UTILITY-PRESERVING FACE REDACTION AND CHANGE DETECTION FOR SATELLITE IMAGERY

by

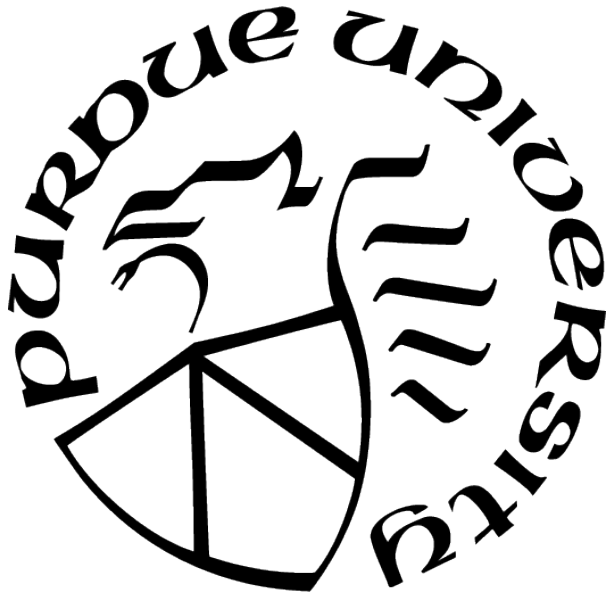
Hanxiang Hao

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Electrical and Computer Engineering

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Edward J. Delp, Chair

School of Electrical and Computer Engineering

Dr. Amy R. Reibman

School of Electrical and Computer Engineering

Dr. Mary L. Comer

School of Electrical and Computer Engineering

Dr. Fengqing M. Zhu

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Petroulis

To my dear wife, Yuting.

ACKNOWLEDGMENTS

I would like to thank my doctoral advisor Professor Edward J. Delp for his support and guidance throughout my entire PhD life. This experience with Prof. Delp not only gave me the knowledge to support my future career, but also changed my mindset to make me think like a graduate student and a research scholar. Thinking like a research scholar can assist my academic career, but most importantly it can also help me solve the problems and obstacles in my future life. I really appreciate the help from Prof. Delp and I am really proud and grateful to have this experience working with him. I would also like to thank Professor Amy R. Reibman for her help on my research. I really enjoyed our weekly meetings with Prof. Reibman, since she always had great ideas and suggestions, which always led to fruitful research outcomes. I would like to thank Professor Mary L. Comer for her insightful feedback and suggestions on my research, especially, for the knowledge I learned from her Random Variables and Signals course that I took in my first semester. It indeed built the foundation on my entire PhD research. I would like to thank Professor Fengqing M. Zhu for her help and support on my research as well. Her lectures and the projects from the Digital Video System course gave me a lot of insights to support my research and future career.

Besides the help from my PhD committee, I would also like to thank all the former and current members of the Video and Image Processing Laboratory (VIPER) for their help and companionship, especially, Dr. David Güera, Dr. Daniel Mas, Dr. Sri Kalyan Yarlagadda, Dr. Khalid Tahboub, Dr. Chichen Fu, Dr. Shuo Han, Dr. Yuhao Chen, Dr. Shaobo Fang, Dr. Soonam Lee, Dr. David Joon Ho, Dr. Javier Ribera Prat, Sriram Baireddy, Emily Bartusiak, Mridul Gupta, János Horváth, Ruiting Shao, Yifan Zhao, Justin Yang, Enyu Cai, Jiaqi Guo, Changye Yang, Han Hu, Yue Han, Liming Wu, Alain Chen, Ziyue (Alan) Xiang, Qingshuang Chen, and Di Chen.

Finally, I would like to thank my family, especially, my wife Yuting Li, my father Xueru Hao, and my mother Airong Sun for their unconditional love and support. Their love and support make all of the impossible possible.

The material in Chapter 2 is based on the research sponsored by the Department of Homeland Security (DHS) under agreement number 70RSAT18FR0000161. The material in Chapter 3 is based on the research sponsored by Lockheed Martin Space. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DHS, the U.S. Government, or Lockheed Martin Space.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory network
RNN	Recurrent Neural Network
GAN	Generative Adversarial Network
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
JPEG	Joint Photographic Experts Group
DCT	Discrete Cosine Transform
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
RGB	Red, Green, and Blue
SSIM	Structural Similarity Index
FID	Fréchet Inception Distance
MSE	Mean Square Error
MAE	Mean Absolute Error
AP	Average Precision
ACM	Affine Combination Module
AdaIN	Adaptive Instance Normalization
SPADE	Spatially-Adaptive Normalization
CSD	Consistency-Based Semi-Supervised Object Detection
SSL	Semi-Supervised Learning
DSM	Digital Surface Model
RPN	Region Proposal Network
GSD	Ground Sample Distance

ABSTRACT

Face redaction is needed by law enforcement and mass media outlets to guarantee privacy. In this thesis, a performance analysis of several face redaction/obscuration methods, such as blurring and pixelation is presented. The analysis is based on various threat models and obscuration attackers to achieve a comprehensive evaluation. We show that the traditional blurring and pixelation methods cannot guarantee privacy. To provide a more secured privacy protection, we propose two novel obscuration methods that are based on the generative adversarial networks. The proposed methods not only remove the identifiable information, but also preserve the non-identifiable facial information (as known as the utility information), such as expression, age, skin tone and gender.

We also propose methods for change detection in satellite imagery. In this thesis, we consider two types of building changes: 2D appearance change and 3D height change. We first present a model with an attention mechanism to detect the building appearance changes that are caused by natural disasters. Furthermore, to detect the changes of building height, we present a height estimation model that is based on building shadows and solar angles without relying on height annotation. Both change detection methods require good building segmentation performance, which might be hard to achieve for the low-quality images, such as off-nadir images. To solve this issue, we use uncertainty modeling and satellite imagery metadata to achieve accurate building segmentation for the noisy images that are taken from large off-nadir angles.

1. INTRODUCTION

1.1 Utility-Preserving Face Redaction

Face redaction or obscuration techniques are often used by law enforcement and mass media outlets to provide a privacy protection. As shown in Figure 1.1, law enforcement agencies use these identity obscuration techniques to avoid exposing the identities of bystanders or officers. Similarly, Google StreetView also relies on these techniques to protect the privacy of the identities presented in the scene. Figure 1.2 shows several face obscuration methods, including pixelation, blurring, and blacking out. Blacking out the entire face region is rarely used in real-world applications, because its visual effect is unpleasant, especially if there are many faces in the scene that need to be obscured.

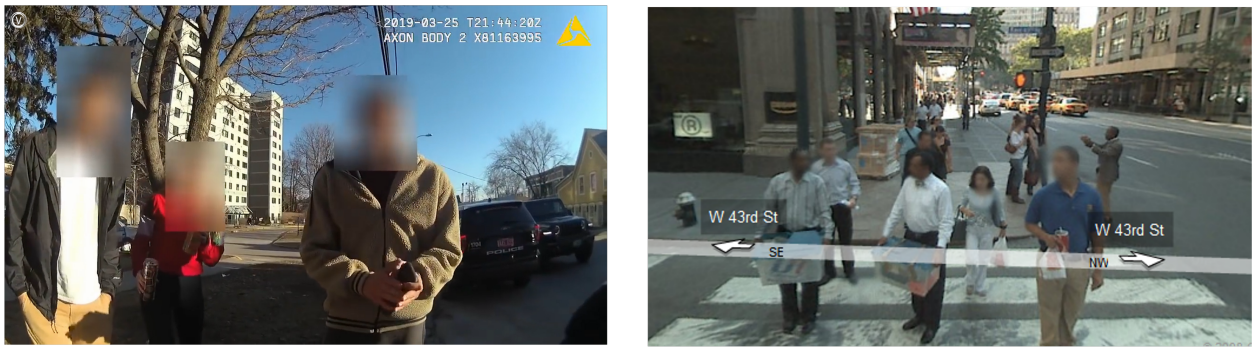


Figure 1.1. Common scenarios that need for face obscuration techniques. Left: the video footage from police body-worn camera; right: Google StreetView.

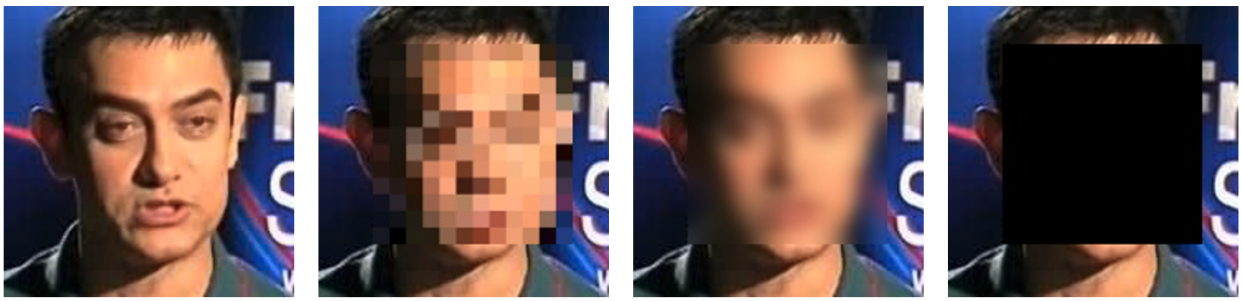


Figure 1.2. Examples of face obscuration techniques. From left to right: original, pixelation, blurring, and blacking out.

Although these obscured identities are hard for us to recognize, advanced face recognition system might still be able to identity them. Due to recent advances in the field of machine learning, especially deep learning, obscuration methods such as blurring and pixelation are not guaranteed to conceal identity. Recent work [1]–[3] shows that advanced machine learning approaches are still able to recognize the identity after the common obscuration methods, such as blurring and pixelation. Figure 1.3 shows the results obtained from such machine learning-based attacker. We show that the attacker can still identify the correct identities after applying the aforementioned obscuration methods. The obscured face is shown on the left side of each example, while the eight query faces are listed on the right side of each example. The bar next to each query face indicates the feature distance of the query face to the obscured face as computed by the attacker. By comparing the feature distances, the obscured face still has the closest distance to the correct identity as highlighted by the green distance bar. Hence, a better face obscuration method that can guarantee identity obscuration is needed. In this thesis, we present a utility-preserving generative model, UP-GAN, that is able to provide an effective face obscuration, while preserving facial utility. By utility-preserving we mean preserving facial features that do not reveal identity, such as age, gender, skin tone, pose, and expression. The proposed method is not only able to remove the identifiable information, but also keep the utility information intact.

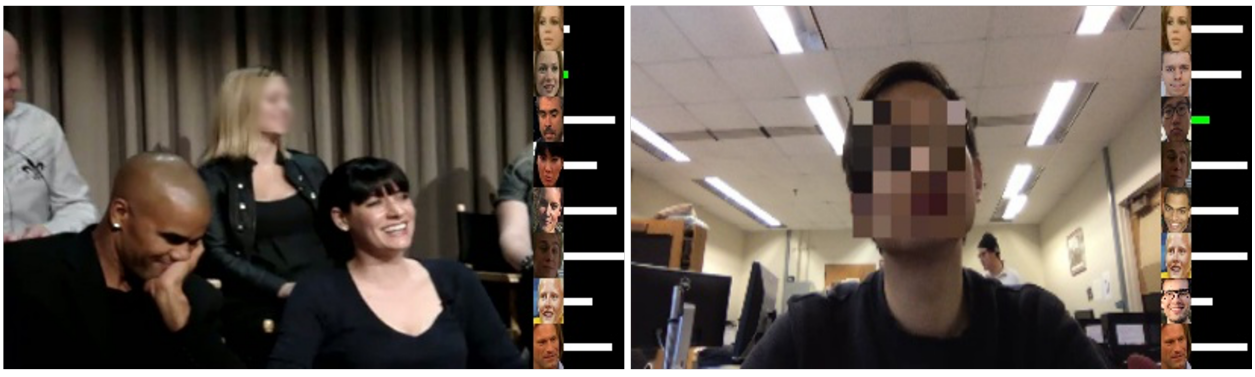


Figure 1.3. Machine learning-based attacker can still recognize the correct identities from the obscured images. The image on the left shows the result obscured by Gaussian blurring. The image on the right shows the result obscured by pixelation. In both cases, the machine learning-based attacker correctly identifies the obscured identities.

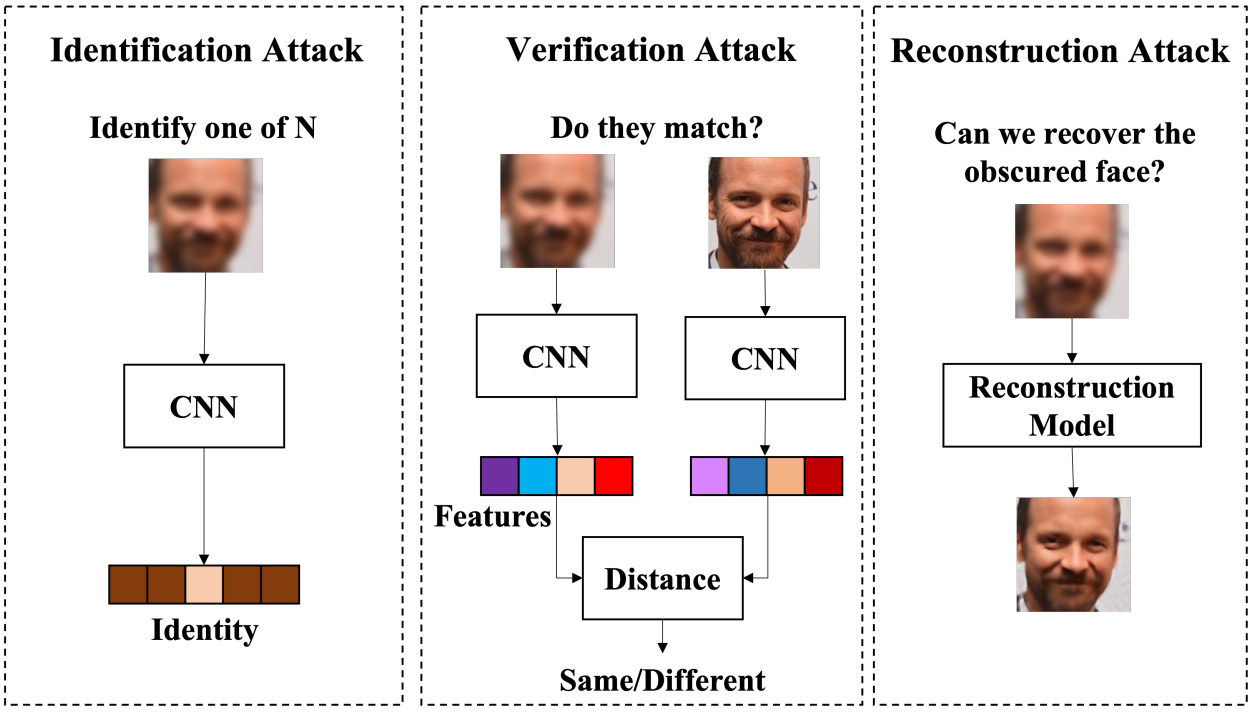


Figure 1.4. Proposed face attack scenarios. To provide a comprehensive analysis, we present three attack scenarios: identification attack, verification attack, and reconstruction attack.

Although in Figure 1.3, we show that machine learning-based attacker is able to identify the obscured identity, a more systematical analysis is still needed to examine these obscuration methods with advanced machine learning and deep learning-based attackers. In order to examine the effectiveness of the proposed method as well as other common obscuration methods, we provide a systematically measurement that is based on recent deep learning models to assess the face obscuration performance of a given technique. In this thesis, we measure the obscuration performance of eight obscuration techniques including common blurring and pixelation methods and machine learning based methods. We do so by attacking the obscured faces in three scenarios: obscured face identification, verification, and reconstruction as shown in Figure 1.4. Face identification attack can be implemented as a standard identity classification task, which has been analyzed by the previous work [1]–[3]. By mapping faces to known identities in different threat models, we can analyze the vulnerability of each obscuration method using advanced deep learning identification methods. However, the requirement of known identities weakens this type of analysis, since query

faces usually come from unknown identities. To overcome this, we propose the verification attack scenario. Specifically, we want to measure the similarity of an unknown obscured face to clear target faces, as shown in Figure 1.3. Since it allows recognizing unseen identities, this scenario is more realistic. Lastly, a reconstruction scenario is proposed to visualize how well we can recover the true identity using the remaining information from the obscured images. Threat modeling is also considered in each attack scenario to provide a vulnerability analysis for each studied obscuration technique. Based on our evaluation, we show that the proposed UP-GAN achieves a more robust obscuration of identity than the compared methods.

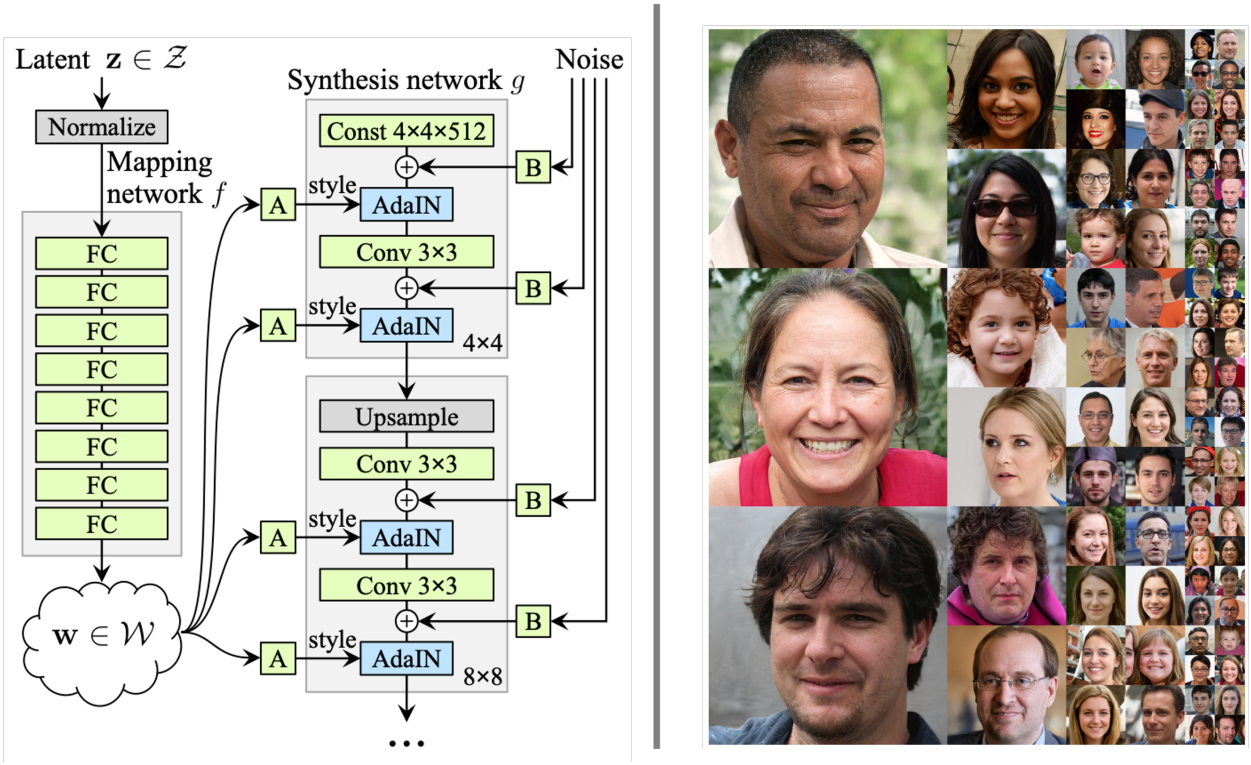


Figure 1.5. StyleGAN model structure (left) and generated faces (right). As proposed in [4], StyleGAN is able to produce photo-realistic face images with high resolution (up to 1024×1024).

With the fast development of generative adversarial networks (GANs), recent methods, such as ProGAN [5] and StyleGAN [4], can generate photo-realistic synthetic face images. As shown in Figure 1.5, StyleGAN [4] is able to produce photo-realistic face images with high image resolution (up to 1024×1024). These high-resolution photo-realistic synthetic faces provide us a new way to

further improve the image quality generated by the previously proposed UP-GAN. In this thesis, we propose a face reenactment model using these high-fidelity synthetic faces to achieve a utility-preserving face obscuration. Given a synthetic face with the target utility information obtained from the original face image (*e.g.*, age, gender, and skin tone), the proposed face reenactment model can animate the synthetic face with the target facial expression and head pose for face obscuration. By doing so, we can decouple the process of generating synthetic identity and editing the facial expression/head pose. And then, we can enforce the model only focusing on producing photo-realistic face editing (*i.e.*, face reenactment) given the synthetic face image. Based on our experiment analysis, we show that the proposed method achieves photo-realistic face reenactment compared to the previous methods.

1.2 Change Detection For Satellite Imagery

Natural disasters cause severe damage to our society. To save lives and reduce damage, we need an accurate situational information and a fast, effective response. Widely available, high resolution satellite images enable emergency responders to estimate locations, causes, and severity of damage. Quickly and accurately analyzing the extensive amount of satellite imagery available, though, requires an automatic approach. In this thesis, we propose a change detection model – a multi-class deep learning model with an attention mechanism – to assess damage levels of buildings given a pair of satellite images depicting a scene before and after a disaster, as shown in Figure 1.6. We evaluate the proposed method on xView2 dataset, a large-scale building damage assessment dataset. xView2 dataset defines four levels of building change:

- *No Damage*: Undisturbed and no sign of water, structural damage, shingle damage, or burn marks.
- *Minor Damage*: Building partially burnt, water surrounding the structure, volcanic flow nearby, roof elements missing, or visible cracks.
- *Major Damage*: Partial wall or roof collapse, encroaching volcanic flow, or the structure is surrounded by water or mud.

- *Destroyed*: Structure is scorched, completely collapsed, partially or completely covered with water or mud, or no longer present.

Examples of the buildings with different damage levels can be found in Figure 1.7. The proposed method needs to compare the difference between the two input images in order to accurately segment the building region and assign the correct damage level. Based on our experiment analysis, we show that the proposed approach achieves accurate damage scale classification and building segmentation results, simultaneously.

Figure 1.6. An example of building damage assessment task given a pair of satellite images taken from a scene before and after a disaster. Left: before disaster. Right: after disaster.

However, only considering the changes of building appearance sometime is not enough to capture all information required for a change detection application. In many real-world applications, detecting the change of building height is also important. Therefore, in this thesis, we also propose a height estimation model to detect the changes of building height. Obtaining the ground truth building height requires the access of LiDAR sensor to obtain the digital surface model of a given scene as shown in Figure 1.8, which is not available in many situations. To solve this issue, the proposed method is designed to estimate building height based on building shadows and satellite imagery metadata, such as solar elevation and azimuth angles without relying on height

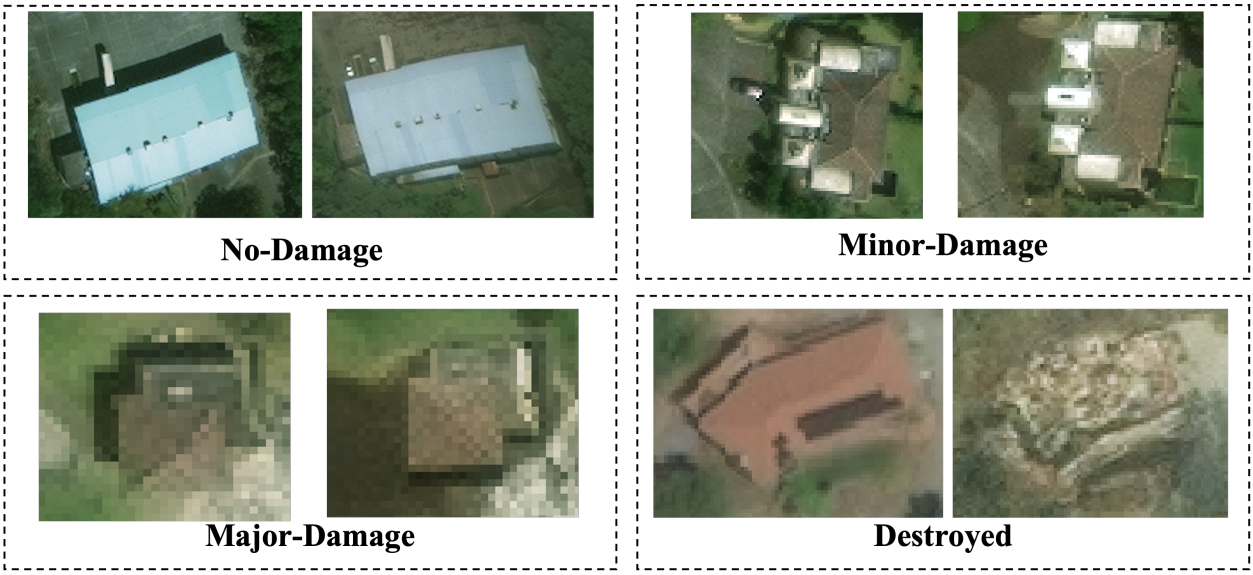


Figure 1.7. Examples of different damage levels defined in [6].

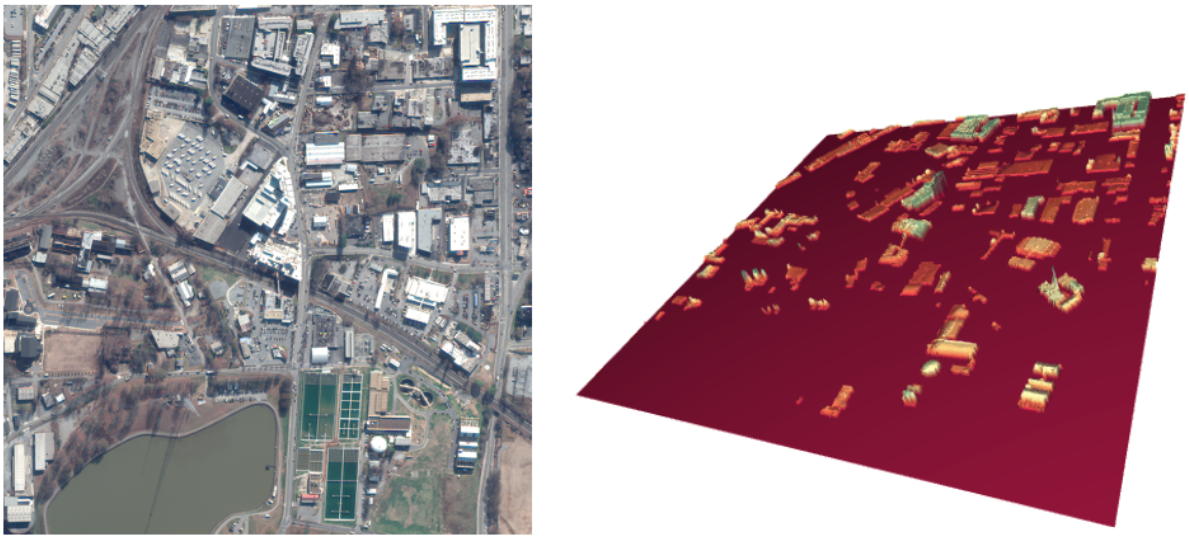


Figure 1.8. Digital surface model (right) obtained from Urban Semantic 3D dataset [7].

annotations. More specifically, our method contains three steps: 1) supervised building instance detection; 2) semi-supervised shadow instance detection; 3) unsupervised building height estimation. Due to the widely available labels for building detection, we use a supervised instance segmentation method to obtain building instances. The shadow instance detection task aims to find shadow instances paired with building instances. Given a satellite image and its detected building

instance mask, the model outputs the shadow instances associated with the corresponding building instances. Because of the lack of building shadow annotation, we train the method in a semi-supervised manner that requires fewer training labels. Given the building and shadow association, we can estimate the building height with satellite metadata, such as solar azimuth and elevation angles, and ground sample distance. Building height estimation is done by maximizing the overlap between the theoretical shadow region given a query height and the detected shadow instance region. We qualitatively and quantitatively show that the proposed method achieves accurate building height estimation.

Figure 1.9. Satellite images with low off-nadir angle (left) and high off-nadir angle (right). The image with high off-nadir angle is noisy and blurry, which is challenging for the existing methods to provide an accurate segmentation.

Both proposed methods for detecting 2D appearance change and 3D height change require accurate building segmentation. Most existing segmentation methods focus on the case where the images are taken from directly overhead (*i.e.*, low off-nadir/viewing angle). These methods often fail to provide accurate results on satellite images with larger off-nadir angles due to the higher noise level and lower spatial resolution. As shown in Figure 1.9, compared to the image with low off-nadir angle, the image with high off-nadir angle is noisier and blurrier. In this thesis, we propose a method that is able to provide accurate building segmentation for satellite imagery captured

from a large range of off-nadir angles. Based on Bayesian deep learning, we explicitly design our method to learn the data noise via aleatoric and epistemic uncertainty modeling. Satellite image metadata (*e.g.*, off-nadir angle and ground sample distance) is also used in our model to further improve the result. We show that with uncertainty modeling and metadata injection, our method achieves better performance compared to the baseline method, especially for the noisy images taken from large off-nadir angles.

1.3 Contributions Of This Thesis

In this thesis, we developed new methods for face redaction and satellite imagery change detection. The main contributions of the thesis are listed as follows:

- Utility-Preserving Face Redaction

    1. A performance analysis of face obscuration approaches is proposed.
    2. The analysis is based on three attack scenarios: obscured face identification, verification, and reconstruction.
    3. We analyze these attacks based on two widely used deep learning models, VGG19 [8] and ResNet50 [9] in different threat model conditions.
    4. We show that the traditional obscuration methods, such as blurring and pixelation can not guarantee privacy protection.
    5. To provide a more secured privacy protection, we propose two novel obscuration methods that are based on the generative adversarial networks.
    6. With qualitative and quantitative analysis, we show that the proposed methods can not only remove the identifiable information, but also preserve the non-identifiable facial information, such as facial expression, age, skin tone and gender.

- Change Detection For Satellite Imagery

    1. We develop a multi-class deep learning model with attention technique that accurately classifies damage levels of buildings based on 2D appearance changes in satellite imagery.

2. We demonstrate that the proposed model achieves better results for building damage scale classification than other methods while simultaneously achieving accurate building segmentation results.
3. To detect the changes from 3D building height, we propose a building height estimation model.
4. The proposed method can estimate building height based on building shadows and solar angles without relying on height annotations.
5. We qualitatively and quantitatively show that the proposed method achieves accurate building height estimation.
6. To provide a more reliable building segmentation method as required in the previously proposed change detection methods, we present a model that can provide accurate building segmentation even for the low quality satellite images captured from large off-nadir angles.
7. Both uncertainty modeling and satellite imagery metadata are used in the proposed method to achieve a good building segmentation performance, especially for the noisy images taken from large off-nadir angles.

1.4 Publications Resulting From This Thesis

Conference Papers

- **Hanxiang Hao**, David Güera, Amy R. Reibman, Edward J. Delp, "A Utility-Preserving GAN for Face Obscuration", Proceedings of the International Conference on Machine Learning, Synthetic Realities: Deep Learning for Detecting AudioVisual Fakes Workshop, June 2019, Long Beach, CA.
- **Hanxiang Hao**, David Güera, János Horváth, Amy R. Reibman, Edward J. Delp, "Robustness Analysis of Face Obscuration", Proceedings of the International Conference on Automatic Face and Gesture Recognition, November 2020, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Amy R. Reibman, Edward J. Delp, "FaR-GAN for One-Shot Face Reenactment", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, AI for Content Creation Workshop, May 2020, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Emily Bartusiak, Latisha Konz, Kevin LaTourette, Michael Gribbons, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "An Attention-Based System for Damage Assessment Using Satellite Imagery", Proceedings of International Geoscience and Remote Sensing Symposium, July 2021, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Emily Bartusiak, Mridul Gupta, Kevin LaTourette, Latisha Konz, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "Building Height Estimation via Satellite Metadata and Shadow Instance Detection", Proceedings of SPIE 11729, Automatic Target Recognition XXXI, April 2021, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Kevin LaTourette, Latisha Konz, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "Improving Building Segmentation Using Uncertainty Modeling and Metadata Injection", Proceedings of ACM SIGSPATIAL: International Conference on Advances in Geographic Information Systems, November 2021, Virtual Conference.

1.5 Other Publications Not Related to This Thesis

Book Chapters

- **Hanxiang Hao**, Emily R. Bartusiak, David Güera, Daniel M. Montserrat, Sriram Baireddy, Ziyue Xiang, Sri K. Yarlagadda, Ruiting Shao, János Horváth, Justin Yang, Fengqing M. Zhu, Edward J. Delp, "Handbook of Digital Face Manipulation and Detection - From DeepFakes to Morphing Attacks", Advances in Computer Vision and Pattern Recognition, Springer, 2022 (To Be Published)

Conference Papers

- Daniel M. Montserrat, **Hanxiang Hao**, Sri K. Yarlagadda, Sriram Baireddy, Ruiting Shao, János Horváth, Justin Yang, Emily R. Bartusiak, David Güera, Fengqing M. Zhu, and Edward J. Delp, "Deepfakes Detection with Automatic Face Weighting", Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2020, Virtual Conference.

- János Horváth, Daniel M. Montserrat, **Hanxiang Hao**, and Edward J. Delp, "Manipulation Detection in Satellite Images Using Deep Belief Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2020, Virtual Conference.

- János Horváth, Sriram Baireddy, **Hanxiang Hao**, Daniel M. Montserrat, and Edward J. Delp, "Manipulation Detection in Satellite Images Using Vision Transformer", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2021, Virtual Conference.

