets are fused after either the "Conv3" or "Fc4" layer.

by leveraging HR information throughout the *entire* training process. Our method does not need to pre-train the network; instead, we learn the entire network from scratch, and minimize the classification loss function directly while still incorporating HR information as shown in the equations above. Overall, we extend this model in two aspects: first, we consider shared filters in the fully-connected layers (previously only convolutional layers were considered for filter sharing). Second, we adapt this method for action recognition in fused two-stream ConvNets. We also report results for semi-coupled two-stream ConvNets across various fusion operators.

Implementation details

Two-stream fusion network. Conventional standard-resolution ConvNet architectures can be ill-suited for eLR images due to large receptive fields that can sometimes be larger than the eLR image itself. To address this issue, we have designed an eLR ConvNet consisting of 3 convolutional layers, and 2 fully-connected layers as shown in Fig. 2·11. We have tried many variations, but found that larger models do not improve performance. Also, the model in (Ryoo et al., 2016) is larger than ours, but achieves lower CCR. We base both our spatial and temporal streams on this ConvNet,

and explore fusion operations after either the "Conv3" or "Fc4" layer. We train all networks from scratch using the Matconvnet toolbox (Vedaldi and Lenc, 2015). The weights are initialized to be zero-mean Gaussian with a small standard deviation of 10^{-3}. The learning rate starts from 0.05 and is reduced by a factor of 10 after every 10 epochs. Weight decay and momentum are set to 0.0005 and 0.9 respectively. We use a batch size of 256 and perform batch normalization after each convolutional layer. At every iteration, we perform data augmentation by allowing a 0.5 probability that a given image in a batch is reflected across the vertical axis. Each RGB frame in the spatial stream corresponds to 11 stacked frames of optical flow. This stacked optical-flow block contains the current, the 5 preceding, and 5 succeeding optical flow frames. To regularize these networks during training, we set the dropout ratio of both fully-connected layers to 0.85.

Semi-coupled ConvNets. In Section 2.3.2, we have discussed how to incorporate filter sharing in a semi-coupled network. However, it is not obvious how many filters should be shared in each layer. To discover the proper proportion of filters we should share, we conducted a coarse grid search for the coupling ratio c_n from 0 to 1 with a step size of 0.25. The coupling ratio is defined as:

$$c_n = \frac{k^n}{D^n}, \quad n = 1, \cdots, 5 \tag{2.11}$$

where the two ConvNets are uncoupled when $c_n = 0$ ($n = 1, \cdots, 5$). For the step sizes that we consider, a brute force approach would be unfeasible, as the total number of two-stream networks to train would be $5^5 = 3125$. Therefore, we follow the methodology used in (Wang et al., 2016) to monotonically increase the coupling ratios with the increasing layer depth. This is inspired by the notion that the disparity between eLR and HR domains is reduced as the layer gets deeper (Glorot et al., 2011; Wang et al., 2014). For all our experiments, we used the following coupling ratios: $c_1 = 0$,

$c_2 = 0.25$, $c_3 = 0.5$, $c_4 = 0.75$, and $c_5 = 1$. We determined these ratios by performing a coarse grid search on a cross-validated subset of the IXMAS dataset (subjects 2, 4, 6).

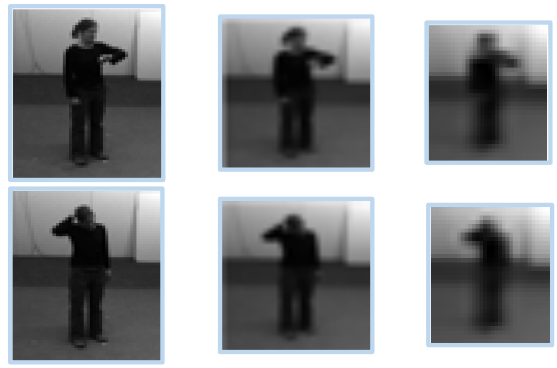
Normalization. In our experiments, we apply a variant of mean-variance normalization to each video $\mathbf{v}_{i,j}[t], i, j = 1, \cdots, R, t = 1, \cdots, T$, where R is the spatial size, T is the temporal length, and $\mathbf{v}_{i,j}[t]$ denotes the grayscale value of pixel (i, j) at time t, as follows:

$$\hat{\mathbf{v}}_{i,j}[t] = \frac{\mathbf{v}_{i,j}[t] - \mu_{i,j}}{\sigma}. \tag{2.12}$$

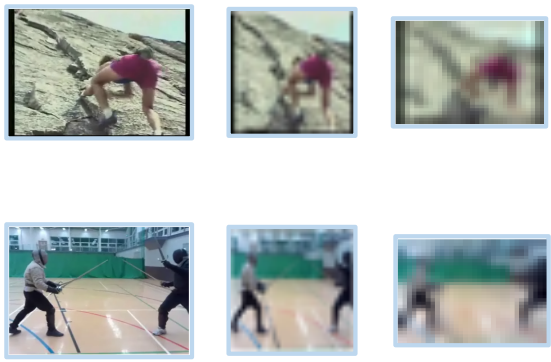
Above, $\mu_{i,j}$ denotes the empirical mean pixel value across time for the spatial location (i, j), and σ denotes the empirical standard deviation across all pixels in one video. The subtraction of the mean emphasizes a subject's local dynamics, while the division by the empirical standard deviation compensates for the variability in subject's clothing.

Optical flow. As discussed earlier, we use a stacked block of optical flow frames as input to the temporal stream. We follow (Wu et al., 2016) and use colored optical-flow frames. First, we compute optical flow between two consecutive normalized RGB frames (Liu et al., 2009). The computed optical flow vectors are then mapped into polar coordinates and converted to hue and saturation based on the magnitude and orientation, respectively. The brightness is set to one. As a reminder, the eLR optical flow is computed from the interpolated 32×32 pixel eLR frames. Further, we subtract the mean of the stacked optical flows to compensate for global motion as suggested in (Simonyan and Zisserman, 2014a).

Source code: More implementation details as well as source code are available on project web site (Chen, 2017).



(a) IXMAS



(b) HMDB

Figure 2·12: Sample frames from IXMAS and HMDB datasets. (a) From left to right are original frames, and resized 32×32 and 16×12 frames from the IXMAS dataset. (b) From left to right are original frames, and resized 32×32 and 12×16 frames from the HMDB dataset. Note that we resize the IXMAS dataset to 16×12 and the HMDB dataset to 12×16 in order to preserve the original aspect ratio. We use 32×32 resized videos as HR data. The 16×12 (12×16) eLR videos are upscaled using bi-cubic interpolation to 32×32 interpolated-eLR video which is used in our proposed semi-coupled fused two-stream ConvNet architecture.

2.3.3 Experimental Results

Datasets

In order to confirm the effectiveness of our proposed method, we conducted experiments on two publicly-available video datasets. First, we used the ROI sequences from the multi-view IXMAS action dataset, where each subject occupies most of the

field of view (Weinland et al., 2010). This dataset includes 5 camera views, 12 daily-life motions each performed 3 times by 10 actors in an indoor scenario. Overall, it contains 1,800 videos. To generate the eLR videos (thus eLR-IXMAS), we decimated the original frames to 16×12 pixels and then upscaled them back to 32×32 pixels using bi-cubic interpolation (Fig. 2·12). The upscaling operation does not introduce new information (fundamentally, we are still working with 16×12 pixels) but ensures that eLR frames have enough spatial support for hierarchical convolutions to facilitate filter sharing. On the other hand, we generate the HR data by decimating the original frames straight to 32×32 pixels. We perform *leave-person-out* cross validation in each case and compute correct classification rate (CCR) and standard deviation (StDev) to measure performance.

We also test our algorithm on the popular HMDB dataset (Kuehne et al., 2011) used for video activity recognition. The HMDB dataset consists of 6,849 videos divided into 51 action categories, each containing a minimum of 101 videos. In comparison to IXMAS, which was collected in a controlled environment, the HMDB dataset includes clips from movies and YouTube videos, which are not limited in terms of illumination and camera position variations. Therefore, HMDB is a far more challenging dataset, especially when we decimate to eLR, which we herein refer to as eLR-HMDB. In our experiments, we used the three training-testing splits provided with this dataset. Note that since there are 51 classes in the HMDB dataset, the CCR based on a purely random guess is 1.96%.

Results for eLR-IXMAS

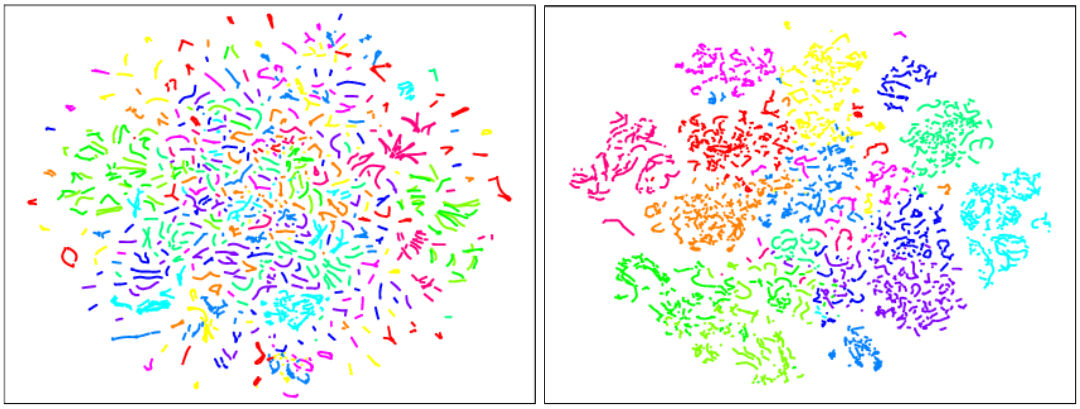
We first conduct a detailed evaluation of the proposed paradigms on the eLR-IXMAS action dataset. For a fair comparison, we follow the image resolution, pre-processing and cross-validation as described in (Dai et al., 2015). We first resize all video clips to a fixed temporal length $T = 100$ using cubic-spline interpolation.

Table 2.5 summarizes the action recognition accuracy on the eLR-IXMAS dataset. We report the CCR for separate spatial and temporal ConvNets, as well as for various locations and operators of fusion, with and without eLR-HR coupling. We also report the baseline result from (Dai et al., 2015) which employs a nearest-neighbor classifier. We first observe that dedicated spatial or temporal ConvNet outper-

Table 2.5: Action recognition performance of different ConvNet architectures against baseline on the eLR-IXMAS dataset. "Spatial & Temp avg" has been performed by averaging the temporal and spatial stream predictions. The best performing method is highlighted in bold.

Method	Fusion Layer	eLR-HR coupling?	CCR	StDev
Baseline (Dai (Dai et al., 2015))	-	-	80.0%	6.9%
Spatial Network	-	No	88.6%	6.2%
Temporal Network	-	No	91.6%	4.9%
Spatial&Temp avg	Softmax	No	92.0%	6.0%
Concat Fusion	Fc 4	No	92.2%	5.2%
	Fc 4	Yes	92.5%	5.5%
	Conv 3	No	92.2%	5.2%
	Conv 3	Yes	93.3%	5.6%
Conv Fusion	Fc 4	No	92.0%	5.8%
	Fc 4	Yes	93.1%	5.2%
	Conv 3	No	93.3%	4.0%
	Conv 3	**Yes**	**93.7%**	**4.5%**
Sum Fusion	Fc 4	No	92.2%	5.5%
	Fc 4	Yes	92.8%	7.1%
	Conv 3	No	93.0%	4.7%
	Conv 3	Yes	93.6%	4.0%

forms the benchmark result from (Dai et al., 2015) by 8.6% and 11.6% respectively, which validates the discriminative power of a ConvNet. If we equally weigh these two streams ("Spatial&Temp avg"), we can see that the fusion only marginally improves recognition performance. Secondly, we can see that fusing after the "Conv3" layer



(a) eLR-IXMAS pixel-wise time series features (Dai et al., 2015)

(b) eLR-IXMAS ConvNet features after 'Fc 5' layer

Figure 2·13: 2-D t-SNE embeddings (Maaten and Hinton, 2008) of features for the eLR-IXMAS dataset. A single marker represents a single video clip and is color-coded by action type. (a) Embeddings of pixel-wise time series features (Dai et al., 2015). (b) Embeddings of the last fully-connected layer's output from our best performing ConvNet.

provides a consistently better performance than fusing after the "Fc4" layer. In our preliminary experiments, we also found that fusing after the "Conv3" layer was consistently better than fusing after the "Conv2" layer, which suggests that there is an ideal depth (which is not too shallow or too deep in the network) for fusion. Regarding which fusion operator to use, we note that all 3 operators we consider provide comparable performance after the "Fc4" layer. However, if we fuse features after the "Conv3" layer, convolutional fusion performs best.

As for the effectiveness of semi-coupling in the networks using HR information, we can see that eLR-HR coupling consistently improves recognition performance. Our best result on IXMAS is 93.7%, where we find that without coupling, our performance drops by 0.4%. This result is very close to that achieved by using *only* HR data in both training and testing, which is 94.4% CCR. Effectively, this should be an upper-bound, in terms of performance, when using eLR-HR coupling in training but testing *only* on eLR data. That the performance gap between HR and eLR is small may

Table 2.6: Comparison of the number of parameters of our best-performing action recognition ConvNet as compared to those of the standard-resolution image classification ConvNets.

Network	Task	Input resolution	# param
Ours	Action Rec.	$32 \times 32 \times 3$	0.84M
AlexNet	Image Class.	$224 \times 224 \times 3$	60M
VGG-16	Image Class.	$224 \times 224 \times 3$	138M
VGG-19	Image Class.	$224 \times 224 \times 3$	144M

be explained by the distinctiveness of actions and the controlled indoor environment (static cameras, constant illumination, etc.) in the IXMAS dataset. Additionally, the fine details (e.g., hair, facial features), that are only visible in HR, are not critical for action recognition.

We also conduct experiment to evaluate the privacy-preserving performance at eLR. We train a person identifier using the best-performing ConvNet architecture (showed in bold in Table. 2.5), where two of the three repetitions of each action for each person are used for training and the remaining videos are used for evaluation. The resulting identification CCR is 23.10% which is only about two times random guess performance (10%), indicating that reducing the spatial resolution to extremely low level can protect user's privacy to a large extent.

In order to qualitatively evaluate our proposed model, we visualize various feature embeddings (for action recognition) of the eLR-IXMAS dataset. We extract output features of the "Fc5" layer from the best-performing ConvNet (shown in bold), and project them to 2-dimensional space using t-SNE (Maaten and Hinton, 2008). For comparison, we also apply t-SNE to the pixel-wise time series features proposed in our benchmark (Dai et al., 2015). As seen in Fig. 2·13, the feature embedding from our ConvNet model is visually more separable than that of our baseline. This is not surprising, as we are able to consistently outperform the baseline on the eLR-IXMAS dataset.

Regarding the number of parameters, our ConvNet designed for eLR videos needs about 100 times less parameters than state-of-the art ConvNets designed for image classification like AlexNet (Krizhevsky et al., 2012), VGG-16, and VGG-19 (Simonyan and Zisserman, 2014b) (Table 2.6). In consequence, this significantly reduces the computation cost of training and testing compared to these standard-resolution networks.

Results for eLR-HMDB

In this section, we report the results of our methods on eLR-HMDB. Note that, for this dataset, we only report results for fusion after the "Conv3" layer, based on our observations from eLR-IXMAS. We follow the same pre-processing procedure as used for eLR-IXMAS except that we do not resize the video clips temporally for the purpose of having a fair comparison with the results reported in (Ryoo et al., 2016). Our reported CCR is an average across the three training-testing splits provided with this dataset.

First, we measure the performance of a dedicated spatial-stream ConvNet and a dedicated temporal-stream ConvNet. As shown in Table 2.7, using only the appearance information (spatial stream) provides 19.1% accuracy. If optical flow is used alone (temporal stream), performance drops to 18.3%. This is likely because videos in the HMDB dataset are unconstrained; camera movement is not guaranteed to be well-behaved, thus resulting in drastically different optical-flow quality across videos. Such variations are likely to be amplified in eLR videos. We then evaluate the same three fusion operators after the "Conv3" layer. Compared to the average of predictions from a dedicated spatial network and a dedicated temporal network, fusing the temporal and spatial streams improves the recognition performance by 0.8%, 0.9% and 1.8% with concatenation, convolution, and sum fusion, respectively. The performance improvement from fusion is not significant. This, however, is consistent with the observations in (Feichtenhofer et al., 2016).