- Emily R. Bartusiak, **Hanxiang Hao**, Michael Jacobs, Nhat X. Nguyen, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "A Stochastic Grammar Approach to Predict Flight Phases of a Hypersonic Glide Vehicle", Proceedings of the IEEE Aerospace Conference, March 2022, Montana, USA

2. UTILITY-PRESERVING FACE REDACTION

2.1 A Utility-Preserving GAN for Face Obscuration

2.1.1 Overview

Major developments in the machine learning field have uncovered severe flaws in current face obscuration approaches. As shown by [2], machine learning methods are able to defeat Gaussian blurring or pixelation based obscuration methods. These obscuration techniques have been widely used by Internet news outlets, social media platforms, and government agencies. An extreme resort to prevent information leaking is to simply blacking out the entire facial region by setting all pixels in the facial area to a fixed value. However, this approach is rarely used because its visual effect is unpleasant, especially if there are many faces to be redacted. Besides the identifiable information, facial images also contain information that does not reveal identity, such as age, gender, and skin tone. Often, we want to preserve these features in many applications involving visual understanding and data mining [10].

Figure 2.1. Obscuration effect of the proposed method. First row: original faces; second row: obscured faces.

New obscuration methods are needed to remove identifiable facial information, while preserving the features that do not convey identity. The proposed method, utility-preserving GAN (UP-GAN), aims to provide an effective obscuration by generating faces that only depend on the non-identifiable facial features. In this thesis, we define utility as the facial properties such as age, gender, skin tone, pose, and expression. We choose these properties because in practice, when

dealing with a large number of identities, knowing these properties from the obscured images cannot reveal identity. One can also choose other properties to retain for different applications. As shown in Figure 2.1, UP-GAN is able to obscure the original faces by replacing them with synthetic faces that have the same utility.

2.1.2 Related Work

Standard approaches, such as pixelation and Gaussian blurring, achieve good obscuration performance in terms of human perception. However, McPherson *et al.* [2] proposed a deep learning method with a simple structure that is able to defeat these obscuration techniques. To provide better obscuration performance, a variety of approaches have been proposed to balance the need to remove identifiable information while preserving utility information.

k-same Methods. This family of approaches first groups faces into clusters based on non-identifiable information such as expression, and then generates a surrogate face for each cluster. These methods can guarantee that any face recognition system cannot do better than $1/k$ in recognizing who a particular image corresponds to [11], where k is the minimum number of faces among all clusters. This property is also known as k-anonymity [12]. In [13] and [11], they simply compute the average face for each cluster. Therefore, their obscured faces are blurry and cannot handle various facial poses. In [10], the use of an active appearance model [14] to generate more realistic surrogate faces is presented. A generative neural network, k-same-net, that directly generates faces based on the cluster attributes is described in [15]. These two methods are able to produce more realistic obscured faces with the property of k-anonymity, but cannot handle different poses.

GAN Methods. Generative adversarial network (GAN) [16] methods can provide more realistic faces. Their discriminator is designed to guide the generator by distinguishing real faces from generated faces. In [17], a model that produces obscured faces directly from original faces based on conditional-GAN [18] is proposed. They use a contrastive loss to enforce the obscured face to be different than the input face. However, since they need to directly input the original faces, the obscuration performance is not guaranteed. [19] present a two-stage model that is able to generate an obscured face without the original identifiable facial information, which prevents the leakage of identifiable information directly from faces. GANs have also been used for face manipulation in

videos. These techniques aim to create believable face swaps without tampering traces, by altering age [20] or skin color [21]. To prevent scenarios where these videos are used to create political distress or fake terrorism events, [22] design a deep learning model that is able to detect the altered frames using both the spatial and temporal information.

Our proposed method tries to leverage the advantages of both types of methods. To achieve k-anonymity, it is designed to generate faces that depend only on the utility information without directly accessing original faces. Since it is also a GAN-based method, with the discriminator guidance, it is able to produce more realistic faces than the k-same methods.

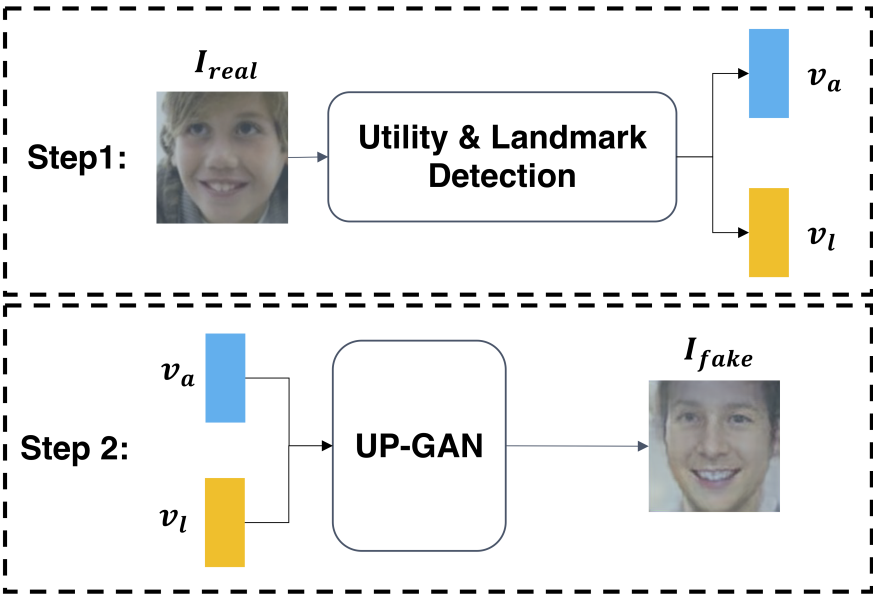


Figure 2.2. Inference block diagram of the UP-GAN model.

2.1.3 Proposed Method

Recall that, in this implementation, we choose age, gender, skin tone, pose, and expression as the utility to be preserved. To better formulate our problem, we further divide the utility into two parts: attributes and landmarks. Attributes define the static part of the utility information that does not change with facial movement. Landmarks define a set of points of interest that describe the facial pose and expression. Figure 2.2 shows the inference workflow of the proposed method. In

order to obtain obscured faces, we first use an auxiliary system to detect the utility information: attribute vector v_a and landmark vector v_l from the original face I_{real}. Since, in this thesis, we are not focusing on this auxiliary system, we use the UTKFace dataset [23] which provides the needed attributes (age, gender, and skin tone) and landmarks (7 points) to train and test our model. The fake face I_{fake} is then generated by the UP-GAN model using the attribute and landmark vectors. Given that the generated face has the same pose and expression, we can swap it with the original face to perform de-identification using face swapping algorithms [24]–[26]. Figure 2.1 shows the swapping results using [24].

Figure 2.3 shows the generator architecture of the UP-GAN model, which is based on the architecture proposed by [27]. Similar to the previous work, UP-GAN jointly learns the fake face and its binary mask. However, we modify the structure of the fully-connected layers to input the attribute and landmark vectors. As suggested by [15], we also add a max pooling layer with stride 1 for dimension reduction before generating the output image and mask. More specifically, we first use two fully-connected layers to encode the input vectors and then apply de-convolution, followed by another convolution layer to upsample the feature maps. The de-convolution layer contains an upsampling layer with stride 2 and a convolution layer with a kernel size of 5. For the following convolution layer after the de-convolution layer, we choose the kernel size to be 3. Note that the final output size of the generated face is $128 \times 128 \times 3$ and the size of the binary mask is $128 \times 128 \times 2$.

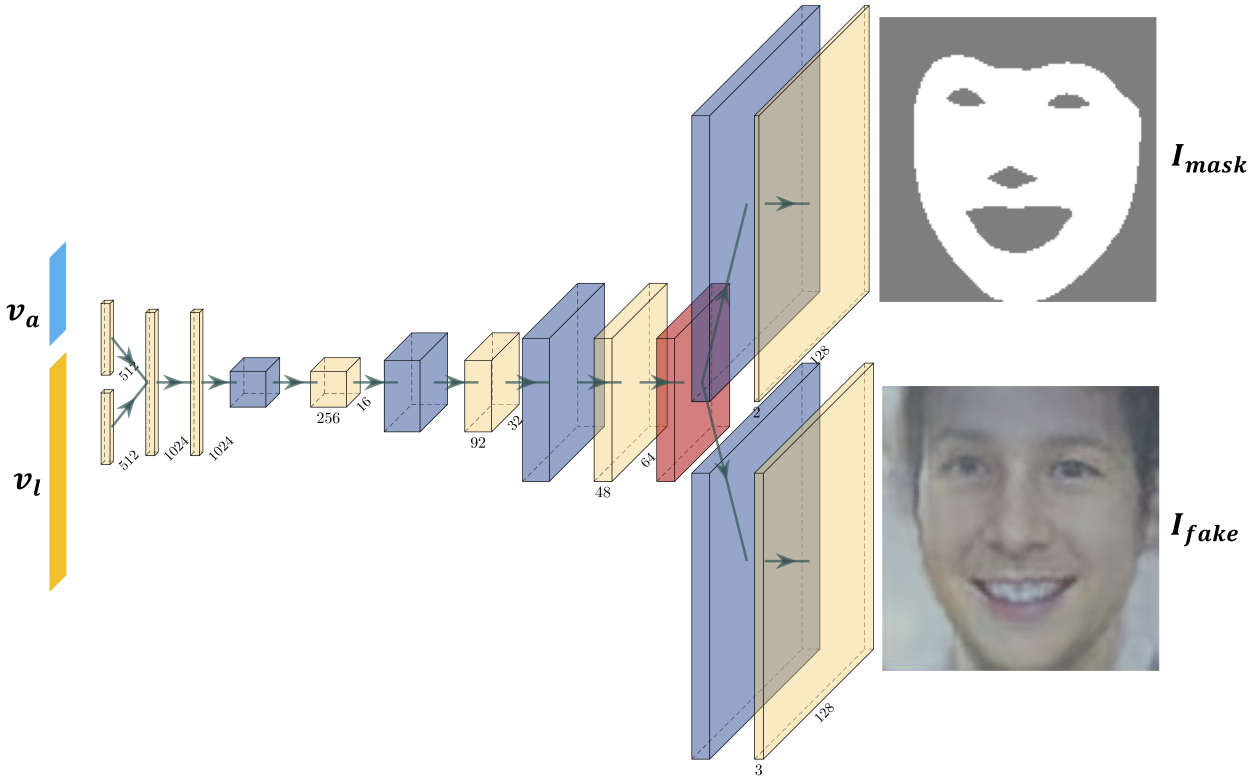


Figure 2.3. Generator architecture of the UP-GAN model. Yellow vectors indicate the activation of fully-connected layers. Blue blocks indicate the activation from de-convolution layers (upsampling + convolution). Yellow blocks show the activation from following convolution layers after the de-convolution layer. The red block shows the output from the max pooling layer.

The loss functions for the generator G and discriminator D are defined as:

$$\mathcal{L}_G = \mathbb{E}_{v_a, v_l}[\log D(G(v_a, v_l)))] + \lambda_1 \mathcal{L}_2 + \lambda_2 \mathcal{L}_M + \lambda_3 \mathcal{L}_P,$$
$$\mathcal{L}_D = \mathbb{E}_{I_{real}}[\log D(I_{real})] + \mathbb{E}_{v_a, v_l}[\log (1 - D(G(v_a, v_l)))],$$

where

$$\mathcal{L}_2 = \|I_{real} - I_{fake}\|_2^2,$$
$$\mathcal{L}_M = -\frac{1}{N} \sum_{i=1}^{N} y_i \log (p_i) + (1 - y_i) \log (1 - p_i).$$
$$\mathcal{L}_P = \sum_{l \in \Omega} \|\phi_l(I_{fake}) - \phi_l(I_{real})\|_2^2,$$

$\mathcal{L}_2$ is the reconstruction loss for learning the image content. $\mathcal{L}_M$ is the binary cross entropy loss for learning the facial mask where p_i is the predicted probability of the i-th pixel in the binary mask, y_i is the ground truth label, and N is the total number of pixels. $\mathcal{L}_P$ is the perceptual loss for learning the facial details, where Ω is a collection of convolution layers from the perceptual network and ϕ_l is the activation from the l-th layer. The perceptual loss was originally proposed by [28] for learning high level features extracted from a network pretrained on the ImageNet dataset [29]. In this thesis, the perceptual network is pretrained on a face identification dataset to enforce that the generated face contains similar facial features to the original face. More specifically, we choose the pretrained VGG-19 network [8] and finetune it with the FaceScrub dataset [30]. Lastly, λ_1, λ_2, and λ_3 are the scalar weights for their corresponding losses. Note that in our implementation, we have chosen $\lambda_1 = 5$, $\lambda_2 = 1$, and $\lambda_3 = 1$ to ensure that the terms in $\mathcal{L}_G$ are within the same numerical order of magnitude.

2.1.4 Experiment

In this section, we will evaluate different loss functions and analyze the obscuration performance of the generated faces compared to Gaussian blurring, pixelation, k-same method and k-same-net method.

Datasets. The UTKFace dataset [23] contains 23,708 images with annotations of 68-point facial landmarks and attributes of age, gender, and skin tone. The range of age provided by the dataset is from 0 to 110, the possible values for gender are 0 (male) and 1 (female) and the possible values for race are 0 (white), 1 (black), 2 (Asian), 3 (Indian) and 4 (Hispanic, Latino, Middle Eastern, etc.). To obscure the identifiable information present in the facial landmarks, we reduce the input landmark points from 68 points to 7 points. These include the centers of the eyes, the center of the nose, and four points around the mouth. Therefore, the dimensionality of the attribute vector is 3 and of the landmark vector is 14. From the perspective of k-anonymity, reducing landmark points is similar to increasing k. When we increase k, the size of each cluster also increases, since they are grouped based on attribute and landmark vectors. Therefore, the upper bound of identification rate ($1/k$) decreases, meaning that the obscuration performance improves.

To verify the obscuration performance, we use the FaceScrub dataset for face identification. Note that this dataset contains 106,806 images from 530 identities. As this dataset does not provide attributes and landmarks, we use fixed attribute values and detect facial landmarks using the *Dlib* toolkit [31]. We can produce fake faces using the fixed attributes and detected landmarks. We then use a face identification model (VGG-19) to determine if we are able to identify these generated faces.

Data Augmentation. To prevent UP-GAN from simply memorizing the original face and replicating the output face using the input vectors, we use data augmentation on the original image I_{real} to increase its variation. First, we use elastic distortion [32] to add variety to the facial landmarks. As shown in Figure 2.4, the wave-like structure distorts the landmark points (e.g. the shape of the mouth). We also add random rotations, ranging from °-30 to °30, to increase the variation of facial poses.

Results and Discussion. In Figure 2.5, we compare the results using different loss functions to show the effectiveness of training with the perceptual network and binary mask. We can also see

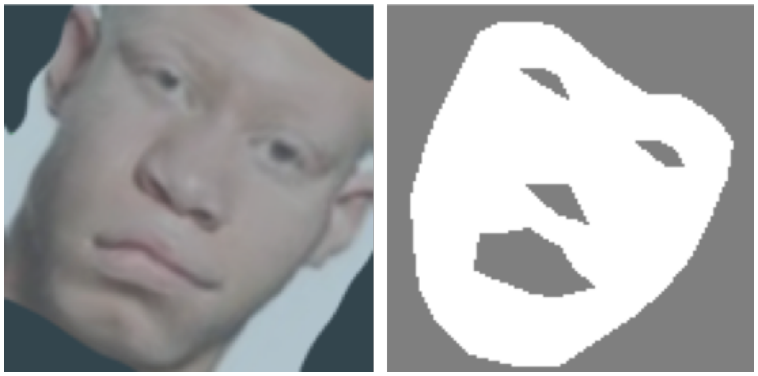


Figure 2.4. Example of the augmented face with elastic distortion and random rotation (left) and its binary mask (right).

that, compared to the original face, the generated face with adversarial loss and L_2 reconstruction loss can preserve the facial utility. However, the facial details such as the outlines are partially missing. By adding the mask loss, we can enhance the facial boundary, like the cheek and chin. If we add the perceptual loss, the generated face visually looks more realistic with fewer ripple-like artifacts.

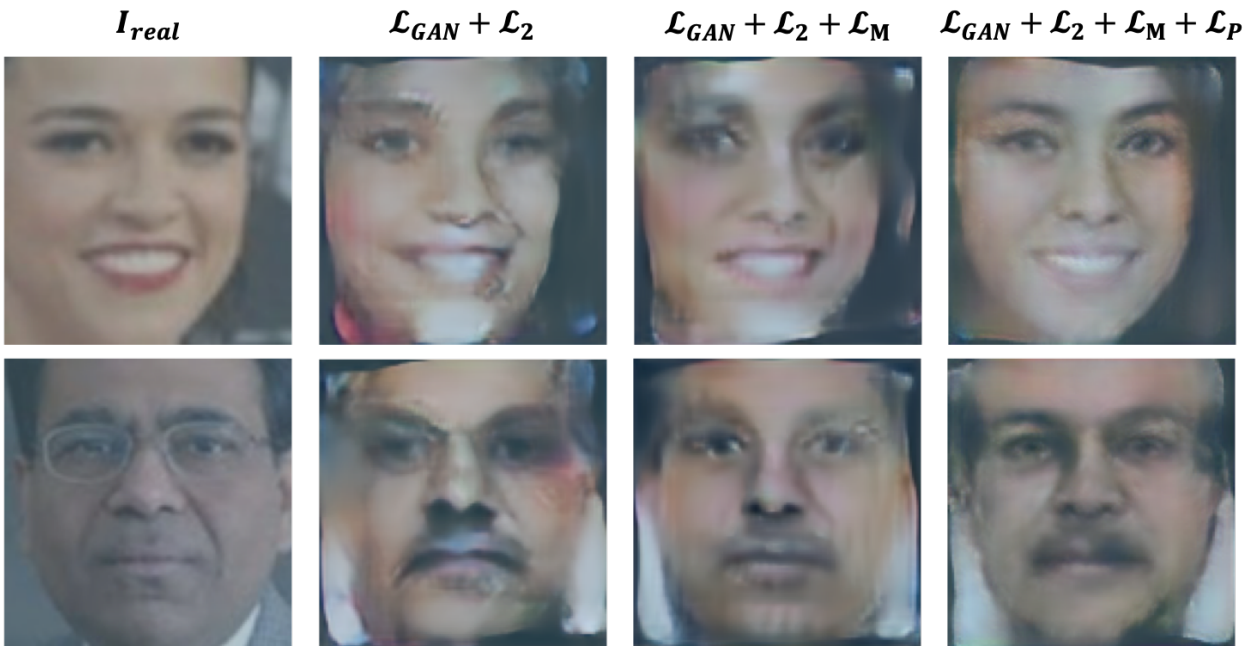


Figure 2.5. Generated faces with different loss functions.

Figure 2.6 shows the generated faces from the proposed method with various landmark information. Given the target landmark vector detected from the original image, UP-GAN is able to correctly generate the fake face with the target landmark. By doing so, we show that the proposed method can accurately retain the face expression and head pose given only 7-point facial landmarks. Furthermore, we also did another experiment of different landmarks given a fixed utility value. As shown in Figure 2.7, our model is able produce fake faces with different pose and expression given different facial landmarks.

Figure 2.6. UP-GAN results with different landmark information.

Figure 2.7. UP-GAN results of facial landmark interpolation.

Figure 2.8 shows the generated faces with various attribute information (*i.e.*, age, gender, and skin tone). Given the target attribute vector obtained from the original image, UP-GAN is able to correctly generate the fake face with the target attribute information. We also provide the interpolation comparison of the generated faces given different utility values as shown in Figure 2.9. The first row shows the interpolation of age: 0, 26 and 52. Note that the mask on the right side of each generated face shows that our model is able to change the facial outline without changing the expression (the position of mouth, eyes, and nose) given a different age value. This intuitively makes sense, since facial landmarks are not completely independent of age. For example, an infant usually has a wider facial outline than an adult. The second and third rows show the interpolation of gender and skin tone. In these two cases, the facial outline does not change much, because gender and skin tone are more independent from the facial outline than age.

Figure 2.8. UP-GAN results with different attribute information.

We also evaluate the obscuration performance to see how well UP-GAN can conceal the original faces. Note that we will provide a more comprehensive obscuration analysis in Section 2.2 with more compared methods and attacking scenarios. We consider two threat models: I) the attacker (identifier) has no information about the obscuration methods and II) the attacker knows the obscuration methods. In threat model I, we train the identifier on the pristine images and test it on the obscured faces. In threat model II, we train and test the identifier on both clear and obscured im-

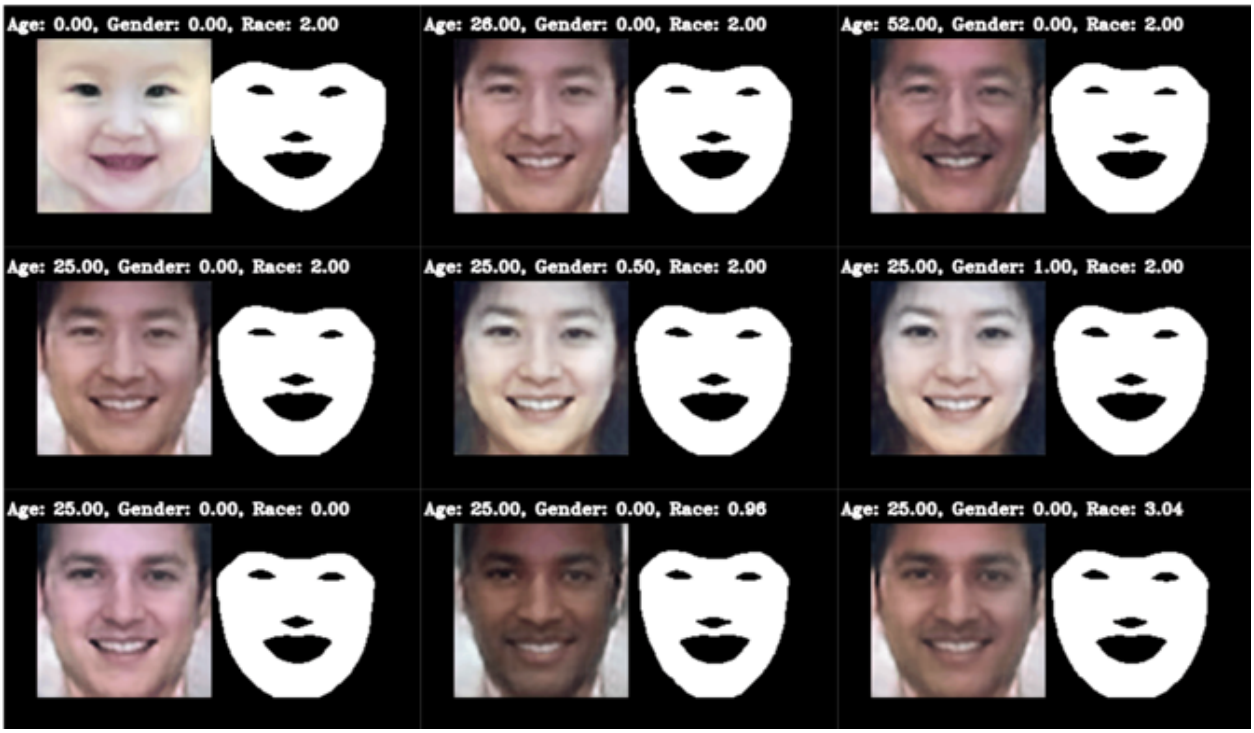


Figure 2.9. UP-GAN results of utility interpolation. Top row: age interpolation; middle row: gender interpolation; bottom row: skin tone interpolation.

ages. To provide a fair comparison with the other obscuration methods, we use the generated faces I_{fake} as the obscured images, but we do not swap them into the original images. This is because the unobscured area (non-facial region) may contain identifiable information. Figure 2.10 shows the visual quality of the obscured images with different methods including Gaussian blurring, pixelation, k-same method [11], k-same-net method [15], and UP-GAN. For the k-same method, we first use k-nearest neighbors to find k references and use their average as the output fake face. We modify the input layers of the k-same-net method to input the same attribute and landmark vectors as UP-GAN. The obscured face from k-same method is blurry (*e.g.*, the areas of eyes), although the skin tone is preserved. The result from k-same-net method contains more facial structures, but compared to UP-GAN, the facial boundary is not clear. To further quantify the visual performance, we compute the Fréchet inception distances (FID) [33] of the obscured faces. With the assumption that the real and obscured faces are two sets of realizations coming from two distributions, FID measures the distance of these two distributions. Therefore, we can use FID to estimate how real-

istic the obscured faces are. As shown in Table 2.1, UP-GAN achieves the minimum FID value, which confirms that the obscured face has the best visual quality.

Table 2.1. Face identification accuracy and FID of the obscured faces for different obscuration methods. Note that the method "None" means no obscuration and $k = 10$ for the k-same method.

Method	Threat Model T_1	Threat Model T_3	FID
None	0.955	0.955	-
Gaussian-5	0.914	0.979	212.36
Gaussian-15	0.360	0.983	386.31
Gaussian-25	0.046	0.923	358.86
Pixelation-5	0.010	0.897	154.26
Pixelation-15	0.003	0.694	576.28
Pixelation-25	0.003	0.191	486.41
k-same	0.003	0.028	91.41
k-same-net	0.003	0.238	252.90
UP-GAN	0.004	0.245	68.78

Figure 2.10. Examples of obscured faces. Top row: original image, k-same ($k = 10$) and Gaussian blurring (kernel sizes: 5, 15 and 25). Bottom row: k-same-net, UP-GAN and pixelation (pixel sizes: 5, 15 and 25).

Table 2.1 also compares the obscuration performance of UP-GAN against other methods. For the threat model I, Gaussian blurring with kernel size 5 and 15 fail to provide an effective obscuration, while all other methods achieve good performance. For the threat model II, the obscuration performance degrades for all methods, while pixelation with pixel size 25, k-same, k-same-net, and UP-GAN still achieve relatively good results. Since the k values for k-same-net and UP-GAN

methods depend on the input vectors, their identification accuracies are similar. However, as shown in Figure 2.10, for pixelation-25 there are only 5×5 blocks representing the facial region. As with k-same and k-same-net, the visual quality of pixelation-25 is worse than UP-GAN.

Figure 2.11 shows four obscuration results using the proposed UP-GAN with face swapping [24]. The image on the left of each example is the final obscuration result, while the two small images on the right is the outputs from UP-GAN (*i.e.*, generated face and its corresponding mask). By swapping the generated face to the original image, we can obscure the original face while preserving the landmark information (*i.e.*, facial expression and head pose) and attribute information (*i.e.*, age, gender, and skin tone).

Figure 2.11. UP-GAN results of face obscuration. By swapping the generated face to the original image, we can obscure the original face while preserving the facial utility information.

2.2 Robustness Analysis of Face Obscuration

2.2.1 Overview

From TV news to Google StreetView, object obscuration has been used in many applications to provide privacy protection. Law enforcement agencies use obscuration techniques to avoid exposing the identities of bystanders or officers. To remove identifiable information, Gaussian blurring or pixelation methods are commonly used. Median filtering is also used due to its simple implementation and its non-linearity, which translates into higher information distortion when compared to linear filters such as the Gaussian filter. These simple obscuration techniques are able to successfully prevent humans from recognizing the obscured objects. Previous work [1]–[3] shows that machine learning approaches can still identify these objects using the subtle information left in the obscured images. More robust and effective techniques have been described including k-same methods [10], [11], [13], [15], [34] which are able to provide a secured obscuration while preserving non-identifiable information. Reversible obscuration [35]–[37] is another type of method to prevent the leakage of privacy information from unauthorized viewers when sharing an image on social media. These methods are designed to achieve privacy-preserving image sharing by encrypting the images published online. Only the viewer with the correct decoding key is able to access the image. In this section, we focus on the robustness analysis of several obscuration techniques for face redaction. We study these obscuration methods to answer the following question: "Is there any remaining identifiable information from the obscured faces to enable re-identification?".

Although several of these approaches are widely used by news outlets, social media platforms, and government agencies, their performance has not been objectively measured. The lack of a formal study of these obscuration techniques makes it hard to evaluate the quality of redaction systems. As shown by McPherson *et al*. [2], a simple deep learning model can identify individuals from their highly pixelated and blurred faces. This indicates that human perception is no longer the gold standard to examine the effectiveness of obscuration methods. To provide a better way to examine a given obscuration method, we need to consider it in a controlled environment that can determine how well identifiable information can be extracted from the obscured face. We design three scenarios: obscured face identification, verification, and reconstruction. Figure 2.12 shows the results from the reconstruction attack for the eight studied obscuration methods. To analyze

the vulnerability of these methods, we examine multiple threat models based on an attacker's knowledge of the obscuration method used. Our simplest threat model assumes that the attacker has no information of these obscuration methods. In the most challenging scenario, we consider that the attacker knows the exact type of the obscuration method and its hyperparameters. These previously unexplored threat models are necessary to offer a complete vulnerability analysis under realistic situations.

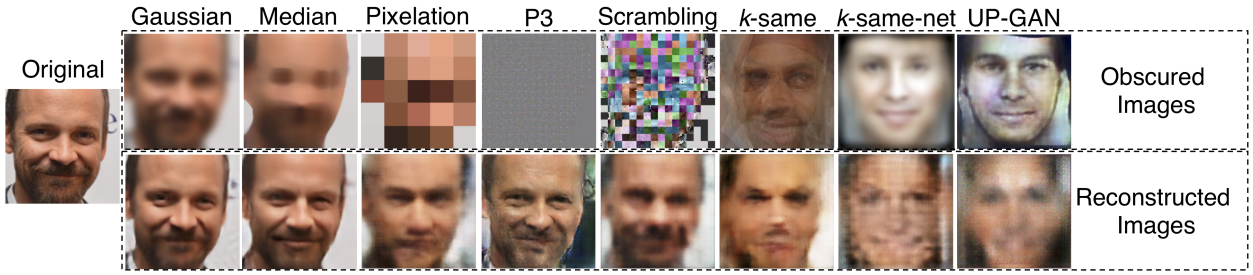


Figure 2.12. Reconstruction of obscured images using Pix2Pix [38] as described in Section 2.2.3. Although the obscured images are hard to recognize, deep learning models can still recover the person's identity. For Gaussian, median, and P3, we can clearly recognize the person from their recovered images.

2.2.2 Related Work

Face Obscuration Methods. Gaussian blurring and pixelation are frequently used in many applications. However, these techniques are not reliable. As we will show in Section 2.2.5, Gaussian blurring even with a large kernel size is still not able to defend against the some of our attacks. An extreme example of blurring to prevent information leaking is to simply black out the entire facial region by setting all pixels in the facial area to a fixed value. This approach is rarely used because its visual effect is unpleasant, especially if there are many faces in the scene that need to be redacted.

To address some of these issues, k-same methods [10], [11], [13], [15], [34] have been proposed to balance the removal of identifiable information while preserving non-identifiable facial features. These methods attempt to group faces into clusters based on personal attributes such as age, gender, or facial expression. Then, a template face for each cluster is generated. These methods can fulfill the requirement of k-anonymity [39]. They are able to guarantee that any face recognition system

cannot do better than $1/k$ in recognizing to whom a particular image corresponds, where k is the minimum number of faces among all clusters [11]. In Newton *et al.* [13] and Gross *et al.* [11], they simply compute the average face for each cluster. Therefore, the obscured faces are blurry and cannot handle various facial poses. Du *et al.* [10] use the active appearance model [14] to learn the shape and appearance of faces. Then, they generate a template face for each cluster to produce obscured faces with better visual quality. A generative neural network, k-same-net, that directly generates faces based on the cluster attributes is described in [15]. To produce more realistic faces, generative adversarial network (GAN) [16] have been used, since its discriminator is designed to guide the generator by distinguishing real faces from generated faces. Hao *et al.* [34] propose a method based on conditional GAN [18] that can generate a synthetic face given the facial landmarks and cluster attributes without the original image.

Face completion is alternative approach to achieve face obscuration. It first blocks the facial region and then completes the blocked region with a synthetic face without accessing the original facial information. Sun *et al.* [19] propose a GAN-based method by generating a fake face to complete the blocked region. The generator is able to predict the face appearance based on the body pose and surrounding environment.

Besides the methods above that permanently remove the identifiable information, reversible obscuration methods [35]–[37] are also needed for the purposes of privacy-preserving image sharing. These reversible obscuration methods split the image information into two parts: 1) the public part which contains most volume, but not meaningful content and 2) a secret part that stores the image decoding key. Therefore, when publishing an image to social media, the public and secret parts can be stored separately to avoid the leakage of images to unauthorized viewers. Ra *et al.* [35] propose a method, P3, which is based on the JPEG encoding framework. They separate the DCT coefficients in the JPEG encoding process based on a predefined threshold value to generate the public and secret images. Yuan *et al.* [37] propose a scrambling method that further reduces the data storage in the secret part. Instead of thresholding, they randomly flip the sign of DCT coefficients and store the result as the public image. For the secret part, they only need to store the random seed to recover the original image.

Privacy Analysis of Obscuration Methods. Although Gaussian blurring and pixelation are widely used, these methods might still leak sensitive information. Dufaux and Ebrahimi, and Sah

et al. [1], [3] provide an analysis of the obscuration performance of simple identifiers and show the ineffectiveness of current obscuration methods. By using a simple deep learning model, McPherson *et al*. [2] also show that obscured images still contain enough information to perform accurate identification. They uncover the identity obscured with blurring, pixelation, and P3 methods. For the 16×16 pixelation method, they achieve a top-5 identification accuracy of 98.75% for the AT&T dataset [40] and 72.23% for the FaceScrub dataset [30]. Oh *et al*. [41] also propose a semi-supervised model that is able to identify the face under large variations in pose. Their model is based on a conditional random field (CRF), which not only infers the individual faces (unary part) but also deduces the identities based on other visible faces (pairwise part). Therefore, when the unary part is weak due to the obscuration, the identifiable information from other visible faces is able to help improve the deduction of the obscured face through the connections from the CRF.

To extend the previous literature [1]–[3], [41], we first consider the face identification scenario. By mapping faces to known identities in different threat models, we analyze the vulnerability of each obscuration method using advanced deep learning identification methods. However, the requirement of known identities weakens this type of analysis, since query faces usually come from unknown identities. To overcome this, we provide a threat analysis under a more realistic setup: the face verification scenario. Specifically, we want to measure the similarity of an unknown redacted face to clear target faces. Since it allows recognizing unseen identities, this scenario is more realistic. Lastly, a reconstruction scenario is proposed to visualize how well we can recover the true identity using the remaining information from the obscured images.

2.2.3 Proposed Method

To evaluate the performance of the obscuration methods, we first introduce three threat models based on the amount of knowledge about the obscuration method that is available to the attackers. Then we describe the three attacks: obscured face identification, verification, and reconstruction.

Threat Modeling. In our model, the attacker aims to identify the redacted faces based on the information still present in the obscured images. We design three threat models, which vary on how much information about the used obscuration approach is available to the attacker.

- Threat model T_1 assumes the attacker has no information of any obscuration method, which means that the attacker is only able to learn the facial features used for identification from clear faces. During the testing phase, it extracts the facial features from the obscured faces directly.

- Threat model T_2 assumes the attacker is aware of some obscuration methods, but not the same method used in the testing phase. *i.e.*, the attacker is trained on both clear and obscured images and tested with the obscured images of the obscuration methods not used in the training set. This model assumes the attacker does not know the exact obscuration method being used, which is the same as T_1. However, it provides more information to the attacker, since different obscuration methods may share similarities in terms of identifying facial features.

- Threat model T_3 assumes the attacker knows the exact type of the obscuration method and its hyperparameters, like the kernel size of Gaussian blurring. Compared to T_1 and T_2, T_3 is the strongest attack, since it provides the attacker with the most information of the obscuration method to identify identities.

Obscured Face Identification Attack. For the obscured face identification attack, we assume a fixed number of identities. We treat this identification problem as a classification problem where the number of classes is equal to the number of identities. In this section, we evaluate the performance of different obscuration methods based on different backbone deep learning models, such as VGG19 or ResNet50 in order to have a more generalizable conclusion.

Obscured Face Verification Attack. The obscured face verification attack is defined as: given an obscured face and a clear face, decide if the two faces come from the same person or not. Previous work [1]–[3], [41] only considers the identification scenario, which assumes all identities are in the dataset. However, in many cases, we cannot assume the obscured identity is in any dataset. For example, the attackers may want to find out if the obscured face from a TV news is a person they know. Therefore, face verification attack is more stringent.

In order to solve this verification problem, we project the image into a low-dimension latent vector, where faces from the same person are closer together than faces from different people. Therefore, by comparing the distance of the latent vectors, we can determine if the two faces

are from the same person or not. To improve the accuracy, we use the Additive Angular Margin loss (also known as ArcFace) [42] to obtain highly discriminative features for face recognition. ArcFace simultaneously reduces intra-class difference and enlarge inter-class difference of the embedding vectors. We choose ArcFace because it yields the best facial recognition performance among the traditional softmax loss [43], contrastive loss [44], triplet loss [45], and other angular space losses, like SphereFace [46] and CosFace [47]. Specifically, ArcFace is designed to enforce a margin between the distance of the sample to its class center and the distances of the sample to the other centers from different classes in angular space. Given an input image (either clear image or obscured image), we first embed it as a low-dimension vector $\mathbf{x} \in \mathbb{R}^d$ using a deep learning model. Define an auxiliary projection weight $\mathbf{W} \in \mathbb{R}^{d \times n}$, where n is the number of unique identities in the dataset. We further normalize the embedding vector and projection weight as $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ and $\hat{\mathbf{W}} = \frac{\mathbf{W}}{\|\mathbf{W}\|}$, respectively. The normalized embedding vector then is projected onto $\mathbb{R}^n$ as follows

$$\hat{\mathbf{W}}^T \hat{\mathbf{x}} = \|\hat{\mathbf{W}}\| \|\hat{\mathbf{x}}\| \cos \boldsymbol{\theta} = \cos \boldsymbol{\theta},$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is a vector of angular distance from $\hat{\mathbf{x}}$ to $\hat{\mathbf{W}}$. The normalized embedding vector is then re-scaled by multiplying a scalar s to make it distributed on a hypersphere with a radius of s. The ArcFace loss function of a single sample is then calculated using softmax cross entropy as follows

$$L = -\log \frac{e^{s \cos(\theta_t + m)}}{e^{s \cos(\theta_t + m)} + \sum_{j=1, j \neq t}^{n} e^{s \cos(\theta_j)}},$$

where m is the additive angular margin penalty between $\mathbf{x}$ and $\mathbf{W}$, θ_t is the angle of the target class of the input image. Note that the computation of the ArcFace loss is only used to aid the training process. For inference, we compute the embedding vectors from the clear face $\boldsymbol{x}_c$ and obscured face $\boldsymbol{x}_o$ using the same deep learning model. We then compare the angular distance after normalization to a predefined threshold value to determine the verification result. The threshold value can be obtained based on the value that maximizes the verification accuracy on the validation set.

Obscured Face Reconstruction Attack. As we will show in Section 2.2.5, highly obscured images still contain identifiable information. To examine the amount of remaining information

in obscured images, we design a reconstruction attack to visualize how well we can recover the original image. We apply a conditional generative adversarial network, Pix2Pix [38], to perform this image reconstruction attack. Given the obscured images, the generator is trained to reconstruct the clear image guided by the discriminator and the L_2 distance loss. To quantify the reconstruction performance, we compute the mean square error (MSE) over pixel-wise differences. We also compute the identification accuracy based on a face recognition model which is pretrained with clear images. This test provides us a way to quantify and visualize the amount of identifiable information leaked from the obscuration methods.

2.2.4 Evaluated Methods

In this thesis, we propose to analyze eight obscuration methods. These methods include three traditional methods (Gaussian blurring, median blurring, and pixelation), three k-same based methods (k-same, k-same-net, and UP-GAN) and two privacy-preserving image sharing methods (P3 and scrambling). Examples of obscured faces using these methods are shown in Figure 2.12. We use Gaussian-5 representing the experiment of Gaussian blurring with kernel size of 5.

Traditional obscuration methods. We evaluate the three obscuration methods including Gaussian blurring, median blurring and pixelation methods for four different kernel (pixel) sizes of 5, 15, 25, and 35. We use the OpenCV function *cv2.getGaussianKernel* to compute the kernel of Gaussian blurring. Note that the Gaussian standard deviation is defined as

$$\sigma = 0.3 * \left(\left(\frac{w-1}{2} - 1 \right) + 0.8 \right),$$

where w is the kernel size. The pixelation method is implemented by image downsampling and upsampling using nearest-neighbor interpolation.

k-same based obscuration methods. k-same based methods aim to obscure identifiable information while preserving the non-identifiable information (also known as utility information). Algorithm 1 shows the workflow of the k-same based methods, which is based on [11]. In this thesis, we choose $k = 10$. We evaluate three k-same based methods: the original k-same method [11], k-same-net [15], and UP-GAN [34]. We model the obscuration process as follows.

Suppose we have a clear face dataset $\mathcal{M}_c$ and an obscuration function f mapping the clear image I_c to the obscured image I_o by $I_o = f(I_c)$. We use this mapping function building an obscured face dataset $\mathcal{M}_o$ based on $\mathcal{M}_c$. Based on [11], we also need to assume the dataset $\mathcal{M}_c$ has no two images coming from the same identity to make Algorithm 1 k-anonymous. The k-same based methods require the function f mapping k nearest neighbors from the clear images to a single obscured image. For example, considering the original k-same method, the obscured face is obtained by averaging the k nearest neighbors in the image space. Therefore, the $x_{1,\dots,k}$ from Algorithm 1 in this case are the clear images.

k-same-net is a deep learning model that generates faces given the cluster attributes. UP-GAN has similar architecture to k-same-net with the same input cluster attributes. However, it improves the generated image quality using its discriminator and the perceptual loss. For both k-same-net and UP-GAN, the $x_{1,\dots,k}$ from Algorithm 1 are the cluster attributes. Therefore, the input attribute to the models is the average of the k nearest neighbors in the attribute space.

As proposed by [34], we choose UTKFace dataset [23], which contains the required utility values (age, gender, and skin tone) and facial landmarks to train k-same-net and UP-GAN. The utility values are defined as facial features that do not reveal identity, such as age, gender, skin tone, pose, and expression [34]. For the purpose of obscuration evaluation, we test these two methods on the FaceScrub dataset [30] and LFW dataset [48], with a fixed utility values (26 years old, male, and white) and 7-point facial landmarks obtained by *Dlib* toolkit [31][1]. These points include the centers of the eyes, the center of the nose, and four points around the mouth. Note that since the two datasets contain different faces from the same identity, the k-anonymity property in this case may not hold.

Privacy-preserving image sharing methods. Privacy-preserving image sharing methods are designed to encrypt the content of the original image when publishing to social media. To recover the original images, the encrypted images need a key to decrypt the content. We evaluate two methods: P3 [35] and scrambling [37]. Both of them are based on the manipulation of DCT coefficients in the JPEG framework. After obtaining the DCT coefficients from 8×8 image patches, P3 separates the AC coefficients given a predefined threshold value. It then stores the coefficients that are smaller than the threshold value as the public image. The secret image contains the DC

[1]↑Since we fix the utility values, in this case, $x_{1,\dots,k}$ are the facial landmark vectors.

Algorithm 1: Workflow of the k-same based methods.

Input: Clear face dataset $\mathcal{M}_c$, privacy constant k with $|\mathcal{M}_c| \geq k$
Output: Obscured face dataset $\mathcal{M}_o$
$\mathcal{M}_o \leftarrow \emptyset$;
for $i \in \mathcal{M}_c$ **do**
  if $|\mathcal{M}_c| < k$ **then**
    $k = |\mathcal{M}_c|$;
  end
  Select the k nearest neighbors $x_1, \ldots, x_k \in \mathcal{M}_c$;
  $x_o \leftarrow \frac{\sum_{m=1}^{k} x_m}{k}$;
  Add k copies of x_o to $\mathcal{M}_o$;
  Remove $x_1, \ldots, x_k$ from $\mathcal{M}_c$;
end

coefficients and the AC coefficients that are higher than the threshold value. In this thesis, we choose the threshold value as 10. For the scrambling method, it first evenly and randomly flips the DCT coefficients and stores the result as the public image. For the secret part, it only stores the random seed. Therefore, it can restore the image by undoing the flipping process based on the random seed. In this thesis, we scramble both DC and AC DCT coefficients for all YUV components, which is the high-level scrambling as proposed by [37].

2.2.5 Experiment

Datasets. We use the FaceScrub dataset [30] and LFW dataset [48] in this thesis. The FaceScrub dataset [30] is a benchmark for face identification which contains 106,863 images from 530 identities. The LFW dataset [48] is a benchmark for face verification which contains 13,233 images from 5,749 identities. We use the LFW dataset for the verification attack and the FaceScrub dataset for the identification and reconstruction attacks. We choose the FaceScrub dataset for the reconstruction attack, since it contains more images for each identity with various poses. For the identification attack, we split the images from each identity into training, validation, and testing sets with the ratio of $6:2:2$. For the verification and reconstruction attacks, we split the identities into three groups for the purpose of training, validation, and testing with the same ratio. We do so

to verify if the verification and reconstruction models are able to recover unknown identity instead of just memorizing faces.

Obscured Face Identification Attack. This attack is designed to quantify the obscuration performance in the face identification scenario. To have a more generalizable conclusion, we run the experiments based on two widely used backbone models, VGG19 and ResNet50. The input images are resized to 128×128 and the output is the softmax score for classification. We use the stochastic gradient descent (SGD) optimizer with weight decay of 0.0001. The initial learning rate is 0.001 and starts to decay linearly with the ratio of 0.02 after 50 epochs.

Based on the three threat models, we design the experiments as follows. In the first experiment for T_1, the identifier is trained with the set of clear images and tested with obscured images. In the second experiment for T_2, the identifier is trained on both clear and obscured images and tested with the obscured images of the obscuration method not used in the training set. Although including both clear and obscured images during training downgrades the testing accuracy of the clear images, it achieves better testing accuracy of obscured images than training without the clear images. Therefore the attackers are trained with clear and obscured images jointly The intuition of threat model T_2 is to verify if we can enforce the attacker to learn more robust features from this complex dataset. This can be seen as data augmentation. Specifically for the three traditional methods, we use the obscured images from two methods during training and use the other one for testing. For the k-same based methods and privacy-preserving image sharing methods, we train on all three traditional methods. Jointly training on clear and obscured images provides a better accuracy compared to learning from the obscured images themselves. In the third experiment for T_3, each identifier is trained on both clear and obscured images and tested with the obscured images using the same obscuration method. We train and test the attackers for different obscuration methods separately, since different obscuration methods have different features. For example, pixelation has sharp block artifacts, while Gaussian blurring is more smoothing.

Obscured Face Identification Attack Result. Table 2.2 shows the identification accuracy from different obscuration methods and threat models. The lower the identification accuracy, the better the performance of the obscuration method. The results of the clear images under T_2 and T_3 are obtained by training on all three traditional methods and testing on the clear images.

Table 2.2. Top-1 accuracy of the identification attack. The method *Clear* means the identification of the clear image. The lower the accuracy, the better the obscuration method.

Method	Setting	Threat Model T_1		Threat Model T_2		Threat Model T_3	
		VGG19	ResNet50	VGG19	ResNet50	VGG19	ResNet50
Clear	-	0.838	0.890	0.886	0.884	0.886	0.884
Gaussian	5	0.787	0.853	0.829	0.909	0.891	0.867
	15	0.106	0.219	0.548	0.773	0.863	0.847
	25	0.010	0.030	0.236	0.573	0.830	0.819
	35	0.007	0.009	0.152	0.430	0.811	0.798
Median	5	0.786	0.855	0.883	0.907	0.913	0.907
	15	0.185	0.229	0.735	0.823	0.889	0.885
	25	0.025	0.035	0.357	0.489	0.856	0.842
	35	0.011	0.014	0.213	0.270	0.805	0.798
Pixelation	5	0.055	0.208	0.408	0.606	0.877	0.884
	15	0.004	0.003	0.008	0.008	0.651	0.643
	25	0.003	**0.002**	0.005	0.004	0.461	0.408
	35	0.004	**0.002**	0.004	0.005	0.373	0.323
k-same [11]	10	0.012	0.012	0.013	0.012	**0.050**	**0.063**
k-same-net [15]	-	0.091	0.081	0.091	0.088	0.095	0.092
UP-GAN [34]	-	0.091	0.082	0.090	0.088	0.093	0.088
P3 [35]	10	**0.001**	**0.002**	**0.002**	**0.002**	0.678	0.579
Scrambling [37]	-	0.002	**0.002**	0.004	0.003	0.784	0.750

We first compare the same method and same backbone model with different threat models. As the attackers get more information (*i.e.*, from T_1 to T_3), the identification accuracy increases. This means that the identifiable information left in the obscured images can still be learned by the attackers given proper training data. For example, the accuracy of Gaussian-35 with VGG19 increases from 0.007 to 0.811 for T_1 and T_3, respectively. Therefore, Gaussian blurring completely fails to provide privacy for T_3, although visually speaking a human is not able to identify someone from the obscured images. A similar conclusion can be drawn for median blurring. Although pixelation with a large pixel size can achieve a relatively good performance, comparing the results from T_1 to T_3, the attacking accuracy still improves a lot. *e.g.*, for pixelation-35 with VGG19, the accuracy increases from 0.004 to 0.373, for T_1 and T_3, respectively. The three k-same based methods achieve a good obscuration performance even for T_3. For the privacy-preserving image sharing methods, although they achieve the best performance under T_1 and T_2, they still fail to provide a good obscuration under T_3. Surprisingly, even for the scrambling method which involves a random flipping process, the attackers can still extract useful features for accurate identification. Note that these conclusions do not change for different backbone models.

Considering T_1 itself, besides Gaussian-5 and median-5, all methods achieve an effective obscuration on both VGG19 and ResNet50 models. This means that the attackers fail to extract identifiable information from the obscured images if they solely learn from the clear image. For the three traditional methods, the obscuration performance gets better (*i.e.*, identification accuracy gets lower) as the kernel size increases. The original k-same method achieves the best obscuration performance among the three k-same based methods. For k-same-net and UP-GAN, since they allow the input of utility information to generate obscured faces, their obscuration performance is a little bit worse than the original k-same method. Both of the privacy-preserving image sharing methods achieve the performance of randomly guessing, which means the attackers cannot extract any identifiable information from the obscured images.

For T_2, by introducing more informative training set, all traditional methods have worse performance, besides pixelation-25 and pixelation-35, which are relatively close to the results obtained from T_1. The obscuration performance of Gaussian and median blurring drops significantly (*i.e.*, the identification accuracy greatly increases). Because the two methods share similar blurring effects, the attackers can learn more robust features from the augmented training set. The augmented

training set contains both obscured and clear images as previously mentioned. For the k-same based methods and privacy-preserving image sharing methods, compared to T_1, the augmented training set still does not provide useful knowledge for the attackers.

For T_3, both attackers achieve the strongest attack for all cases. Even for pixelation-35, which only contains 9 distinct pixel values, both attackers can still achieve a identification accuracy over 0.5, which is much bigger than the accuracy of randomly guessing (0.002). The three k-same based methods achieve the best obscuration performance by a great margin when compared to other methods. Surprisingly, the two privacy-preserving image sharing methods have a much worse performance compared to their performance in T_1 and T_2. Even successfully concealing the identifiable information in terms of human perception, both methods fail to provide effective obscuration.

Therefore, based on the results from the identification attack, the k-same based methods (k-same, k-same-net, and UP-GAN) achieve the best obscuration performance.

Obscured Face Verification Attack. The input images are resized to 128×128. According to [42], we choose the dimension of the embedding vector as 512 and margin m as 0.5. However, if we use the re-scale factor $s = 64$ as suggested by the original paper, we are not able to obtain a stable result. Therefore, after several experiments, we empirically choose the re-scale factor as $s = 11$ for VGG19 and $s = 8$ for ResNet50, which provides the best performance according to the validation set. The batch size is chosen as 128. We choose the SGD optimizer with a weight decay of $5e^{-4}$. The learning rate starts at 0.1 and is divided by 10 at the epochs of 6, 11, and 16. For the training of P3 and scrambling, we reduce the starting learning rate to 0.05 due to convergence issues. We implement the experiments based on the three threat models. Since the face verification problem is just a binary classification problem, we choose the area under the curve (AUC) of the receiver operating characteristic (ROC) curve to examine the performance.

During testing we need to obtain pairs of faces with the same identity and pairs of faces with different identities. Due to the large number of combinations of valid pairs from the testing set, in our implementation, we only compute all valid pairs within each mini-batch (128 images which are coming from 64 identities). Furthermore, we run testing 10 times with different combinations of image pairs. The average AUC is been reported in Table 2.3. The standard deviation for the tests

ranges from $[0.004, 0.093]$. Therefore, we can directly use the average AUC to compare different experiments because of the small variation.

Obscured Face Verification Attack Result. Table 2.3 shows the verification AUC from different obscuration methods, threat models and backbone models. The lower the AUC, the better the performance of the obscuration method. We first compare the same method and same backbone model to different threat models. As the attackers get more information (from T_1 to T_3), the verification AUC increases. Take Gaussian-35 with ResNet50 as an instance. The AUC increases from 0.629 to 0.962 for T_1 and T_3, respectively. Note that the AUC for randomly guessing is 0.5. This means that although Gaussian-35 can successfully defend from the attack under T_1, after introducing the obscured data in the training set, the attackers can still extract enough identifiable information to achieve a high accuracy verification. For the k-same based methods, similar to the identification attack, they achieve a robust obscuration performance even for T_3. For the privacy-preserving image sharing methods, both of them succeed in T_1 and T_2, but fail to obscure the identities under T_3. Note that for different backbone models, although there is a small performance difference, choosing different models does not affect the conclusions reached above.

Consider different methods with the same threat model and backbone model. For the traditional methods, a similar conclusion to the identification attack can be drawn. As the kernel (pixel) size increases, the AUC decreases for all cases, especially for pixelation-35 with VGG19 which achieves the best performance among the traditional methods. The k-same based methods achieve good results for all threat models and both attackers, which agrees with the conclusion from the identification attack. Although the privacy-preserving image sharing methods can conceal identities well under T_1 and T_2, for the stronger T_3, both of them fail to provide effective obscuration.

Therefore, based on the results from the verification attack, the k-same based methods (k-same, k-same-net, and UP-GAN) achieve the best obscuration performance.

Obscured Face Reconstruction Attack. Previously, we show that most of the obscuration methods fail to remove all identifiable information. In this reconstruction attack, we use the remaining information from these obscured images to recover the clear image. If the remaining information has a strong correlation with the information from the clear image, we can reconstruct the original face with a high accuracy. We choose Pix2Pix [38] which is a GAN model designed for image-to-image translation as our reconstruction model.

Table 2.3. AUC of ROC for the verification attack. The lower the AUC, the better the obscuration method.

Method	Setting	Threat Model T_1		Threat Model T_2		Threat Model T_3	
		VGG19	ResNet50	VGG19	ResNet50	VGG19	ResNet50
Clear	-	0.983	0.981	0.983	0.981	0.983	0.981
Gaussian	5	0.971	0.993	0.933	0.966	0.900	0.967
	15	0.756	0.878	0.909	0.950	0.901	0.952
	25	0.572	0.742	0.865	0.942	0.879	0.948
	35	0.512	0.629	0.852	0.918	0.893	0.962
Median	5	0.959	0.963	0.954	0.960	0.922	0.954
	15	0.675	0.835	0.918	0.945	0.913	0.948
	25	0.570	0.668	0.853	0.884	0.908	0.931
	35	0.539	0.592	0.832	0.837	0.877	0.933
Pixelation	5	0.806	0.865	0.909	0.921	0.955	0.954
	15	0.543	0.542	0.794	0.709	0.856	0.926
	25	0.510	0.539	0.681	0.709	0.793	0.890
	35	0.505	0.530	0.530	0.598	0.630	0.792
k-same [11]	10	0.573	0.580	0.573	0.588	0.695	0.768
k-same-net [15]	-	0.505	**0.493**	0.503	0.504	0.497	**0.492**
UP-GAN [34]	-	**0.500**	0.499	0.506	**0.490**	**0.494**	0.497
P3 [35]	10	0.503	0.502	**0.496**	0.500	0.524	0.899
Scrambling [37]	-	0.544	0.549	0.595	0.568	0.951	0.928

Assume that the obscured images and clear images come from two distinct distributions. The reconstruction model aims to find a mapping function from the obscured image distribution to the clear image distribution. To quantify the reconstruction performance, we choose mean square error (MSE) as the metric to calculate pixel-wise distance between the clear image and the reconstructed image. The value range of the clear image and reconstructed image is $[0, 1]$. To evaluate similarity of the identifiable information from the reconstructed image and the clear image, we use the identification accuracy obtained from the ResNet50 model which is pretrained on the clear images. This is the same setting as T_1, since the attacker is trained with clear images and tested with obscured images.

Obscured Face Reconstruction Attack Result. Figure 2.12 shows the reconstruction results from the eight obscuration methods. Visually, the three k-same based methods can successfully prevent reconstruction compared with other methods. Although the privacy-preserving image sharing methods can prevent identification in terms of human perception, the reconstruction model can still recover the images fairly accurately, especially for P3. For the three traditional methods, pixelation-25 achieves a better obscuration performance compared to Gaussian-25 and median-25.

Table 2.4 shows the results of the face reconstruction attack. Note that setting *Clear* means we input clear images to Pix2Pix model to achieve an identity mapping. The exact MSE for the clear image is 0.000144 and the exact MSE for Gaussian-5 is 0.000289. For the three traditional methods, with the kernel size increases, the reconstruction MSE increases and the identification accuracy decreases. Compared to the identification attack of T_1, this reconstruction process can help the attackers achieve a stronger attack, since the accuracy from the reconstructed images is higher than the obscured images for most cases. The k-same based methods achieve both high MSE and low identification accuracy. Compared to the three k-same methods, the two privacy-preserving image sharing methods are vulnerable to the reconstruction attack, because of their low MSE.

Therefore, as with the conclusion in the identification and verification attack, these two methods also fail to conceal identity on this reconstruction attack and the k-same based methods (k-same, k-same-net, and UP-GAN) achieve the best obscuration performance.

Table 2.4. MSE and identification accuracy of the reconstruction attack. The arrows next to *MSE* and *Accuracy* indicate that the higher the MSE and the lower the identification accuracy are, the better the obscuration method is.

Method	Setting	MSE↑	Accuracy↓
Clear	-	0.000	0.849
Gaussian	5	0.000	0.824
	15	0.001	0.707
	25	0.002	0.519
	35	0.002	0.367
Median	5	0.001	0.774
	15	0.003	0.356
	25	0.004	0.152
	35	0.007	0.102
Pixelation	5	0.004	0.439
	15	0.014	0.043
	25	0.022	0.013
	35	0.031	0.006
k-same [11]	10	0.029	0.005
k-same-net [15]	-	**0.064**	0.018
UP-GAN [34]	-	0.059	**0.003**
P3 [35]	10	0.013	0.339
Scrambling [37]	-	0.018	0.042

2.3 Utility-Preserving Face Obscuration via Face Reenactment

2.3.1 Overview

Previously, we introduced a deep learning-based model, UP-GAN, that is able to generate synthetic faces that preserve the utility information while also removing identifiable information from the original faces. By swapping the generated face back on the original image, we can produce an effective obscuration that not only removes personal identifiable information, but also retains the information that does not reveal identity, such as expression, age, gender and skin tone. As shown in Figure 2.13, the proposed method is required to generate a synthetic identity appearance and transfer the appearance to the target pose at the same time. In order to generate a synthetic identity with consistent appearance given the same appearance information but different pose information, the model needs to decouple the appearance and pose information when generating the synthetic image. Based on our experiment, the proposed method may fail to produce a consistent identity for certain poses and expressions, like rotating head or opening mouth. Therefore, we need to improve our model by explicitly decoupling the synthetic appearance generation process and the facial pose transformation process.

With the fast development of generative adversarial networks (GANs), recent methods, such as ProGAN [5] and StyleGAN [4], can generate photo-realistic synthetic face images. These high-fidelity synthetic faces can be used for our face obscuration task. Figure 2.14 shows the block diagram of the approach we propose to solve the previously mentioned issue. First, a face dataset that contains the high quality synthetic faces is generated. With the high-fidelity face generator, such as ProGAN [5] and StyleGAN [4], the variety of synthetic faces with different age, gender, and skin tone can be generated. Since these synthetic faces are not from real person, we will use these faces as the surrogate appearance information for redaction, instead of generating the synthetic appearance within our model. Given an input face to be redacted, we find its closet matched face in the synthetic dataset as the surrogate appearance. The matching process is done by simply finding the best match of the utility information from the synthetic faces to the input face. The proposed face reenactment model needs to transfer the synthetic face given to the target facial pose and expression given by the original input face. Our reenactment model is required to transfer pose and expression in one-shot (*i.e.*, transfer the pose and expression based on only

one input image). By doing so, the synthetic appearance generation process and the facial pose transformation process are decoupled. Therefore, we are able to produce a consistent synthetic identity from different target poses and expressions.

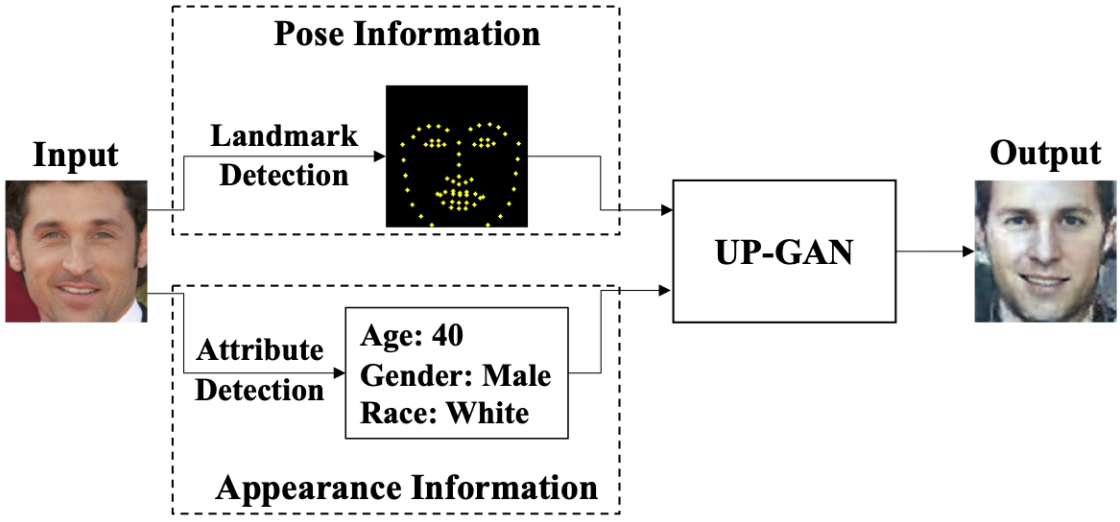


Figure 2.13. The Block Diagram of the proposed UP-GAN.

Figure 2.15 shows the reenacted faces produced by the proposed method. Given an input face image of a source identity, the proposed one-shot face reenactment model, FaR-GAN, is able to transform the expression from the input image to any target expression. The reenacted faces have the same expression captured by the target landmarks, while also retaining the same identity, background, and even clothes as the input image. Therefore, the proposed one-shot face reenactment model requires no assumption about the source identity, facial expression, head pose, and image background.

2.3.2 Related Work

Face Reenactment by 3D Modeling. Modeling faces in 3D helps in accurately capturing their geometry and movement, which in turn improves the photorealism of any reenacted faces. Thies *et al*. [49] propose a real-time face reenactment approach based on the 3D morphable face model (3DMM) [50] of the source and target faces. The transfer is done by fitting a 3DMM to both faces and then applying the expression components of one face onto the other [51]. To achieve

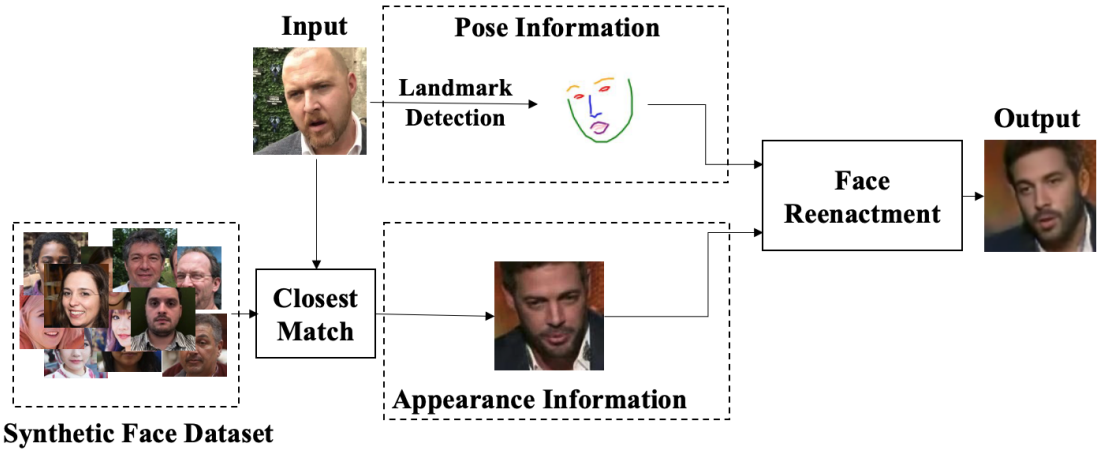


Figure 2.14. The Block Diagram of Use Face Reenactment Method for Utility-Preserving Face Obscuration.

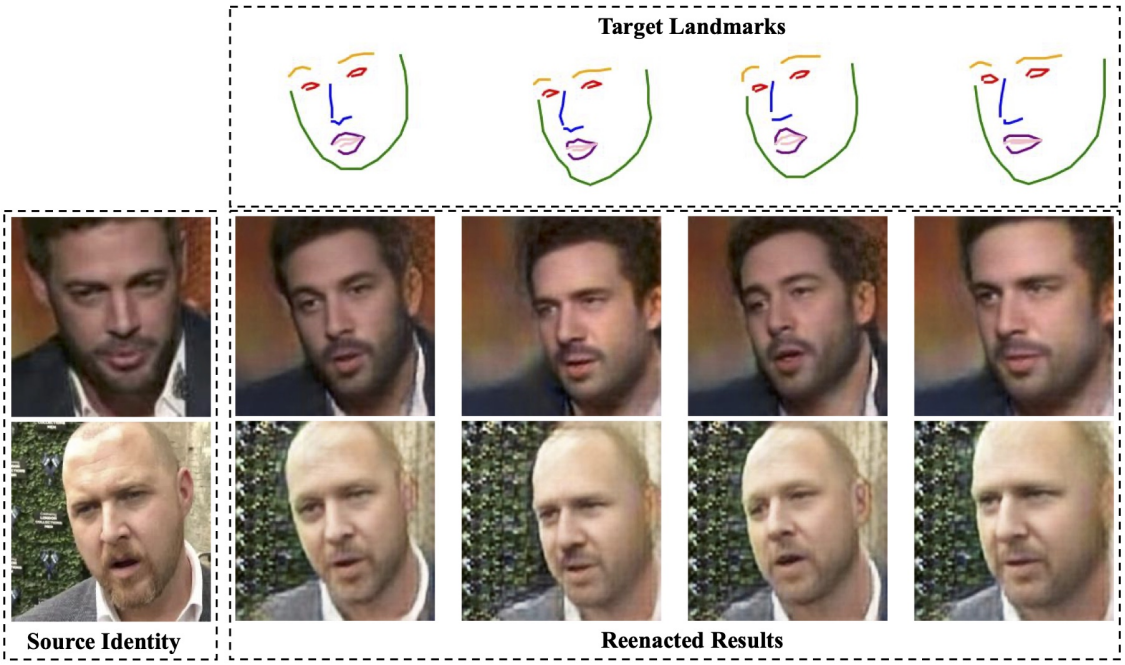


Figure 2.15. One-shot face reenactment results from the proposed model. The proposed method takes a face image from a source identity (visualized on the left column) and a target landmark mask (visualized on the top row), and then outputs the face of the source identity but with the target expression.

face synthesis based on imperfect 3D model information, they further improve their method by introducing a learnable feature map (*i.e.*, neural texture) alongside the UV map from the coarse 3D model as input to the rendering system [52]. During 2D rendering, they also design a learnable neural rendering system that is based on U-Net [53] to output the 2D reenacted image. The entire rendering pipeline is end-to-end trainable.