Table 2.7: Action recognition performance of different ConvNet architectures and current state-of-the-art method on the eLR-HMDB dataset. The two-stream networks are all fused after the "Conv3" layer. The best method is highlighted in bold.

Method	eLR-HR coupling?	CCR
Spatial Network	No	19.1%
Temporal Network	No	18.3%
Spatial & Temp avg	No	19.6%
Concat Fusion	No	20.4%
	Yes	27.1%
Conv Fusion	No	20.5%
	Yes	27.3%
Sum Fusion	No	21.4%
	Yes	**29.2**%
ConvNet feat + SVM(Ryoo et al., 2016)	-	18.9%
ConvNet feat + ISR + SVM(Ryoo et al., 2016)	-	20.8%
ConvNet + hand-crafted feat + ISR + SVM(Ryoo et al., 2016)	-	28.7%

When fusion is combined with eLR-HR coupling, the gains are significant. We achieve large performance gains from 20.4% to 27.1% using concatenation fusion, 20.5% to 27.3% using convolutional fusion, and 21.4% to 29.2% using sum fusion. Such notable improvements validate the discriminative capabilities of semi-coupled fused two-stream ConvNets. Compared to the state-of-the-art results reported in (Ryoo et al., 2016), our approach is able to outperform their ConvNet feature-only method by 8.4%. We also exceed the performance of their best method, that uses an augmented hand-crafted feature vector, by 0.5%.

Please note that HMDB dataset does not come with identity labels. Thus we cannot evaluate the privacy protection performance of our methods on this dataset.

2.4 Discussions

In this chapter, we investigated three visual analysis tasks at extremely low resolutions. For head pose estimation, we showed that a monocular camera with 10×10 resolution can provide estimates that have about twice error of state-of-the-art methods at full resolution. Even at 5×5 resolution, reasonable results are still attainable. As for indoor occupant localization, we have demonstrated that on average using six single-pixel color sensors is capable of localizing a single individual to within 35 cm of the true position on a floor area of 2.37m$\times$2.72m. This could be useful for applications such as visible light communication. Regarding action recognition, we proposed multiple eLR ConvNet architectures, each leveraging and fusing spatial and temporal information. Further, in order to leverage HR videos in training we incorporated eLR-HR coupling to learn an intelligent mapping between the eLR and HR feature spaces. We achieved state-of-the-art performance on two public datasets at low resolutions.

Although we achieved promising results, we also observed noticeable discrepancy between task performance at low resolutions and high resolutions. It seems inevitable that task performance will drop when data resolution decreases.

Chapter 3

Methodologies Based on Invariant Representation Learning

In the previous chapter, we showed that concerns about privacy can be partially addressed by significantly reducing the camera resolution. This, however, degrades recognition accuracy. Another extreme approach is to withhold releasing the imagery data altogether and only release estimates of the utility information.

However, the smart room scenario calls for scalability to various visual analysis tasks. However, this approach demands installation of a customized multi-task recognition algorithm on each local camera, which precludes "future-proofing" because the types of specified utilities may change over time. As a result, the local cameras will require to be updated every time a new task utility needs to be accommodated. Additionally, this approach provides no visual utility.

In order to address the aforementioned concerns, we propose a third radically different approach: seamlessly *replace* the private information in an image without significantly degrading its visual quality or the ability to accurately infer the utility information for the task of interest. Meanwhile, a low-dimensional identity-invariant utility-preserving image representation is created. Both the generated image and representation can be safely sent to cloud for processing or released to public (if necessary). Specifically, this chapter introduces two novel methodologies that leverage the Variational Auto-Encoder (VAE) (Kingma and Welling, 2013) and the Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to learn an image representa-

tion which is invariant to a specified factor of variation (e.g., user-identity) while enabling the synthesis of a utility-equivalent, realistic version of the source image with a different specified factor value.

3.1 Background Material

Before introducing the proposed methods in detail, we first provide the following necessary background material.

3.1.1 Invariant Representation Learning

Invariant representations/features, by definition, have reduced sensitivity in the direction of invariance. This is the goal of building features that are insensitive to variation in the data that are uninformative to the task at hand (Bengio et al., 2013). Invariant representation learning has been extensively studied in various contexts. For instance, transformation-invariant feature learning has deep roots in computer vision; features are often designed for a specific case, e.g., rotation or scale invariance. Early examples include hand-crafted features such as HOG (Dalal and Triggs, 2005) and SIFT (Lowe, 1999). More recently, deep Convolutional Neural Networks (CNNs) appear to be exceptionally effective in learning transformation-invariant representations (Cheng et al., 2016; Cohen and Welling, 2016; Soatto and Chiuso, 2014)

An emerging research area aims to learn domain-invariant representations which compensate domain mismatch (Hoffman et al., 2013; Ganin et al., 2016; Li et al., 2018b; Johansson et al., 2019). Another line of research aim to build fair, bias-free classifiers that also attempt to learn representations invariant to "nuisance variables", which could potentially induce bias or unfairness (Li et al., 2014; Zafar et al., 2015; Edwards and Storkey, 2015; Xie et al., 2017). (Li et al., 2014) proposed to obtain fairness by imposing l_1 regularization between representation distributions for data with different nuisance factors of variation. The Variational Fair Auto-Encoder (Zafar

et al., 2015) tackles the same task using a VAE with maximum mean discrepancy regularization. Particularly relevant to our work are the methods proposed in (Edwards and Storkey, 2015; Xie et al., 2017), which also incorporate adversarial training in a VAE framework. Our models differ in that we apply adversarial training in the image space instead of the latent space. This creates better quality images. In addition, we propose a cyclic training process in the second model to further improve the quality of invariance.

3.1.2 Disentangled Representation Learning

Disentangled representation learning is closely related to invariant representation learning, as both attempt to separate factors of variation in the data. The major difference between them is that invariant representations eliminate unwanted factors in order to reduce sensitivity in the direction of invariance, while disentangled representations try to preserve as much information about the data as possible (Bengio et al., 2013). A number of models have been proposed in the literature to learn disentangled representation for imagery data. Early work (Tenenbaum and Freeman, 2000) proposed a bilinear model to separate content and style for face and text images. Method proposed in (Ghahramani, 1995) utilized E-M algorithm to discover the independent factors of variation of the underlying data distribution. Manifold learning was also leveraged in (Elgammal and Lee, 2004) to explicitly separate body configuration and appearance. In (Desjardins et al., 2012), a method based on Restricted Boltzmann Machines were developed to separate factors of variations in images. An autoencoder augmented with simple regularization during training was proposed and demonstrated to learn latent factors of variation (Cheung et al., 2014). In (Kingma et al., 2014; Higgins et al., 2017; Burgess et al., 2018), Variational Auto-Encoder (VAE) based methods were proposed to learn an interpretable factorised representation in the latent space. A recent work (Harsh Jha et al., 2018) proposed to use

cycle-consistency in a VAE framework to disentangle the latent space into two complementary subspaces in a semi-supervised setting.

Adversarial training was also employed for disentangled representation learning. Adversarial auto-encoder (Makhzani et al., 2015) uses a semi-supervised approach to disentangle style and content of images. It learns to disentangle label information from the latent space by providing additional labels as input to the decoder. Models in (Edwards and Storkey, 2015; Hadad et al., 2018; Lample et al., 2017) directly apply adversarial training to latent space within a VAE in order to learn invariance to attributes. However, methods with sole pixel-wise reconstruction objective in the image space tend to produce blurry images. Recent works (Mathieu et al., 2016; Szabó et al., 2017) both combine auto-encoders with adversarial training to disentangle specified and unspecified information into two subspaces. Indeed, the resulting unspecified representation is equivalent to an invariant representation that is disentangled from the specified factor. However, their methods lack necessary constraints over the latent space. Thus, the disentanglement quality falls short in comparison to other methods (Harsh Jha et al., 2018).

In this thesis, we also compare our methods with disentangled representation learning methods that learn to produce, for a given input image, two latent vectors. One of the latent vectors captures information related to the unspecified factors of variation and is, in an ideal scenario, devoid of any information related to the specified factor of variation (e.g., identity information). This latent vector is the counterpart of the latent invariant representation in our methods.

3.1.3 Conditional Image Generation

Recent advances in image modeling with neural networks have made it feasible to create realistic-looking images with desired attributes, conditioned on different types of source information such as class label, text and image. PixelCNN (Van den Oord

et al., 2016) was proposed as a image density model that can be conditioned on any vector including class labels or latent embeddings created by other networks. (Sohn et al., 2015) developed a conditional variational auto-encoder (CVAE) which is a conditional directed graphical model whose input observations (e.g., face images) modulate the prior on Gaussian latent variables that generate the outputs. The popular GAN architecture is able to produce convincing image samples (Goodfellow et al., 2014). However, it lacks the capability to control its outputs, since the outputs only depend on input random noise. Auxiliary-Classifier GAN (AC-GAN) (Odena et al., 2017) is a variant of GAN architecture. It adds more structure to the GAN latent space along with a specialized cost function, which enables AC-GAN to generate images from a particular class. Concurrently, Pix2Pix (Isola et al., 2017) and CycleGAN (Zhu et al., 2017) show success in paired and unpaired image-to-image translation using adversarial networks, respectively. The task of text to image translation also has been successfully tackled in (Zhang et al., 2017) using stacked GANs. The idea of combining VAE and GAN was first proposed in (Larsen et al., 2015) for better-quality image generation. Later, conditional VAE-GANs (Bao et al., 2017; Di and Patel, 2017; Shang and Sohn, 2019) were proposed for synthesizing images in fine-grained categories. While our models are related to the aforementioned works in terms of using VAE and GAN, our goals are very different. We explicitly optimize our models to create invariant image representations. Once trained, our models becomes conditional image generators.

3.1.4 Variational Autoencoder Network

A VAE network consists of two neural networks: an encoder network (Enc) and a decoder network (Dec). The encoder is a randomized mapping of a data sample $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$ while the decoder is a randomized mapping $\boldsymbol{z}$ from a latent

representation back to data space:

$$\boldsymbol{z} \sim Enc(\boldsymbol{x}) = q(\boldsymbol{z}|\boldsymbol{x}) \tag{3.1}$$

$$\widehat{\boldsymbol{x}} \sim Dec(\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z}) \tag{3.2}$$

In practice, these randomized mappings are implemented *via* deterministic maps (given by the neural networks) with additional inputs which provide the source of randomness. For example, it is common to set $\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{w}$ where the vector $\boldsymbol{\mu}_{\boldsymbol{x}}$ and the square matrix $\boldsymbol{A}_{\boldsymbol{x}}$ are the outputs of a neural network with input $\boldsymbol{x}$, and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, a standard multivariate Gaussian, is the source of randomness. Then, $q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{A}_{\boldsymbol{x}}\boldsymbol{A}_{\boldsymbol{x}}^T)$. VAE networks are trained by *minimizing* a cost function which is additive over all training data samples. The cost function for a single data sample $\boldsymbol{x}$ is given by

$$\mathcal{L}_{\boldsymbol{x}}^{VAE} = -\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] + KL\big(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\big) \tag{3.3}$$

where KL is the Kullback-Leibler divergence and $p(\boldsymbol{z})$, the marginal distribution of the latent representation, is typically taken to be $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The first term encourages the decoder to assign higher probability to the observed data samples $\boldsymbol{x}$. In practice, the expectation in the first term is replaced by an empirical average across a small batch of independent and identically distributed $\boldsymbol{z}$ for a given $\boldsymbol{x}$. The KL term encourages the encoder $q(\boldsymbol{z}|\boldsymbol{x})$ to be close to a target $p(\boldsymbol{z})$ which has sufficient spread (diversity) in the latent space. The KL term has a closed analytic form since both of its arguments are Gaussian (Kingma and Welling, 2013). The total cost across all data samples is typically minimized *via* mini-batch gradient descent.

3.1.5 Generative Adversarial Network

A standard GAN consists of a generator neural network G and a discriminator neural network D that are trained by making them compete in a two-player min-max game. The discriminator network D adjusts its weights so as to reliably distinguish real data samples $\boldsymbol{x} \sim p_d(\boldsymbol{x})$ from fake data samples $G(\boldsymbol{z})$ generated by passing $\boldsymbol{z}$, randomly sampled from some distribution $p_z(\boldsymbol{z})$, through the generator network G. The generator network G adjusts its weights to fool D. The discriminator D assigns probability $D(\boldsymbol{x}) \in [0, 1]$ to the event that $\boldsymbol{x}$ is a "real" training data sample and the probability $1 - D(\boldsymbol{x})$ to the event that $\boldsymbol{x}$ is a "fake" sample synthesized by the generator. The two networks are trained iteratively using a loss function given by

$$\mathcal{L}_{GAN}(G, D) = E_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log D(\boldsymbol{x})] \; + \; E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))] \tag{3.4}$$

with G aiming to minimize $\mathcal{L}_{GAN}(G, D)$ and D aiming to maximize it. In practice, the expectations are replaced by empirical averages over a mini-batch of samples and the loss function is alternately minimized and maximized from one mini-batch to the next as in mini-batch gradient descent.

3.2 Model I: Privacy-Preserving Representation-Learning Variational-GAN (PPRL-VGAN)

3.2.1 Introduction

In this section, we introduce a Privacy-Preserving Representation-Learning Variational Generative Adversarial Network (PPRL-VGAN) for learning a *face image* representation that is explicitly invariant to the *identity information*. At the same time, this representation is discriminative from the standpoint of one specific utility information (e.g., facial expression) and generative as it allows utility-equivalent face image synthesis. We leverage variational generative-adversarial networks (VGANs) to

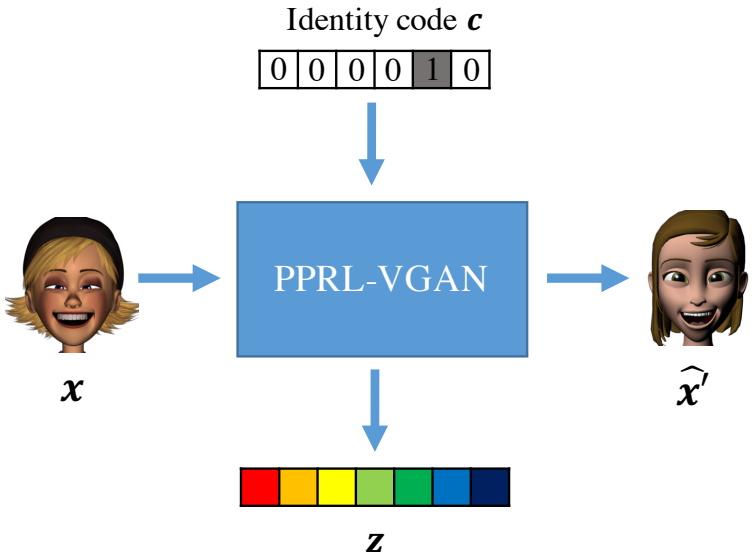


Figure 3·1: Basic functionality of PPRL-VGAN: given an input face image $\boldsymbol{x}$, the network produces an identity-invariant representation $\boldsymbol{z}$, and a utility-preserving face image with another identity specified by identity code $\boldsymbol{c}$.

learn an identity-invariant representation of an image while enabling the synthesis of a utility-equivalent, realistic version of this image with a different identity (Fig. 3·1). Beyond its application to privacy-preserving visual analytics, our approach could also be used to generate realistic avatars for animation and gaming.

3.2.2 Formulation of PPRL-VGAN

Given a face image $\boldsymbol{x}$ with an identity label $y_{id} = 1, ..., N_{id}$ and a label y_u for a specified utility attribute, where N_{id} is the number of distinct subjects, the proposed model has two objectives: 1) to learn an identity-invariant face image representation $\boldsymbol{z}$ that retains the specified utility information, and 2) to synthesize a realistic face image $\widehat{\boldsymbol{x}}'$ with the same specified utility attribute as in $\boldsymbol{x}$ and target identity specified by a one-hot encoded identity code $\boldsymbol{c} \in \{0, 1\}^{N_{id}}$.