Face Reenactment by GANs. Generative adversarial networks have been successfully used in this area due to their ability to generate photo-realistic images. They are able to achieve high quality and high resolution unconditional face generation [4], [5], [54]. ReenactGAN, proposed by Wu *et al*. [55], first maps the face that contains the target expression into an intermediate boundary latent space that contains the information of facial expressions but no identity-related information. Then the boundary information is used for an identity-specific decoder network to produce the reenacted face of the specific identity. Therefore, their model cannot be used for the reenactment of unknown identities.

To solve this issue, few-shot or even one-shot face reenactment methods have also been developed in the recent work [56]–[58]. Wiles *et al*. [56] propose a model, namely X2Face, that is able to use facial landmarks or audio to drive the input source image to a target expression. Instead of directly learning the transformation of expressions, their model first learns the frontalization of the source identity. "Frontalization" is the process of synthesizing frontal facing views of faces appearing in single unconstrained photos [59]. Then it produces an intermediate interpolation map given the target expression to be used for transferring the frontalized face. Zakharov *et al*. [57] present a few-shot learning approach that achieves the face reenactment given a few, or even one, source images. Unlike the X2Face model, their method is able to directly transfer the expression without the intermediate boundary latent space [55] or interpolation map [56]. Zhang *et al*. [58] propose a one-shot face reenactment model that only requires one source image for training and inferencing. They use an auto-encoder-based structure to learn the latent representation of faces, and then inject these features using the SPADE module [60] for the face reenactment task. The SPADE module in our proposed method is inspired by their work. However, instead of using the multi-scale landmark masks used by [58], we use learnable features from convolution layers as the input to the SPADE module.

2.3.3 Proposed Method

Model Architecture. Figure 2.16 shows the generator architecture of the proposed FaR-GAN model. The model consists of two parts: embedder and transformer. The embedder model aims to learn the feature representation of facial expressions given a set of facial landmarks. In this thesis, we adopt a similar color encoding method proposed in [57] to represent the facial landmarks. More specifically, we use distinct colors for eyes, eyebrows, nose, mouth outlier, mouth inlier, and face contour. We also tried to use a binary mask to represent the landmark information (*i.e.*, set 1 for the facial region and set 0 for the background), but it did not give us a better result. We will show the comparison of results with different landmark representations in Section 2.3.4. The transformer model aims to use the landmark features from the embedder model to reenact the input source identity with the target landmarks. The transformer architecture is based on the U-Net model [53]. The U-Net model is a fully convolutional network for image segmentation. Besides its encoder-decoder structure for local information extraction, it also utilizes skip connections (the gray arrows in Figure 2.16) to retain global information.

A similar generator architecture can be found in [57] but with several differences. First, instead of using the embedder to encode appearance information of the source identity, we use it to extract the target landmark information. The embedder model is a fully convolutional network that continuously downsamples the feature resolution with maxpooling or average-pooling layers. Therefore, the spatial information of the input image will be lost due to the downsampling process. To encode the appearance information of the source identity, the output features are required to represent a large amount of information including the identifiable information, hair style, body parts (neck and shoulders), and even background. Therefore, it is challenging for the embedder model to learn precise appearance information with the loss of the spatial information. In our approach, we use the embedder model to encode the facial landmarks, which contains much less information than the aforementioned appearance features. Moreover, instead of outputting a single 1D embedding vector [57], we use the embedder features from all resolutions obtained after the downsampling process. With this, we can assure the embedder features contain the required spatial information for expression transformation.

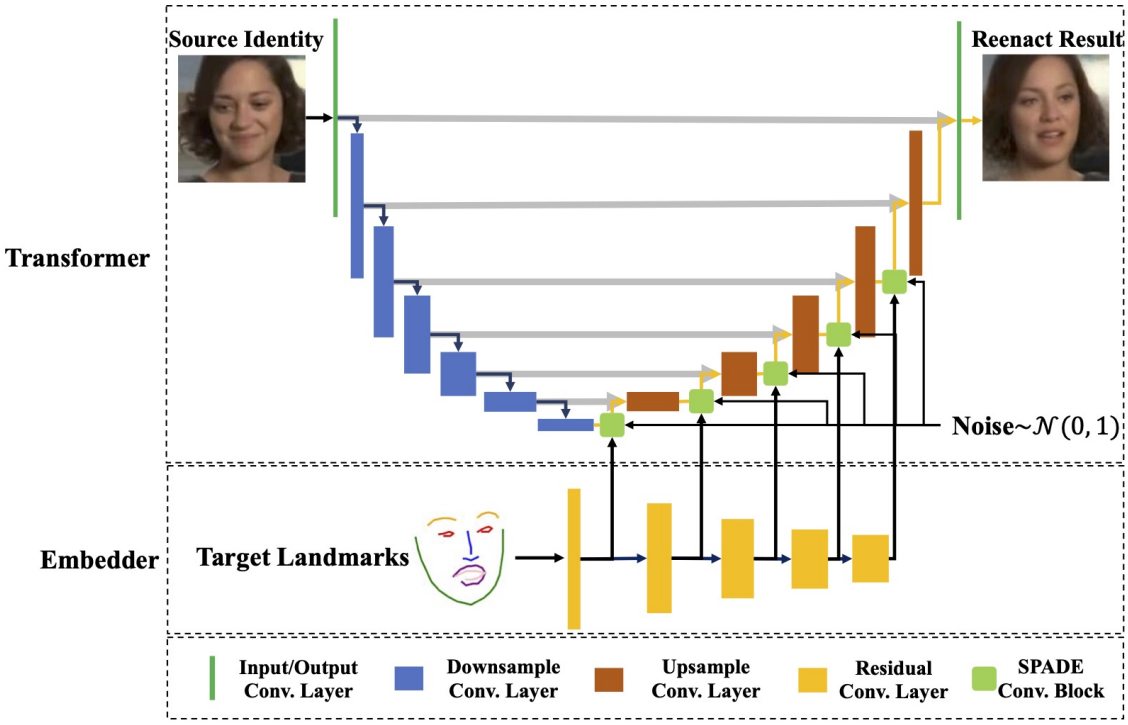


Figure 2.16. The generator architecture of the proposed FaR-GAN model. Given target facial landmarks and a arbitrary source identity, the proposed model learns to transfer facial expression for the source identity. The embedder model learns the feature representation of the facial expression defined by the landmarks. The transformer model uses the features from the embedder to generate a new face of the source identity but has the same facial expression as the target landmarks.

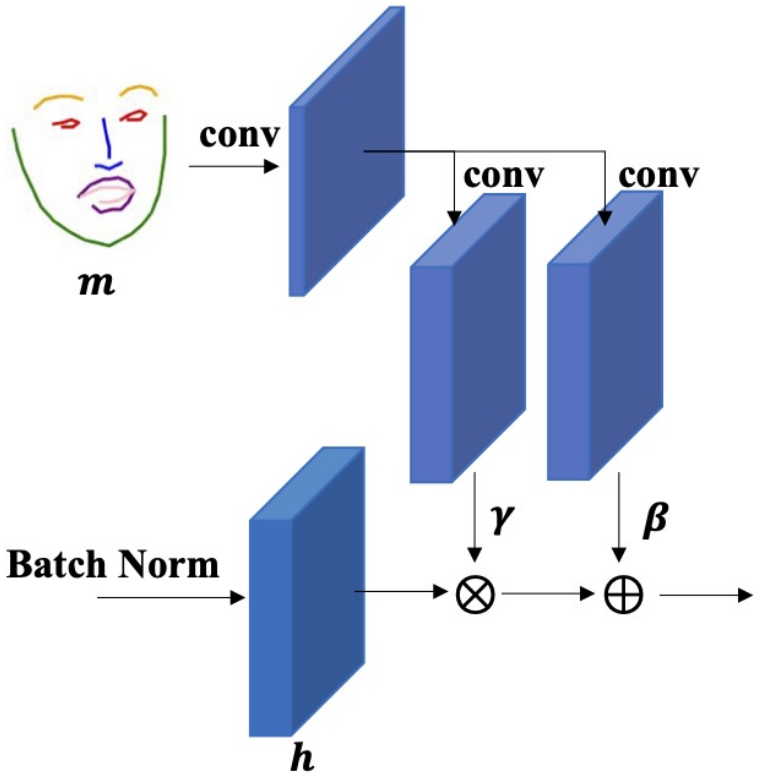


Figure 2.17. Architecture of the SPADE module. This figure is based on [60]. SPADE module maps the input landmark mask to the modulation parameters γ and β through a set of convolution layers. Then the element-wise multiplication and addition are used for γ and β, respectively to the batch-normalized input feature.

The adaptive instance normalization (AdaIN) module has been successfully used for face generation in previous work [4], [54], [57]. In [57], they use AdaIN modules to inject the appearance information into the generator model to produce the reenacted face by assigning a new bias and scale of the convolution features based on the embedder features. However, since we need to inject landmark information, which comes from a sparse landmark mask, we cannot simply adopt the AdaIN module in our method. This is because, the instance normalization (*e.g.*, AdaIN) tends to wash away semantic information when applied to uniform or flat segmentation masks [60], such as our input landmark masks. Instead, we propose using the spatially-adaptive normalization (SPADE) [60] module to inject the landmark information. As the name indicates, the SPADE module is a feature normalization approach that uses the learnable spatial information from the input features. Similar to batch normalization [61], the input convolution features are first normalized in a channel-wise manner, and then modulated with a learned scale and bias, as shown in Figure 2.17. The output of the SPADE module can be formulated, as shown in Equation 2.1.

$$\gamma_{c,x,y}(\mathbf{m})\frac{h_{n,c,x,y}-\mu_c}{\sigma_c}+\beta_{c,x,y}(\mathbf{m}) \tag{2.1}$$

where $\mathbf{m}$ is the input landmark mask or intermediate convolution features from the embedder, $h_{n,c,x,y}$ is the input convolution feature from mini-batch $n \in N$, channel $c \in C$, dimension $x \in W$, and dimension $y \in H$, $\gamma_{c,x,y}$ is the new scale, and $\beta_{c,x,y}$ is the new bias. The mean μ_c and standard deviation σ_c of the activation in channel c are defined in Equation 2.2 and 2.3.

$$\mu_c = \frac{1}{NHW}\sum_{n,x,y} h_{n,c,x,y} \tag{2.2}$$

$$\sigma_c = \sqrt{\frac{1}{NHW}\sum_{n,x,y}(h^2_{n,c,x,y}-\mu^2_c)} \tag{2.3}$$

This SPADE module has been successfully used for the face reenactment task in [58]. As shown in Figure 2.16, in our method, the input to the SPADE block is the convolution features from the embedder network. In [58], they use a group of multi-scale landmark masks as the input to the SPADE blocks, instead of the deep features from our proposed method. However, in our experiment, if we use these multi-scale masks instead of deep features as input to the SPADE

Figure 2.18. The artifacts of using multi-scale masks as input to the SPADE module. The landmark contours are still visible in the output images of the transformer model.

blocks, the output reenacted faces will contain the artifacts from the input landmark contours, as shown in Figure 2.18. Similar to [62], we use the features from the embedder network to inject the landmark information into the transformer model.

There are many aspects in human portraits that can be regarded as stochastic, such as the exact placement of hairs, stubble, freckles, or skin pores [4]. Inspired by StyleGAN [4], we introduce stochastic variation into our transformer model by injecting noise. The noise injection is executed for each resolution of the decoder part of the transformer model. More specifically, we first sample an independent and identically distributed standard Gaussian noise map z of size $H \times W$, where H and W are the spatial resolution of the input feature. Then a noise block with the number of channels C is obtained by scaling the noise map z with a set of learnable scaling factors for each channel. We inject the noise block by adding it element-wise with the input features.

We adopt the design of [18], [38] for our discriminator. More specifically, the input to our discriminator is the reenacted face concatenated with the target landmark mask, or the ground truth face image with its corresponding landmark mask. Therefore, the discriminator aims to guide the generator to produce a realistic face and also faces with the correct target landmarks. In Section 2.3.4, we will provide an ablation study to show the importance of the discriminator.

Loss Function. The proposed model including both embedder and transformer is trained end-to-end. Assume we have a set of videos that contain the moving face/head of multiple identities. We denote $\mathbf{x}_i(t)$ as the i-th video and t-th frame. Assume $\mathbf{x}_i(t_1)$ and $\mathbf{x}_i(t_2)$ are two random frames

from a video. Therefore, the two frames $\mathbf{x}_i(t_1)$ and $\mathbf{x}_i(t_2)$ contain the same identity but with different facial expressions and head poses. We formulate our generator function G as follows:

$$\hat{\mathbf{x}}_i(t_2) = G(\mathbf{x}_i(t_1), \mathbf{m}_i(t_2)) \tag{2.4}$$

where $\mathbf{m}$ is the landmark mask. The generator loss function is defined in Equation 2.5.

$$\mathcal{L}_G = \mathcal{L}_{adv} + \mathcal{L}_{L1} + \mathcal{L}_p + \mathcal{L}_{id} \tag{2.5}$$

where

$$\begin{aligned}
\mathcal{L}_{adv} &= \mathbb{E}_{\mathbf{x}_i,\mathbf{m}_i}[(D(\hat{\mathbf{x}}_i(t_2), \mathbf{m}_i(t_2)) - 1)^2] \\
\mathcal{L}_{L1} &= \|\hat{\mathbf{x}}_i(t_2) - \mathbf{x}_i(t_2)\| \\
\mathcal{L}_p &= \sum_{l \in \Phi} \|\phi_l(\hat{\mathbf{x}}_i(t_2)) - \phi_l(\mathbf{x}_i(t_2))\| \\
\mathcal{L}_{id} &= \sum_{l \in \Psi} \|\psi_l(\hat{\mathbf{x}}_i(t_2)) - \psi_l(\mathbf{x}_i(t_2))\|.
\end{aligned}$$

$\mathcal{L}_{adv}$ is the generator adversarial loss, which is based on LSGAN [63]. We compared the results from the vanilla-GAN [16], LSGAN [63], and WGAN-GP [64] and chose LSGAN based on the visual quality of reenacted images. $\mathcal{L}_{L1}$ is the pixel-wise L1 loss to minimize the pixel difference of the generated image and the ground truth image. $\mathcal{L}_p$ is the perceptual loss for minimizing the semantic difference, which was originally proposed by [28]. Φ is a collection of convolution layers from the perceptual network and ϕ_l is the activation from the l-th layer. In this thesis, the perceptual network is a VGG-19 model [8] pretrained on the ImageNet dataset [29]. To enforce the reenacted face to have the same identifiable information as the input source identity, we add an identity loss $\mathcal{L}_{id}$, which is similar to the perceptual loss, but with a VGGFace model [43] pretrained for face verification.

The discriminator loss function is based on the LSGAN loss function, which is defined as follows:

$$\begin{aligned}
\mathcal{L}_D = &\mathbb{E}_{\mathbf{x}_i,\mathbf{m}_i}[(D(\hat{\mathbf{x}}_i(t_2), \mathbf{m}_i(t_2)))^2] + \\
&\mathbb{E}_{\mathbf{x}_i,\mathbf{m}_i}[(D(\mathbf{x}_i(t_2), \mathbf{m}_i(t_2)) - 1)^2]
\end{aligned} \tag{2.6}$$

Implementation Details. As shown in Figure 2.16, the convolution layers in the embedder model are a set of residual convolution layers [9]. This is adopted from [57], which also adds the spectral normalization [65] layers to stabilize the training process. The transformer network consists of input/output convolution layers, downsampling convolution layers, upsampling convolution layers, and SPADE convolution blocks. The input/output convolution layers only contain convolution layers; so the feature resolutions do not change. The downsampling convolution layers consist of an average-pooling layer, convolution layer, and spectral normalization layer. The upsampling convolution layers consist of a de-convolution layer followed by a spectral normalization layer to upsample the feature resolution by a factor of 2. The SPADE convolution block contains the noise injection layer followed by the SPADE module. For the discriminator, we use the same structure proposed by [38], with the two downsampling convolution layers.

Previously, the self-attention mechanism has been successfully used for GANs that generate high quality synthetic images [66]. To ensure that the generator learns from a long-range of information within the entire input image, we adopt the self-attention module in both the generator and discriminator. More specifically, for the generator, we place the self-attention module after the upsampling convolution layers of the feature resolutions of 32×32 and 64×64, which is similar to the implementation in [57]. For the discriminator, we place the self-attention module after the second downsampling convolution layer.

During training, in order to balance the magnitude of each term in the loss function, we choose the weights for $\mathcal{L}_{L1}$, $\mathcal{L}_p$, and $\mathcal{L}_{id}$ as 20, 2, and 0.2, respectively. These weights could be different when using different datasets or different perceptual networks. We use the Adam optimizer [67] for both the generator and discriminator with the initial learning rate as $5e^{-5}$. The learning rate decays linearly and decreases to 0 after 100 epochs.

2.3.4 Experiment

Dataset. We use the VoxCeleb1 dataset [68] for training and testing our method. It contains 24,997 videos from 1251 different identities. The dataset provides cropped face images extracted at 1 frame per second and we resize these images to 256×256. *Dlib* package [31] is used for

extracting 68-point facial landmarks. We split the identities into training and testing sets with the ratio of $8:2$ in order to assure that our model is generalizable to new identities.

Experimental Results. We compare the proposed method against two methods, the X2Face model [56] and the few-shot talking face generation model (Few-Shot) [57]. X2Face contains two parts: an embedder network and a driver network. Instead of directly mapping the input source image to the reenacted image, their embedder learns to frontalize the input source image and the driver network produces a interpolation map given the target expression to transform the frontalized image. To compare with the X2Face model, we use their model with pretrained weights provided by the authors and evaluate on the VoxCeleb1 dataset. The Few-Shot model also contains two parts: an embedder network and a generator network. As described in Section 2.3.3, their embedder learns to encode the appearance information of the source image, while the generator learns to generate the reenacted image given the appearance information and target landmark mask. For the Few-Shot model, since the authors only provide the testing results, we directly use these results for comparison. Both the X2Face and Few-Shot method require two stages of training. The first stage uses two frames from the same video, while the second stage requires the frames from two different videos. By doing so, they can ease the training process at the beginning by using the frames that contain the same identity and similar background information. Then for the second stage, they use the frames from two different videos to ensure that the reenacted face contains the same identifiable information as the input source identity. As mentioned in Section 2.3.3, the proposed method requires only the first stage training.

In this section, we provide both qualitative and quantitative results comparison. For the quantitative analysis, we use the following metrics to evaluate the reenacted images in terms of image quality and the performance of the preservation of source identity:

- Structured Similarity Index (SSIM) [69]: we use SSIM to measure the image quality of the reenacted images. SSIM measures low-level similarity between the ground truth images and reenacted images [57], such as color and shape. The higher the SSIM is, the better the quality of the generated images are.

- Fréchet-Inception Distance (FID) [33]: we use FID to measure the image quality of the generated images. It measures perceptual realism based on an InceptionV3 network that was

pretrained on ImageNet dataset for image classification (the weights are fixed during the FID evaluation). Given a set of synthetic images (*i.e.*, reenacted images) and a set of real images (*i.e.*, video frames), FID computes their statistical difference based on the InceptionV3 features. Therefore, it measures both high frequency and low frequency components. FID has been used for image quality evaluation in many work [4], [5]. In this thesis, the FID score is computed using the default setting [2] (using the final average pooling features from the InceptionV3 network). The lower the FID is, the better the quality of the generated images are.

- Identity Cosine Similarity (CSIM): a good face reenactment model needs to preserve the source identity when generating the reenacted images. In this thesis, we use CSIM to measure the identity similarity between the reenacted images to the source images. CSIM first encodes the reenacted and source images using a face recognition network and then computes the cosine distance of the encoding vectors to measure the similarity of the identities between the source image and reenacted image. Following the work [57], we use ArcFace [42] for the the face recognition network. The higher the CSIM is, the more similar the identities from the source and reenacted images.

Table 2.5. Quantitative comparison of the proposed method with the compete methods.

Method	SSIM↑	FID↓	CSIM↑
X2Face [56]	0.68	45.8	0.16
Few-Shot [57]	0.67	43.0	0.15
FaR-GAN (proposed)	0.68	27.1	0.48

Table 2.5 shows the results of the proposed and compared methods. The SSIM, FID, and CSIM scores of the compared methods are obtained from the original paper [57]. We first consider the evaluation of image quality using SSIM and FID. Although the SSIM results are similar for all three methods, the proposed method outperforms the compared methods in terms of FID. Figure 2.19 shows the qualitative comparison from the testing set. The results from X2Face contains wrinkle

[2]↑The implementation is in https://github.com/mseitzer/pytorch-fid

Figure 2.19. Face reenactment results from the compared and proposed methods.

artifacts, because it uses the interpolation mask to transfer the source image, instead of directly learning the mapping function from the source image to the reenacted image. Although the X2Face result in the first row shows its effectiveness when the change of head pose is relatively small, the results in the second and third rows show that the wrinkle artifacts get more visible when the background becomes complex and the change of head pose is larger. Both Few-Shot method and the proposed method obtain the results with a good visual quality, including transferring accurate target expression and also preserving the background information. Due to the proposed method of injecting the noise into the transformer network, the reenacted faces contain more high frequency information than the Few-Shot model, especially for the woman's hair from the third testing case. Because the FID computes the statistical difference from a collection of synthetic images and real images, it measures both high frequency and low frequency components. Therefore, the proposed method achieves much lower FID than the two compared methods. We use the identity cosine similarity (CSIM) to measure the model performance of source identity preservation. As shown in

Table 2.5, the proposed method achieves the maximum CSIM score with a large margin compared to the two competed methods. This shows that the proposed method can more effectively preserve the source identity when generating the reenacted images.

2.3.5 Ablation Study

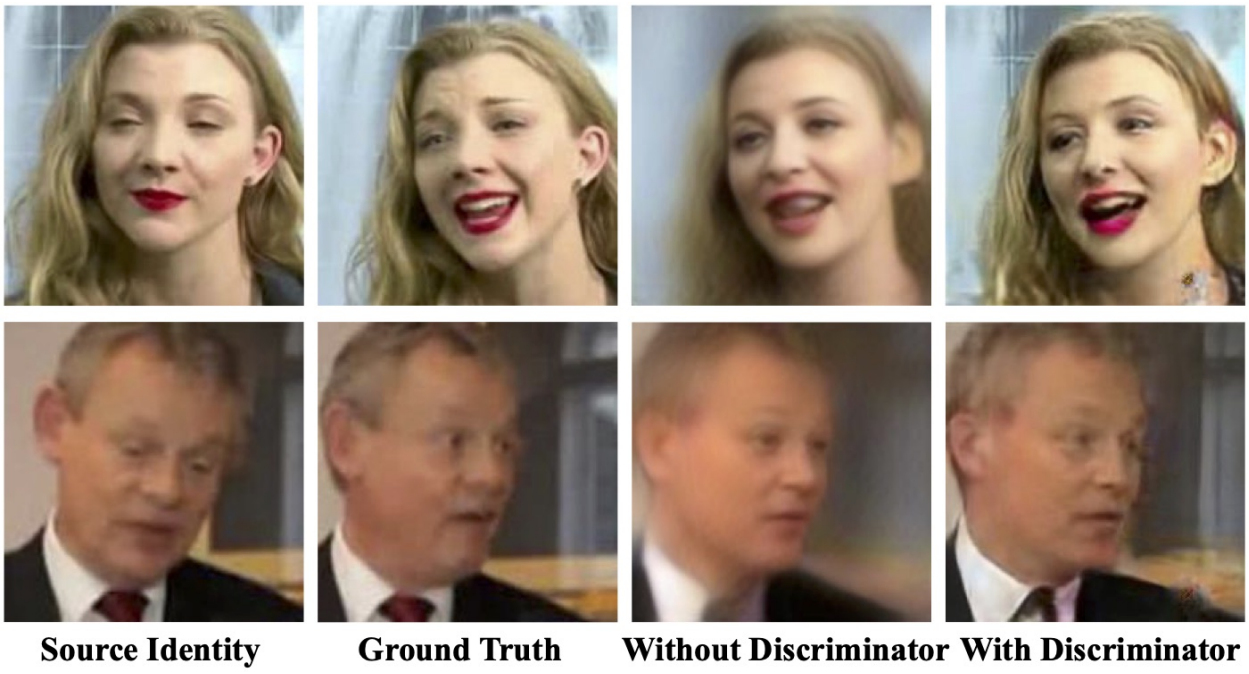


Figure 2.20. Ablation study of the use of discriminator.

Figure 2.20 shows our results with and without the discriminator. The result with discriminator contains more details, like hair, teeth, and background, compared to the result without discriminator. Therefore, the discriminator does guide the generator (both embedder and transformer) in producing better synthetic images.

To show the effectiveness of the choice of embedder landmark representation: the contour-based mask or binary mask, we evaluate the image quality of generated images based on the aforementioned SSIM and FID metrics, as shown in Table 2.6 An example of the landmark binary mask is shown in Figure 2.21. Although the SSIM scores are similar, the FID score of the binary mask is much higher than the contour-based representation. Due to the use of different colors for different

Table 2.6. SSIM and FID results of the proposed method with different landmark representations.

Method	SSIM↑	FID↓
FaR-GAN (Mask)	0.67	52.1
FaR-GAN (Contour)	0.68	27.1

Figure 2.21. An alternative landmark representation using a binary mask.

parts of the facial components, the contour-based mask provides additional information for the embedder to treat different parts of face separately. Thus, it can achieve a better understanding of facial pose and expression.

Table 2.7. SSIM and FID results of the proposed method with different model components.

Method	SSIM↑	FID↓
FaR-GAN (w/o attention and w/o noise)	0.67	63.9
FaR-GAN (w/ attention and w/o noise)	0.66	35.3
FaR-GAN (w/ attention and w/ noise)	0.68	27.1

Table 2.7 shows the ablation study of different model components regarding to the image quality of the generated images, including self-attention module and noise injection module. Although

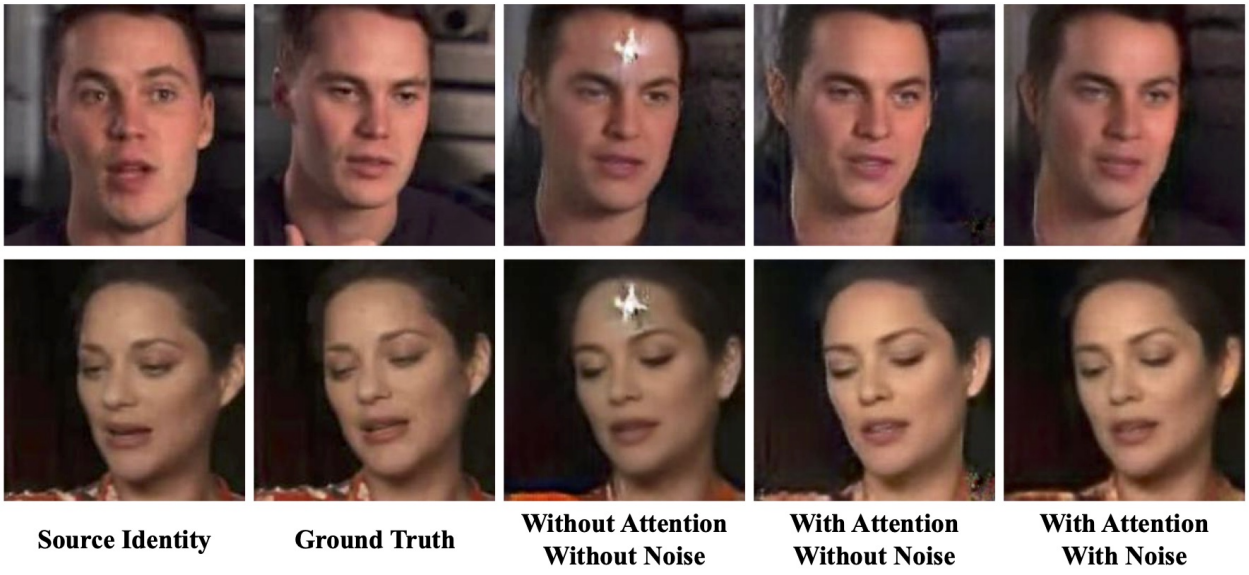
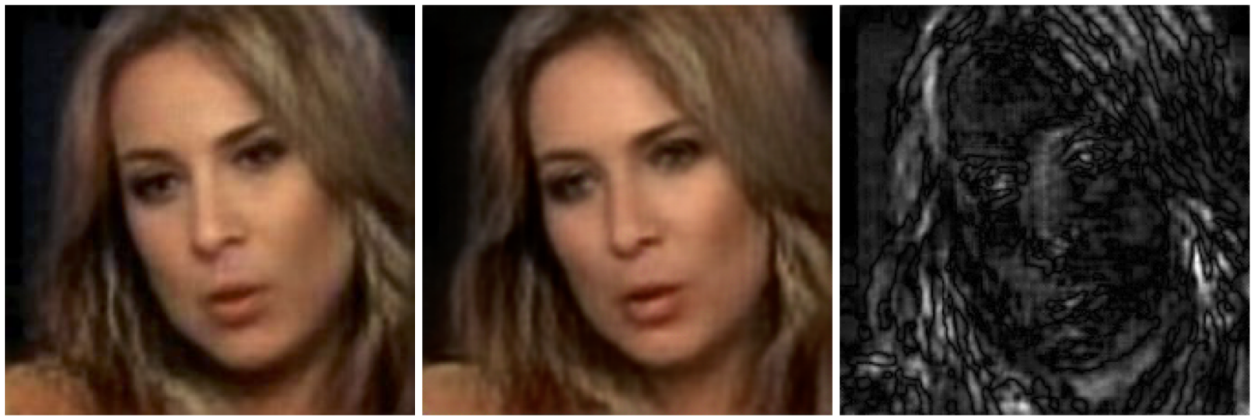


Figure 2.22. Ablation study of different model component settings.

the SSIM scores show the similar performance of the three experiments, the FID scores indicate the improvement when using these components. Adding the self-attention module reduces the FID from 63.9 to 35.3 and with the noise injection module, the FID drops to 27.1. Therefore, the two components indeed help improve the model performance. We also show the visual comparison of these experiments in Figure 2.22. The results without self-attention and noise injection contain blob-like artifacts that are also mentioned in [54]. In general, both of the results with and without noise injection achieve a good visual quality. However, the results without noise injection have some artifacts, as seen in the ear region in the first example and right shoulder region in the second example. As shown in Figure 2.23, noise injection can improve the reenacted image quality by adding high frequency details in the hair region. Therefore, the model with both self-attention and noise injection modules achieves the best image quality.



Without Noise Injection **With Noise Injection** **Difference Image**

Figure 2.23. Noise injection improves the image quality by adding high frequency details. The difference image has been rescaled for better visualization.

3. CHANGE DETECTION FOR SATELLITE IMAGERY

3.1 An Attention-Based System for Damage Assessment Using Satellite Imagery

3.1.1 Overview

Natural disasters wreak havoc on nations. They kill approximately 90,000 people every year and affect 160 million people around the globe [70]. Furthermore, areas afflicted by weather and climate disasters sustain significant physical, social, and economic devastation. Short-term effects of disasters evolve into long-term ramifications that linger for years [70], [71]. Considering economic consequences alone reveals staggering figures. For example, the 2010 Haiti earthquake inflicted approximately $7.8 billion - $8.5 billion in damages to infrastructure [72]. In 2019, the United States endured fourteen distinct natural disasters whose overall damages each exceeded $1 billion [73]. Environmental climate analyses also indicate that the frequency and brutality of natural disasters will increase in the future due to climate change and rising greenhouse gas emissions [71], [74]. The impact of disasters is immediate and far-reaching.

With the increase in severity and regularity of disasters, preparation for disaster recovery and emergency resource planning is needed now more than ever. Emergency responders require rapid and reliable situational details to save disaster victims while ensuring their own safety during rescue efforts. Moreover, accurate damage estimates assist responders in determining evacuation plans and in preventing secondary disasters caused by collapses of damaged buildings. In the long run, damage assessment estimates also empower planning efforts for building and infrastructure repairs.

Very high resolution (VHR) satellite imagery is increasingly available due to an ever-expanding fleet of commercial satellites, such as DigitalGlobe's WorldView satellites [75]. VHR imagery enables detailed assessment of disaster damage at the building level. With recent improvements in machine learning methods, especially deep learning approaches, rapid analysis of large amounts of VHR satellite imagery is feasible, facilitating damage estimation and disaster relief efforts. In this section, we propose a deep learning model, Siam-U-Net-Attn, to quickly and accurately estimate the damage of a disaster. Our approach analyzes two satellite images of the same scene, acquired before and after the disaster. It then produces a mask showing buildings with labels that indicate different damage scale levels, as depicted in Figure 3.1.

Figure 3.1. Damage scale classification components. From left to right and top to bottom: pre-disaster input image, post-disaster input image, ground truth mask, and damage scale classification output mask. The green areas illustrate buildings with *no damage*, and the pink areas reveal *destroyed* buildings.

3.1.2 Related Work

The proposed method achieves building damage scale classification by localizing the building area and determining the level of individual building damage. Due to limited amounts of labeled data, most research addressing damage scale classification instead simplifies this multi-class task to a two-class change detection operation, which assigns a binary label, *damage* or *no-damage*, to each building. Existing research approaches that focus on change detection fall into several broad categories [76].

Algebra-based change detection techniques perform mathematical operations on image pixels to obtain a difference image. Such approaches, including image differencing [77] and change vector analysis [78], involve a threshold selection process to determine which components changed in a scene. Algebra-based change detection methods are relatively simple to implement, but they do not provide contextual information about the detected changes.

Transform-based change detection approaches transform event images. Image transforms, including a standard Principal Component Analysis (PCA) approach [79], strive to determine pertinent information for the change detection task. While transforming the images enables analysis of change in a different dimensionality, it also presents challenges in labeling regions of change in the event images themselves.

Classification-based change detection methods usually rely on larger amounts of labeled data. They easily extend to the multi-class damage scale classification task considered in this thesis. Gueguen *et al.* [80] propose a damage detection method that uses a tree-of-shapes representation [81] to capture contextual/spatial features. Other types of contextual features are used in [82], including normalized difference and soil adjusted vegetation indexes. Deep neural networks are also been used for contextual feature extraction. Xu *et al.* [83] and Fujita *et al.* [84] describe several models for this objective, including a single-stream model and a double-stream model (*i.e.*, Siamese network). Their models evaluate two input images of a scene, before and after a disaster. They then produce a single binary classification label, indicating whether the image contains *damage* or *no-damage*. Similarly, Nex *et al.* [85] propose a binary classification model based on DenseNet [86], modified to use dilated convolution [87] to achieve a larger receptive field. Mou *et al.* [88] and Lyu *et al.* [89] introduce Recurrent Neural Networks (RNNs) to jointly

learn spectral-spatial-temporal features for change detection. Connors *et al.* [90] design a semi-supervised method that uses a Variational Autoencoder (VAE) [91] to infer change detection labels without ground truth for every training instance. An unsupervised method proposed by Liu *et al.* [92] uses active learning [93] to construct training samples and a graph convolutional network [94] for change detection. However, none of these approaches produce pixel-wise classification masks.

Some approaches strive to construct building damage classification masks in an unsupervised manner. Sublime *et al.* [95] use an autoencoder model to learn the trivial differences (*e.g.*, illumination changes) between pre-disaster and post-disaster images. Then the non-trivial changes (*e.g.*, changes caused by the disaster) can be detected from the high reconstruction error. Doshi *et al.* [96] first train a building semantic segmentation model (supervised) for the pre-disaster and post-disaster images. Then they compare the difference between the corresponding building masks for damage assessment (unsupervised). Similarly, Jong *et al.* [97] utilize U-Net [53] for the building segmentation task. During change detection inferencing, they collect two sets of features from the trained U-Net (*i.e.*, , activations of different layers in the U-Net), given two query images. Then, the difference of the two sets of features forms the change detection map. Therefore, they do not need the ground truth of building changes. Another approach is a deep convolutional coupling network proposed by Liu *et al.* [98] that uses both optical and radar images for unsupervised change detection. They use an ad-hoc weight initialization for the network based on noise models of the optical and radar images to assist the model in learning the proper features during training.

Supervised classification methods constitute the final category of solutions for the change detection task. Demir *et al.* [99] propose a method that only requires the annotation of one image in a time series. They train a supervised classification model using a dataset constructed by an active learning approach [93]. Rudner *et al.* [100] use more information by fusing multi-resolution, multi-sensor, and multi-temporal information for flooded building segmentation. Chu *et al.* [101] apply deep belief networks (DBNs) [102] to produce a change detection map. Two DBNs are used for extracting features from the image regions that contain changes and do not contain changes, respectively. They compare the feature distances obtained from the two DBNs for each image patch to construct the change detection map. Papadomanolaki *et al.* [103] combine the U-Net model with a Long Short-Term Memory (LSTM) [104] model in order to use temporal information from multiple frames of satellite imagery. Compared to approaches that use only two input frames, their

model achieves better performance. Daudt *et al.* [105] propose using an encoder-decoder-based architecture to produce the change detection map. The decoder upsamples features extracted from the encoder to generate a mask indicating damage levels throughout the region under analysis. They also improve on this performance in [106] by combining the semantic segmentation task with the change detection task to achieve multi-task learning. They use two U-Net models in total; one for each task. The semantic segmentation U-Net utilizes one image (captured either before or after the change event) to produce the segmentation mask of objects of interest. The change detection U-Net utilizes two images (*i.e.*, , one taken before the change event and one taken after the change event) as well as the features extracted from the semantic segmentation model to produce the change detection mask. By fusing the features together, they achieve better performance in the change detection task. Weber *et al.* [107] propose a Siamese-based method inspired by Mask R-CNN [108]. They first use a shared ResNet model [9] to extract the features from the pre-disaster and post-disaster images. Then, they feed the concatenated features to the semantic segmentation head from Mask R-CNN to obtain the damage scale classification mask.

Inspired by [105], [106], [109], [110], we propose a model that combines the U-Net model with the Siamese model for multi-task learning. Different from the previous work [105], [109], [110], we use a U-Net model to learn the semantic segmentation of buildings while using the Siamese model to learn the damage scale classification. In doing so, we achieve multi-task learning of segmentation and change detection simultaneously. The use of the Siamese model allows us to reduce both the number of learned parameters and the size of the model in comparison to [106]. More specifically, we use a shared encoder for the segmentation and change detection tasks instead of two separate encoders as proposed in [106]. Additionally, we introduce a self-attention module that improves performance by incorporating long-range information from the entire image.

3.1.3 Proposed Method

We propose a Siam-U-Net-Attn model for damage classification and building segmentation, as shown in Figure 3.2. It is inspired by [53], [105]. One element of this architecture is a U-Net model that analyzes a single input image and produces a segmentation mask showing building locations in the input image. The U-Net model is a fully convolutional network that was proposed by [53] for

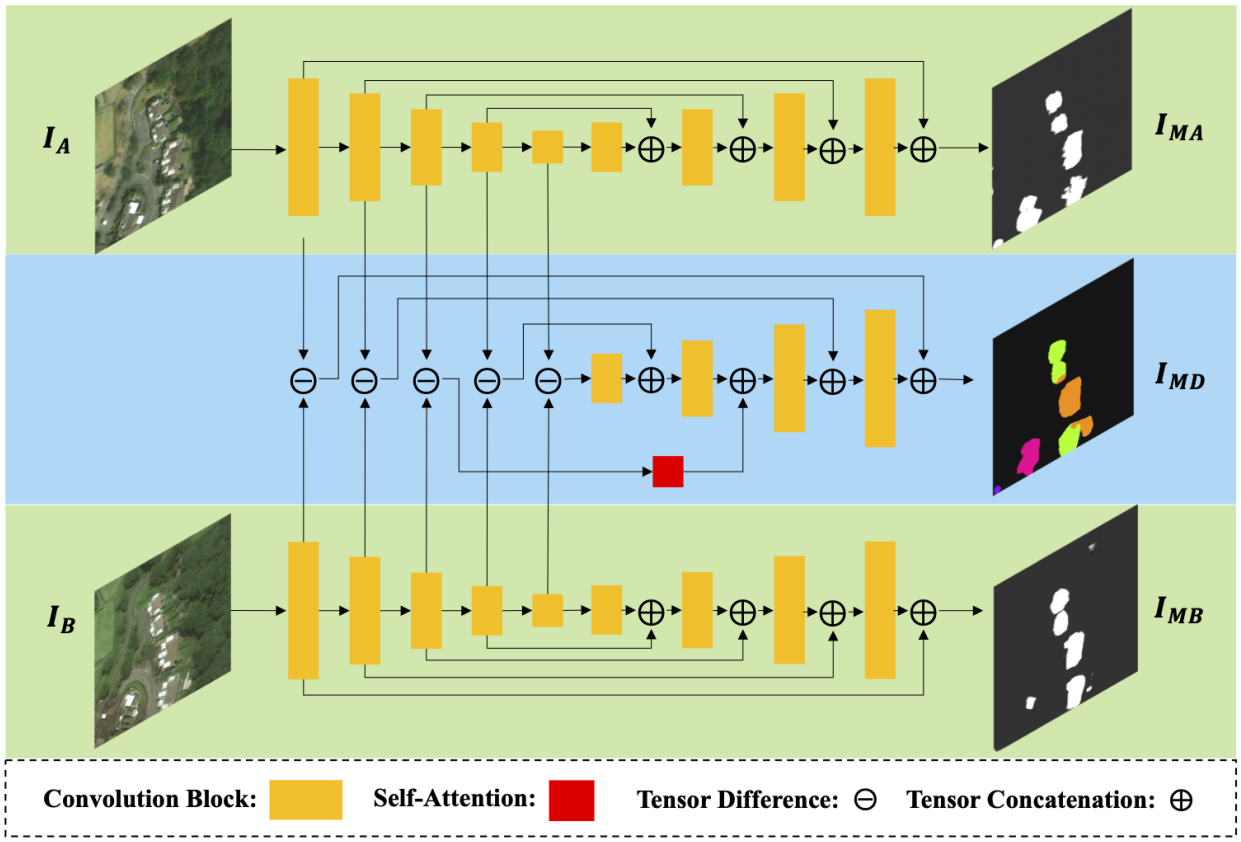


Figure 3.2. Architecture of proposed method: Siam-U-Net-Attn-diff. I_A and I_B are the pre-disaster and post-disaster input images. I_{MA} and I_{MB} are the corresponding output building segmentation masks. I_{MD} is the output damage scale classification mask. The green regions highlight the U-Net model. The blue region shows the decoder of the Siamese network.

image segmentation. Besides its encoder-decoder structure for local information extraction, it also utilizes skip connections to retain global information. A single U-Net model analyzes input frames I_A and I_B, which depict the same scene pre-disaster and post-disaster, respectively. Since the U-Net focuses on the building segmentation objective, it is agnostic to the disaster. In other words, we can use the same model for both pre-disaster and post-disaster images to produce binary masks I_{MA} and I_{MB}, corresponding to their respective input frames. The two green regions in Figure 3.2 indicate the shared U-Net model for I_A and I_B.

The features extracted from the encoder regions of the U-Net model also assist in the damage scale classification task. The two-stream features produced by the U-Net encoder and a new, separate decoder constitute the Siamese network, shown as the blue region in Figure 3.2. In the Siamese network, we compare features from the two input frames to detect the damage levels of buildings. Differencing and channel-wise concatenation are two methods to compare the two-stream features. By comparing features from the two frames, the Siamese model evaluates the differences between the features in order to assess the damage levels. Figure 3.2 shows the architecture of the Siam-U-Net-Attn in difference mode (*i.e.*, Siam-U-Net-Attn-diff). The Siam-U-Net-Attn in concatenation mode (*i.e.*, Siam-U-Net-Attn-conc) can be obtained by replacing the difference operations with channel-wise concatenation operations. In Section 3.1.5, we will compare the performance of the proposed model in difference and concatenation modes.

Analyzing a building by itself is not sufficient for accurate damage level classification. It is also necessary for the network to consider the area surrounding buildings in its assessment. For example, natural disasters such as floods may not damage a building's roof, but water surrounding the building may indicate interior damage. Since convolution is a local operation that can only access local neighborhoods, we use a self-attention module [66], [111] to capture long-range information. Figure 3.3 illustrates the mechanism of the self-attention module introduced in [66]. Assume the input feature map is $\mathbf{x} \in \mathbb{R}^{D \times N}$, where N is the flattened size of feature map along the height and width dimensions (*i.e.*, $N = H \times W$) and D is the number of channels of the input features. To compute the attention map, we first transform the input features into two feature spaces by:

$$\mathbf{f}(\mathbf{x}) = \mathbf{W_f x}, \quad \mathbf{g}(\mathbf{x}) = \mathbf{W_g x}.$$

The attention map is calculated as

$$\mathbf{a}(\mathbf{x}) = \text{Softmax}(\mathbf{f}(\mathbf{x})^T \mathbf{g}(\mathbf{x})).$$

The Softmax function is computed along the second dimension to normalize each row of the attention map. We then apply the attention map to the input features as:

$$\mathbf{o}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{a}(\mathbf{x})^T,$$

where $\mathbf{h}(\mathbf{x}) = \mathbf{W}_\mathbf{h}\mathbf{x}$.

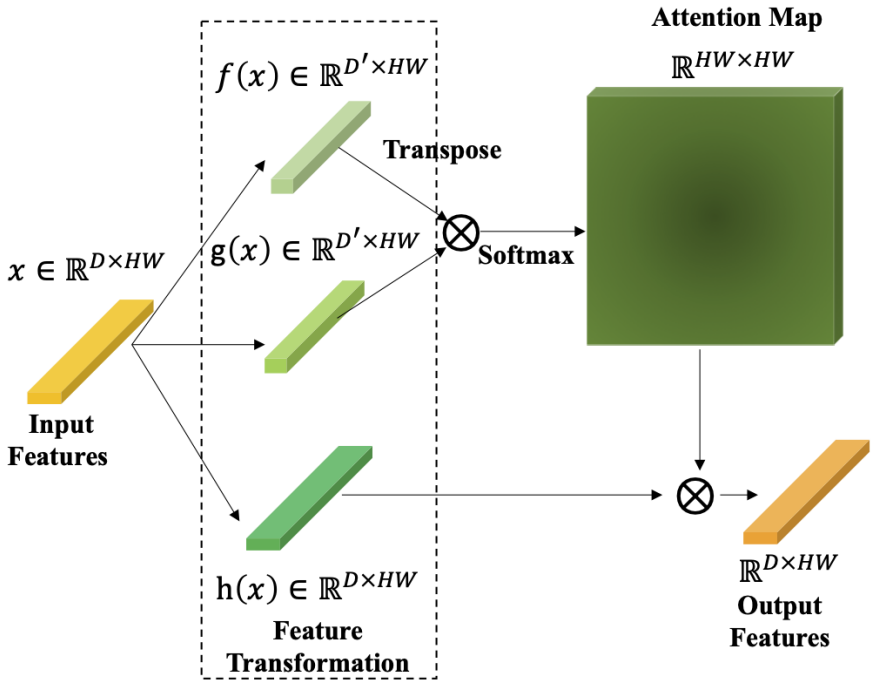


Figure 3.3. Architecture of the self-attention module. This modified figure is based on [66].

$\mathbf{W}_\mathbf{f} \in \mathbb{R}^{D \times D}$, $\mathbf{W}_\mathbf{g} \in \mathbb{R}^{D \times D}$, and $\mathbf{W}_\mathbf{h} \in \mathbb{R}^{D \times D}$ are trainable parameters that are implemented as the convolution operation with a kernel size of 1×1. Based on [66], we choose $D = D/8$ to

reduce memory usage. The final output of the self-attention module is a weighted summation of the original input with the attention feature:

$$\mathbf{y}(\mathbf{x}) = \gamma \mathbf{o}(\mathbf{x}) + \mathbf{x},$$

where $\gamma \in \mathbb{R}$ is also a learnable parameter. Therefore, each value of the self-attention output contains information of every input feature provided by the attention map. As shown in Figure 3.2, the model invokes a self-attention module after merging the features from the two input frames. It is important to note that the attention map from the self-attention module requires a lot of memory for large-resolution features, so we place the module in a low resolution layer of size 32×32 to reduce the memory usage.

3.1.4 Dataset

In this thesis, we use the xView2 dataset [6] for both training and testing. This dataset is designed for the task of building damage assessment and covers a wide variety of disaster events, including tsunamis, earthquakes, and volcanic eruptions. The training and validation data contains 2,799 pairs of pre-disaster/post-disaster, multi-band images with resolution 1024×1024 pixels. For testing, we use the xView2 challenge testing set that contains 933 pairs of images. The dataset contains ground truth building masks and classification labels indicating damage levels of the buildings. Buildings are labeled as *no-damage*, *minor-damage*, *major-damage*, or *destroyed*. [6] describes the scoring method used to assign damage levels to buildings to create the ground truth masks.

To reduce the memory usage during training and testing, we use image patches of size 256×256 as the inputs to our system. We crop every satellite image into 16 non-overlapping patches, each sized 256×256. More specifically, we first separate the full-resolution images into training and validation sets and then crop the full-resolution images into patches. This procedure ensures that the training and validation sets do not contain patches from the same full-resolution image. During training, we also use data augmentation methods (*i.e.*, horizontal/vertical flipping, random color jittering, and random cropping) to reduce overfitting. Random color jittering and cropping are applied independently to pre-disaster and post-disaster images to simulate poor image normalization

and registration. In our experiments, random cropping is implemented by upsampling the input image to 286×286 pixels and then randomly cropping it to a size of 256×256.

3.1.5 Experiment

As shown in Figure 3.2, our model consists of five convolution blocks for the encoder and decoder components. Inspired by the work in [112], we use SENet [113] with a ResNeXt50 [114] backbone. We utilize cardinality of 32 and internal dimension of 4 for the ResNeXt50 model. Note that the SE-ResNeXt-50 model is pretrained with ImageNet [115] for image classification. Each upsampling block consists of upsampling with bilinear interpolation, convolution, batch normalization, and ReLU layers. The final output damage scale classification mask has five channels, one for each of the four damage levels plus one *background* label.

Two-stage training is implemented to facilitate better learning. In the first stage, the U-Net component (used for building segmentation) is trained on only pre-disaster images. We do so because the damaged buildings that appear in post-disaster images may adversely affect the model's performance on the building segmentation task. In the second stage, the entire model (*i.e.*, U-Net and Siamese components) is finetuned to learn both building segmentation and damage scale classification with pre- and post-disaster images. As shown in Section 3.1.6, two-stage training yields better results on these two tasks overall.

Table 3.1. Class balancing weights. Weights of the binary cross entropy loss and multi-label cross entropy loss for the imbalanced building segmentation and damage scale classification tasks.

Weights	**Label**				
	0	1	2	3	4
w_s stage 1	1	1	-	-	-
w_s stage 2	1	10	-	-	-
w_d	1	10	30	30	30

For the loss functions, we use weighted binary cross-entropy loss and multi-label cross-entropy loss for the building segmentation loss $\mathcal{L}_s$ and damage scale classification loss $\mathcal{L}_d$, respectively, which are defined as:

$$\mathcal{L}_s = -(w_{s,1} y_s \log p_s + w_{s,0}(1 - y_s) \log (1 - p_s))$$
$$\mathcal{L}_d = -\sum_{c=1}^{5} w_{d,c} y_d(c) \log p_d(c)$$

y_s and p_s are the ground truth label and the detected building segmentation probability, respectively, while $y_d(c)$ and $p_d(c)$ are the ground truth label and the detected classification probability for damage scale c. w_s and w_d are weights applied to each class to address the class imbalances present in our dataset. Table 3.1 shows the empirical weights we use. For Stage 1, the binary cross-entropy loss with equal weights is used to train the U-Net. For Stage 2, we address the imbalanced classes issue by assigning higher weights to the building classes since most areas in our images do not contain any buildings. We also consider the frequency of damaged and undamaged buildings in xView2. Undamaged buildings are more common than damaged buildings in this dataset. Therefore, we select larger weights for the damaged building classes ($c = 2, 3, 4$) compared to the non-damaged buildings ($c = 1$) in the damage scale classification loss $\mathcal{L}_d$. As shown in Section 3.1.6, we can achieve a better damage classification performance with the weighted loss functions. The final loss function for Stage 1 is only the building segmentation loss for the pre-disaster images, and the final loss function for Stage 2 is the summation of the building segmentation loss and the damage scale classification loss.

The Adam optimizer [67] is used for training. We train our model for 50 epochs in Stage 1 with an initial learning rate of 0.001. The learning rate linearly decays to 0.0005 in the final epoch. In Stage 2, we train our model for another 100 epochs with an initial learning rate of 0.0001. This time, the learning rate linearly decays to 0 in the final epoch.

Since the models operate on image patches, the model results must be stitched together to create a full-resolution mask corresponding to the original image dimensions. We use a moving-window approach to infer full-resolution images from patches with overlapping regions. This inferencing method is only performed on images in the testing dataset, solely for the purpose of producing better and more coherent visual results. The goal of using overlapping regions is to reduce abrupt

edges at the boundaries of adjacent patches. The stride for the moving-window is 64 pixels in both the vertical and horizontal directions. The model analyzes these patches and produces corresponding segmentation maps. Next, we use a voting strategy for each pixel contained in the overlapping regions to determine the final segmentation mask. More specifically, we sum the probabilities of each class to calculate five overall probabilities that a specific pixel belongs to each of the damage level classes. Then, we label the pixel under consideration as the class with the maximum probability.

Table 3.2. Quantitative performance comparison. The damage scale classification performance (harmonic means of F1 scores for all damage scales), the building segmentation F1 scores, and overall F1 score for the proposed and compared methods.

Method	Damage F1	Segmentation F1	Overall F1
FC-EF [105]	0.451	0.732	0.535
FC-Siam-diff [105]	0.447	0.722	0.530
FC-Siam-conc [105]	0.487	0.752	0.567
Siam-Mask-RCNN [107]	0.697	**0.835**	0.738
Siam-U-Net-Attn-diff	**0.714**	0.823	**0.747**
Siam-U-Net-Attn-conc	0.707	0.817	0.740

To validate our method, we compare our results with those of two previous works [105], [107]. Daudt *et al.* [105] proposed three models: fully convolutional early fusion (FC-EF), fully convolutional Siamese-difference (FC-Siam-diff), and fully convolutional Siamese-concatenation (FC-Siam-conc). The FC-EF model is essentially the U-Net model we described in Section 3.1.3. Its input is I_A and I_B after concatenation along their channels. The FC-Siam-diff and FC-Siam-conc models utilize the Siamese model without the U-Net decoder used in the proposed method. These methods are designed for the change detection task and thus operate in a binary classification fashion. To compare these models with our proposed method, we change their output layers from binary classification layers to multi-class classification layers. Because the authors of these methods do not provide details about training parameters in their papers (*i.e.*, optimizer and learning rate), we train their models with the same specifications utilized for our method. Weber *et al.* [107] proposed a Siamese-based model based on Mask R-CNN [108]. They feed pre-disaster and post-

disaster images through a ResNet-50 backbone [9] with shared weights. Then they concatenate the ResNet-50 features for building damage scale classification.

Table 3.2 shows a quantified comparison of damage scale classification and building segmentation results from the xView2 challenge testing set [6]. To evaluate performance, we use the same evaluation metrics proposed by the challenge. The evaluation metric F1_s for the building segmentation task is defined as:

$$\text{F1}_s = \frac{2TP_s}{2TP_s + FP_s + FN_s}$$

where the TP_s, FP_s, and FN_s are the number of true-positive, false-positive, and false-negative pixels of segmentation results for the entire testing set. Since the compared methods only produce multi-class damage scale classification masks, we binarize their outputs to create segmentation masks for comparison purposes. The evaluation metric F1_d for the damage scale classification task is defined as the harmonic mean of the F1 scores for the four damage scales:

$$\text{F1}_d = \frac{4}{\sum_{c\in\{1,2,3,4\}}(\text{F1}_c + \epsilon)^{-1}},$$

where $\epsilon = 10^{-6}$ to avoid zero division and F1_c is the F1 score for the class c, which is defined as:

$$\text{F1}_c = \frac{2TP_c}{2TP_c + FP_c + FN_c}.$$

The TP_c, FP_c, and FN_c are the number of true-positive, false-positive, and false-negative pixels of the class c for the testing set. Note that this testing set does not include background pixels; it only includes pixels from the foreground as determined by the building segmentation ground truth. As used in the xView2 challenge, we define the final overall F1 as the weighted combination of 30% segmentation F1 and 70% damage F1.

Our proposed models achieve better overall performance than all compared methods, as can be seen in the third column of Table 3.2. The Siam-U-Net-Attn-diff model achieves the best performance overall and produces slightly better results than the model in concatenation mode. Additionally, our two proposed approaches outperform the compared methods for the damage

scale classification task. With the help of the self-attention module, the proposed methods produce better damage scale classification results using long-range information. On the other hand, the Siam-Mask-RCNN model achieves slightly better segmentation results than our proposed method. However, its overall performance is still worse than our proposed method because the overall F1 score weighs the damage scale classification performance more than the segmentation performance. Also, our proposed methods do achieve better performance than the three methods from [105] for the building segmentation task. Thus, both of the proposed models achieve better overall scores than all compared methods.

Figure 3.4 shows the damage scale classification results from the proposed models of the xView2 testing set. The first row shows a case with a single *destroyed* building and many buildings with *no-damage*. Both of the proposed methods, especially the Siam-U-Net-Attn-diff model, achieve accurate damage scale classification and segmentation for most of buildings, including the small objects. The second row depicts a more difficult case with high building density. In this example, there are a few *destroyed* buildings and plenty of buildings with *no-damage*. Despite the difficulty of this example, the two proposed methods also provide very good building classification and segmentation results. Based on the results of these two cases, we show from a visual analysis perspective that the proposed methods correctly localize building pixels and assign the correct damage labels to them. Thus, our proposed methods adeptly compare the degree of difference between pre-disaster and post-disaster images.

The third row depicts an even more challenging example with both shadow and cloud coverage. Although shadows engulf the buildings in the middle of the post-disaster image, the models still correctly classify the buildings as *no-damage*. Therefore, the proposed models are able to successfully distinguish between changes due to inflicted damage and changes due to illumination. Another challenge is present in this example in the form of cloud coverage. Although the clouds block some of the buildings in the bottom and top-right regions of the post-disaster image, our results show that both of the models still classify the occluded buildings as *no-damage*. Ground truth labels are not available for these buildings, so we assume that they are *no-damage* buildings, consistent with the buildings in the rest of the image. Based on this assumption, both methods still correctly assign the *no-damage* label to the buildings.

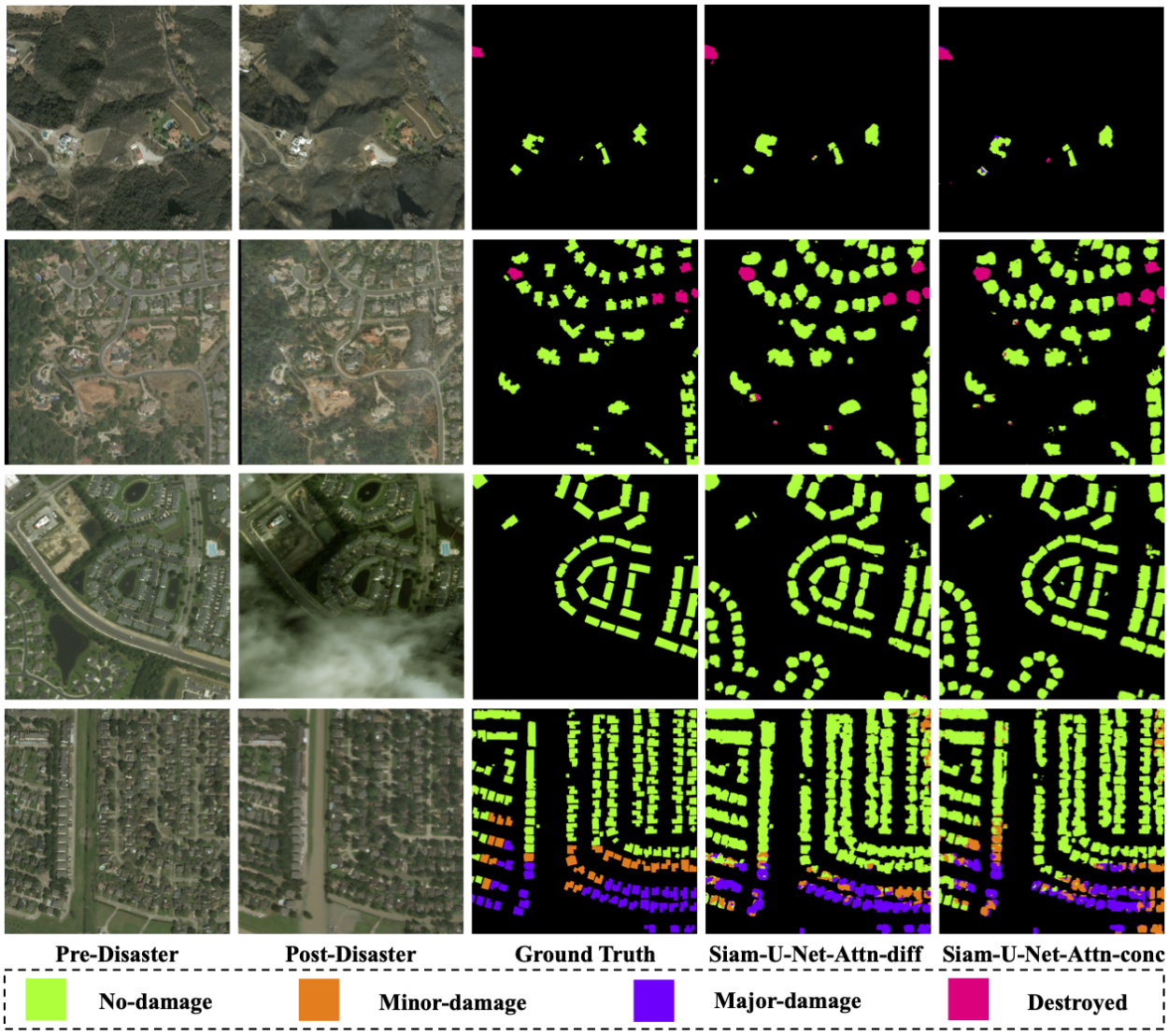


Figure 3.4. Damage scale classification results. From left to right: pre-disaster image patch, post-disaster image patch, ground truth mask, output of Siam-U-Net-Attn-diff model, and output of Siam-U-Net-Attn-conc model. Each row depicts a different scene from the xView2 challenge testing set.

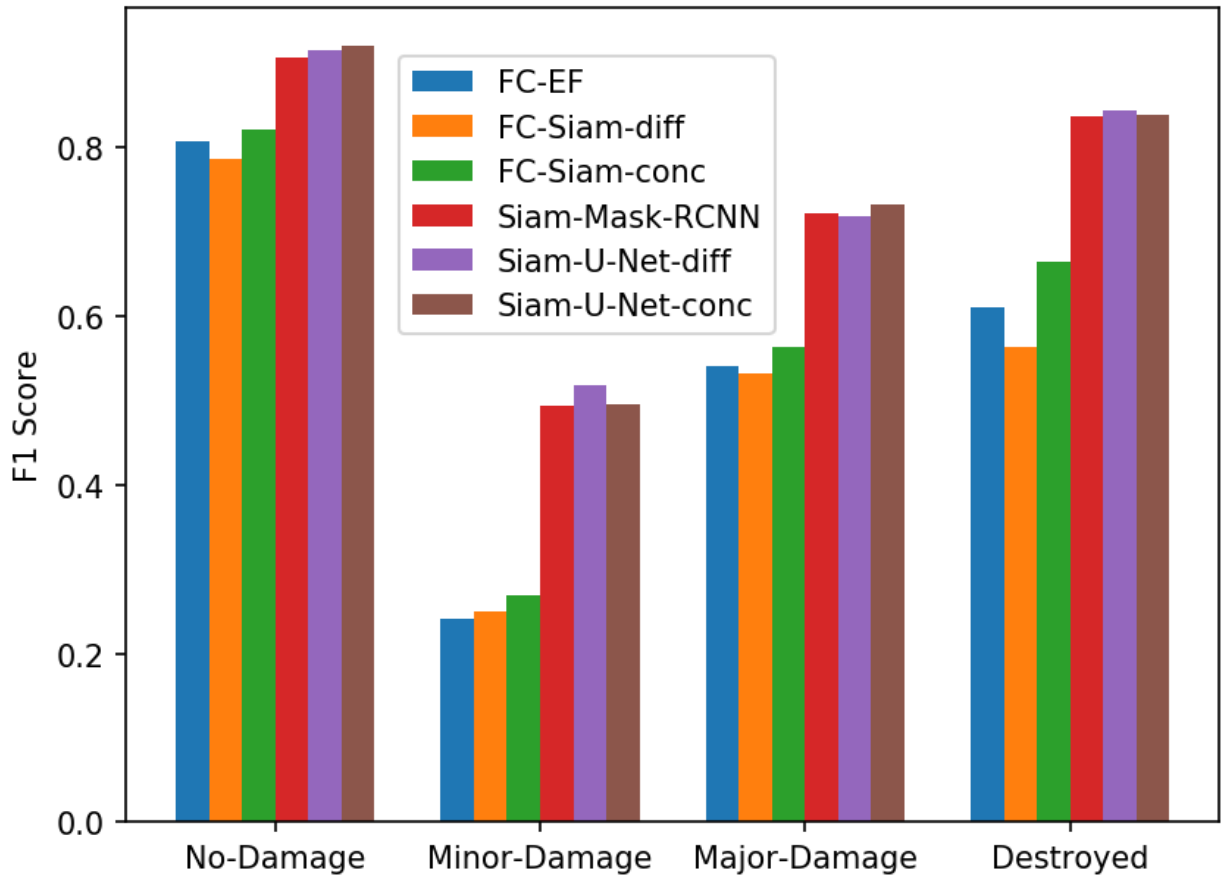


Figure 3.5. F1 scores based on damage scale level. These results indicate F1 scores for each damage scale level.