Discriminator: Different from the discriminator network in a conventional GAN, the discriminator $D = (D^1, D^2, D^3)$ in PPRL-VGAN is a multi-task estimator consisting

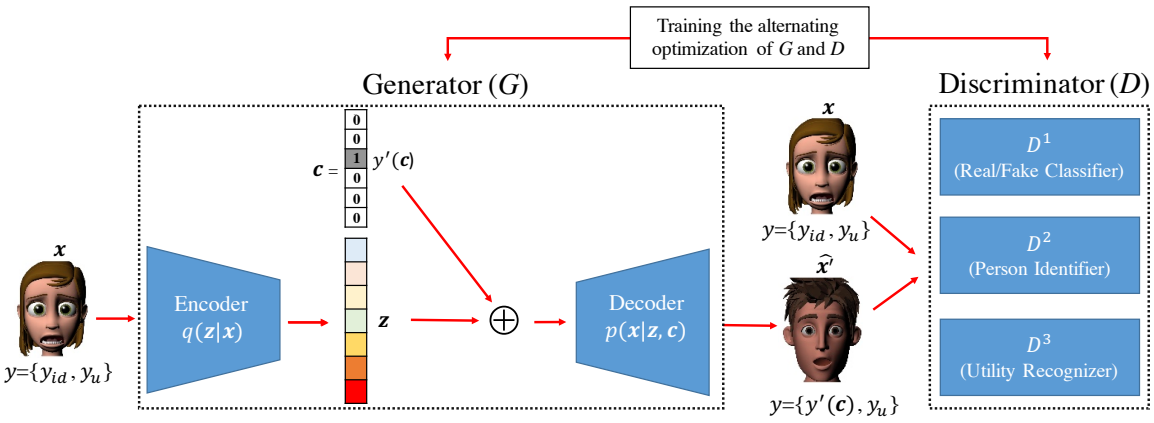


Figure 3·2: Schematic diagram of the proposed PPRL-VGAN ($\oplus$ represents concatenation).Training alternates between optimizing the weights of D keeping G fixed and vice-versa. Both original and synthesized images with their labels are used during training.

of three separate neural networks (Fig. 3·2): 1) the D^1 network classifies an input face image $\boldsymbol{x}$ as real or synthetic, 2) the D^2 network estimates the identity of the person in the input face image, and 3) the D^3 network recognizes the specified utility attribute. The weights of the networks in D are trained to classify real face image inputs $\boldsymbol{x}$ as real and accurately recognize the person's identity and the utility attribute of interest. They are also trained to classify synthetic image inputs $\widehat{\boldsymbol{x}}'$ as fake.

Specifically, if the specified utility attribute y_u is categorical, then we adjust the network weights to *maximize* the following *discriminator cost function*:

$$\begin{aligned}\mathcal{L}_D(G,D) =& \lambda_1^D \left\{ E_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log D^1(\boldsymbol{x})] - E_{\boldsymbol{x}\sim p_d(\boldsymbol{x}),\boldsymbol{c}\sim p(\boldsymbol{c})}[\log(D^1(G(\boldsymbol{x},\boldsymbol{c})))] \right\} + \\ & E_{(\boldsymbol{x},\boldsymbol{y})\sim p_d(\boldsymbol{x},\boldsymbol{y})}[\lambda_2^D \log D^2_{y_{id}}(\boldsymbol{x}) + \lambda_3^D \log D^3_{y_u}(\boldsymbol{x})] \end{aligned} \tag{3.5}$$

where D_i^2, D_i^3 are the predicted probabilities of the ith class for identity and the specified utility attribute, respectively. The tuning parameters λ_1^D, λ_2^D and λ_3^D control the relative importance between image quality, identity recognition, and expression recognition objectives. Whereas if y_u is continuous, we instead adjust the network

weights to *maximize* the following *discriminator cost function*:

$$\mathcal{L}_D(G,D) = \lambda_1^D \left\{ E_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\log D^1(\boldsymbol{x})] - E_{\boldsymbol{x}\sim p_d(\boldsymbol{x}),\boldsymbol{c}\sim p(\boldsymbol{c})}[\log(D^1(G(\boldsymbol{x},\boldsymbol{c})))] \right\} + E_{(\boldsymbol{x},\boldsymbol{y})\sim p_d(\boldsymbol{x},\boldsymbol{y})}[\lambda_2^D \log D^2_{y_{id}}(\boldsymbol{x}) - \lambda_3^D ||D^3(\boldsymbol{x}) - y_u||^p] \tag{3.6}$$

where D^3 is a regressor that is trained to produce an approximation of y_u.

Generator: In contrast to the generator in a conventional GAN which directly maps a "noise" vector to a synthesized image, the generator G in a PPRL-VGAN maps a real input image $\boldsymbol{x}$ with identity y_{id} and the specified utility attribute y_u to a synthesized output image $\widehat{\boldsymbol{x}}' = G(\boldsymbol{x},\boldsymbol{c})$ with a target identity $y'(\boldsymbol{c})$ and the same utility attribute y_u. This is accomplished *via* a VAE-like encoder-decoder structure. Specifically, the encoder aims to learn an image representation $\boldsymbol{z}$ from $\boldsymbol{x}$ *via* a randomized mapping $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x})$ parameterized by the weights of the encoder neural network. Similarly to a VAE, the cost function for training the generator includes KL divergence between a prior distribution on the latent space $p(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ and the conditional distribution $q(\boldsymbol{z}|\boldsymbol{x})$. The training attempts to minimize this KL term. The generator cost function also includes a term that encourages the decoder to learn to synthesize a face image $\widehat{\boldsymbol{x}}' \sim p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c})$ that can fool D into classifying it as a real face image having the same specified utility attribute y_u as the input image $\boldsymbol{x}$, but with a target identity $y'(\boldsymbol{c})$ determined by $\boldsymbol{c}$. Specifically, if y_u is categorical the generator network weights are adjusted during training to *minimize* the following *generator cost function*:

$$\mathcal{L}_G(G,D) = - E_{(\boldsymbol{x},\boldsymbol{y})\sim p_d(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{c}\sim p(\boldsymbol{c})}[\lambda_1^G \log(D^1(G(\boldsymbol{x},\boldsymbol{c}))) + \lambda_2^G \log(D^2_{y'(\boldsymbol{c})}(G(\boldsymbol{x},\boldsymbol{c}))) + \lambda_3^G \log(D^3_{y_u}(G(\boldsymbol{x},\boldsymbol{c})))] + \lambda_4^G KL\big(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\big) \tag{3.7}$$

where λ_1^G, λ_2^G, λ_3^G and λ_4^G are tuning parameters of the loss functions for D^1, D^2, D^3 and KL divergence respectively. A key difference compared to the cost in Eq. 3.3

is that the first term (reconstruction error) in Eq. 3.3 has been replaced with a perceptual loss term for the discriminator D^1 in Eq. 3.7.

If y_u is continuous, the generator network weights are adjusted to *minimize* the following *generator cost function*:

$$\begin{aligned}\mathcal{L}_G(G,D) = &- E_{(\boldsymbol{x},\boldsymbol{y})\sim p_d(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{c}\sim p(\boldsymbol{c})}[\lambda_1^G \log(D^1(G(\boldsymbol{x},\boldsymbol{c}))) + \lambda_2^G \log(D^2_{y'(\boldsymbol{c})}(G(\boldsymbol{x},\boldsymbol{c})))\\ &- \lambda_3^G||D^3(G(\mathbf{x},\mathbf{c})) - y_u||^p] + \lambda_4^G KL\big(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\big) \qquad (3.8)\end{aligned}$$

the training alternates between maximizing Eq. 3.5 (or Eq. 3.6) with respect to the weights of the networks in D and minimizing Eq. 3.7 (or Eq. 3.8) with respect to the weights of the networks in G. As the target identity code $\boldsymbol{c}$ ranges over all N_{id} distinct subjects, N_{id} synthetic images $\widehat{\boldsymbol{x}'}$ are produced for each training or test image $\boldsymbol{x}$. As in the training of VAEs and GANs, the expectations are approximated by empirical averages computed from a mini-batch of training examples.

Over successive training epochs, G learns to fit the true data distribution and creates a realistic face image that can fool D^1 having the specified utility attribute as the input image, which can be correctly recognized by D^3, and identity $y'(\boldsymbol{c})$, which can be correctly recognized by D^2. As the latent code $\boldsymbol{c}$ determines the identity of $\widehat{\boldsymbol{x}'}$, the encoder is encouraged to disentangle the identity information from $\boldsymbol{z}$. Moreover, as $\widehat{\boldsymbol{x}'}$ retains information about the specified factor, the encoder is also encouraged to embed as much information about the specified factor as possible into $\boldsymbol{z}$. As a consequence, $\boldsymbol{z}$ is a generative representation that is not only invariant to identity, but also discriminative for recognizing the specified utility attribute.

3.2.3 Experimental Results

Datasets

In order to validate the effectiveness of the proposed model, we conducted experiments on two facial expression datasets: FERG (Aneja et al., 2016) and MUG (Aifanti et al., 2010), and a synthetic human head pose dataset: UPNA Synthetic (Larumbe et al., 2017).

FERG is a database of cartoon characters with annotated facial expressions containing 55,769 annotated face images of six characters. The images for each character are grouped into 7 types of cardinal expressions, *viz.* anger, disgust, fear, joy, neutral, sadness and surprise. The MUG database is video-based. It consists of realistic image sequences of 86 subjects performing the same 7 cardinal expressions. For the sake of computational efficiency, we chose 8 subjects with the most image samples as our training and testing data. In each image sequence, we removed the first and last 20 frames which mostly correspond to the neutral expression. We used 11,549 images in total. In experiments with these two facial expression datasets, we randomly selected (without replacement) 85% images of each expression from each subject for the training set. The remaining 15% of images were used as testing data. We also resized each RGB image to 64×64-pixel resolution.

The UPNA Synthetic dataset contains 10 videos for each of 10 subjects. In total, this dataset includes 35,990 frames. Ground-truth continuous head pose angles (yaw, pitch, roll) and a face-centered bounding box are provided for each frame. In our experiments, we first cropped each frame using the provided bounding box and then resized it to 64×64-pixel resolution. We randomly chose (without replacement) 85% frames from each video to populate the training set. The remaining frames were used for evaluation.

The specified utility attribute we investigated for FERG and MUG is facial ex-

pression, while the three head pose angles are considered as the utility attributes of interest for UPNA Synthetic.

Training Details

We used the same network architecture for both facial expression datasets and only modified D^3 for UPNA Synthetic to make it produce real-valued estimates of the head pose angles. Details of PPRL-VGAN structure are listed in Table 3.1. We implemented our algorithm in Keras (Chollet, 2015) and trained all networks from scratch. The weights were initialized to be zero-mean Gaussian with a small standard deviation of 10^{-2}. We used a batch size of 256 and performed batch normalization after each convolutional/deconvolutional layer except the last deconvolutional layer in the decoder. We set $\alpha = 0.2$ for LeakyReLU's across the network. We used RMSprop optimizer (Hinton et al., 2012) with a learning rate of 0.0002. We observed that network training is very sensitive to the choice of the tuning parameters in the generator and discriminator cost functions. We optimized these parameters using grid search. We found that the following values: $\lambda_1^D = 0.25$, $\lambda_2^D = 0.5$, $\lambda_3^D = 0.25$ for discriminator training and $\lambda_1^G = 0.108$, $\lambda_2^G = 0.6$, $\lambda_3^G = 0.29$, $\lambda_4^G = 0.002$ for generator training work well. In conventional GANs, it is common to optimize the discriminator more frequently than the generator.

However, we update the generator twice as frequently as the discriminator during training because the identity and utility attribute labels used in PPRL-VGAN provide additional supervision signals that help the discriminator training, whereas such signals are unavailable for the generator training. The source code, additional implementation details and more experimental results are available on our project website (Chen, 2018).

Table 3.1: Architecture of PPRL-VGAN. ↓ and ↑ represent down- and upsampling operations, respectively. D^1, D^2 and D^3 share the weights of all convolutional layers and of the first fully-connected layer.

Layer	Encoder	Decoder	Discriminator
1	$5 \times 5 \times 32$ conv. ↓, BNorm, LeakyReLU	2048 FC layers $\xrightarrow{\text{Reshape}}$ $4 \times 4 \times 128$, LeakyReLU	$5 \times 5 \times 32$ conv, BNorm, LeakyReLU
2	$5 \times 5 \times 64$ conv. ↓, BNorm, LeakyReLU	$5 \times 5 \times 256$ deconv. ↑, BNorm, LeakyReLU	$5 \times 5 \times 64$ conv, BNorm, LeakyReLU
3	$5 \times 5 \times 128$ conv. ↓, BNorm, LeakyReLU	$5 \times 5 \times 128$ deconv. ↑, BNorm, LeakyReLU	$5 \times 5 \times 128$ conv, BNorm, LeakyReLU
4	$5 \times 5 \times 256$ conv. ↓, BNorm, LeakyReLU	$5 \times 5 \times 64$ deconv. ↑, BNorm, LeakyReLU	$5 \times 5 \times 256$ conv, BNorm, LeakyReLU
5	128 fully-connected (FC), Linear	$5 \times 5 \times 3$ deconv, tanh	256 fully-connected, LeakyReLU
6			D^1: 1 FC , D^2: N_{id} FC , D^3: 1 or 3 FC

Threat Scenarios

We evaluate privacy-preserving performance of the proposed PPRL-VGAN under three threat scenarios.

Attack scenario I: This is a simple scenario in which the attacker has access to the unaltered training set $(\boldsymbol{x}_{train}, y_{id}^{train})$. However, the attacker's test set consists of all images in the original test set *after* they have been passed through the trained PPRL-VGAN network. Thus, the attacker never gets to see the original test image $\boldsymbol{x}_{test}$ but only its privacy-protected version $\widehat{\boldsymbol{x}'}_{test}$. Also, the test set for the attacker contains all N_{id} distinct privacy-protected versions $\widehat{\boldsymbol{x}'}_{test}$ of each $\boldsymbol{x}_{test}$ corresponding to N_{id} distinct values of the identity code $\boldsymbol{c}$.

Attack scenario II: This is a more challenging scenario (from the perspective of protecting privacy) where the attacker has access to the privacy-protected training images $\widehat{\boldsymbol{x}'}_{train}$ and knows their underlying ground-truth identities y_{id}^{train}. Therefore, the attacker can train an identifier on training images that have the same type of identity-protecting transformation as the test images. If the proposed privacy-preserving transformation is weak and the identifier has sufficient learning capacity, it may be possible for a trained identifier to correctly predict the underlying ground-truth identity even from a privacy-protected test image. Similarly to scenario I, there

are N_{id} images for each training and testing image.

Attack scenario III: In this scenario, the attacker gets access to the encoder network and can obtain the latent representation $\boldsymbol{z}$ for any image $\boldsymbol{x}$. Then, if the produced latent representation is not void of identity traits, the attacker can train an identifier using $(\boldsymbol{z}_{train}, y_{id}^{train})$ and apply it to $\boldsymbol{z}_{test}$ for identification. Although more challenging than scenario II, because the attacker can access the "more pristine" $\boldsymbol{z}$, there are fewer training and test samples available since the identity code $\boldsymbol{c}$ does not enter into the picture and thus there is no N_{id}-fold dataset expansion. Moreover whereas $\widehat{\boldsymbol{x}}'_{train}$ resembles a real image, $\boldsymbol{z}$ needs not (and typically does not).

In terms of utility, we train a dedicated estimator in each scenario with the available format of training data and the corresponding ground-truth utility attribute labels. Then, we apply this estimator to test data and measure the recognition performance.

Privacy Preservation versus Data Utility

Results for facial expression datasets: We first report the evaluation results for the facial expression datasets with respect to privacy preservation and data utility. We use correct classification rate (CCR) in person identification to measure how much privacy is preserved (the lower the CCR, the better) and also in facial expression recognition to measure the utility of data (the higher the CCR, the better). Table 3.2 summarizes the performance of the proposed approach on the FERG and MUG datasets under a privacy-unconstrained scenario (training and testing sets are both unaltered), under a random-guessing attack and under the three attack scenarios described earlier. In each scenario, the identification and facial expression are estimated separately by different neural network classifiers.

For attack scenario I, we train an identifier using the original training set $(\boldsymbol{x}_{train}, y_{id}^{train})$ and apply it to privacy-protected test images $\widehat{\boldsymbol{x}}'_{test}$. The identifier has

Table 3.2: Person identification and facial expression recognition performance in different scenarios on FERG and MUG datasets.