The last row showcases an example containing *no-damage*, *minor-damage*, and *major-damage* buildings. The yellowish region in the post-disaster image is a flooded region. Based on the damage scale definitions from [6], buildings entirely surrounded by the flood region should be classified as *major-damage* buildings, while buildings partially surrounded by the water or close to the water should be classified as *minor-damage*. Based on our results, the proposed methods classify most of the buildings correctly. However, there are still some misclassifications involving *minor-damage* and *major-damage* buildings. In order to correctly classify these buildings, the model not only needs to measure the damage of buildings themselves but also quantify the distance between the damage indicators (*e.g.*, flooding region, mud, or volcano flow) and the buildings. In general, the proposed models still struggle with differentiating between *minor-damage* and *major-damage* buildings.

For further analysis, we plot the damage F1 for each damage scale level in Figure 3.5. Overall, the proposed methods perform better than the compared methods for all damage scale levels. Most of the methods achieve the best performance on buildings with *no-damage* and achieve the worst performance on buildings with *minor-damage*. As mentioned earlier, *minor-damage* buildings present the greatest challenge to a classification model because these cases do not usually exhibit visible damage on the buildings themselves. Damage assessment experts from [6] consider buildings as *minor-damage* when they are partially surrounded by indicators such as flooding regions, volcano flow, or burned trees. Similarly, buildings should be classified as *major-damage* buildings when such elements completely surround them. Thus, these two similar damage scale levels present a more significant challenge to damage scale classification models.

Table 3.3. Ablation study of self-attention module.

Method	Damage F1	Segmentation F1	Overall F1
Siam-U-Net-diff	0.675	0.822	0.719
Siam-U-Net-Attn-diff	0.714	0.823	0.747
Siam-U-Net-conc	0.701	0.820	0.737
Siam-U-Net-Attn-conc	0.707	0.817	0.740

An ablation study of the self-attention module was conducted to demonstrate its effectiveness. The results of the study are shown in Table 3.3. The network incorporates the self-attention module in the Siamese part of the network, so attention does not significantly affect the building segmentation results. Table 3.3 reveals that the model with the self-attention module outperforms the model without attention on the damage scale classification task. A more significant increase in performance is observed for the Siam-U-Net-Attn-diff model. Therefore, these results indicate that using attention to more explicitly leverage information from the entire image improves the damage scale classification performance without downgrading the building segmentation results. The utility of the self-attention module can also be visualized. We portray an attention map in Figure 3.6 to demonstrate the effectiveness of the self-attention module. For a given query location (*i.e.*, , the red point in the post-disaster image patch), we obtain the corresponding attention map. Pixel values in the attention map indicate the importance of that pixel to the query point. The brighter a pixel is, the more important it is for classifying the query point. In the area shown in the example, the brownish-yellowish area in the post-disaster image patch indicates the flooding region. According to the attention map, the self-attention model highlights this flooding area, which aids the model in classifying the buildings' damage levels.

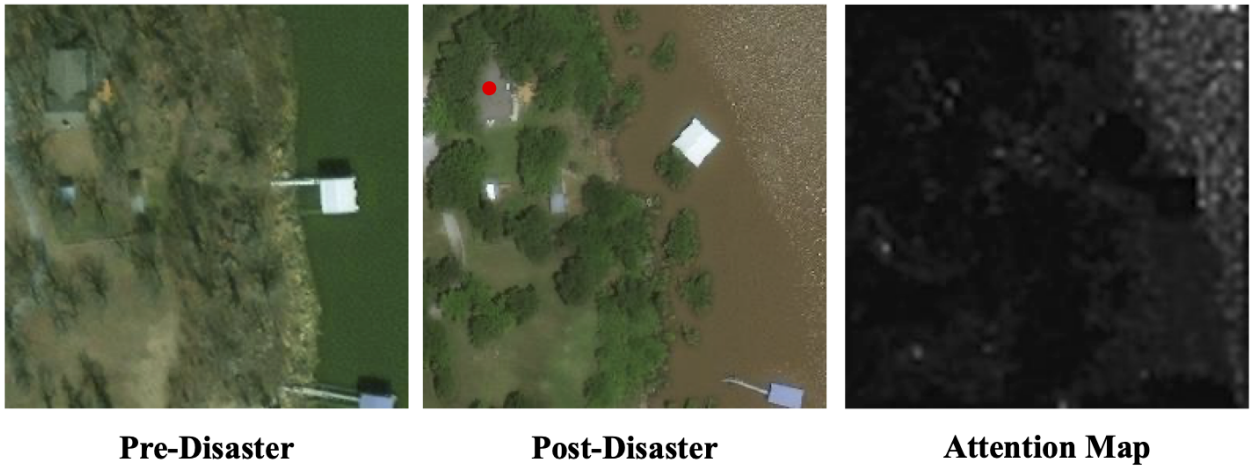


Figure 3.6. Attention map visualization. From left to right: pre-disaster image patch, post-disaster image patch with query point (*i.e.*, , the red point), and attention map associated with the given query point. Brighter regions in the attention map signify greater importance of those pixels to the classification of the query point.

3.1.6 Ablation Study

Ablation Study on Two-Stage Training. Table 3.4 shows the results of single-stage training and two-stage training. Recall that in two-stage training, we first train the U-Net part of the model for building segmentation and then include the Siamese model with the weighted loss function for building damage scale classification. For single-stage training, we train the entire model (U-Net and Siamese parts) together with the same weighted loss function. As shown in Table 3.4, two-stage training yields better overall performance for both Siam-U-Net-diff and Siam-U-Net-conc model. Although it does not have a significant effect on the damage scale classification task, it greatly improves the building segmentation performance. This is because the U-Net model is first trained on the segmentation task and then finetuned with the damage scale classification task, which reduces the training complexity compared to when learning both tasks at the same time.

Table 3.4. Ablation study on two-stage training. Two-stage training can improve the building segmentation performance.

Model	Setting	Damage F1	Segmentation F1	Overall F1
Siam-U-Net-diff	Single-Stage	0.711	0.796	0.736
	Two-Stage	0.714	0.823	0.747
Siam-U-Net-conc	Single-Stage	0.710	0.803	0.738
	Two-Stage	0.707	0.817	0.740

Ablation Study on Weighted Loss Function. We use the weighted loss function to address the class imbalance issue by considering the frequency of damaged and undamaged buildings in the xView2 dataset. Undamaged buildings are more common than damaged buildings in this dataset. Therefore, we select larger weights for the damaged building classes compared to those for the non-damaged buildings in the damage scale classification loss. As shown in Table 3.5, with the weighted loss function, both Siam-U-Net-diff and Siam-U-Net-conc achieve better overall performance than the experiments without weighted loss function. Due to the uneven weights of the building segmentation loss, the model focuses more on the building area, which causes larger building masks and a lower segmentation F1 score, as shown in Figure 3.7. However, despite the downgrade of performance on the segmentation task, using the weighted loss function

causes a significant improvement in the damage scale classification task, leading to better overall performance.

Table 3.5. Ablation study on weighted loss function. The weighted loss significantly improves the performance in the damage scale classification task. Thus, despite the slight downgrade in building segmentation performance, the overall F1 score also improves greatly.

Model	Setting	Damage F1	Segmentation F1	Overall F1
Siam-U-Net-diff	No Weight	0.601	0.862	0.680
	Weighted	0.714	0.823	0.747
Siam-U-Net-conc	No Weight	0.608	0.864	0.685
	Weighted	0.707	0.817	0.740

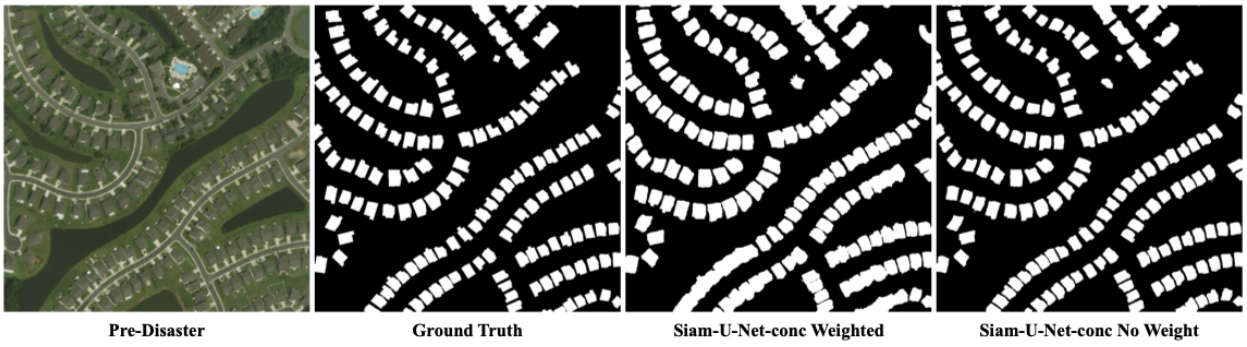


Figure 3.7. Building segmentation result with and without weighted loss Function. From left to right: pre-disaster image, building segmentation ground truth, segmentation output of Siam-U-Net-Attn-conc model with weighted loss, and segmentation output of Siam-U-Net-Attn-conc model without weighted loss.

3.2 Building Height Estimation via Satellite Metadata and Shadow Instance Detection

3.2.1 Overview

The previous proposed method considers the building appearance changes in images. However, only considering the changes of building appearance sometime is not enough to capture all information required for a change detection application. In many real-world applications, detecting the

change of building height is also important. Therefore, in this thesis, we also propose a building height estimation model to detect the building height changes. Understanding the 3D geometry of buildings via satellite imagery plays an important role in applications including urban growth analysis, building footprint detection for off-nadir imagery, and change detection. Traditional approaches estimate building 3D models (*e.g.*, above ground height) by relying on LiDAR sensors or a digital surface model (DSM), which might not be available in many scenarios. Building height can also be estimated via multi-view stereo, but obtaining a stereo image set is also a challenging task in many situations.

To address this issue, we focus on single-view building height estimation using RGB satellite imagery. Without using other information, reconstructing height information from a single image is an ill-posed problem, since the 3D information is lost when projecting onto 2D image. The previous work [7], [116]–[118] investigates this problem by learning from large-scale datasets with height annotation from LiDAR and/or DSMs. This data provides prior knowledge for the model to learn a proper mapping function between 2D satellite imagery and 3D building models. However, these data-driven based methods might not be generalizable to new data, since this prior knowledge learned from the data is likely to change for different datasets, such as images taken from different places or at different times. Another group of methods aim to use more reliable clues instead of learning solely from the data. Prior approaches [119]–[125] as well as our proposed method learn to estimate building height using building shadows and satellite image metadata. By detecting a building and its corresponding shadow area, given the solar direction information from satellite image metadata, we can estimate the above ground height for each building. The benefit of this type of method is that as long as the building and shadow detection are accurate, we can achieve a relatively accurate height estimation. These methods are more generalizable than the data-driven based methods and they work well with new data.

In this section, we present a method for building height[1] estimation using building shadows and satellite metadata[2]. Figure 3.8 shows the block diagram of the proposed estimation method. Since the shadow labels are hard to obtain in publicly available datasets, we design a multi-stage instance detection method to detect building and shadow instances with less required shadow annotation

[1] ↑We will use the term *building height* to mean *building above ground height* in the rest of this section.
[2] ↑We will use the term *satellite metadata* to mean *satellite image metadata* in the rest of this section.

than the previous work [126]. Then we use the detected instances with satellite metadata, including ground sample distance, solar angles, and satellite angles, to estimate building height. Building height estimation is done by maximizing the overlap between the projected shadow region given a query height and the detected shadow region.

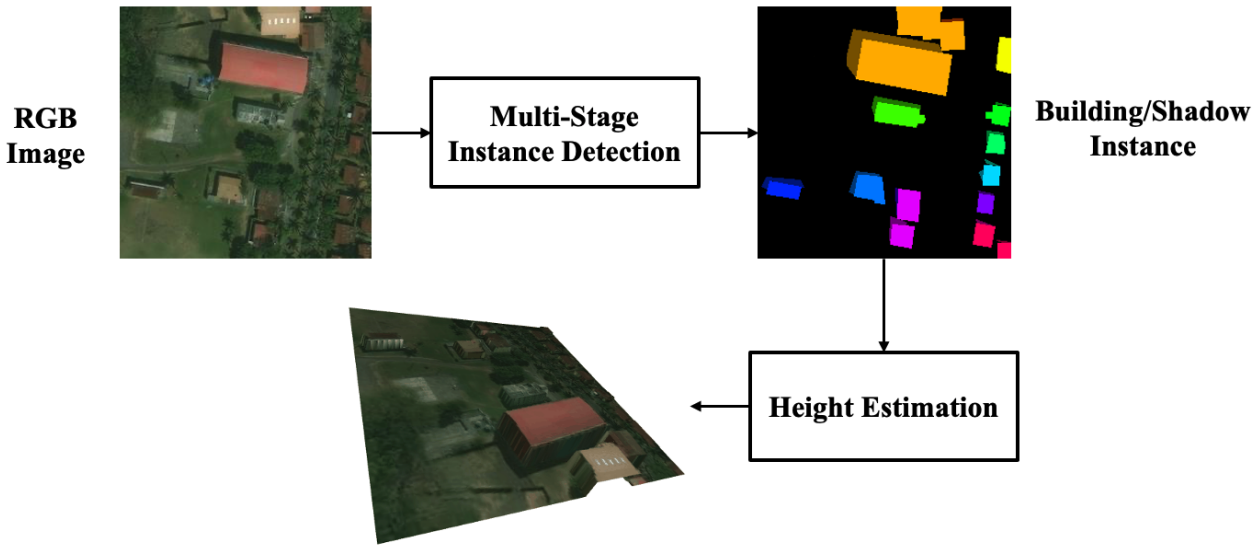


Figure 3.8. The block diagram of the proposed building height estimation method. Building and shadow instances are detected using a multi-stage instance detection model. Then, the height of each building is estimated using the detected instance masks and satellite metadata.

3.2.2 Related Work

In this section, we briefly review the previous approaches for single-view building height estimation. We group the previous work into two main categories: *direct learning-based approach* and *shadow detection-based approach*.

Height Estimation via Direct Learning [7], [116]–[118]. The direct learning-based approach tries to directly predict depth (*i.e.*, building height), supervised by the ground truth depth annotation (*e.g.*, LiDAR sensor). Although directly predicting building height from a single image is an ill-posed problem, Mou *et al.* [116] model this problem as a posterior distribution estimation $p(y|x)$, where x is the observed RGB image and y is the predicted DSM (*i.e.*, building height, in our case). To resolve this ill-posed problem, one can learn this posterior distribution as a determin-

istic mapping function $f : x \rightarrow y$ through the regression loss and directly using the data. In doing so, Mou *et al*. first show the possibility of estimating building height from a monocular remote sensing image. Semantic segmentation and depth estimation have a strong correlation, since both tasks require understanding the boundary of an object and the spatial relationship between objects. Srivastava *et al*. [117] propose a multi-task model that aims to learn both tasks jointly. Their model uses a shared CNN model for feature extraction that splits into two output layers to learn the semantic segmentation task and the DSM regression task. They show that this multi-task strategy yields better results than learning the tasks separately. Adversarial learning [127] is another potentially useful tool to solve this ill-posed problem by introducing a stochastic process to the model to learn the posterior function we mentioned earlier. Ghamisi *et al*. [118] introduce conditional generative adversarial networks (cGANs) [128] to this task. More specifically, they use a generator similar to a U-Net [53] to learn the posterior function $p(y|x)$, where x is the observed RGB image and y is the predicted DSM. They use the discriminator to differentiate the predicted DSM and real DSM obtained using a LiDAR sensor, which helps in learning the high-frequency details. They also use L_1 reconstruction loss to learn the low-frequency correctness. Christie *et al*. [7] also propose a U-Net like model for building height estimation. Their model jointly predicts image-level building orientation (caused by large satellite looking angle), height estimation, and flow vector magnitude. The flow vector can be obtained from building orientation and vector magnitude. It aims to map the pixels from the original image to their ground level (*i.e.*, mapping off-nadir image to on-nadir image). Similar to the multi-task method we mentioned earlier [117], the learning of the flow vector has a strong correlation to height estimation. Due to the large looking angle for some satellite images (*i.e.*, off-nadir imagery), the flow vector magnitude for tall buildings is greater than short buildings. Therefore, by learning these tasks jointly, they can achieve an accurate building height estimation.

Height Estimation via Shadow Detection [119]–[125]. A shadow is cast when an object blocks light from an illumination source. The physics of the illumination source position and the length of the shadow cast can inform us of the physical characteristics of the object, such as its height. The first step of this process is detecting shadows in overhead imagery. There has been existing work in using spectral features to identify shadows in multispectral data. A simple approach is using histogram thresholding to distinguish darker regions from brighter ones;

some of the techniques used are simple linear thresholding and bi-modal histogram splitting [129]. However, this approach can lead to dark objects being misidentified as shadows. Researchers subsequently developed a spectral ratio between hue and intensity to leverage color information to detect shadows [130]. More recently, contour and level set-based approaches [131], [132] have been used to segment shadow regions and have shown to be adaptable to topological changes. Once the shadows are identified, the second step is using the knowledge of the position of the sun to estimate building height. Irvin and McKeown used shadows along with the solar elevation angle to estimate the height of buildings [133]. A model can be constructed to show the projected shadow cast by the building if it had a certain height given the position of the sun. Kadhim and Mourshed [124] show that comparing this projected shadow with the detected shadow will allow for a good estimation of the building height.

3.2.3 Proposed Method

We propose a method for building height estimation using building and shadow instance detection with satellite metadata. As shown in Figure 3.8, the proposed approach contains two steps: 1) multi-stage building and shadow instance detection and 2) building height estimation. Although previous work also relied on shadow information for building height estimation, they mainly used either supervised detection methods [119], [122] or unsupervised detection methods [120], [123], [124]. Supervised methods can provide accurate detection accuracy, but require a heavy workload for annotations, like building and shadow instance labels. Unsupervised methods remove the requirement of annotation, but are sensitive to data noise, like changing of illumination, different satellite sensors, or urban growth. To solve this issue, we use a multi-stage approach for building and shadow instance detection, which uses both supervised and semi-supervised training schemes. By doing so, this requires less annotated samples than a fully supervised approach, while also being less sensitive to data noise compared to unsupervised methods.

Building and Shadow Instance Detection. As shown in Figure 3.9, our multi-stage instance detection contains two detection stages for building instance detection and shadow instance detection. These two stages rely on different training schemes, based on the availability of annotation for different tasks. We use Mask R-CNN [134] as the instance detection models in both stages.

In the first stage, we use a fully supervised method for building instance detection, since there are many large-scale publicly available datasets for building footprint and instance detection [6], [135]–[137]. We can use the building annotations from these datasets to train a Mask R-CNN for the building instance detection task.

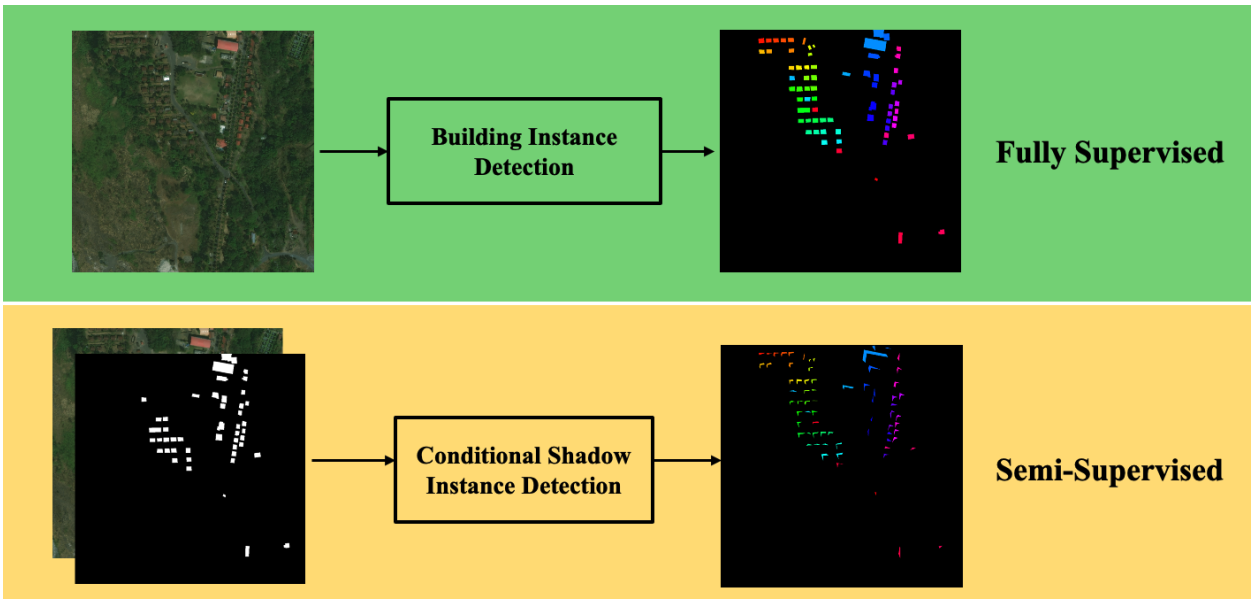


Figure 3.9. The block diagram of the multi-stage instance detection. Due to the sufficient annotation of building footprint detection in publicly available datasets, in the first stage, we use fully supervised training for building instance detection. To reduce the required shadow labels, in the second stage, we use semi-supervised training for shadow instance detection. The shadow detection model is conditional on the previously detected buildings, as it detects the shadows corresponding to the input building regions.

Shadow annotations are much harder to find from publicly available datasets. Although, there are unsupervised approaches [120], [123], [124] for shadow detection, they mainly focus on satellite spectral analysis and the detected shadow region is not robust and accurate. We also need to find the shadow instance for each building instance, which is not specifically obtained by these approaches. Wang *et al*. [126] introduce a task known as *Instance Shadow Detection*, which detects objects and their corresponding shadow instances. Although this instance shadow detection task is useful in our scenario, their method is a fully supervised approach that requires the labels of object instance, shadow instance, and their association relationship. These supervised methods require a

time-consuming annotation process, which is not suitable for our scenario. Our approach needs to detect a large number of small objects (*i.e.*, building instances and their shadow instances), which would require even more time to annotate and would be more susceptible to annotation errors. To address this, we extend the Mask R-CNN model to a conditional shadow detection model, as shown in the yellow block of Figure 3.9. Given a satellite image and its corresponding building detection mask obtained from the first stage, our shadow detection model aims to detect the shadow instances for the input building area. This forces the shadow detection model to only focus on the shadows around the input building regions, instead of detecting all shadow areas within the image. For this conditional shadow detection model, we concatenate the RGB satellite image with the binary building mask along the channel dimension as the input to Mask R-CNN.

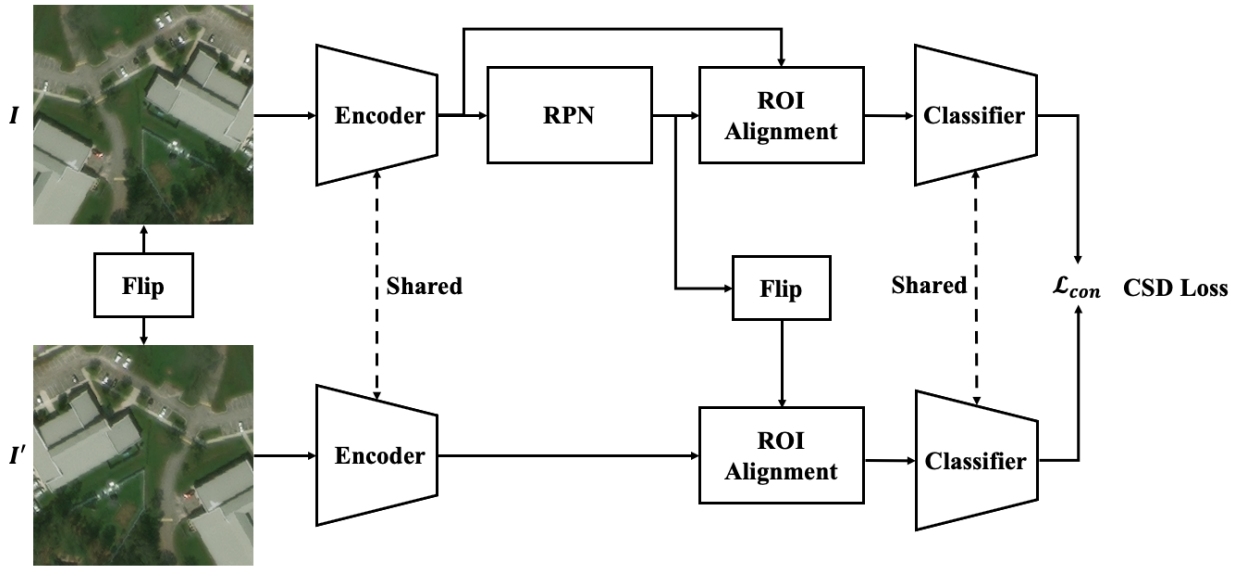


Figure 3.10. Consistency-based semi-supervised object detection (CSD) loss [138]. Given an unlabeled image, both original and its horizontally-flipped images are fed to a shared CNN encoder for feature extraction. Then the detected bounding boxes from the region proposal network (RPN) from the original image are used for both original and flipped features for feature cropping (*i.e.*, ROI Alignment). The cropped features are then fed to a shared classifier for object classification. The CSD loss is computed to minimize the output difference between the original and flipped results. For the bounding box detection task, we simply replace *Classifier* with the bounding box regression module.

Shadow Instance Detection with Semi-Supervised Learning. We use semi-supervised learning (SSL) so that shadow detection succeeds even with a small number of shadow annotations. More specifically, we use the consistency-based semi-supervised object detection (CSD) loss as proposed by Jeong *et al.* [138] during training. As shown in Figure 3.10, CSD loss minimizes the output difference between original and flipped images without annotation. Given an RGB image I and its horizontally flipped version I', we pass both images into the same CNN encoder Φ to extract features $\Phi(I)$ and $\Phi(I')$. Then the feature encoding of the original image $\Phi(I)$ is used with a region proposal network (RPN) [139] to extract bounding boxes for feature cropping. As described by Jeong *et al.* [138] , we horizontally flip these bounding boxes before cropping from the flipped image. The feature cropping is done by ROI Alignment [134] , which can be formulated as $r(\Phi(I), h)$ and $r(\Phi(I'), h')$ for the original and flipped images, respectively. Note that h is the bounding boxes from RPN and h' is obtained by horizontally flipping h. Then these cropped features are input to the classification layers for object classification, which can be formulated as $g(r(\Phi(I), h))$ and $g(r(\Phi(I'), h'))$ for the original and flipped images, respectively. In our case, we use binary classification to classify the object as either *shadow* or *background*. Similarly, for the bounding box detection task, we input the cropped features into the bounding box regression module to obtain detected bounding boxes, which can be formulated as $f(r(\Phi(I), h))$ and $f(r(\Phi(I'), h'))$ for the original and flipped images, respectively.

As mentioned earlier, the CSD minimizes the output difference between the original and flipped images. According to Jeong *et al.* [138] , we define the CSD loss for the classification task as:

$$\mathcal{L}_{cls} = JS(g(r(\Phi(I), h)), g(r(\Phi(I'), h'))), \tag{3.1}$$

where $JS(\cdot,\cdot)$ is the Jensen–Shannon divergence measuring the similarity between two probability distributions. In our case, the distribution is the output probability maps. For the bounding box regression task, we define the CSD loss as:

$$\begin{aligned}\mathcal{L}_{box} = \frac{1}{4}(&\|f(r(\Phi(I),h))_{\delta x} - (-f(r(\Phi(I'),h'))_{\delta x})\|^2 + \\ &\|f(r(\Phi(I),h))_{\delta y} - f(r(\Phi(I'),h'))_{\delta y}\|^2 + \\ &\|f(r(\Phi(I),h))_{\delta w} - f(r(\Phi(I'),h'))_{\delta w}\|^2 + \\ &\|f(r(\Phi(I),h))_{\delta h} - f(r(\Phi(I'),h'))_{\delta h}\|^2),\end{aligned} \tag{3.2}$$

where $f(\cdot,\cdot)_{\delta\cdot}$ is the bounding box offset output with respect to anchor boxes [139]. Note that we flip the sign of $f(r(\Phi(I'),h')_{\delta x}$ in Equation 3.2 to consider the flipping process we mentioned previously. During training, for each mini-batch, we sample the labeled and unlabeled data with equal probability. Then we use the original supervised loss functions [134] for the labeled data and use the CSD loss functions ($\mathcal{L}_{cls}$ and $\mathcal{L}_{box}$) for both labeled and unlabeled data.

Building Height Estimation. Before estimating building height, we need to pair the detected building and shadow instances. Although the conditional shadow instance detection method is able to detect shadows corresponding to the input building mask, it can detect multiple shadow instances for a single building instance or detect a shadow instance that belongs to multiple building instances. To properly pair the shadow and building instances, we use the solar azimuth angle as the direction to find the best shadow instance for each building and to find the best building instance for each shadow instance. First, we compute the distances between a building instance and all detected shadow instances[3]. Then, we move the building instance along the solar azimuth direction for a pre-defined maximum building-shadow distance (we choose 20 pixels in our experiment) to find the shadow instances it intersects with. The best shadow instance is the intersected shadow instance with minimum building-shadow distance. We can use the same procedure as described above to find the best building instance for each shadow instance. Then we combine the best building to shadow pairs and the best shadow to building pairs to match each building instance with a set of shadow instances.

[3]↑We use the geometry distance function implemented in https://shapely.readthedocs.io/en/stable/manual.html#object.distance.

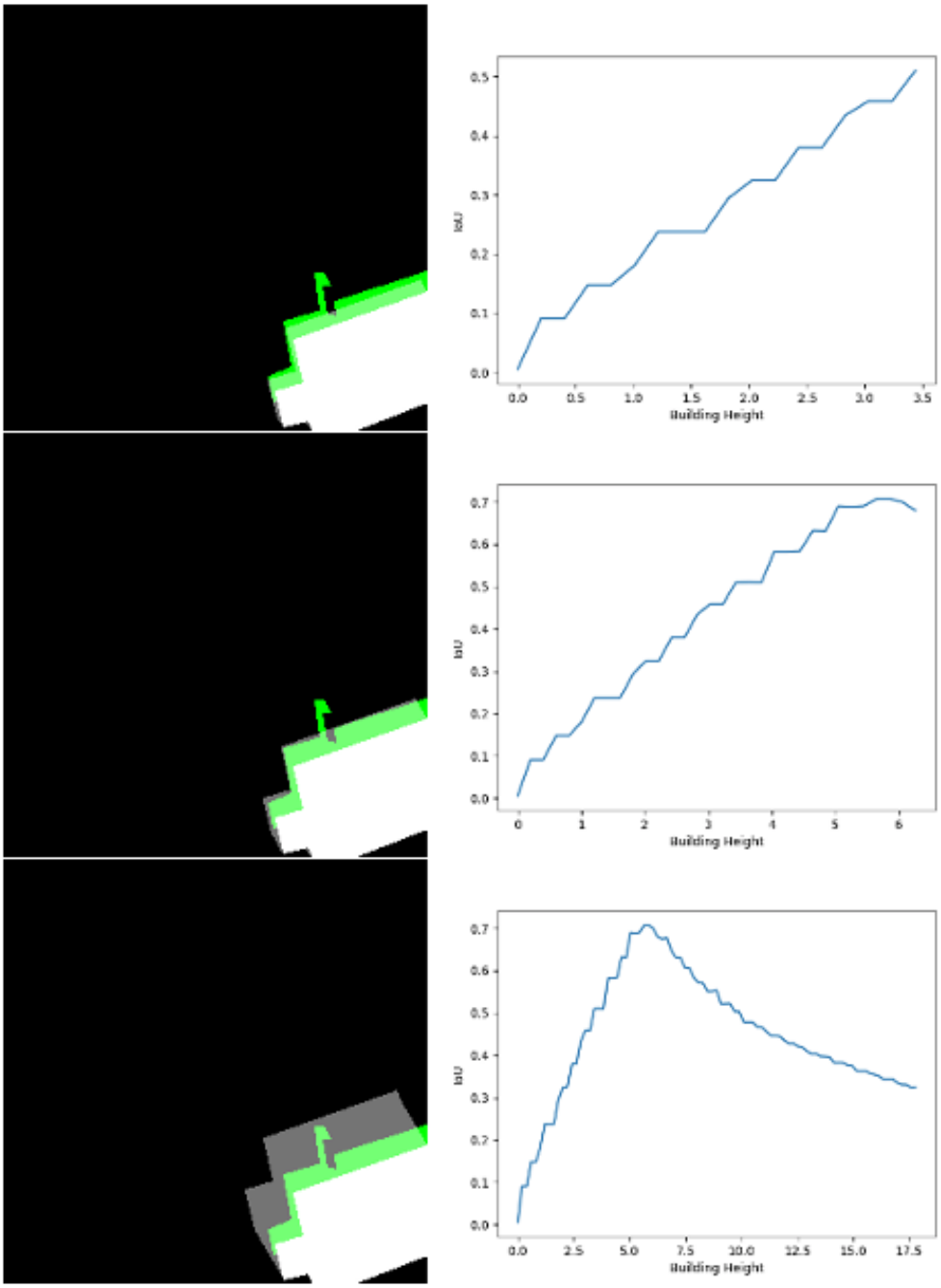


Figure 3.11. The approach used for estimating building height given detected building instance (white) and corresponding shadow instance (green). By enumerating different heights, we can compute the projected shadow area (gray) and the best building height can be obtained by maximizing IoU between the detected shadow area (green) and the projected shadow area (gray).

With the paired building and shadow instances, we can estimate the height for each building instance. Similar building height estimation approaches exist in the literature [120], [124]. Figure 3.11 shows the building height estimation method. Given the detected building region (white), we can estimate the projected shadow region (gray) given different height values. This projection process requires satellite metadata information, including ground sample distance, solar elevation, and solar azimuth. The estimated building height can be obtained by maximizing the intersection over union (IoU) between the projected shadow region (gray) and detected shadow region (green), as shown in the plots from Figure 3.11.