Scenario	Identification		Expression Recognition	
	FERG	MUG	FERG	MUG
Privacy Unconstrained	100%	100%	100%	87.90%
Random Guess	16.67%	12.50%	14.29%	14.29%
Attack Scenario I	17.01%	12.80%	93.02%	82.33%
Attack Scenario II	28.30%	22.08%	95.00%	85.14%
Attack Scenario III	22.42%	20.62%	100.00%	87.58%

the same structure as D^2 (Fig. 3·2). We first observe that the identification CCRs are 17.01% for FERG and 12.80% for MUG. Both are close to a random guess (16.67% for FERG since there are 6 characters and 12.50% for MUG since we selected 8 subjects). However, the same classifier applied to the privacy-unconstrained test images results in 100% identification performance on both datasets. Such a huge performance gap confirms the proposed model effectively protects users' privacy when the attacker has no information about the applied privacy-preserving transformation. For utility evaluation, we train a dedicated facial expression classifier, with the same structure as D^3, using $(\boldsymbol{x}_{train}, y_u^{train})$ pairs and test it on $\widehat{\boldsymbol{x}'}_{test}$ images. The resulting expression recognition accuracies are 93.02% for FERG and 82.33% for MUG. These results are close to those achieved in the privacy-unconstrained scenario, which indicates that the synthesized images look realistic and retain the expression of the input images.

In attack scenario II, we use the privacy protected training data $\widehat{\boldsymbol{x}'}_{train}$ and the corresponding ground-truth identity labels to train an identity recognizer and the ground-truth expressions to train a facial expression classifier (having the same architectures as in scenario I). We first observe that the identification accuracy in scenario II is about 11% higher than that of a random guess for both datasets, which suggests that some identity-related information is leaked into the synthesized images, but this is still much lower than in the privacy-unconstrained scenario. With respect to facial expression recognition, the performance in scenario II is consistently better than that

in scenario I. This is likely because the number of training samples in scenario II is N_{id} times that in scenario I, which benefits the training of the facial expression classifier.

In attack scenario III, we assume the attacker can access the latent representations of the training and probe images. We simulate this attack scenario by training an identifier using $(\boldsymbol{z}_{train}, y_{id}^{train})$ and test it on $\boldsymbol{z}_{test}$. However, as $\boldsymbol{z}_{train}$ is a 1-D vector, the 2-D ConvNet classifiers we used before are not suitable. We have experimented with 3 classifiers for $\boldsymbol{z}_{train}$, namely a Support Vector Machine (SVM), a customized 1-D ConvNet and a customized Artificial Neural Network (ANN). The customized ANN (3 hidden layers, each with 256 nodes) performed best in terms of identification and expression recognition accuracy. Therefore, only results for the customized ANN classifier are reported. As shown in Table 3.2, the identification performance is reduced in comparison with scenario II. However, the expression recognition performance in scenario III is the best among the three attack scenarios. Effectively, this suggests that the learned image representation $\boldsymbol{z}$ contains crucial facial expression information, but is largely disentangled from the identity information.

Results for head pose dataset: We then report the experimental results for UPNA Synthetic. We use correct classification rate (CCR) and mean absolute error (MAE) to measure the performance of identification and head pose estimation, respectively. In all three attack scenarios, a low identification CCR and a small head pose estimation error are favored. The identification and head pose estimation performance are summarized in Table 3.3. In the privacy unconstrained scenario, both training and testing data are unaltered. The resulting identification CCRs upper-bound the attainable identification accuracy while the resulting head pose estimation MAEs lower-bound the attainable estimation error. On the other hand, the resulting identification CCR from the random guessing scenario lower-bound the identification performance

and the resulting head pose estimation MAEs from the "Median" estimate (the median value of ground truth across the entire training set) upper-bound the attainable estimation error.

Table 3.3: Person identification and head pose estimation performance in different scenarios on UPNA Synthetic dataset.

Scenario	Identification	Head Pose Estimation		
		Yaw°	Pitch°	Row°
Privacy Unconstrained	100%	0.69 ± 0.54	0.77 ± 0.80	0.50 ± 0.46
Random Guess/Median	10.00%	5.10 ± 6.70	4.98 ± 5.02	4.68 ± 6.88
Attack Scenario I	13.20%	2.89 ± 2.61	2.47 ± 2.24	2.03 ± 2.41
Attack Scenario II	20.57%	2.88 ± 2.56	2.34 ± 2.18	2.10 ± 2.38
Attack Scenario III	24.17%	2.47 ± 2.12	2.27 ± 2.01	2.01 ± 2.42

In attack scenario I, we observe that the identification performance of the proposed model is close to pure chance (10%). This indicates our model succeeds in protecting identity when the attacker has no knowledge about the applied privacy-protection transformation. As for head-pose estimation, we can see our model consistently outperforms the "Median" estimate by about 2 degrees. In attack scenario II, the identification CCR is higher than that in attack scenario I by 7%. This indicates a small amount of the identity information has leaked into the synthesized images, this is consistent with the observations on the facial expression datasets. However, the resulting CCRs are still much lower than those in the privacy unconstrained scenario. In terms of preserving head pose, the proposed model again significantly outperforms the "Median" estimate. As for attack scenario III, the identification and pose estimation performance are similar to that in scenario II. Overall, the quantitative results show that PPRL-VGAN performs well for privacy-preserving head pose estimation in various attack scenarios.

Image Synthesis

Identity Replacement/Expression Transfer: In addition to producing an identity-invariant image representation, PPRL-VGAN can be applied to an input face image of any identity to synthesize a realistic, utility-equivalent output face image of a target identity specified by the latent code $\boldsymbol{c}$ (see Fig. 3·3). This may also be equivalently viewed as "transferring" a facial attribute from one face to another. Unlike in a standard GAN, the synthesized image contains a lot of detail about the target identity due to the incorporation of the identifier D^2 and the utility attribute estimator D^3.

Face Image Synthesis without Input Image: Once trained, our model can also synthesize face images without using an input image. This is due to the constraint we impose on the encoder which forces the distribution of the latent representation to follow a prior distribution (in our experiments: $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$). To generate a new face image, we simply sample a latent vector from the prior distribution and concatenate it with an identity code. Then, we feed the concatenated vector into the decoder for image generation. As shown in Fig. 3·4, the synthesized images are realistic and the identities are consistent with the identity code $\boldsymbol{c}$. While the current model is incapable of controlling the specified utility attribute like facial expression of a generated image when no input image is given, we believe the synthesized images are useful for other applications, e.g, augmenting the original dataset.

Face Image Synthesis for Left-Out Expression: In order to further evaluate the generative capacity of PPRL-VGAN, we conducted experiments where we intentionally left out all samples of a specific facial expression e from subject i in training (images of expression e from other subjects are still used) and then synthesized the left-out expression for subject i after the model had been trained. This was done by feeding the generator G an image with expression e from subject j, $j \neq i$, and an

Figure 3·3: Examples of identity replacement for MUG (top) and UPNA Synthetic (bottom). In each row, from left to right, is an input image followed by synthesized images with identity code $\boldsymbol{c}_i, i = 1, ..., N_{id}$.

identity code $\boldsymbol{c}_i$ with ith entry equal to 1 and all other entries 0.

Figure 3·5 shows examples of left-out expression synthesis. While artifacts are clearly visible, the synthesized images capture the essential traits of a left-out expression, thus validating the generative capacity of PPRL-VGAN.

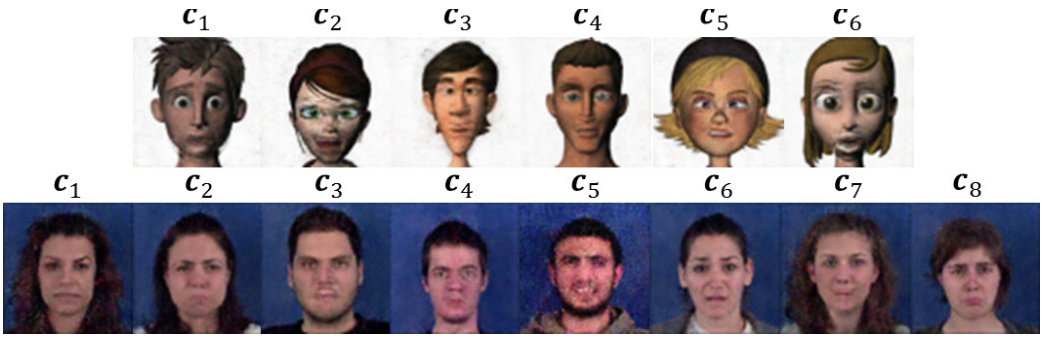
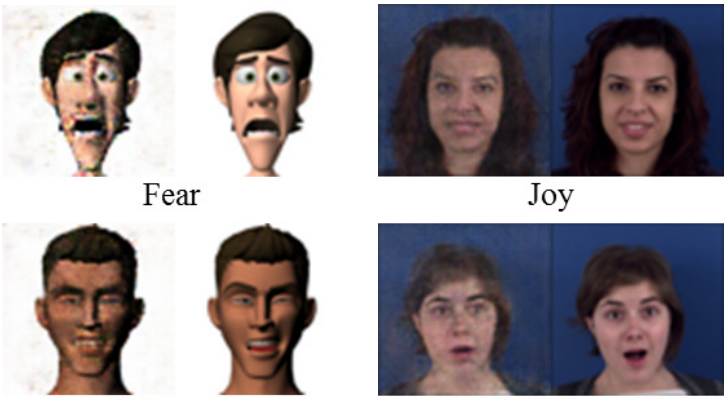


Figure 3·4: Image synthesis without input image; $\boldsymbol{z}$ is sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with identity code $\boldsymbol{c}_i, i = 1, ..., N_{id}$.



(a) FERG **(b)** MUG

Figure 3·5: Image synthesis of left-out expressions (left: synthesized image of a left-out expression; right: corresponding ground-truth image).

Expression Morphing: Facial expression morphing is a challenging problem because a human face is highly non-rigid and significantly deforms across expressions. Most methods perform face morphing in image space. Here, we leverage the latent representation and apply linear interpolation in latent space. Let $\boldsymbol{x}_{\text{initial}}$, $\boldsymbol{x}_{\text{final}}$ be a pair of source images with different expressions for subject i and $\boldsymbol{z}_{\text{initial}}$, $\boldsymbol{z}_{\text{final}}$ their corresponding latent representations. First, we linearly interpolate $\boldsymbol{z}_{\text{initial}}$ and $\boldsymbol{z}_{\text{final}}$ in the latent space to obtain a series of new representations $\boldsymbol{z}_{\text{interp}}$ as follows:

$$\boldsymbol{z}_{\text{interp}} = (1 - \alpha)\boldsymbol{z}_{\text{initial}} + \alpha \boldsymbol{z}_{\text{final}}, \quad \alpha \in [0, 1] \tag{3.9}$$

Then, we feed $\boldsymbol{z}_{\text{interp}}$ and identity code $\boldsymbol{c}_i$ into the decoder to synthesize images. Figure 3·6 shows two examples of expression morphing. We can see that in both cases, the facial expression changes gradually from left to right. These smooth semantic changes indicate the model is able to capture salient expression characteristics in $\boldsymbol{z}$.

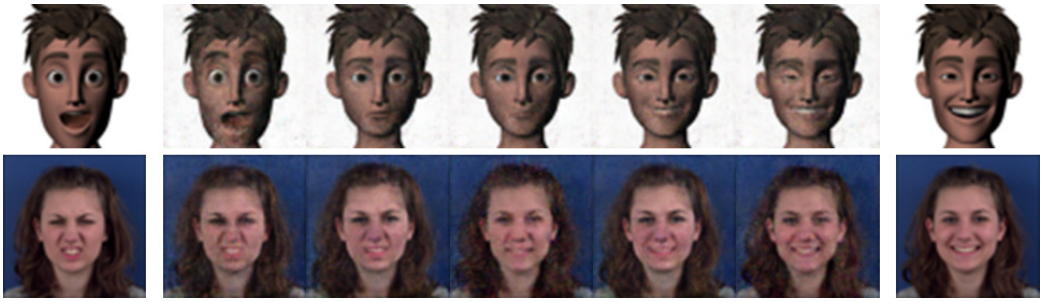
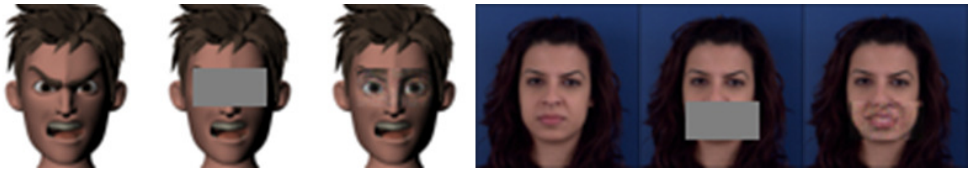


Figure 3·6: Examples of expression morphing for FERG (top) and MUG (bottom) datasets. The first and last images in each row are the source images, while those in-between are synthesized by linear interpolation in latent space.

Image completion: PPRL-VGAN can be also applied to an image completion task. We tested two different masks (Fig. 3·7): one covering the eyebrows, eyes and nose, and the other covering the mouth (each mask occupies $\sim$ 7% of the image). To complete the missing content of a query image $\boldsymbol{x}_q$ of subject j, we first pass $\boldsymbol{x}_q$ to the encoder to produce a latent representation $\boldsymbol{z}_q$. Then, we feed $\boldsymbol{z}_q$ and $\boldsymbol{c}_i$ to the decoder for synthesizing a new image $\widehat{\boldsymbol{x}}' \sim Dec(\boldsymbol{z}_q, \boldsymbol{c}_i)$. Finally, we replace the missing pixel values of $\boldsymbol{x}_q$ with values from corresponding locations in $\widehat{\boldsymbol{x}}'$.

Examples of both successful and unsuccessful image completions are shown in Fig. 3·7. Figure 3·7a shows examples for which our model was able to accurately estimate the missing image content. This demonstrates that our model learns correlations between different facial features, for example that opening the mouth is likely to appear jointly with raising eyebrows. However, our model occasionally fails (Fig. 3·7b). One possible reason for this is that some critical facial features (e.g., lowered eyebrows and narrowed eyes in the angry expression) are missing. A distortion

(a) Examples of successful image completion



(b) Examples of unsuccessful image completion

Figure 3·7: Example of image completion for FERG and MUG datasets. From left to right: original image, masked image and image completion result. Note that the original images are excluded from the training set.

may also occur when a face in the synthesized images is not accurately aligned with the one in the query image.

3.3 Model II: Invariant Representation Learning Variational-GAN (IRL-VGAN)

3.3.1 Introduction

In section 3.2, we presented a framework, namely PPRL-VGAN, for privacy-preserving representation learning and face image synthesis. Experimental results on both facial expression datasets and a head pose dataset demonstrate that PPRL-VGAN strikes a balance between preservation of privacy and data utility. However, PPRL-VGAN is designed to retain only one specific utility information (with labels available for training) in the representations. Moreover, in PPRL-VGAN a discriminator is trained for each factor of variation so the number of model parameters grows linearly with the number of factors.

This section introduces our second invariant representation learning model. We call this model Invariant-Representation-Learning Variational-GAN (IRL-VGAN).

IRL-VGAN uses a single discriminator which only requires labels of the specified factor of variation (e.g., identity). Moreover, it is designed to automatically capture *all* unspecified factors of the data (e.g., pose and illumination) into the representation with no need for corresponding labels. Specifically, IRL-VGAN is a cyclically-trained adversarial network for learning mappings from image space to a latent representation space and back such that the latent representation is invariant to a *specified* factor of variation. The learned mappings also assure that the synthesized image is not only realistic, but has the same values for *unspecified* factors as the original image and a desired value of the specified factor. We encourage invariance to a specified factor, by applying adversarial training using a variational autoencoder in the image space. We strengthen this invariance by introducing a cyclic training process (forward and backward pass). We also propose a new method to evaluate conditional generative networks. It compares how well different factors of variation can be predicted from the synthesized, as opposed to real, images. We demonstrate the effectiveness of this approach on factors such as identity, pose, illumination or style on three datasets and compare it with state-of-the-art methods. Finally, we provide a performance comparison of IRL-VGAN and PPRL-VGAN on privacy-preserving head pose estimation task.

3.3.2 Formulation of IRL-VGAN

Let $\mathbf{X}$ denote the image domain and $\mathbf{Y} = \{y_1, ..., y_K\}$ a set of possible factors of variation associated with data samples in $\mathbf{X}$, where K is the number of factors. Given an image $\mathbf{x} \in \mathbf{X}$ and one specified factor y_s, where $y_s \in \{1, ..., N_s\}$ and N_s is the number of possible classes, our proposed approach has two objectives: 1) to learn a latent representation $\mathbf{z}$ which is invariant to the specified factor but preserves the other unspecified factors of variation, and 2) to synthesize a realistic sample $\widehat{\mathbf{x}}'$ which has the same unspecified factors as $\mathbf{x}$ and a desired specified factor value which is

determined by an input class code $\mathbf{c}(y'_s)$, where $y'_s \in \{1, ..., N_s\}$ is generated from a distribution $p(y'_s)$ and $\mathbf{c}(\cdot)$ is a one-hot encoding function. For simplicity, we consider here the case where y_s is categorical, but our approach can be extended to continuous y_s.