It is computationally intensive if we choose to enumerate all height values within a range of interest. To speed up this process, we assume the intersection over union (IoU) distribution (*i.e.*, IoU vs. height) to be concave within the range of interest. Then we can use a one-dimensional search method for convex/concave functions to find this global maximum with less query time[4]. In this thesis, we use Brent's method [140] implemented in *SciPy* [141] to find the global maximum in a bounded range of interest.

The approach discussed above assumes the satellite looking angle (*i.e.*, off-nadir angle) is 0, which usually is incorrect. To obtain a more accurate height estimation, we refine the previous estimated height using the satellite looking angle, based on the method shown by Liasis *et al.* [123]. Figure 3.12 illustrates the height refinement. Given the previously estimated building height h, we can compute the average visible shadow length sl_1 by

$$sl_1 = \frac{h}{\tan(a)}, \tag{3.3}$$

where a is the solar elevation angle. Similarly, the average hidden shadow length sl_2 can be obtained by

$$sl_2 = \frac{h}{\tan(b)}, \tag{3.4}$$

where b is the satellite elevation angle. Then we can find the real shadow length by $sl = sl_1 + sl_2$ and the refined building height h_{refine} can be obtained by

$$h_{refine} = \tan(a) \cdot sl. \tag{3.5}$$

[4]↑Query time is the number of times we compute IoU given a certain query height value.

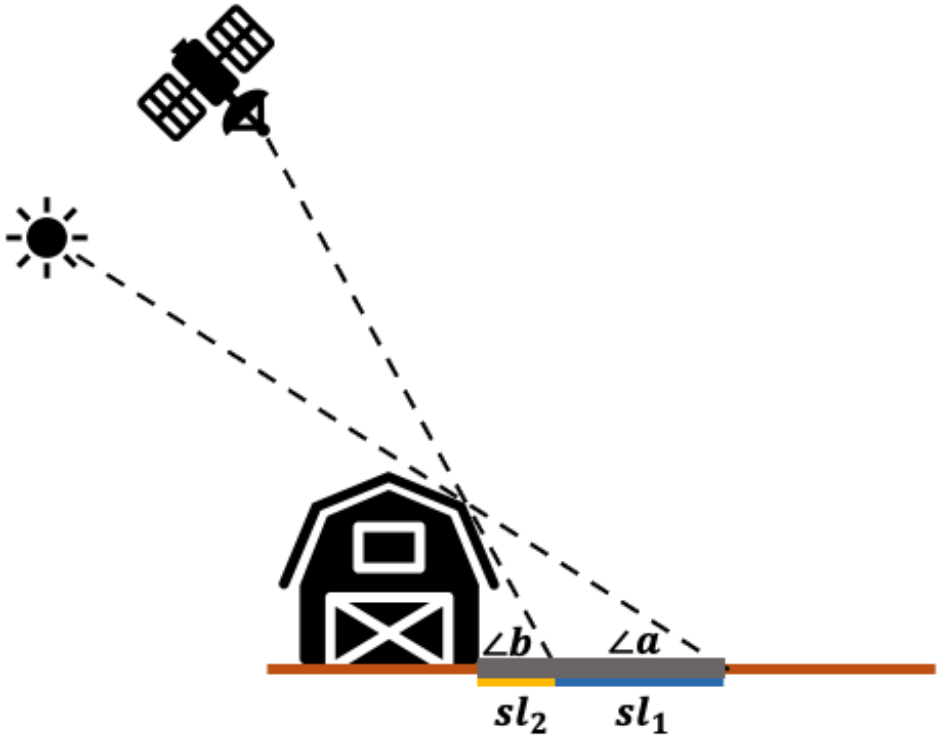


Figure 3.12. Illustration of building height refinement. This method takes the satellite elevation angle into account to refine the estimated building height obtained previously. $\angle a$ and $\angle b$ are the solar and satellite elevation angles. sl_1 is visible shadow length (blue section), while sl_2 is hidden shadow length (yellow section) blocked by the building.

Note that this refinement process is only valid when $\angle a < \angle b < 90^\circ$ or $\angle a > \angle b > 90^\circ$. For the invalid cases, we directly use the estimated height without refinement.

3.2.4 Dataset

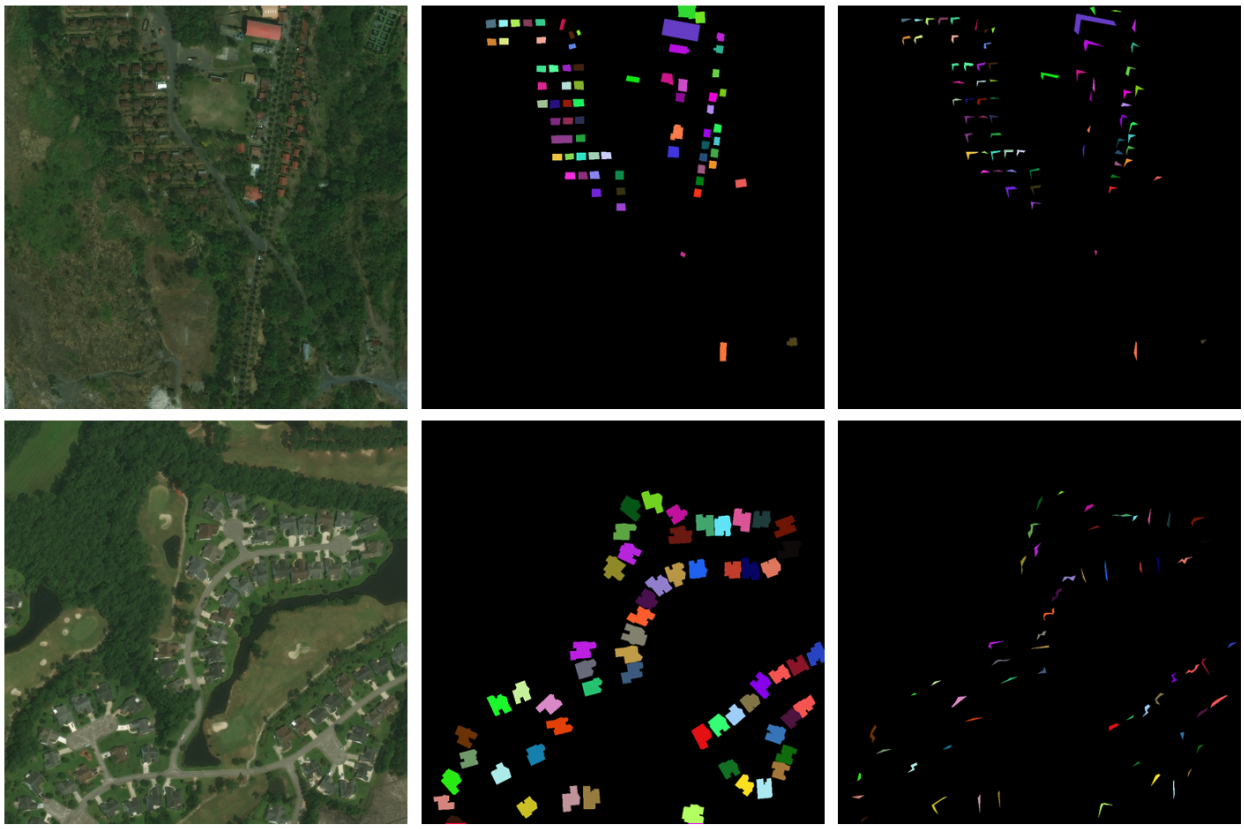


Figure 3.13. Two examples from our shadow instance detection dataset. From left to right: RGB satellite image, building instance mask, and corresponding shadow instance mask.

We use the large-scale satellite dataset xView2 [6] to train our method for the building instance detection task. This dataset is designed for the task of building damage assessment and covers a wide variety of disaster events. It has 5,598 images with a resolution of 1024×1024. This dataset also contains the required satellite metadata and the annotations of building instance detection. For the shadow instance detection task, we manually labeled the shadow instances given the building instances for 150 images from the xView2 dataset. As shown by the examples in

Figure 3.13, the shadow annotations are labeled based on the building annotations, since for the shadow detection task we described in Section 3.2.3, our model aims to detect the shadows conditionally on the building instances. Different colors shown in the building and shadow masks indicate different object instances. As shown in Table 3.6, our shadow detection dataset contains 1,854 shadow instances for training and 620 shadow instances for testing. Although the size of our shadow detection dataset is much smaller than the building instance detection dataset, the use of the conditional shadow detection model and semi-supervised learning can ensure our model still achieves a good performance as shown in Section 3.2.5. To reduce the memory use during training and testing, instead of inputting the original full resolution images to our model, we crop the images into 256×256 patches. During training, we crop the image patches with an overlapping ratio of 50%. During testing, we use a moving-window based inferencing approach with the same overlapping ratio of 50% to obtain the result for the full-resolution images. By using this moving-window based approach, we can reduce the abrupt changes near the edges of each image patch. For the overlapped area, we merge the detected instances using the *unary_union* function from *Shapely*[5].

Table 3.6. Statistics of our shadow instance detection dataset.

Dataset Split	# of Image	# of Building Instance	# of Shadow Instance
Training	100	1854	1854
Testing	50	620	620

3.2.5 Experiment

Figure 3.14 shows the results of building instance detection and conditional shadow instance detection. Compared to the ground truth in the second and third columns, we show that both models achieve accurate instance detection. To show the improvement of our multi-stage instance detection method and semi-supervised learning, we compare our result with the Instance Shadow Detection model [126] we mentioned in Section 3.2.3. Wang *et al.* [126] propose a Light-guided Instance Shadow-object Association (LISA) model, which is an extension of Mask R-CNN [134].

[5] ↑https://shapely.readthedocs.io

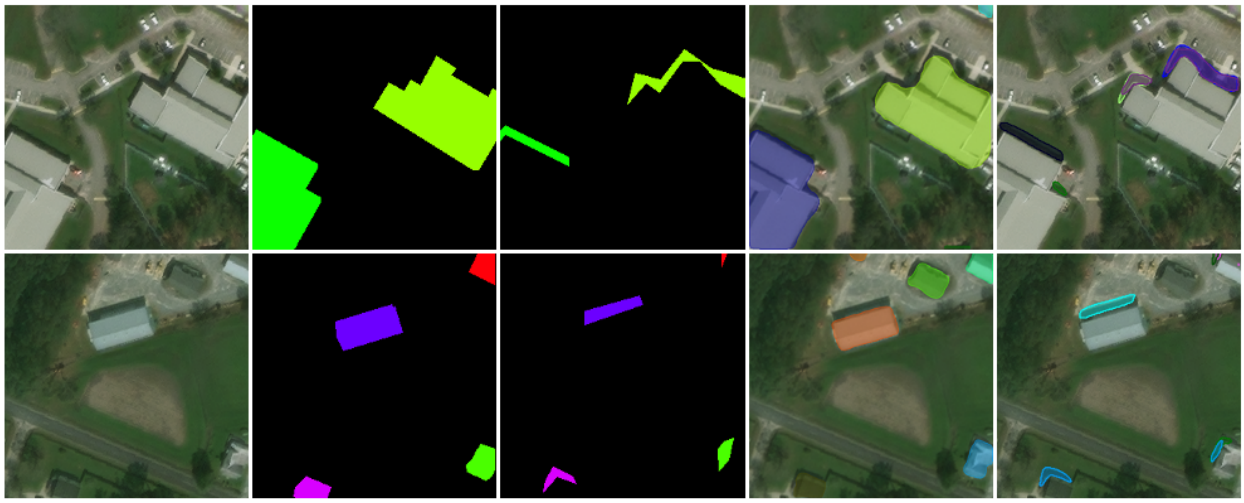


Figure 3.14. Multi-stage instance detection. From left to right: RGB satellite image, ground truth building instance, ground truth shadow instance, detected building instance, and detected shadow instance.

Given an RGB image, their model can detect object and shadow instances and their associations. Based on their official implementation[6], we train their model on our shadow instance detection dataset mentioned in Section 3.2.4.

Tables 3.7 and 3.8 show the quantitative comparison between our method and LISA [126] using the COCO evaluation metrics [7]. All of the results are evaluated using the testing set we described in Section 3.2.4. More specifically, we use average precision (AP) with different IoU threshold value τ to determine if the object is matched with the ground truth or not. For example, AP_{50} is the AP with $\tau = 0.5$ and AP is the mean of APs with $\tau \in \{0.5, 0.55, 0.60, ..., 0.95\}$.

As shown in Table 3.7a and Table 3.7b, the proposed method achieves better building instance detection results for both bounding box detection and mask segmentation tasks. This is because our method is trained with a much larger building instance detection dataset (*i.e.*, entire xView2 dataset). However, LISA is only able to use the smaller shadow instance detection dataset as proposed in Section 3.2.4 since it requires not only building annotation to train but also shadow annotation and building-shadow association. Therefore, due to the multi-stage design, our building

[6] ↑https://github.com/stevewongv/InstanceShadowDetection
[7] ↑https://github.com/cocodataset/cocoapi

instance detection model can be easily trained on large-scale building footprint detection dataset to achieve better performance on the testing set.

Table 3.8a and Table 3.8b show the quantitative comparison of LISA with the proposed conditional shadow instance detection with CSD loss (*SSL Ours*) and without CSD loss (*SL Ours*). Compared to LISA, our conditional shadow instance detection model achieves better results for all APs with different τ, and this holds even for our model without SSL training. Although for the shadow detection task, both LISA and our conditional shadow instance detection model use the same training set as mentioned in Section 3.2.4, with the input building mask obtained from the building instance detection stage, the detection model can obtain the region of interest information (*i.e.*, building region) directly from the input, which reduces the task complexity compared to detecting shadow instances from only the RGB image. Therefore, our models achieve better performance than the compared method. Moreover, with the help of CSD loss, we can further improve the result for both bounding box detection and mask segmentation tasks by including the unlabeled data. This shows that for our small shadow instance detection dataset, the semi-supervised learning method leverages the unlabeled data to enable our model to generalize better to new scenes.

Table 3.7. Testing average precision (AP) result of building instance detection.

(a) Testing average precision (AP) result of building bounding box detection.

Experiment	AP (%)	AP_{50} (%)	AP_{75} (%)
LISA [126]	39.093	65.222	41.598
Ours	**40.149**	**67.961**	**42.180**

(b) Testing average precision (AP) result of building mask detection.

Experiment	AP (%)	AP_{50} (%)	AP_{75} (%)
LISA [126]	35.660	64.193	35.335
Ours	**37.589**	**67.427**	**36.257**

Table 3.8. Testing average precision (AP) result of shadow instance detection.

(a) Testing average precision (AP) result of shadow bounding box detection.

Experiment	AP (%)	AP_{50} (%)	AP_{75} (%)
LISA [126]	5.746	17.627	2.105
SL Ours	18.846	51.455	8.536
SSL Ours	**24.698**	**63.912**	**11.572**

(b) Testing average precision (AP) result of shadow mask detection.

Experiment	AP (%)	AP_{50} (%)	AP_{75} (%)
LISA [126]	1.337	5.437	0.347
SL Ours	6.161	27.069	0.310
SSL Ours	**7.901**	**33.858**	**1.308**

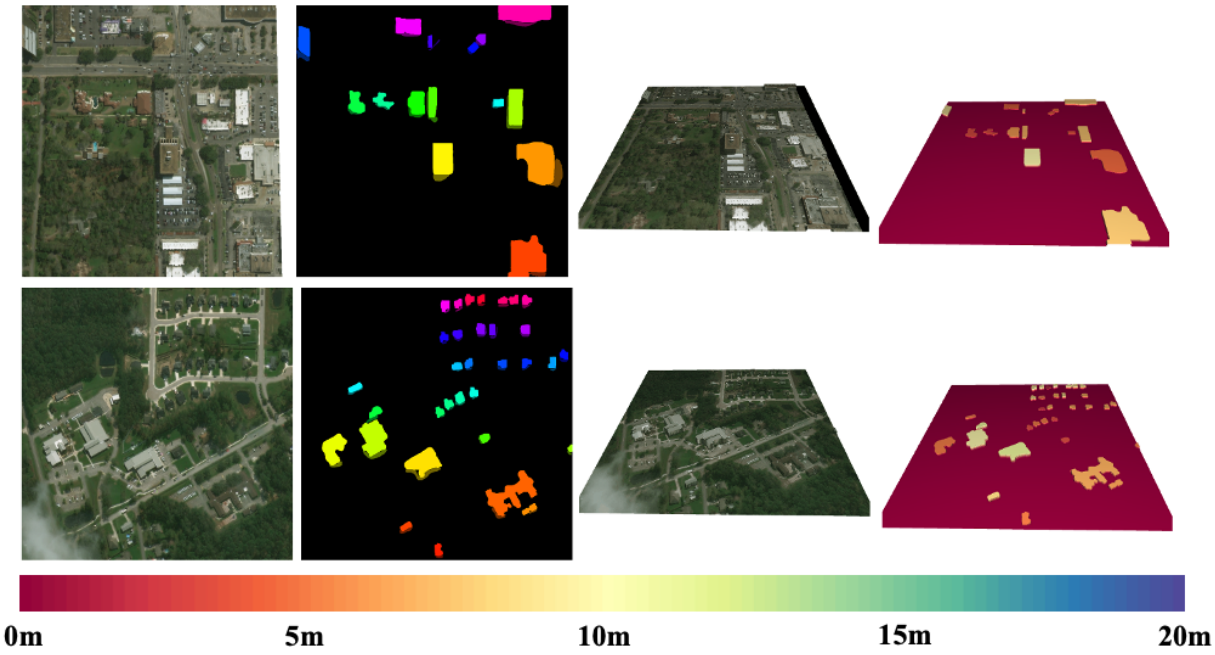


Figure 3.15. Building height estimation. From left to right: RGB satellite image, building-shadow instance detection result, height estimation with texture rendering, and height estimation with pseudo-color rendering (color legend is below the figures).

Figure 3.15 shows the results of building height estimation using the detected building and shadow instances and satellite metadata. From the height estimation result (third and fourth columns), we show that our method achieves good height estimation based on visual analysis. For example, the building in the center of the example on the first row is much higher than the buildings around it. Based on the pseudo-color result, the height estimation for the building on the center is about 10m, while the result buildings are around 5m. Similar results can be seen in the example in the second row. Please see Figure 3.16 for more height estimation results from our dataset.

To better evaluate our height estimation results, we test our model on a 3D reconstruction dataset known as the Urban Semantic 3D dataset [7]. Since building instance and shadow instance annotations are not available in the dataset, we cannot finetune our model on this dataset.

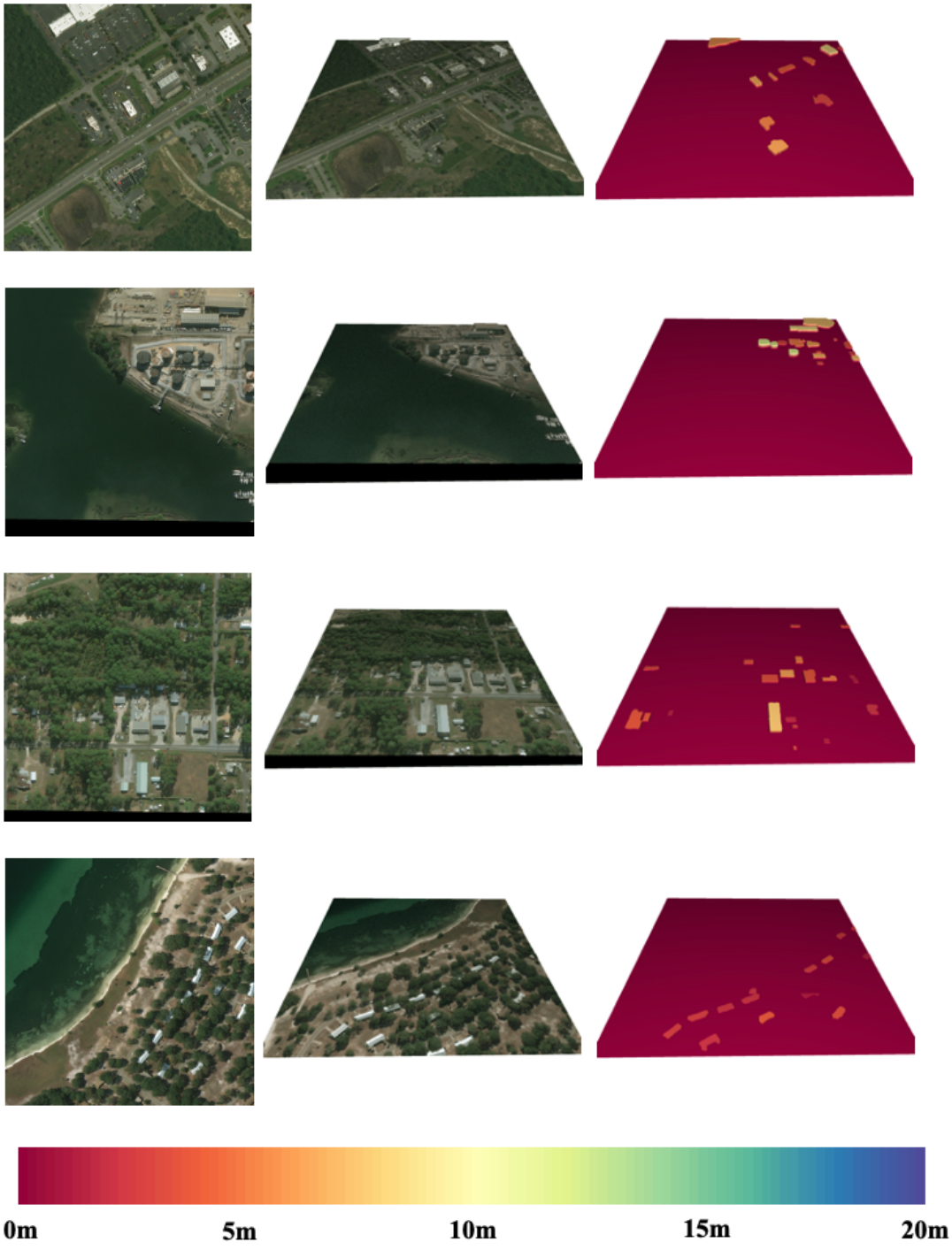


Figure 3.16. Building height estimation result from xView2 dataset. From left to right: RGB satellite image, height estimation with texture rendering, and height estimation with pseudo-color rendering (color legend is below the figures).

Therefore, we directly test our model on this data to evaluate our method[8]. The results shown in Figure 3.17 indicate that our model still achieves good performance on this new dataset even without training. Compared to the ground truth obtained from LiDAR sensor, our model can obtain a good height estimation. This is especially true for larger buildings, as their shadow regions are relatively larger than those of other buildings, making them easier to detect with the shadow detection model. Since we did not finetune the model on this new dataset, there are still several missed building instances and shadow instances. As we assign a single height value for an entire building area, we are not able to obtain detailed height information within each building area, such as the height difference between pixels from a building roof. Table 3.9 shows the quantitative results of the proposed method with and without the height refinement process discussed in Section 3.2.3. We use mean absolute error (MAE) and root mean square error (RMSE) to evaluate our method. Figure 3.18 shows the ground truth building height statistics of the data we used in our experiments. Compared to the range of ground truth height, the proposed method achieves relatively low height estimation error. From the third column of Table 3.9, we can also see that with the height refinement process, we can further reduce the estimation error for the building regions. However, this refinement process will exaggerate the error for the false positive buildings, which causes the increase of all-region MAE and RMSE (the second column of Table 3.9). Overall, based on these results, we show that our method achieves a really promising height estimation performance even without finetuning. Please see Figure 3.19 for more height estimation results from Urban Semantic 3D dataset.

Table 3.9. Quantitative evaluation (in meter) of height estimation on a subset of Urban Semantic 3D dataset [7] (Atlanta region). *MAE* and *RMSE* are computed for all regions; *MAE of Buildings* and *RMSE of Buildings* are computed for all building regions determined by ground truth.

Method	MAE	MAE of Buildings	RMSE	RMSE of Buildings
Before Refinement	1.28	8.12	5.09	13.66
After Refinement	1.35	8.02	5.27	13.51

[8]↑Due to the large number of buildings in the dataset, we only test our model on a subset of the Urban Semantic 3D dataset to reduce evaluation time.

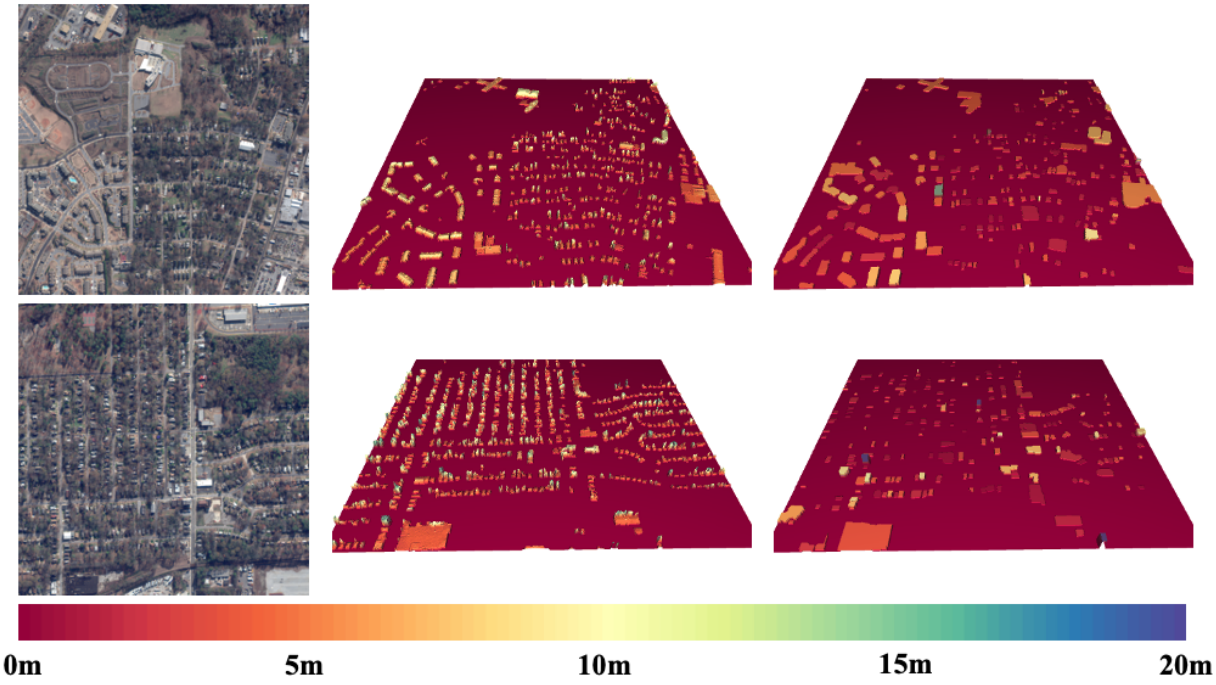


Figure 3.17. Building height estimation result for Urban Semantic 3D Dataset. From left to right: RGB satellite image, building height ground truth from LiDAR sensor, and building height estimation result (color legend is below the figures).

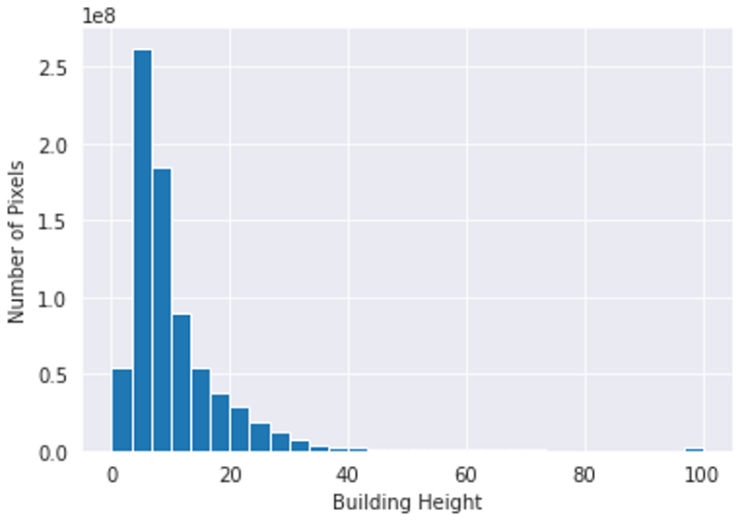


Figure 3.18. Ground truth building height histogram/distribution of the data we used in our experiment.

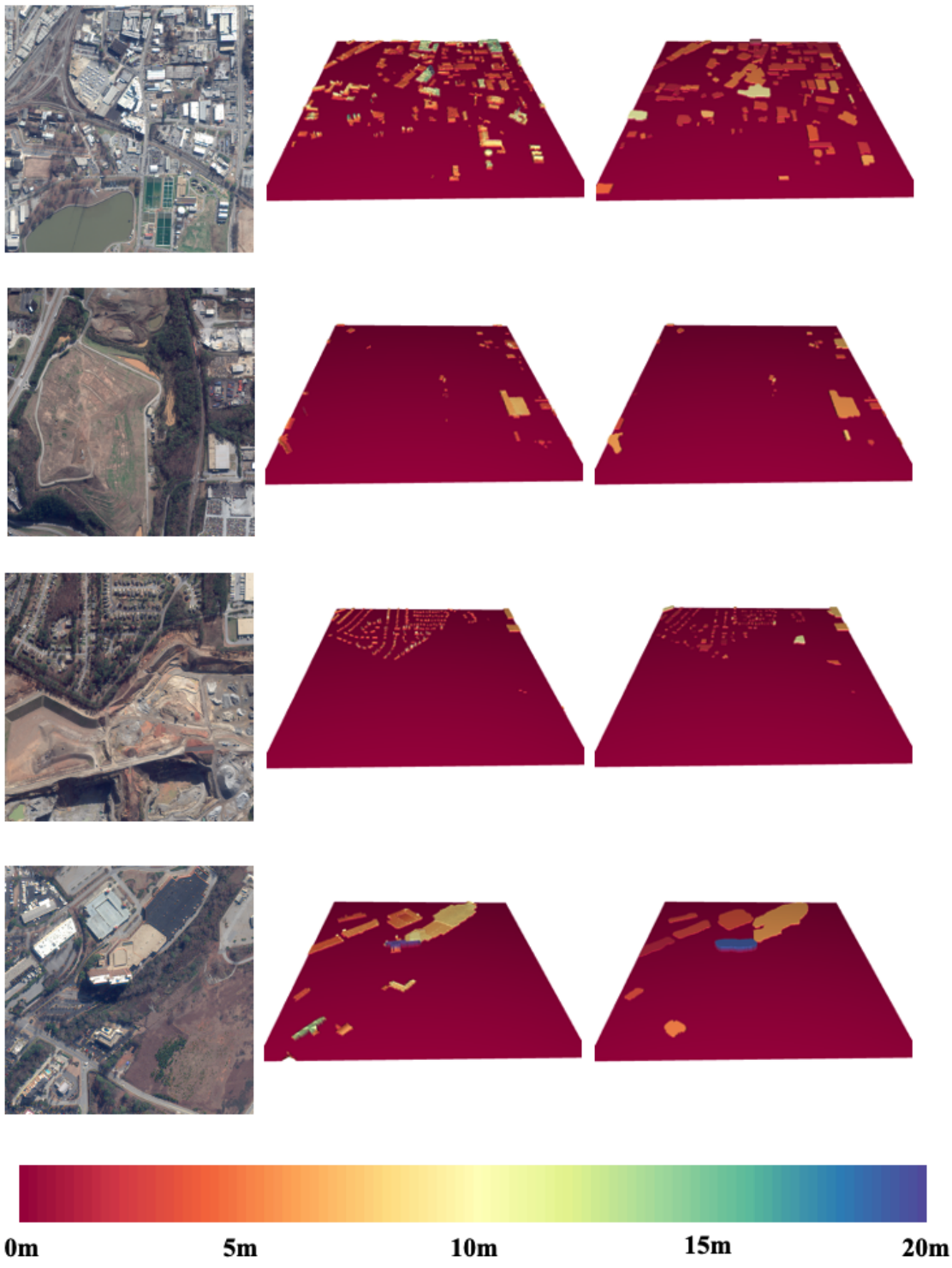


Figure 3.19. Building height estimation result for Urban Semantic 3D Dataset. From left to right: RGB satellite image, building height ground truth from LiDAR sensor, and building height estimation result (color legend is below the figures).

3.3 Improving Building Segmentation Using Uncertainty Modeling and Metadata Injection

3.3.1 Overview

Both previously proposed methods for 2D appearance change and 3D height change require accurate building segmentation. However, this might be hard to achieve for the low-quality images, such as off-nadir images. In general, object segmentation for satellite imagery has been studied extensively because of the availability of large-scale datasets [6], [135], [136], [142], [143] and computational resources. Although many existing methods achieve accurate segmentation results, using them in real-world applications is still challenging. Unlike many segmentation tasks for natural images, such as the COCO dataset [144] and Cityscapes dataset [145], real-world object segmentation for satellite imagery often faces challenges in identifying small, visually heterogeneous objects (e.g. cars and buildings) with varying orientation and density in images [135]. For example, it is even hard for humans to detect the small buildings inside the forest area from the images in Figure 3.20, because of the low lighting condition and the similar colors of the buildings compared to their surrounded trees. Furthermore, due the changes of satellite viewing angle, the appearance of target objects can vary dramatically, including changes in lighting intensity, object resolution, and image noise level. As the input images in Figure 3.20 show, from small viewing angle (first row) to large viewing angle (second row), the overall image intensity and image quality changes significantly. Therefore, to be able to successfully use the segmentation models in real-world applications, addressing the aforementioned challenges is necessary.