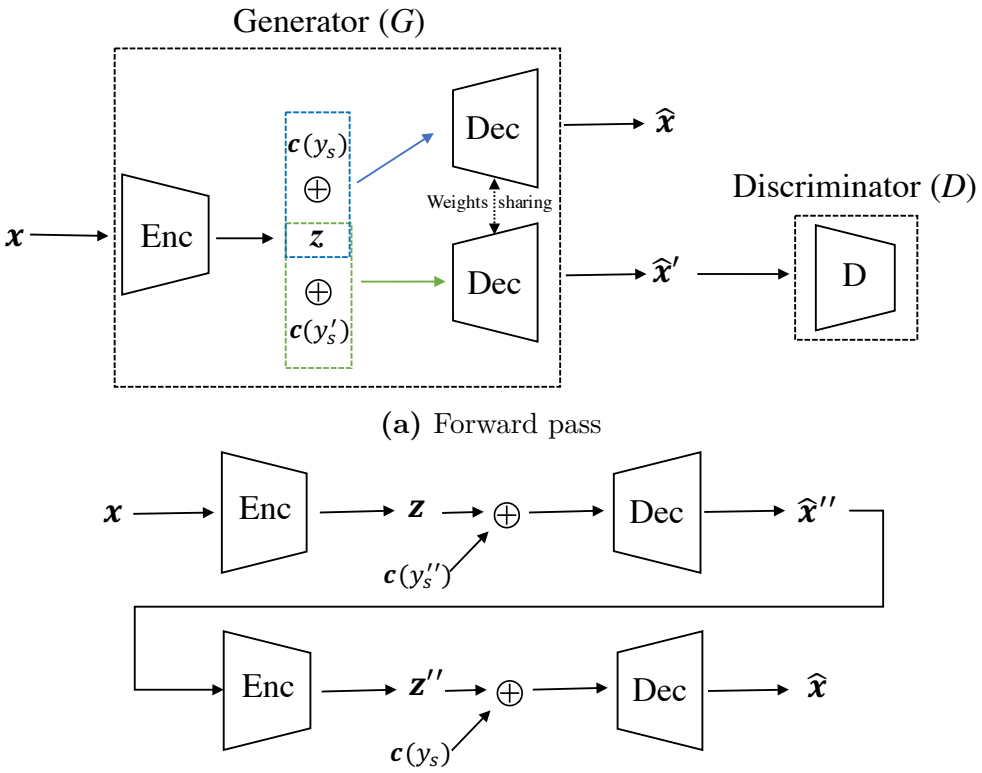
Generator: We structure the generator in the proposed model similarly to a variational auto-encoder (Fig. 3·8). The encoder (Enc) aims to create a low-dimensional data representation $\mathbf{z} = Enc(\mathbf{x})$ *via* a randomized mapping $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x})$ parameterized by the weights of the encoder's neural network. On the other hand, the decoder (Dec) is responsible for learning a mapping function $\widehat{\mathbf{x}}' \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y'_s))$ that can map the latent representation $\mathbf{z}$ in combination with with class code $\mathbf{c}(y'_s)$ back to the image space. The latent space is regularized by imposing a prior distribution, in our experiments a normal distribution $r(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Discriminator: Different from the discriminators in conventional GANs, the discriminator D in our model is a multi-class classifier. The output of the discriminator $D(\mathbf{x}) \in \mathbb{R}^{N_s+1}$ are the predicted probabilities of each class corresponding to N_s different values of the specified factor and an additional "fake" class.

Forward pass: First, we sample an image $\mathbf{x}$ from the training set and pass it through the encoder to generate a latent representation $\mathbf{z}$. The decoder is trained to produce a reconstruction of the input $\widehat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y_s))$ and also to synthesize a new data sample $\widehat{\mathbf{x}}' \sim p(\mathbf{x}|\mathbf{z}, \mathbf{c}(y'_s))$ that can fool the discriminator D into classifying it as the specified class y'_s. Specifically, the weights of the generator network are adjusted to *minimize* the following cost function:

$$\mathcal{L}_G^{fw}(G, D) = -\lambda_1^G E_{\mathbf{x}\sim p(\mathbf{x}), y'_s \sim p(y'_s)} \big[\log D_{y'_s}\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big)\big] + \\ \lambda_2^G E_{(\mathbf{x}, y_s)\sim p(\mathbf{x}, y_s)} \big[||\mathbf{x} - G(\mathbf{x}, \mathbf{c}(y_s))||_2^2\big] + \lambda_3^G KL(p(\mathbf{z}|\mathbf{x})||r(\mathbf{z})) \qquad (3.10)$$

where $p(\mathbf{x}, y_s)$ denotes the joint distribution of the real image and the specified factor



(a) Forward pass

(b) Backward pass

Figure 3·8: Schematic diagram of the proposed model ($\oplus$ represents concatenation): (a) forward pass in which training alternates between optimizing G and D; (b) backward pass that only optimizes G. Note: the label for image synthesis is denoted by y_s' in forward pass and y_s'' in backward pass.

in the training data, $p(\mathbf{x})$ the corresponding marginal distribution of the real image, $p(y'_s)$ a distribution of the specified factor used to synthesize a "fake" image, D_i is the predicted probability of the i-th class, and $\lambda_1^G, \lambda_2^G, \lambda_3^G$ are weighting factors.

The discriminator aims to correctly classify a real training sample $\mathbf{x}$ to its ground-truth class value y_s of the specified attribute but, when given a synthetic sample $\widehat{\mathbf{x}}'$ from the generator, it attempts to classify it as fake. This is accomplished by adjusting the weights of the discriminator by *maximizing* the following cost function:

$$\mathcal{L}_D^{fw}(G, D) = \lambda_1^D E_{(\mathbf{x},y_s)\sim p(\mathbf{x},y_s)}[\log D_{y_s}(\mathbf{x})] + \lambda_2^D E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}[\log D_{N_s+1}(G(\mathbf{x}, \mathbf{c}(y'_s)))] \tag{3.11}$$

where λ_1^D and λ_2^D are tuning parameters.

The weights of the networks in G and D are updated in an alternating order. Over successive training steps, G learns to fit the true data distribution and reconstruct the input image as well as synthesize realistic images that can fool D. The generator objective (second term in Eq. (3.10)) encourages the encoder to pass as much information about the unspecified factors as possible to the latent representation. Since the class code $\mathbf{c}$ determines the specified factor value of $\widehat{\mathbf{x}}'$, the encoder is also encouraged to eliminate information about the specified factor of $\mathbf{x}$ in the latent representation. The encoder may, however, fail to disentangle the specified and unspecified factors of variation and the decoder may still learn to synthesize images according to the class code $\mathbf{c}$ by ignoring any *residual* information about the specified factor that is contained within the representation. To avoid such a *degenerate* solution, we use a backward pass to further constrain the latent space.

Backward pass: This pass requires a synthesized image $\widehat{\mathbf{x}}''$ of class y''_s generated from a real image $\mathbf{x}$ of class y_s. We intentionally choose $y''_s \neq y_s$ so that $\widehat{\mathbf{x}}''$ and $\mathbf{x}$ carry different specified factor values. Two latent representations $\mathbf{z} = Enc(\mathbf{x})$ and $\mathbf{z}'' =$

$Enc(\widehat{\mathbf{x}}'')$ can be computed by passing, respectively, $\mathbf{x}$ and $\widehat{\mathbf{x}}''$ through the encoder (see Fig. 3·8b). If the encoder fails to transmit information about unspecified factors from the input to its latent representation, or if it retains considerable information about the specified factor in the latent space, then we would expect the representations $\mathbf{z}$ and $\mathbf{z}''$ to have a large pairwise distance. In addition, we would like to encourage the generator to reconstruct the input $\mathbf{x}$ from its synthetic version $\widehat{\mathbf{x}}''$ in combination with a class code $\mathbf{c}(y_s)$ that encodes the ground-truth label of the specified factor of $\mathbf{x}$. These considerations motivate optimizing the generator in the backward pass by minimizing the following cost function:

$$\mathcal{L}_G^{bw} = E_{(\mathbf{x},y_s)\sim p(\mathbf{x},y_s),y_s''\sim p(y_s'')}\left[\lambda_1^{bw}||\mathbf{z}-\mathbf{z}''||_1 + \lambda_2^{bw}||\mathbf{x}-G(\hat{\mathbf{x}}'',\mathbf{c}(y_s))||_2^2\right] \tag{3.12}$$

$$\mathcal{L}_G^{bw} = E_{(\mathbf{x},y_{id})\sim p(\mathbf{x},y_{id}),y_{id}''\sim p(y_{id}'')}\left[\lambda_1^{bw}||\mathbf{z}-\mathbf{z}''||_1 + \right. \tag{3.13}$$

$$\left. \lambda_2^{bw}||\mathbf{x}-G(\hat{\mathbf{x}}'',\mathbf{c}(y_{id}))||_2^2\right] \tag{3.14}$$

where λ_1^{bw} and λ_2^{bw} are two weighting factors. The first term in Eq. (3.14) penalizes the generator if $\mathbf{z}$ is not close to $\mathbf{z}''$. The second term encourages the synthesized $\widehat{\mathbf{x}}$ to resemble $\mathbf{x}$.

Essentially, the forward pass translates $\mathbf{x}$ to a synthetic image $\widehat{\mathbf{x}}'' = G(\mathbf{x},\mathbf{c}(y_s''))$ followed by a backward transform $\hat{\mathbf{x}} = G(\widehat{\mathbf{x}}'',\mathbf{c}(y_s))$, such that $\widehat{\mathbf{x}} \simeq \mathbf{x}$. This cyclic training process assists the model in generating good quality images and further encourages invariance to the specified factor in the latent space.

3.3.3 Experimental Results

We evaluate the performance of IRL-VGAN on three image datasets: 3D Chairs (Aubry et al., 2014), YaleFace (Lee et al., 2005) and UPNA Synthetic (Ariz et al., 2016). We first conduct a quantitative evaluation of the degree of invariance

in the latent space by training dedicated neural networks (one per factor) to predict the values of the specified and certain unspecified factors (that have ground-truth labels) from the latent representation. The factor prediction accuracies quantify how much information about each factor has been preserved in the latent representation. If the model succeeds in eliminating all information about the specified factor and preserving all information about unspecified factors, we should expect the prediction accuracy for the specified factor to be close to pure chance and the prediction accuracies for the unspecified factors to be nearly perfect. We also evaluate the quality of the image generation process. Unlike previous works (Hadad et al., 2018; Harsh Jha et al., 2018), which only provide a qualitative evaluation through visual inspection of the synthesized images, we propose a new method to quantitatively assess the capability of a conditional generative model to synthesize realistic images while preserving unspecified factors. We will present details of the proposed evaluation method and associated experimental results in a later section.

We compare our model with two state-of-the-art methods (Hadad et al., 2018; Harsh Jha et al., 2018) that learn to produce, for a given input image, two latent vectors (as opposed to just one in our method). One of the latent vectors captures information related to the unspecified factors of variation and is, in an ideal scenario, devoid of any information related to the specified factor of variation. This latent vector is the counterpart of the latent invariant representation in our method. For synthesizing an image with a desired value for the specified factor, the methods in (Hadad et al., 2018; Harsh Jha et al., 2018) require an additional surrogate image which has the desired value for the specified factor. They would then *substitute* the latent vector of the specified factor in the original image with that of the surrogate image and then decode the result. Our approach, in contrast, uses a class code (as opposed to a surrogate image) to explicitly set the value of the specified factor in

the synthesized image. In our experiments, we compare the latent vectors for the unspecified factors from the competing methods and the latent representation from our method in terms of their ability to predict the specified and unspecified factors which indicates the quality of invariance. We used the publicly available source code to implement both benchmarks, but slightly modified their network architectures to ensure that all three competing models have similar numbers of parameters. We also did parameter tuning for each method for each of the three datasets.

Additional results of performance comparison between IRL-VGAN and PPRL-VGAN regarding privacy-preserving head pose estimation are presented in section 3.3.4.

Datasets

3D Chairs: This dataset includes 1,393 3D chair styles rendered on a white background from 62 different viewpoints that are indexed by two values of angle θ and 31 values of angle ϕ. Each image is annotated with the chair identity indicating its style as well as viewpoint (θ, ϕ). For each chair style, we randomly picked 50 images (out of 62) to populate the training set, and used the remaining 12 images in the testing phase. This gives, in total, 69,650 images in the training set, and 16,716 images in the test set.

YaleFace: This dataset consists of gray-scale frontal face images of 38 subjects under 64 illumination conditions. In our experiments, we randomly chose 54 images (out of 64) from each subject for training, and use the rest as the test set for performance evaluation.

UPNA Synthetic: This is a synthetic human head pose database. It consists of 12 videos for each of 10 subjects; 120 videos in total with 38,800 frames. Ground-truth *continuous* head pose angles (yaw, pitch, roll) are provided for each frame. We randomly selected 85% of the frames from each video for each subject for the training

and used the remaining 15% for testing.

For computational efficiency, in our experiments, we resized each RGB image to 64×64-pixel resolution for all three datasets. Table. 3.4 summarizes the specified and unspecified factors of variation that we investigate across the three datasets.

Table 3.4: Specified and unspecified factor(s) of variation investigated in the three datasets.

Dataset	Specified factor	Unspecified factor(s)
3D Chairs (Aubry et al., 2014)	Chair style	View orientation (θ, ϕ)
YaleFace (Lee et al., 2005)	Identity	Illumination Cond.
UPNA Synthetic (Ariz et al., 2016)	Identity	Head pose

Quality of invariance

We follow previous methodology (Harsh Jha et al., 2018; Hadad et al., 2018) and train dedicated neural network estimators to predict the specified and unspecified factors of variation based on the learned latent representations generated by each competing model. We use correct classification rate (CCR) and mean absolute error (MAE) to measure the performance of classification tasks and regression tasks, respectively.

In the 3D Chairs dataset, we regard chair style as the specified factor and the viewing orientation angles as the unspecified factors. Since both orientation angles are discrete, we treat viewing orientation estimation as a classification problem. As shown in Table 3.5, all three competing models manage to reduce the style information contained within the latent representation to a large extent (very low style prediction CCR values). However IRL-VGAN outperforms the benchmark models, in terms of the ability to predict the viewing orientation angles, by a large margin (about 11–28% CCR improvement for ϕ and 9–13% CCR improvement for θ). We also observe that the backward pass significantly improves invariance, i.e., style prediction CCR decreases from 3.21% to 0.79%.

Table 3.5: Classification CCRs for predicting chair style and discrete viewing orientation angles (θ, ϕ) based on the latent representations of 3D Chairs dataset. *Lower* is better for style classification. *Higher* is better for orientation classification.

Method	**Style**	$\boldsymbol{\theta}$	$\boldsymbol{\phi}$
Random guess	0.07%	50%	3.22%
(Hadad et al., 2018)	0.77%	68.92%	50.23%
(Harsh Jha et al., 2018)	0.70%	64.22%	43.75%
IRL-VGAN	0.79%	78.17%	71.90%
IRL-VGAN w/o backward pass	3.21%	74.37%	69.45%

Table 3.6: Classification CCRs for person identification and illumination condition recognition based on the latent representations for the YaleFace dataset. *Lower* is better for person identification and *higher* is better for illumination condition recognition.

Method	**Identity**	**Illumination Condition**
Random guess	2.63%	1.56%
(Hadad et al., 2018)	4.68%	77.80%
(Harsh Jha et al., 2018)	5.50%	32.36%
IRL-VGAN	6.97%	85.50%
IRL-VGAN w/o backward pass	12.36%	85.40%

Table 3.7: Classification CCRs for person identification, and MAE/standard deviation for head-pose estimation based on the latent representations for the UPNA Synthetic dataset. *Lower* is better for both tasks.

Method	**Identity**	**Yaw°**	**Pitch°**	**Roll°**
Random guess/ Median	10%	5.10 ± 6.70	4.98 ± 5.02	4.68 ± 6.88
(Hadad et al., 2018)	15.80%	2.77 ± 2.00	2.43 ± 2.10	1.19 ± 1.43
(Harsh Jha et al., 2018)	18.83%	2.42 ± 2.52	2.88 ± 2.71	1.65 ± 2.35
IRL-VGAN	18.05%	2.12 ± 2.12	2.23 ± 2.10	1.16 ± 1.24
IRL-VGAN w/o backward pass	33.40%	2.10 ± 2.08	2.20 ± 2.06	1.29 ± 1.43

Table 3.6 summarizes the performance of each model on the YaleFace dataset. In this case, subject identity is considered as the specified factor and illumination condition as the unspecified factor of variation. We first observe that the identification performance of the three models is comparable and close to a random guess, which

suggests the competing models perform equally well in creating representations that are invariant to identity. For the recognition of illumination condition, the classification CCR for our model is 85.50%, which again surpasses the two benchmark CCRs by about 8% and 53% in accuracy. Such large performance gaps suggest that the invariant representation learned by our model is better, than the competing alternatives, in preserving information about unspecified factors of variation. Lastly, we observe that the identification CCR of the complete IRL-VGAN model is about 5% less than that of using forward pass only, which again verifies the effectiveness of the backward pass.

In the case of UPNA Synthetic dataset, the specified and unspecified factors of variation used in evaluation are subject identity and head pose, respectively. Head pose is defined as a three-dimensional angular value (yaw, pitch, roll) in continuous space. Thus, we train neural-network based regressors to estimate head pose and report the mean and standard deviation of the absolute errors for yaw, pitch and roll angles separately. Detailed evaluation results are shown in Table 3.7. In terms of identification accuracy, the performance of the three methods is similar (no more than 3% difference in CCR or about 2-3 times that of a random guess). For our model, the incorporation of backward pass greatly helps to reduce identification CCR from 33.40% to 18.05%. As for head-pose estimation, we use "Median" estimate as a baseline, i.e., the median value of ground truth across the entire training set. We note that our model slightly, but consistently, outperforms the benchmarks, and significantly outperforms the median estimate. This once again confirms the effectiveness of our model in preserving information pertaining to the unspecified factors in the latent representation while discarding information related to the specified factor.

Quality of image generation

Many studies have proposed measures to evaluate generative models for image synthesis. Some of them attempt to quantitatively evaluate models while some others emphasize qualitative approaches, such as user studies (e.g., visual examination). However, such subjective assessment may be inconsistent and not robust as human operators may fail to distinguish subtle differences in color, texture, etc. In addition, such a measure may favor models that can merely memorize training samples. In terms of quantitative methods, some studies proposed to use measures from image quality assessment literature such as SSIM, MSE and PSNR. However, they require a corresponding reference real image for each synthesized one. Other widely-adopted reference-free quantitative measures like Inception Score (Salimans et al., 2016) and Fréchet Inception Distance (Heusel et al., 2017) are designed for generic GANs and only measure how realistic a GAN's output is. Thus, they are not suitable for conditional models that aim to generate samples from a particular class. Several quantitative evaluation methods have been proposed for conditional generative models. For example, it was proposed to feed fake colorized images (of real grayscale images) to a classifier that was trained on real color images (Zhang et al., 2016). If the classifier performs well, this indicates that the colorization is accurate.

Inspired by the previous studies that use an off-the-shelf classifier to assess the realism of synthesized data, we propose a quantitative method that utilizes a number of attribute estimators to evaluate the quality of conditional generative models. The intuition is that a good generative model for learning an invariant/disentangled representation should have the capability to explicitly and accurately control the specified factor value when it generates a novel image. Furthermore, it should precisely transfer the other unspecified factors of variation from the source image to its synthetic version. Therefore, we can evaluate a model by measuring how well the different

factors of variation in the synthesized images can be predicted *via* estimators that are pretrained on the real images.

Specifically, we train a number of attribute estimators $\mathcal{F}^j$, where $j \in \{1, ..., K\}$, on the original training sets of real images. For each (real) test image $\mathbf{x}$ having specified and unspecified factors of variation y_j, $j \in \{1, \ldots, K\}$, we synthesize a new version $\widehat{\mathbf{x}}' = G(\mathbf{x}, \mathbf{c}(y'_s))$ using the generator, where y'_s is sampled at random, independently of $\mathbf{x}, y_s$, from a distribution $p(y'_s)$. The image $\widehat{\mathbf{x}}'$ thus synthesized is passed to the pretrained estimators to obtain a prediction for each attribute (whether specified or unspecified). If a factor of variation y_j is categorical, then $\mathcal{F}^j(\widehat{\mathbf{x}}')$ is a probability distribution over the set of all possible values that factor can take. In particular, $\mathcal{F}^j_{y_j}(\widehat{\mathbf{x}}') = p(y_j|\widehat{\mathbf{x}}')$. If y_j is continuous, then $\hat{y_j} := \mathcal{F}^j(\widehat{\mathbf{x}}')$ is a numerical value which should be approximately equal to y_j. In order to quantify performance, we introduce the following *Generator Label Score* (*GLS*) for both discrete and continuous factors of variation. For a categorical unspecified factor y_j,

$$GLS := E_{(\mathbf{x},y_j)\sim p(\mathbf{x},y_j),y'_s\sim p(y'_s)}\big[\mathcal{F}^j_{y_j}\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big)\big]$$

whereas for a categorical specified factor y_s,

$$GLS := E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}\big[\mathcal{F}^j_{y'_s}\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big)\big].$$

For a quantitative unspecified factor y_j,

$$GLS := E_{(\mathbf{x},y_j)\sim p(\mathbf{x},y_j),y'_s\sim p(y'_s)}||\mathcal{F}^j\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big) - y_j||^p$$

whereas for a quantitative specified factor y_s,

$$GLS := E_{\mathbf{x}\sim p(\mathbf{x}),y'_s\sim p(y'_s)}||\mathcal{F}^j\big(G(\mathbf{x}, \mathbf{c}(y'_s))\big) - y'_s||^p.$$

For a good conditional generative model, the value of *GLS* should be high for every categorical factor of variation (whether specified or unspecified) and low for every quantitative factor. Although quantitative, *GLS* need not correlate well with the subjective quality of synthesized images as perceived by humans. It is also worth mentioning that *GLS* provides a vector of values for a given set of factors, but they can be converted to a single value if the relative importance of each attribute is known.

In order to compute *GLS*, we use the three competing models to create, separately, synthetic versions of test images for each dataset. For the proposed model, the input image $\mathbf{x}$ and class code $\mathbf{c}$ provide the necessary information about unspecified and specified factors, respectively. Thus, we synthesize a new version for each test image by passing it through the generator in combination with a randomly-generated class code. For the benchmark methods, we follow the procedure described in the respective papers to generate new images. In order to generate a new sample, we combine the unspecified latent representation of a test image and the specified latent representation of another image randomly picked from the same test set.

Tables 3.8, 3.9 and 3.10 report the *GLS* for the three datasets. We first observe that our model consistently achieves better scores compared to the benchmark models. In particular, *GLS* values for the specified factors (chair style and identity) for our model are nearly perfect suggesting that our model manages to accurately alter the specified factor value in the generated images. With respect to unspecified factors of variation, our model yields a high *GLS* value for the illumination condition (0.70) and a low value for head pose (e.g., 1.37 for roll angle). While the achieved scores on viewing orientation (θ, ϕ) for our model are slightly lower than expected, they are still better than those for the benchmarks. This is likely because our model occasionally fails to precisely construct chairlegs or arms (see Fig. 3·9a), which provide important cues for recognizing the viewing orientation. It is worthwhile to mention that the

performance differences are less significant on UPNA Synthetic dataset. One possible reason is that it has the maximum number of training samples per class among the three datasets which could benefit the training of the generator.

Table 3.8: *GLS* values for chair style and viewing orientation (θ, ϕ) for 3D Chairs dataset. *Higher* is better for both factors.

Model	**Chair style**	$\boldsymbol{\theta}$	$\boldsymbol{\phi}$
IRL-VGAN	0.87	0.66	0.57
(Hadad et al., 2018)	0.02	0.56	0.38
(Harsh Jha et al., 2018)	0.02	0.61	0.49

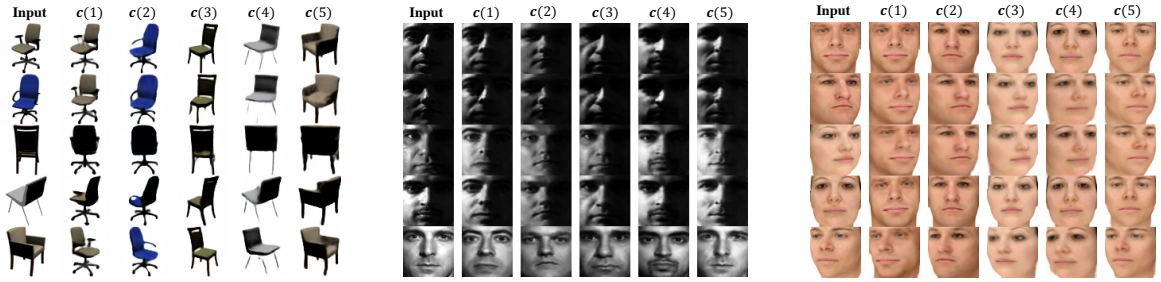
Table 3.9: *GLS* values for identity and illumination condition for YaleFace dataset. *Higher* is better for both factors.

Model	**Identity**	**Illumination cond.**
IRL-VGAN	0.98	0.70
(Hadad et al., 2018)	0.24	0.17
(Harsh Jha et al., 2018)	0.07	0.29

Table 3.10: *GLS* values for identity and head pose (yaw, pitch and roll) for UPNA Synthetic dataset. *Higher* is better for identity. *Lower* is better for head pose.

Model	**Identity**	**Yaw**	**Pitch**	**Roll**
IRL-VGAN	1.00	2.55	2.46	1.37
(Hadad et al., 2018)	0.88	3.51	4.07	3.17
(Harsh Jha et al., 2018)	0.98	2.65	2.84	1.47

In addition to quantitative results, we also show qualitative results of modifying a specified factor of variation within an image using the three competing models (see Figure 3·9). One can see that IRL-VGAN can change a specified factor of variation in an input image, such as face identity or chair style, by adjusting class code **c**. Meanwhile, the other unspecified factors such as orientation, illumination condition or head pose of the input image are largely preserved in its synthetic version. Overall, images generated by IRL-VGAN are realistic although distortions may occur in image details, e.g, chair legs (see the fifth image in the second row of Fig. 3·9a). In contrast, the benchmark methods can only combine the specified factors from one



(a) Image synthesis results for IRL-VGAN

(b) Image synthesis results for the model in (Hadad et al., 2018)

(c) Image synthesis results for the model in (Harsh Jha et al., 2018)

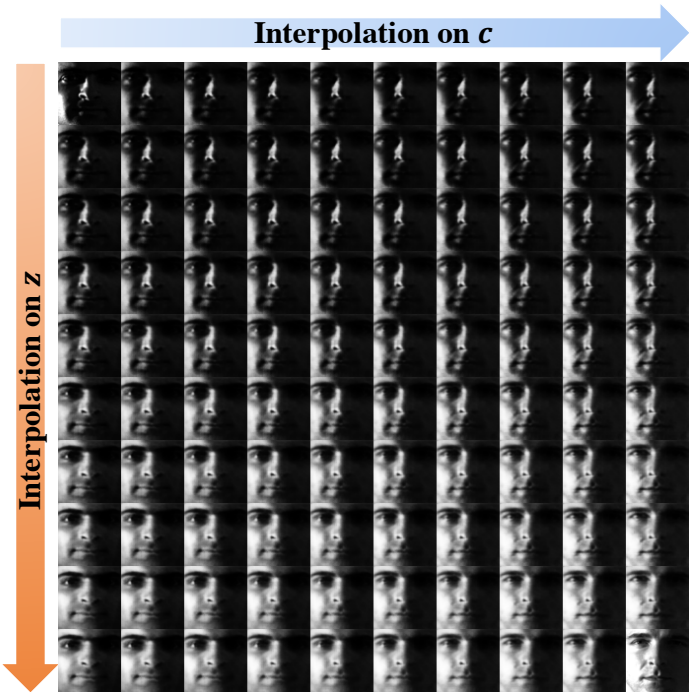
Figure 3·9: Image synthesis by altering the specified factor of variation in 3D Chairs (Aubry et al., 2014), YaleFace (Lee et al., 2005) and UPNA Synthetic (Ariz et al., 2016) (from left to right). (a): The proposed model can modify a specified factor of variation (e.g, chair style) by adjusting the input class code **c**. (b) & (c): Both benchmark models swap the specified latent representation (from the left column images) and the unspecified latent representation (from the top row images) to synthesize new images.

source image and the unspecified factors from another source image to generate a new image. Therefore, they have less flexibility to modify a specified factor of variation to a desired value. Images shown in Figures 3·9b and 3·9c are generated by feeding the specified representations from images in the first row, and the unspecified representations from images in the first column to the decoder. The visual quality of corresponding images is inferior to those from our model; blur and distortions are clearly visible. Furthermore, the benchmark methods are less effective in maintaining certain important factors of variation, e.g., color in the synthesized images (see the generated chair images in Figs. 3·9b and 3·9c).

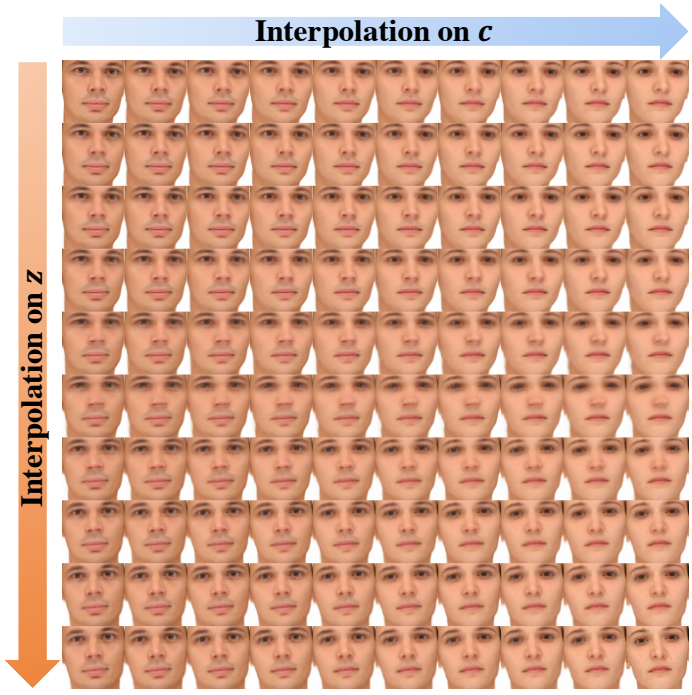
The remarkable consistency of the quantitative and qualitative results confirms the effectiveness of the proposed model in creating realistic images with a desired value for the specified factor and the same unspecified traits as the source images.

Interpolation of synthesis variables

In order to further evaluate the generative capacity of the proposed model, we conducted additional experiments wherein we linearly interpolate between latent representations and class codes of an initial and a final image in order to obtain a series of new image representations and class codes which are then combined and fed to a trained decoder to synthesize new images. Specifically, let $\mathbf{z}_{\text{initial}}, \mathbf{z}_{\text{final}}$ and $\mathbf{c}_{\text{initial}}, \mathbf{c}_{\text{final}}$ denote, respectively, the learned latent representations and class codes of the initial and final images and $\mathbf{c}_{\text{interp}} = (1 - \alpha_c)\mathbf{c}_{\text{initial}} + \alpha_c \mathbf{c}_{\text{final}}$ and $\mathbf{z}_{\text{interp}} = (1 - \alpha_z)\mathbf{z}_{\text{initial}} + \alpha_z \mathbf{z}_{\text{final}}$ their interpolated values, where $\alpha_c, \alpha_z \in [0, 1]$. We synthesize new images by passing $(\mathbf{c}_{\text{interp}}, \mathbf{z}_{\text{interp}})$ to the decoder. Surprisingly, when this is applied to a face dataset, our trained model can generate a sequence of face images that show a seamless transition from one identity into another, i.e., face morphing (rows of Figs. 3·10), and also a seamless transition from one value of an unspecified factor (e.g., illumination, pose) into another (columns of Fig. 3·10).



(a) YaleFace



(b) UPNA Synthetic

Figure 3·10: Linear interpolation results for the proposed model in the latent space ($\mathbf{z}$) and class code space ($\mathbf{c}$). The top-left and bottom-right images are taken from the test set.

Figure 3·11: Image synthesis without input image; $\mathbf{z}$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

This is despite the fact that the model can only see one-hot codes specifying *discrete* identities during training. In Fig. 3·10, the class code is constant within each column while the representation is constant within each row. We observe that when interpolating $\mathbf{c}$, the unspecified factors such as illumination or head pose are consistent, while the specified factor (identity) changes gradually. In contrast, when interpolating $\mathbf{z}$ the specified factor remains unchanged but the unspecified factors transform continuously.

Image synthesis without input image

IRL-VGAN can also synthesize novel images without using an input image as the latent space distribution has been forced to be close to a prior distribution during training. To generate a new image, we first sample a latent vector from a prior distribution (in our experiments: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$). Then, we concatenate it with a class code and feed them into a trained decoder to synthesize a new image. As shown in Fig. 3·11, the synthesized images are realistic and could be useful for applications such as dataset augmentation.

3.3.4 Performance Comparison of IRL-VGAN and PPRL-VGAN on Privacy-Preserving Head Pose Estimation

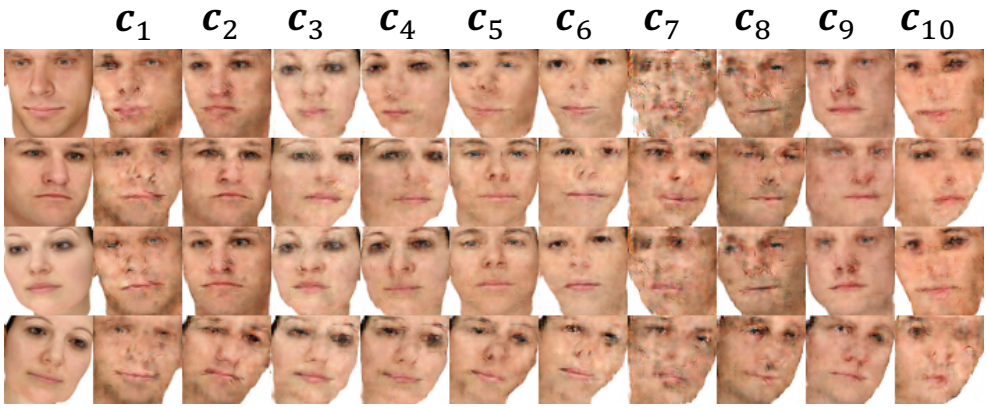
We compare the performance of PPRL-VGAN and IRL-VGAN on the task of privacy-preserving head pose estimation. We use the same evaluation strategy described in section 3.2.3 and provide quantitative and qualitative results of both methods under three privacy-threat scenarios.

Table 3.11: Classification CCRs for person identification and MAE for head pose estimation on UPNA Synthetic. *Lower* is better for both tasks.

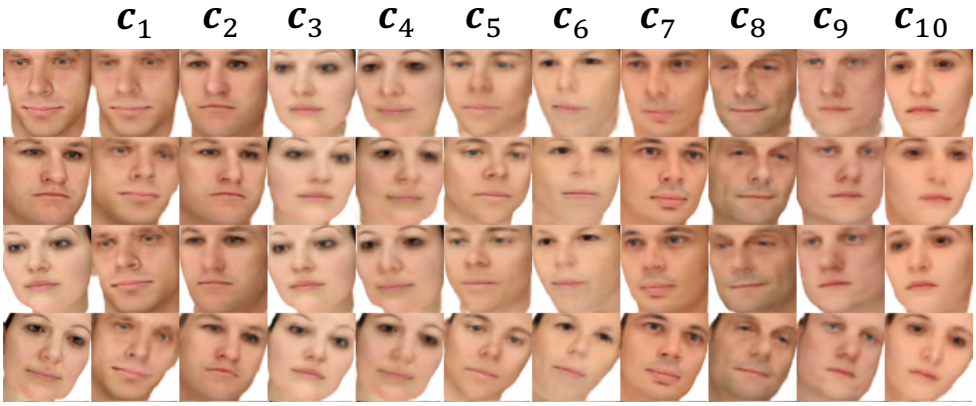
Scenarios	Identification(%)		Yaw°		Pitch°		Roll°	
	IRL-VGAN	PPRL-VGAN	IRL-VGAN	PPRL-VGAN	IRL-VGAN	PPRL-VGAN	IRL-VGAN	PPRL-VGAN
Privacy Unconstrained	100		0.64		0.77		0.50	
Random Guess/Median	10.00		5.10		4.98		4.68	
Attack Scenario I	10.00	13.20	2.55	2.89	2.46	2.47	1.37	2.03
Attack Scenario II	19.53	20.57	2.10	2.88	2.25	2.34	1.18	2.10
Attack Scenario III	18.85	24.17	2.12	2.47	2.23	2.27	1.16	2.01

Table. 3.11 reports the identification and head pose estimation results of the two proposed approaches on the UPNA Synthetic dataset. In each scenario, the identification and head pose are estimated separately by different neural network classifiers. We first observe that the identification CCRs of IRL-VGAN are consistently lower than those of PPRL-VGAN by 1 to 5 percent, which suggests IRL-VGAN is more effective in protecting user's privacy. In terms of head pose estimation, both methods manage to significantly outperform the median estimate, but IRL-VGAN performs slightly better than PPRL-VGAN. This is encouraging because it indicates that better identity-invariant utility-preserving image representations can be achieved even without using labels for utility attributes.

We also provide evaluation results for image synthesis. Figure. 3·12 shows identity replacement examples for both methods. We can observe that the synthesized images



(a) Identity replacement examples for PPRL-VGAN



(b) Identity replacement examples for IRL-VGAN

Figure 3·12: Examples of identity replacement for UPNA Synthetic. In each row, from left to right, is an input image followed by synthesized images with different identity codes.

from both models are realistic-looking, showing the target identities and accurately preserving the original head pose of the input images. While compared to the images generated by PPRL-VGAN, the synthetic face images generated by IRL-VGAN have better visual quality (e.g., contain fewer artifacts). This is also consistent with the *GLS* results shown in Table. 3.12.

Overall, the empirical results suggest both PPRL-VGAN and IRL-VGAN perform well in privacy-preserving head pose estimation. Whereas, IRL-VGAN slightly outperforms PPRL-VGAN in terms of creating identity-invariant, head pose-preserving

Table 3.12: *GLS* values for identity and head pose (yaw, pitch and roll) for UPNA Synthetic dataset. *Higher* is better for identity. *Lower* is better for head pose.

Model	**Identity**	**Yaw**	**Pitch**	**Roll**
IRL-VGAN	1.00	2.55	2.46	1.37
PPRL-VGAN	0.90	2.89	2.47	2.03

image representations and synthesizing novel face images. Furthermore, training IRL-VGAN does not rely on labels for utility attributes. These factors could make IRL-VGAN more favorable in practice.

3.4 Discussions

This chapter presented two invariant representation learning models that can be applied to privacy-preserving visual recognition tasks, e.g., facial expression recognition. The PPRL-VGAN is designed to create an identity-invariant representation of a face image that also permits synthesis of a utility-preserving and realistic version. Training this model requires labels for both identity and utility attributes. The IRL-VGAN is capable to generate an image representation that is invariant to a specified factor of variation (e.g., identity and style), while maintaining *all* unspecified factors. It further promotes invariance using a novel cyclic forward-backward training strategy. Quantitative and qualitative results from a broad set of experiments show that both models perform well in various tasks. IRL-VGAN slightly outperforms PPRL-VGAN despite the fact that it does not need labels for utility attributes. Once trained, both models are also generative as they enable synthesis of a realistic image having a desired value for the specified factor. Finally, both models facilitate image manipulation, such as face morphing.

An alternative approach for protecting user's visual privacy while also providing high utility is to create an "isolated" smart room equipped with standard-resolution

cameras, and a state-of-the-art multi-task visual recognition algorithm which is designed to capture all *currently* specified utilities. This approach could potentially mitigate privacy concerns since no information about the room would be shared. However, the types of specified utilities may change over time and it may be expensive to keep updating the distributed local cameras in order to adapt them for new smart room functionalities.

In contrast, using our proposed IRL-VGAN enables the local cameras to produce low-dimensional image representations and synthesized images that are, in principle, capable of retaining most unspecified utility information. The visual recognition tasks could be done remotely without privacy loss if the room were to only share the generated data from IRL-VGAN. As a result, if the desired utilities change in the future, there would be no need to update the local cameras. All we need to do is to update the recognition algorithms in the remote computing center.

Whereas, it is important to note that even though our IRL-VGAN approach aims to preserve all unspecified attributes, strictly speaking, not all data utilities can be preserved. For example, any utility information tied directly to identity (e.g., gender) could be altered in the privacy-protected data. Only the utility information that is completely independent or very weakly dependent on identity (e.g., activity) would be preserved. In addition, our proposed approaches are based on data-driven algorithms. Therefore, if a utility is inadequately represented in the training set, it is possible that our proposed methods may fail to learn to preserve that utility in the generated representations/images.

It is also worthwhile to mention that, in our experiments we have only demonstrated this capability for 2 utilities. This is primarily because we aimed for a proof-of-concept demonstration and the datasets that we worked with only provide labels for at most two distinct utilities. One promising future direction for further exploration

is to validate the capability of the proposed approach to simultaneously preserve more than two utilities. This could be done, for example, by conducting experiments on large-scale datasets which have multiple face attributes labeled such as CelebA (Liu et al., 2015).

Chapter 4

Concluding Remarks and Outlook

This thesis proposed two distinct approaches for visual analytics that protect user's (visual) privacy while preserving utility for inference task(s). It was motivated by the desire to have reliable and accurate visual analytics methodologies without invasion of privacy, which are critical to achieving the expected benefits of a smart room.

The first approach addresses privacy concerns by significantly reducing camera resolution (e.g., 12×16 pixels). It is a low-complexity approach in terms of sensing modality, data processing and transmission. We conducted proof-of-concept studies for three recognition tasks at extremely low resolutions, namely, human head pose estimation, indoor occupant localization and human action recognition. The impact of spatial resolution on the preservation of privacy and data utility was investigated. Both classical machine learning and modern deep learning algorithms were leveraged to maximize task performance. The empirical results demonstrated that using eLR cameras is suitable for scenarios does not require high accuracy.

The second approach took advantage of the recent advancements in representation learning to design an identity-invariant image representation that also permits synthesis of utility-equivalent realistic image. This approach relies on HR cameras and high-complexity deep learning techniques. Specifically, we proposed two novel models, namely, PPRL-VGAN and IRL-VGAN. Quantitative and qualitative results from a broad set of experiments demonstrate that the generated representations and images from our models largely eliminate the original identity information while accu-

rately preserve the utility information, and therefore are suitable for scenarios calling for high accuracy. Once trained, both PPRL-VGAN and IRL-VGAN are also generative as they enable synthesis of a realistic image having a desired value for the specified factor. Beyond their application to privacy-preserving visual analytics, they also can be used to generate realistic avatars for animation and gaming. In addition, as the proposed models are capable of generating images from a particular class, they can be applied to targeted data augmentation. Last but not least, with necessary modifications on neural network architectures, our models could potentially be used to capture human dynamic information from 3D videos. Thus, they could be used in conjunction with a motion capture system to animate photo-realistic digital character models that can take up the appearance of any person desired.

4.1 Future Directions

This thesis is a first step towards developing privacy-preserving visual analytics methodologies for smart rooms. There exist interesting directions that can be pursued based on our works.

Regarding the eLR-based approaches, we have not considered multiple subjects in the field of view of the sensors. Additionally, our algorithms work under the assumption that ground-truth labels can be attained during the training process, however these measurements can be difficult to obtain in practice. Therefore, extending our approaches to more complicated indoor scenarios and proposing new semi-supervised/unsupervised approaches would be two interesting research directions. Another plausible direction is to incorporate other sensing modalities into the system.

In terms of the representation learning methods, we have only focused on 2D images. It would be interesting to generalize our proposed models to action recognition

based on video where time and action dynamics should be tackled. Having a video as input, we will need to modify the structures of both generator and discriminator of our models to better capture the dynamic information. One possible solution is to leverage Long Short Term Memory (LSTM) type recurrent neural networks (Hochreiter and Schmidhuber, 1997) which are known to be efficient in modeling dynamic temporal behavior. 3D convolution is another possible option given its recent success in action recognition (Hara et al., 2018).

With the rise of advanced generative image models such those based on GANs or VAEs, synthesized images have become photo-realistic to the point where it is often hard for lay people to reliably distinguish them from real images. This could significantly complicate efforts to detect fraudulent impersonation or fake news and even cast serious doubt about the use of image and video data as forensic evidence in courts. Therefore, an interesting future research topic is to develop forensic tools to reliably detect images generated from GANs/VAEs. Given that real images are generated by imaging devices while the synthesized ones are created through a very different pipeline with convolution, activation, etc., in deep neural networks, they are likely to have different statistical properties. One recent work proposed to distinguish real images from synthesized images by comparing statistics of certain color components (Li et al., 2018a). Another possible direction is to leverage the photo-response non-uniformity (PSNU) pattern (Lukáš et al., 2006), which can be used as a device fingerprint and is very difficult to mimic even for a deep neural network.

References

Abe, S. (2005). *Support vector machines for pattern classification*, volume 2. Springer.

Ahn, B., Park, J., and Kweon, I. S. (2015). Real-time head orientation from a monocular camera using deep neural network. In *Computer Vision–ACCV 2014*, pages 82–96. Springer.

Aifanti, N., Papachristou, C., and Delopoulos, A. (2010). The mug facial expression database. In *2010 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4. IEEE.

Aneja, D., Colburn, A., Faigin, G., Shapiro, L., and Mones, B. (2016). Modeling stylized character expressions via deep learning. In *Asian Conference on Computer Vision*, pages 136–153. Springer.

Anwar Saeed, A. A.-H. (2015). Boosted human head pose eestimation using kinect camera. In *2015 International Conference on Image Processing*. IEEE.

Ariz, M., Bengoechea, J. J., Villanueva, A., and Cabeza, R. (2016). A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. *Computer Vision and Image Understanding*, 148:201–210.

Aubry, M., Maturana, D., Efros, A. A., Russell, B. C., and Sivic, J. (2014). Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769.

Badii, A., Al-Obaidi, A., Einig, M., and Ducournau, A. (2013). Holistic privacy impact assessment framework for video privacy filtering technologies. *Signal & Image Processing*, 4(6):13.

Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. (2017). Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2745–2754.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., and Nayar, S. K. (2008). Face swapping: automatically replacing faces in photographs. *ACM Transactions on Graphics (TOG)*, 27(3):39.

Boyle, M., Edwards, C., and Greenberg, S. (2000). The effects of filtered video on awareness and privacy. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 1–10. ACM.

Brkic, K., Sikiric, I., Hrkac, T., and Kalafatic, Z. (2017). I know that person: Generative full body and face de-identification of people in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, volume 1, page 4.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). Easyliving: Technologies for intelligent environments. In *Handheld and ubiquitous computing*, pages 12–29. Springer.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

Butler, D. J., Huang, J., Roesner, F., and Cakmak, M. (2015). The privacy-utility tradeoff for remotely teleoperated robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 27–34. ACM.

Chaaraoui, A. A., Climent-Pérez, P., and Flórez-Revuelta, F. (2012). An efficient approach for multi-view human action recognition based on bag-of-key-poses. In Salah, A. A., Ruiz-del Solar, J., Meriçli, Ç., and Oudeyer, P.-Y., editors, *Human Behavior Understanding*, volume 7559, pages 29–40. Springer Berlin Heidelberg.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009.*, pages 1932–1939. IEEE.

Chen, J. (2017). Boston University: Privacy-Preserving Smart-Room Analytics. vip.bu.edu/projects/vsns/privacy-smartroom.

Chen, J. (2018). vip.bu.edu/projects/vsns/privacy-smartroom/facial-expression-vgan.

Chen, J., Konrad, J., and Ishwar, P. (2018). Vgan-based image representation learning for privacy-preserving facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1570–1579.

Chen, J., Konrad, J., and Ishwar, P. (2019). A cyclically-trained adversarial network for invariant representation learning. *arXiv preprint arXiv:1906.09313*.

Chen, J., Wu, J., Konrad, J., and Ishwar, P. (2017). Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 139–147. IEEE.

Chen, J., Wu, J., Richter, K., Konrad, J., and Ishwar, P. (2016). Estimating head pose orientation using extremely low resolution images. In *2016 IEEE Southwest symposium on image analysis and interpretation (SSIAI)*, pages 65–68. IEEE.

Cheng, G., Zhou, P., and Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415.

Cheung, B., Livezey, J. A., Bansal, A. K., and Olshausen, B. A. (2014). Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*.

Chollet, F. (2015). keras. `https://github.com/fchollet/keras`.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999.

Dai, J., Wu, J., Saghafi, B., Konrad, J., and Ishwar, P. (2015). Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 68–76.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

Deak, G., Curran, K., and Condell, J. (2012). A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16):1939–1954.

Desjardins, G., Courville, A., and Bengio, Y. (2012). Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*.

Di, X. and Patel, V. M. (2017). Face synthesis from visual attributes via sketch using conditional vaes and gans. *arXiv preprint arXiv:1801.00077*.

Drouard, V., Ba, S., Evangelidis, G., Deleforge, A., and Horaud, R. (2015). Head pose estimation via probabilistic high-dimensional regression. In *IEEE International Conference on Image Processing*.

Dufaux, F. (2011). Video scrambling for privacy protection in video surveillance: recent results and validation framework. *SPIE Defense, Security, and Sensing*, pages 806302–806302.

Dufaux, F. and Ebrahimi, T. (2006). Scrambling for video surveillance with privacy. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 160–160. IEEE.

Dufaux, F. and Ebrahimi, T. (2008). Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1168–1174.

Edwards, H. and Storkey, A. (2015). Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*.

Elgammal, A. and Lee, C.-S. (2004). Separating style and content on a nonlinear manifold. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE.

Engelbrecht, K.-P., Schmiedeke, S., Quade, M., and Moller, S. (2015). Designing new experiences in the smart home: Multi-camera person localization framework to document predefined situations. In *2015 International Conference on Intelligent Environments (IE)*, pages 1–8. IEEE.

Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., and Toft, T. (2009). Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer.

Erturk, S. (2007). Multiplication-free one-bit transform for low-complexity block-based motion estimation. *IEEE Signal Processing Letters*, 14(2):109–112.

Fanelli, G., Dantone, M., Gall, J., Fossati, A., and Van Gool, L. (2013). Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Feng, G., Yang, Y., Guo, X., and Wang, G. (2015). A smart fiber floor for indoor target localization. *Pervasive Computing, IEEE*, 14(2):52–59.

Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Adam, H., Neven, H., and Vincent, L. (2009). Large-scale privacy protection in google street view. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2373–2380. IEEE.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Ghahramani, Z. (1995). Factorial learning and the em algorithm. In *Advances in neural information processing systems*, pages 617–624.

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gourier, N., Maisonnasse, J., Hall, D., and Crowley, J. L. (2007). Head pose estimation on low resolution images. In *Multimodal Technologies for Perception of Humans*, pages 270–280. Springer.

Gross, R., Airoldi, E., Malin, B., and Sweeney, L. (2005). Integrating utility into face de-identification. In *International Workshop on Privacy Enhancing Technologies*, pages 227–242. Springer.

Gross, R., Sweeney, L., De la Torre, F., and Baker, S. (2006). Model-based face de-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 161–161. IEEE.

Gunn, S. R. et al. (1998). Support vector machines for classification and regression. *ISIS technical report*, 14.

Hadad, N., Wolf, L., and Shahar, M. (2018). A two-step disentanglement method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–780.

Hamm, J. (2017). Minimax filter: Learning to preserve privacy from inference attacks. *The Journal of Machine Learning Research*, 18(1):4704–4734.

Haque, A., Alahi, A., and Fei-Fei, L. (2016). Recurrent attention models for depth-based person identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1229–1238.

Hara, K., Kataoka, H., and Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555.

Harsh Jha, A., Anand, S., Singh, M., and Veeravasarapu, V. (2018). Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 805–820.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, pages 1415–1424.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural Networks for Machine Learning, Coursera lecture 6e*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hoffman, J., Rodner, E., Donahue, J., Darrell, T., and Saenko, K. (2013). Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Jalal, A., Uddin, M. Z., and Kim, T.-S. (2012). Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics*, 58(3).

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231.

Jia, L. and Radke, R. J. (2014). Using time-of-flight measurements for privacy-preserving tracking in a smart room. *IEEE Transactions on Industrial Informatics*, 10(1):689–696.

Johansson, F. D., Ranganath, R., and Sontag, D. (2019). Support and invertibility in domain-invariant representations. *arXiv preprint arXiv:1903.03448*.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kitahara, I., Kogure, K., and Hagita, N. (2004). Stealth vision for protecting privacy. In *IEEE Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 404–407. IEEE.

Kosba, A. E., Abdelkader, A., and Youssef, M. (2009). Analysis of a device-free passive tracking system in typical wireless environments. In *2009 3rd International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE.

Krinidis, S., Stavropoulos, G., Ioannidis, D., and Tzovaras, D. (2014). A robust and real-time multi-space occupancy extraction system exploiting privacy-preserving sensors. In *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 542–545. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Krumm, J. (2009). *Ubiquitous computing fundamentals*. CRC Press.

Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S. (2000). Multi-camera multi-person tracking for easyliving. In *Third IEEE International Workshop on Visual Surveillance*, pages 3–10. IEEE.

Krumm, J. and Horvitz, E. (2004). Locadio: Inferring motion and location from wi-fi signal strengths. In *mobiquitous*, pages 4–13.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE.

Kunze, K. and Lukowicz, P. (2014). Sensor placement variations in wearable activity recognition. *IEEE Pervasive Computing*, 13(4):32–41.

Kuroiwa, K., Fujiyoshi, M., and Kiya, H. (2007). Codestream domain scrambling of moving objects based on dct sign-only correlation for motion jpeg movies. In *IEEE International Conference on Image Processing*, volume 5, pages V–157. IEEE.

Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., et al. (2017). Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2015). Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*.

Larumbe, A., Ariz, M., Bengoechea, J. J., Segura, R., Cabeza, R., and Villanueva, A. (2017). Improved strategies for hpe employing learning-by-synthesis approaches. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1545–1554.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, K., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698.

Li, H., Li, B., Tan, S., and Huang, J. (2018a). Detection of deep network generated images using disparities in color components. *arXiv preprint arXiv:1808.07276*.

Li, Y., Gong, M., Tian, X., Liu, T., and Tao, D. (2018b). Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, Y., Swersky, K., and Zemel, R. (2014). Learning unbiased features. *arXiv preprint arXiv:1412.5244*.

Lin, T.-Y., RoyChowdhury, A., and Maji, S. (2015). Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457.

Liu, C. et al. (2009). *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157.

Lui, Y. M., Bolme, D., Draper, B. A., Beveridge, J. R., Givens, G., Phillips, P. J., et al. (2009). A meta-analysis of face recognition covariates. *Biometrics: Theory, Applications, and Systems*, pages 1–8.

Lukáš, J., Fridrich, J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Martínez-Ponte, I., Desurmont, X., Meessen, J., and Delaigle, J.-F. (2005). Robust human face hiding ensuring privacy. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, volume 4.

Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., and LeCun, Y. (2016). Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048.

Moussa, M. and Youssef, M. (2009). Smart cevices for smart environments: Device-free passive detection in real environments. In *IEEE International Conference on Pervasive Computing and Communications*, pages 1–6. IEEE.

Murphy-Chutorian, E. and Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626.

Neustaedter, C. and Greenberg, S. (2003). The design of a context-aware home media space for balancing privacy and awareness. In *UbiComp 2003: Ubiquitous Computing*, pages 297–314. Springer.

Neustaedter, C., Greenberg, S., and Boyle, M. (2006). Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(1):1–36.

Newton, E. M., Sweeney, L., and Malin, B. (2005). Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org.

Padilla-López, J. R., Chaaraoui, A. A., and Flórez-Revuelta, F. (2015). Visual privacy protection methods: A survey. *Expert Systems with Applications*, 42(9):4177–4195.

Park, E., Han, X., Berg, T. L., and Berg, A. C. (2016). Combining multiple sources of knowledge in deep cnns for action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE.

Park, S. and Kautz, H. A. (2008). Privacy-preserving recognition of activities in daily living from multi-view silhouettes and rfid-based training. In *AAAI Fall Symposium: AI in Eldercare: New Solutions to Old Problems*, pages 70–77.

Pittaluga, F. and Koppal, S. J. (2015). Privacy preserving optics for miniature vision sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 314–324.

Pittaluga, F. and Koppal, S. J. (2016). Pre-capture privacy for small vision sensors. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2215–2226.

Plagemann, C., Ganapathi, V., Koller, D., and Thrun, S. (2010). Real-time identification and localization of body parts from depth images. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3108–3113. IEEE.

Point, N. (2011). Optitrack. *Natural Point, Inc.,[Online]. Available: http://www. naturalpoint. com/optitrack/.*

Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM.

Raval, N., Machanavajjhala, A., and Cox, L. P. (2017). Protecting visual secrets using adversarial nets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1329–1332. IEEE.

Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180.

Robertson, N. and Reid, I. (2006). Estimating gaze direction from low-resolution faces in video. In *European Conference on Computer Vision (ECCV)*, pages 402–415.

Roeper, D., Chen, J., Konrad, J., and Ishwar, P. (2016). Privacy-preserving, indoor occupant localization using a network of single-pixel sensors. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 214–220. IEEE.

Ryoo, M. S., Rothrock, B., and Fleming, C. (2016). Privacy-preserving egocentric activity recognition from extreme low resolution. *arXiv preprint arXiv:1604.03196*.

Ryoo, M. S., Rothrock, B., and Matthies, L. (2015). Pooled motion features for first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 896–904.

Sadeghi, A.-R., Schneider, T., and Wehrenberg, I. (2009). Efficient privacy-preserving face recognition. In *International Conference on Information Security and Cryptology*, pages 229–244. Springer.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242.

Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, Y.-L., Ekin, A., Connell, J., Shu, C. F., and Lu, M. (2005). Enabling video privacy through computer vision. *IEEE Security & Privacy*, 3(3):50–57.

Shang, W. and Sohn, K. (2019). Attentive conditional channel-recurrent autoencoding for attribute-conditioned face synthesis. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1533–1542. IEEE.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813.

Simonyan, K. and Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576.

Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smola, A. and Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Soatto, S. and Chiuso, A. (2014). Visual representations: Defining properties and deep approximations. *arXiv preprint arXiv:1411.7676*.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491.

Sokolic, J., Qiu, Q., Rodrigues, M. R., and Sapiro, G. (2017). Learning to succeed while teaching to fail: Privacy in closed machine learning systems. *arXiv preprint arXiv:1705.08197*.

Steggles, P. and Gschwind, S. (2005). The ubisense smart space platform. *Adjunct Proceedings of the Third International Conference on Pervasive Computing*, pages 73–76.

Szabó, A., Hu, Q., Portenier, T., Zwicker, M., and Favaro, P. (2017). Challenges in disentangling independent factors of variation. *arXiv preprint arXiv:1711.02245*.

Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283.

Tews, E., Walde, J., and Weiner, M. (2011). Breaking dvb-csa. *Western European Workshop on Research in Cryptology*, page 41.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497. IEEE.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. (2016). Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798.

Vedaldi, A. and Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM.

Wang, B., Liang, W., Wang, Y., and Liang, Y. (2013). Head pose estimation with combined 2d sift and 3d hog features. In *2013 Seventh International Conference on Image and Graphics (ICIG)*, pages 650–655. IEEE.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558.

Wang, W., Cui, Z., Chang, H., Shan, S., and Chen, X. (2014). Deeply coupled auto-encoder networks for cross-view classification. *arXiv preprint arXiv:1402.2031*.

Wang, W., Vong, C.-M., Yang, Y., and Wong, P.-K. (2017). Encrypted image classification based on multilayer extreme learning machine. *Multidimensional Systems and Signal Processing*, 28(3):851–865.

Wang, Z., Chang, S., Yang, Y., Liu, D., and Huang, T. S. (2016). Studying very low resolution recognition using deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4792–4800.

Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102.

Weinland, D., Özuysal, M., and Fua, P. (2010). Making action recognition robust to occlusions and viewpoint changes. In *European Conference on Computer Vision*.

Winkler, T., Erdélyi, A., and Rinner, B. (2014). Trusteye. m4: protecting the sensornot the camera. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 159–164. IEEE.

Wirt, K. (2004). Fault attack on the dvb common scrambling algorithm. Cryptology ePrint Archive, Report 2004/289.

Wu, J., Ishwar, P., and Konrad, J. (2016). Two-stream cnns for gesture-based verification and identification: Learning user style. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Wu, Z., Wang, Z., Wang, Z., and Jin, H. (2018). Towards privacy-preserving visual recognition via adversarial training: A pilot study. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–624.

Xia, L., Chen, C.-C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. IEEE.

Xie, Q., Dai, Z., Du, Y., Hovy, E., and Neubig, G. (2017). Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pages 585–596.

Ye, G. (2010). Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, 31(5):347–354.

Yonetani, R., Naresh Boddeti, V., Kitani, K. M., and Sato, Y. (2017). Privacy-preserving visual learning using doubly permuted homomorphic encryption. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2040–2050.

Youssef, M., Mah, M., and Agrawala, A. (2007). Challenges: device-free passive localization for wireless environments. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 222–229. ACM.

Yu, C.-R., Wu, C.-L., Lu, C.-H., and Fu, L.-C. (2006). Human localization via multi-cameras and floor sensors in smart home. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 3822–3827. IEEE.

Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. (2015). Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*.

Zeng, W. and Lei, S. (2003). Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia*, 5(1):118–129.

Zhang, C., Rui, Y., and He, L.-w. (2006). Light weight background blurring for video conferencing applications. In *2006 IEEE International Conference on Image Processing*, pages 481–484. IEEE.

Zhang, C., Tian, Y., and Capezuti, E. (2012). Privacy preserving automatic fall detection for elderly using rgbd cameras. *Computers Helping People with Special Needs*, pages 625–633.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Ziad, M. T. I., Alanwar, A., Alzantot, M., and Srivastava, M. (2016). Cryptoimg: Privacy preserving processing over encrypted images. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 570–575. IEEE.

CURRICULUM VITAE

Jiawei Chen
8 St Marys St, Boston, MA 02215, USA
garychen@bu.edu

Education

Boston University, Boston, MA, USA *Sep.2015 - Present*
Ph.D Candidate in Engineering in Electrical and Computer Engineering

Duke University, Durham, NC, USA *Sep.2013 - May. 2015*
M.Eng. in Electrical and Computer Engineering

Harbin Institute of Technology, Harbin, China *Sep.2009 - Jul. 2013*
B.Eng. in Electrical and Computer Engineering

Professional Experience

Research Assistant *Sep.2015 - Present*
Visual Information Processing Lab, ECE Department, Boston University

Teaching Assistant *Sep.2016 - May 2017*
Department of Electrical and Computer Engineering, Boston University

Research Intern *Jun. 2018 - Sep. 2018*
Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA

Assistant Engineer Intern *Jul. 2013- Sep. 2013*
Twenty First Century Aerospace Technology Co, Ltd, Beijing, China

Honors and Awards

BU Data Science Award *2016*

Merit Student of Harbin Institute of Technology *2011*

Scholarship of Academic Excellence, HIT, *2009, 2011, 2012*

Publications

- **J. Chen**, J. Konrad, and P. Ishwar. "A Cyclically-Trained Adversarial Network for Invariant Representation Learning." submitted to *IEEE International Conference on Computer Vision (ICCV)*, Oct., 2019.

- H. Kawai, **J. Chen**, P. Ishwar, J. Konrad. "VAE/WGAN-Based Image Representation Learning for Pose-Preserving Seamless Identity Replacement In Facial Images". submitted to *IEEE International Workshop on Machine Learning for Signal Processing*, Oct., 2019.

- **J. Chen**, J. Konrad, and P. Ishwar. "VGAN-Based Image Representation Learning for Privacy-Preserving Facial Expression Recognition". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June, 2018.

- **J. Chen**, J. Wu, J. Konrad, and P. Ishwar. "Semi-Coupled Two-Stream Fusion ConvNets for Action Recognition at Extremely Low Resolutions". *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar., 2017.

- J. Wu, **J. Chen**, P. Ishwar, J. Konrad. "Two-Stream CNNs for Gesture-Based Verification and Identification: Learning User Style." *Bhanu B., Kumar A. (eds) Deep Learning for Biometrics. Advances in Computer Vision and Pattern Recognition.* Springer, Cham

- D. Roeper, **J. Chen**, J. Konrad, and P. Ishwar. "Privacy-preserving, indoor occupant localization using a network of single-pixel sensors". *13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug., 2016.

- **J. Chen**, J. Wu, J. Konrad, and P. Ishwar. "Estimating head pose orientation using extremely low resolution images". *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, Mar., 2016.