Many existing satellite imagery segmentation methods directly adopt approaches that were originally designed for the natural image object segmentation task without considering the previously mentioned challenges. Since most of the publicly available datasets for satellite image segmentation consist of images taken nearly directly overhead (*i.e.*, at-nadir images) [6], [136], [142], [143], these existing methods are able to produce accurate results. However, as mentioned earlier, the accurate results do not guarantee that these methods can be successfully used in real-world applications. To address this issue, in this thesis, we consider the more challenging SpaceNet 4, a multi-view overhead imagery dataset [135] for building segmentation, which focuses on noisy data due to large off-nadir angles. As shown in Figure 3.21, satellite off-nadir angle (*i.e.*, viewing angle) is the angle between the nadir point directly below the satellite and the center of the

imaged scene [135]. Considering images with large off-nadir angles enables us to move one step closer to real-world applications. For example, many satellite images collected during disaster responses or other urgent situations often involve large off-nadir angles. The first set of satellite images taken from Puerto Rico after Hurricane Maria was obtained with the off-nadir angle as $51.9°$ [75]. A large off-nadir angle can cause a significant deterioration in image quality. As shown in Figure 3.20, compared to the image with the smaller off-nadir angle, the image with the larger off-nadir angle is blurrier and noisier. Furthermore, a large off-nadir angle can also cause a change in object appearance. For example, in the same figure, with the smaller off-nadir angle, only building roofs are visible, but with the larger off-nadir angle, both building roofs and their facades are visible, which will cause the change of building area in the satellite images. In the SpaceNet 4 dataset, images of the same scene are taken at different off-nadir angles. All building annotations are labeled based on the images with the smallest magnitude of off-nadir angle ($-7.8°$) and the rest of the images with different off-nadir angles use the same labels as ground truth during training. Therefore, the change of building appearance due to the change of off-nadir angle has an adverse effect when training the model due to the inaccurate ground truth annotations. These challenges are similar to the challenges in domain adaptation, where reliable data is available for training in one scenario, but the model needs to be adapted to new data collected under different scenarios (*e.g.*, different lighting conditions, image noise conditions, or annotation accuracy conditions).

In order to solve these challenges provided by the SpaceNet 4 dataset and real-world applications, we present a building segmentation method with uncertainty modeling and satellite image metadata injection. Our method is able to provide accurate segmentation results when training with noisy images and inaccurate ground truth annotations. More specifically, based on Bayesian deep learning, the proposed method is designed to capture both model and data uncertainty to ignore the image regions with a higher uncertainty level. For example, as shown in Figure 3.20 (we will provide more detailed information in Section 3.3.3), our uncertainty maps highlight the areas with larger image noise (*e.g.*, building boundaries due to the image blur and inaccurate annotation). As the off-nadir angle increases (*i.e.*, from the first row to the second row), the uncertainty level increases, indicating a higher data noise from both image and annotation. Furthermore, satellite image metadata is also considered in our method, as it usually contains useful information to improve model performance. In this thesis, we use ground sample distance (GSD) and off-nadir angle

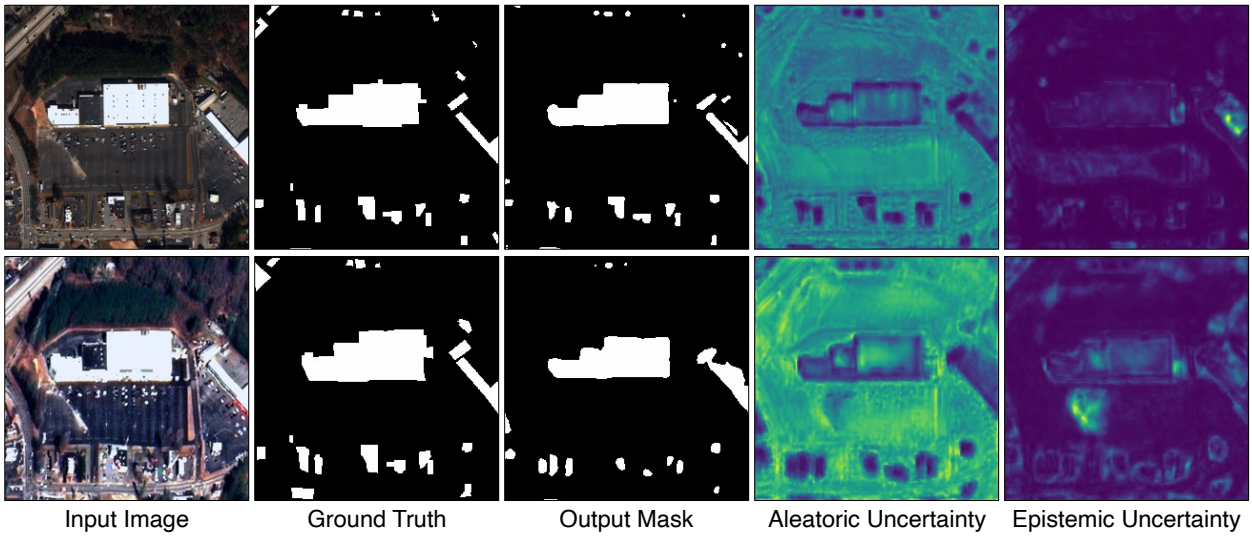


Figure 3.20. Building segmentation results of the proposed method with corresponding uncertainty maps. The first row shows the case with off-nadir angle as $-7.8°$. The second row shows the result of the same scene but with off-nadir angle as $54°$. With a larger off-nadir angle, the input image becomes noisy and blurry. Since aleatoric uncertainty captures the noise inherent in the observations, higher values can be found in the aleatoric uncertainty map from the second case. Class-ambiguous pixels are highlighted in the epistemic uncertainty maps, which often yield incorrect classification results.

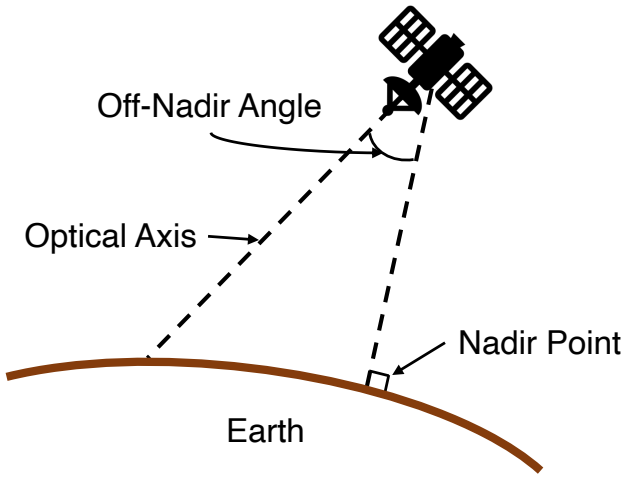


Figure 3.21. Illustration of satellite off-nadir angle.

as input metadata. GSD describes the spatial resolution of the image and a larger GSD usually indicates blurrier and noisier images. As mentioned earlier, different off-nadir angles can also cause the change in image quality. In this thesis, we propose two metadata injection methods to show the effectiveness of using metadata in building segmentation.

3.3.2 Related Work

In this section, we will review the previous work for satellite image building segmentation as well as the methods using uncertainty modeling and metadata injection in satellite imagery.

Building segmentation for satellite imagery. In this thesis, we consider the building segmentation task as a binary semantic segmentation task[9]. Many recent approaches (including our proposed method) are designed based on the U-Net structure [53], because of its good performance in many computer vision tasks [146]–[150]. Here we briefly review several U-Net-based methods of building segmentation for satellite imagery. A large receptive field is important for the segmentation model to detect buildings with different sizes. Therefore, many methods improve the original U-Net by using different techniques to enlarge the receptive field to achieve better performance. Zhang *et al*. [151] extend the U-Net model with dense connections [86] and dilated convolutional layers [87], [150] to reach a large receptive field for capturing the information of large objects. Liu *et al*. [152] incorporate a feature pyramid scene parsing (PSP) network [153] with U-Net to further increasing the receptive field. They use the PSP module to replace the bottleneck layer from U-Net to allow the use of multi-scale features for extracting building footprints of different sizes. Jing *et al*. [154] design a spatial pyramid dilated network for building segmentation by combining the aforementioned PSP network with dilated convolution. In this thesis, as discussed previously, instead of focusing on improving the performance on the at-nadir images, our method aims to deal with the problem of adapting for real-world applications: building segmentation for images with large off-nadir angles, as these images tend to be noisier and blurrier than at-nadir images.

Uncertainty modeling for satellite imagery analysis. Using Bayesian deep learning to model uncertainty has already been seen in satellite imagery analysis. Kampffmeyer *et al*. [155] first introduced Monte Carlo dropout [156] to capture model uncertainty for small object segmenta-

[9]↑Some previous work also considered this task as an instance segmentation task. In this thesis, we will focus on the semantic segmentation task.

tion. Although dropout is rarely used in convolutional neural networks (CNNs) due to the empirically deteriorated performance, they show that adding dropout layers in their fully convolutional encoder-decoder model with Monte Carlo integration during inference can achieve better performance. Our proposed method also uses Monte Carlo dropout; please check Section 3.3.3 for more information. Inspired by this, Bischke *et al.* [157] proposed to use the model uncertainty to address the class imbalance issue in the satellite image segmentation task. The predicted uncertainty for each class is used as the weight in the cross-entropy loss to account for model uncertainty caused by class imbalance. In this thesis, we propose to use not only the model uncertainty (*i.e.*, epistemic uncertainty) as presented in the previous work, but also the data uncertainty (*i.e.*, aleatoric uncertainty) to enable our segmentation model to learn from noisy data.

Injecting metadata for satellite imagery analysis. Satellite image metadata can be used in many satellite imagery analysis tasks, as it usually contains useful information to improve model performance. Pritt *et al.* [158] use a variety of satellite metadata including GSD, off-nadir angle, longitude, and latitude for the image classification task in satellite imagery. They use an ensemble of CNN models for image feature extraction. Then the CNN features are concatenated with the normalized metadata and fed into fully-connected layers for classification. In Section 3.3.3, we will provide a similar concatenation-based metadata injection method with an improvement of metadata feature extraction using multi-layer perceptrons. Christie *et al.* [159] proposed a similar model to fuse the CNN features with normalized metadata for multi-temporal satellite image sequence. Different from the previous work, instead of feeding the fused features to fully-connected layers, these features are fed into a long short-term memory (LSTM) model to accumulate temporal information from different frames to obtain the final classification result. In this thesis, besides the aforementioned concatenation-based method, we will also present an Affine Combination Module-based metadata injection to inject metadata for multiple feature resolutions.

3.3.3 Proposed Method

In this section, we will introduce our building segmentation method with uncertainty modeling and satellite image metadata injection. As shown in Figure 3.22, the proposed method is based on U-Net [53] with multiple outputs. As described later, modeling uncertainty enables our method

to ignore the noisy pixels that are caused by 1) blurry or noisy images; and 2) inaccurate data annotation. Injecting satellite image metadata such as ground sample distance (GSD) and off-nadir angle provides the model with more information to improve its performance. We will provide two metadata injection approaches in this section.

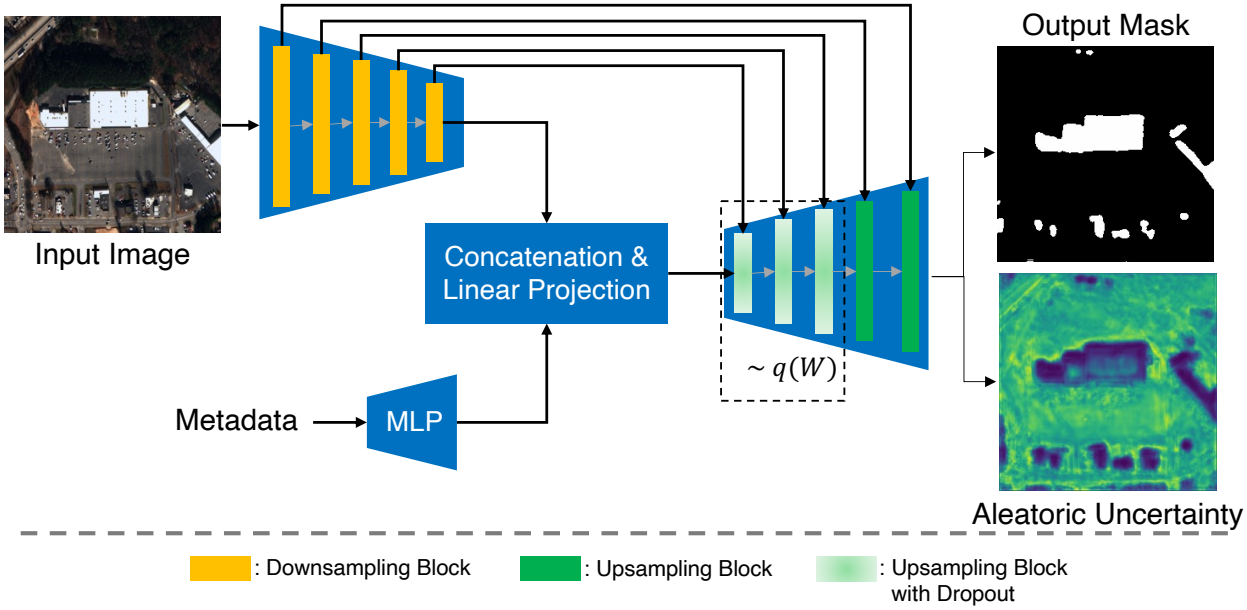


Figure 3.22. The block diagram of the proposed method with uncertainty modeling and concatenation-based metadata injection. $q(W)$ is the dropout variational distribution.

Modeling Uncertainty via Bayesian Deep Learning. Unlike standard deep learning methods, Bayesian deep learning (Bayesian DL) provides a model with the ability to ignore certain data points based on uncertainty. In Bayesian DL, there are two types of uncertainty one can model:

- *Epistemic Uncertainty* describes the uncertainty that is caused by the model ignoring some training data. For example, a segmentation model might miss some building areas with certain colors/textures. Usually, this type of uncertainty can be reduced as more training data is made available.

- *Aleatoric Uncertainty* describes the uncertainty that is inherited from data (*e.g.*, image/sensor noise). Aleatoric uncertainty can be further categorized as *homoscedastic uncertainty*, which is the uncertainty based on the entire dataset, and *heteroscedastic uncertainty*, which is the

uncertainty for each input data point (*i.e.*, each pixel in our case). In this thesis, we will consider heteroscedastic aleatoric uncertainty to accurately model the data noise for different input images.

In the following section, we will review the methods for modeling epistemic uncertainty [160] and aleatoric uncertainty [161], followed by our proposed approach to combine both uncertainties in one model.

Epistemic Uncertainty. In Bayesian DL, to capture the uncertainty from the model (*i.e.*, epistemic uncertainty), we place a distribution over the model parameters. For example, the prior distribution of the model weights for a fully-connected layer, $p(\mathbf{W})$, can be modeled as: $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This is different from the standard deep learning model, which uses deterministic parameters. In Bayesian DL, for each forward pass, including both training and testing, the model parameters will be different due to parameter sampling. Formally speaking, we formulate our building segmentation model as:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}) = p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x})) = S(f^{\mathbf{W}}(\mathbf{x})), \tag{3.6}$$

where $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is the input image, $\mathbf{y} \in \mathbb{R}^{H \times W}$ is the output class label (in our case, it is a binary label indicating foreground or background), $f^{\mathbf{W}} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H \times W}$ is our Bayesian DL model with sampled parameters $\mathbf{W} \sim p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$, and $S : \mathbb{R} \rightarrow \mathbb{R}$ is the sigmoid function applied to each input element.

Estimating the model posterior $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ over the entire training set $(\mathbf{X}, \mathbf{Y})$ is intractable [156], [160]. To evaluate this posterior distribution, following the work [156], [160]–[162], we use *dropout variational inference*. This inference is performed by placing a dropout layer before every convolutional layer (or fully-connected layer). Since dropout can be formulated as a Bernoulli trial by randomly setting the model parameters to zero, [156], [160] show that this dropout distribution over model parameters, $q(\mathbf{W})$, can be used to estimate our model posterior. This is done by minimizing their Kullback-Leibler (KL) divergence via the following loss function during training:

$$\mathcal{L}_{epi}(\mathbf{x}, \mathbf{y}) = \mathcal{L}_{cls}(\mathbf{y}, S(f^{\mathbf{W}}(\mathbf{x}))) + \lambda \|\mathbf{W}\|_2^2, \tag{3.7}$$

where $(\mathbf{x}, \mathbf{y})$ is a pair of training image and its corresponding ground truth label mask, $\mathcal{L}_{cls}(\cdot, \cdot)$ is a classification loss (*e.g.*, binary cross entropy loss in our case), $f^{\mathbf{W}}(\cdot)$ is our model with parameters sampled from the dropout distribution $q(\mathbf{W})$, and λ is a non-trainable hyper-parameter as described in [156]. The second term of Equation 3.7 can be implemented using weight decay [163], which was originally designed for model regularization. During inference, we can estimate the final prediction distribution $p(\mathbf{y}^*|\mathbf{x}^*)$ given a testing image $\mathbf{x}^*$ via Monte Carlo integration as proposed in [156], [160]:

$$p(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{W}) q(\mathbf{W}) d\mathbf{W} \approx S(\frac{1}{T} \sum_{t=1}^{T} f^{\mathbf{W}_t}(\mathbf{x}^*)), \tag{3.8}$$

where $\mathbf{W}_t \sim q(\mathbf{W})$ is the model parameters from each Monte Carlo sample and T is the total number of samples. Equation 3.8 is referred as Monte Carlo dropout as proposed in [156]. Epistemic uncertainty can be visualized by calculating the variance of the Monte Carlo samples:

$$\frac{1}{T} \sum_{t=1}^{T} \left(f^{\mathbf{W}_t}(\mathbf{x}) \odot f^{\mathbf{W}_t}(\mathbf{x}) \right) - \mathbb{E}[f^{\mathbf{W}}(\mathbf{x})] \odot \mathbb{E}[f^{\mathbf{W}}(\mathbf{x})], \tag{3.9}$$

where $\odot$ is the Hadamard product for element-wise multiplication and $\mathbb{E}[f^{\mathbf{W}}(\mathbf{x})] \approx \frac{1}{T} \sum_{t=1}^{T} f^{\mathbf{W}_t}(\mathbf{x})$.

As shown in Figure 3.22, we model the epistemic uncertainty by placing the dropout layers before just the first three decoder layers, instead of all convolutional layers. Since we use a ResNet-34 model [9] pretrained on ImageNet [115] as the CNN encoder, we model this feature extraction process as a deterministic process. Therefore, no dropout layers are used in the CNN encoder. In this thesis, we only model the first three decoder layers as stochastic processes by placing the dropout layers before each convolutional layer in each upsampling block. We do not add dropout layers to the last two decoder layers. This is to reduce the output noise due to the limited number of Monte Carlo samples during inference as shown in Equation 3.8.

Aleatoric Uncertainty. Aleatoric uncertainty captures the noise from training data. As described previously, in this thesis, we consider heteroscedastic aleatoric uncertainty, which captures the noise from each pixel from an input image. We use two additional convolutional layers placed on top of the last decoder layer to obtain the classification logit $f^{\mathbf{W}}(\mathbf{x}) \in \mathbb{R}^{H \times W}$ and aleatoric uncertainty $\sigma^{\mathbf{W}}(\mathbf{x}) \in \mathbb{R}^{H \times W}$, as shown in Figure 3.22. We use the predicted aleatoric uncer-

tainty during training to ignore the pixels with larger uncertainty and address the pixels with less uncertainty. To achieve this, as proposed in [161], we corrupt the predicted logits $f^{\mathbf{W}}(\mathbf{x})$ with Gaussian random noise, where the standard deviation is the predicted aleatoric uncertainty. More specifically, we modify Equation 3.6 by placing a Gaussian distribution over the predicted logits:

$$\begin{aligned} p(\mathbf{y}_{i,j}|\mathbf{x},\mathbf{W}) &= S(\hat{f}^{\mathbf{W}}(\mathbf{x})_{i,j}), \\ \text{where } \hat{f}^{\mathbf{W}}(\mathbf{x})_{i,j} &\sim \mathcal{N}(f^{\mathbf{W}}(\mathbf{x})_{i,j}, (\sigma^{\mathbf{W}}(\mathbf{x})_{i,j})^2). \end{aligned} \tag{3.10}$$

Note that i and j are the pixel coordinates of the output logit and aleatoric uncertainty. We denote $p(\mathbf{y}_{i,j}|\mathbf{x},\mathbf{W})$ with $p_{i,j}$ for simplicity. From Equation 3.10, we can see that with larger aleatoric uncertainty, the Gaussian corrupted logit $\hat{f}^{\mathbf{W}}(\mathbf{x})$ tends to be noisier, which enforces the model to ignore this "random" prediction. With smaller aleatoric uncertainty, the Gaussian corrupted logit $\hat{f}^{\mathbf{W}}(\mathbf{x})$ tends to be closer to the original predicted logit $f^{\mathbf{W}}(\mathbf{x})$, which makes the model to focus on this prediction. Since we use Gaussian corruption, we can facilitate our implementation using the Gaussian reparameterization trick:

$$\hat{f}^{\mathbf{W}}(\mathbf{x})_{i,j} = f^{\mathbf{W}}(\mathbf{x})_{i,j} + \sigma^{\mathbf{W}}(\mathbf{x})_{i,j}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1). \tag{3.11}$$

During training, to capture both uncertainties, we can replace the classification loss $\mathcal{L}_{cls}$ in Equation 3.7 with a binary cross entropy loss with Gaussian corrupted output:

$$\mathcal{L}_{ale}(\mathbf{x},\mathbf{y}) = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{y}_{i,j}\log p_{i,j} + (1-\mathbf{y}_{i,j})\log(1-p_{i,j}), \tag{3.12}$$

where $\mathbf{y}_{i,j}$ is ground truth label at pixel coordinates (i,j) and $p_{i,j} = S(\hat{f}^{\mathbf{W}}(\mathbf{x})_{i,j})$ as shown in Equation 3.10. Therefore, we can obtain the final loss function for learning both epistemic uncertainty and aleatoric uncertainty as:

$$\mathcal{L}_{both}(\mathbf{x},\mathbf{y}) = \mathcal{L}_{ale}(\mathbf{x},\mathbf{y}) + \lambda\|\mathbf{W}\|_2^2. \tag{3.13}$$

We do not need aleatoric uncertainty during inference, as it is used for ignoring noisy pixels during training.

Metadata Injection. Satellite image metadata contains useful information to support many computer vision tasks, such as using solar and satellite azimuth and elevation angles for shadow detection and building height estimation [122], [123], [125], [164], [165]. In this thesis, we consider two types of metadata to improve building segmentation result: (1) ground sample distance (GSD); and (2) off-nadir angle. GSD describes the spatial resolution of the image and a larger GSD indicates blurrier and noisier images due to lower image resolution. Off-nadir angle describes the viewing angle of the satellite camera and a larger off-nadir angle can also cause lower image resolution. In the following sections, we will provide two metadata injection approaches to improve the baseline U-Net model.

Metadata Injection via Feature Concatenation. As shown in Figure 3.22, we first pass the metadata vector to multi-layer perceptrons (MLP) to obtain the output vector ($\mathbf{h} \in \mathbb{R}^D$) for feature extraction and dimension expansion. Then we combine the metadata feature vector with the image features ($\mathbf{v} \in \mathbb{R}^{H \times W \times D}$) obtained from the last CNN encoder layer. To combine metadata and image features, we repeat the metadata feature vector to match the shape of image features: $\mathbf{h} \in \mathbb{R}^{H \times W \times D}$. Then we concatenate the features along the channel dimension as $\mathbf{h_v} \in \mathbb{R}^{H \times W \times 2D}$. The final features can be obtained by linearly projecting the channel dimension back to the input channel dimension: $\mathbf{o} = \mathbf{F}(\mathbf{h_v}) \in \mathbb{R}^{H \times W \times D}$, where $\mathbf{F} : \mathbb{R}^{2D} \rightarrow \mathbb{R}^D$ is applied for each input element and it can be implemented by a convolutional layer with kernel size of 1. We refer to this concatenation-based approach as *MetaCat*.

Metadata Injection via Affine Combination Module. As described above, the previous concatenation-based metadata injection method combines the metadata and image features by channel-wise concatenation following a linear projection layer. By doing so, we augment the image features using the metadata features for every location in the H and W dimensions evenly. However, intuitively, not all image features need to be modified. For example, since we focus on building segmentation, a large forest area should not be considered and modified. To effectively locate the desired regions that need to be modified, we use the Affine Combination Module (ACM) [166] for metadata injection as shown in Figure 3.23. As the name indicates, ACM is based on affine transforms and can be formulated as follows:

$$\mathbf{v} = \mathbf{h} \odot W(\mathbf{v}) + b(\mathbf{v}), \tag{3.14}$$

where $\mathbf{v}$ is the image features obtained from the CNN encoder, $\mathbf{h}$ is either the repeated metadata features $\mathbf{h}$ as previously described or the features from the previous decoder layer, and $W(\cdot)$ and $b(\cdot)$ are convolutional layers as proposed in [166]. From Equation 3.14, we can consider the $W(\mathbf{v})$ term as the metadata-relevant information, since it can directly interact with the metadata features (or the previous decoder features). The $b(\mathbf{v})$ term can be considered as a metadata-irrelevant information that is not modified by the metadata features (or the previous decoder features). As the results that we will provide in Section 3.3.4 indicate, with ACM, we can explicitly decouple the metadata-relevant and metadata-irrelevant information without implicit learning by the model. Following the design from [166], we use multiple ACMs in different feature resolutions in our decoder without changing other parts of the model, as shown in Figure 3.23. We refer to this ACM-based approach as *MetaACM*.

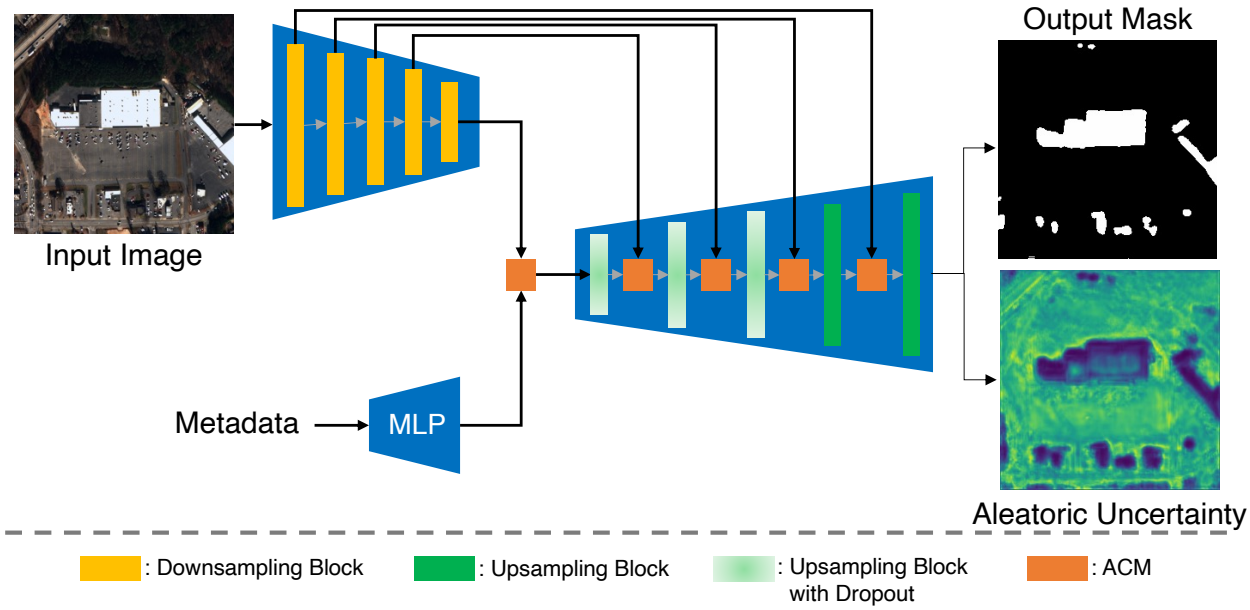


Figure 3.23. The block diagram of the proposed method with uncertainty modeling and ACM-based metadata injection.

3.3.4 Experiment

In this section, we will describe the dataset we used and the model implementation details, and provide experimental results with analysis.

Dataset and Experiment Setting. In this thesis, we use the SpaceNet 4 dataset [135], which is designed for building segmentation with a larger range of off-nadir angles. It contains 4-channel RGB-NIR (Near-Infrared) images with resolutions of 900×900. There are $1,064$ distinct locations in the dataset, with 27 images captured at each location at different off-nadir angles ranging from $-32.5°$ to $54°$, which totals to $28,728$ images. We partition the dataset into training, validation, and testing sets with the ratio of $6:2:2$. Note that when splitting the dataset, we ensure that all images of the same location are assigned to the same partition. This can avoid different partitions sharing images from the same location.

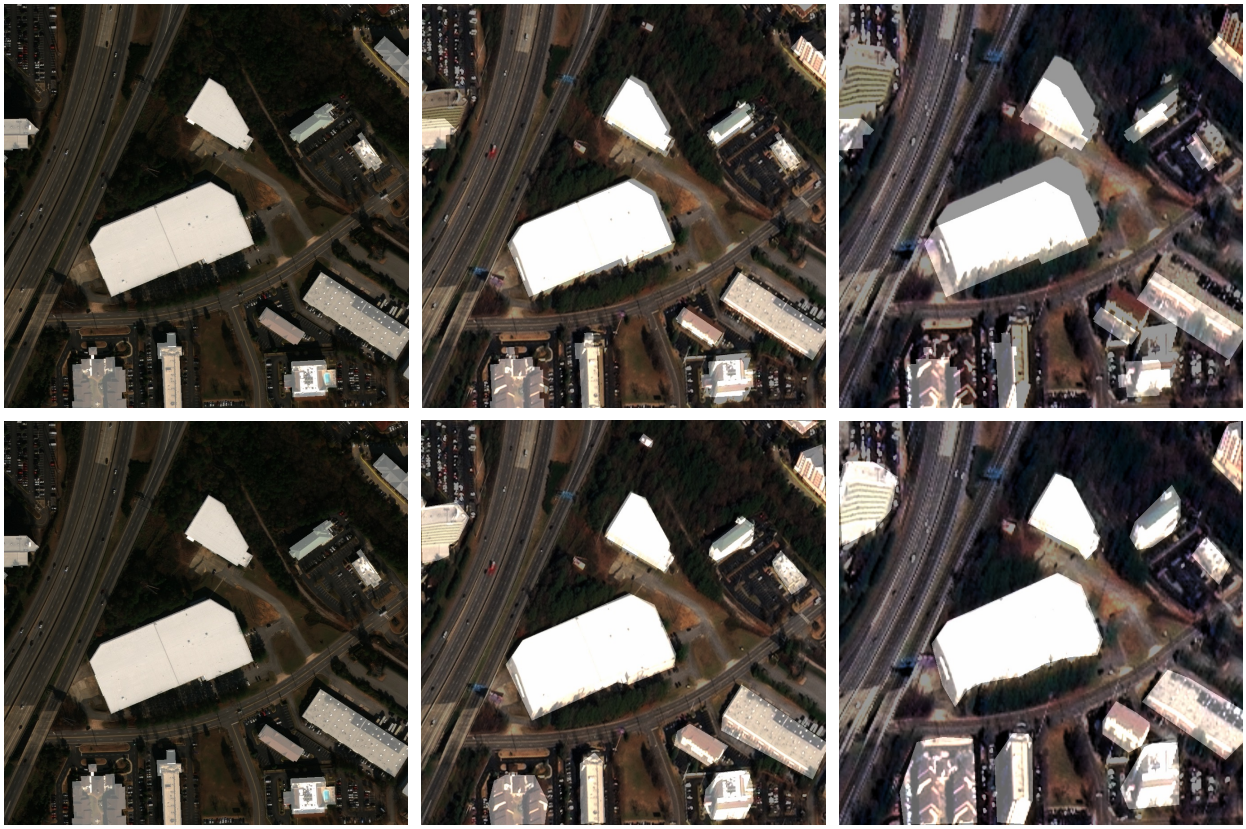


Figure 3.24. Illustration of the building segmentation annotation issue in the original dataset. The light white area is the annotated ground truth area. The first row shows the annotation from the original dataset and the second row shows the annotation we manually labeled.

As mentioned in [135], the building annotations from SpaceNet 4 dataset are obtained from the images with the smallest magnitude of off-nadir angle ($-7.8°$), and the same annotations are used

for other images with different off-nadir angles. As shown in the first row of Figure 3.24, due to the change of viewing angle, the appearance, especially for the tall buildings, changes significantly. For example, with the smaller off-nadir angle, only the building roof is visible, but with larger off-nadir angles, both building roof and facade are visible, which can cause inaccurate annotations. Although the proposed method is designed to deal with the noisy images and annotations, in order to have an accurate testing evaluation, we manually label the testing images with off-nadir angles greater than $40°$, as shown in the second row of Figure 3.24.

To ensure fair comparison between the proposed method and the baseline U-Net, all of our experiments used the same setting, which we will now describe. The downsampling blocks (yellow blocks) in Figure 3.22 and Figure 3.23 are the residual blocks from a ResNet-34 model [9] pretrained on ImageNet [115]. The upsampling blocks (dark green blocks) consist of *bilinear upsampling* $\rightarrow$ *convolution* $\rightarrow$ *batch normalization* $\rightarrow$ *ReLU*. The upsampling blocks with dropout (light green blocks) consist of *bilinear upsampling* $\rightarrow$ *dropout* $\rightarrow$ *convolution* $\rightarrow$ *batch normalization* $\rightarrow$ *ReLU*. Following [161], the dropout rate is set as 0.2. The MLP for metadata feature extraction consists of three blocks, where each block is a fully-connected layer following by a leaky ReLU layer with the slope of 0.2. During training, to allow for a larger batch size as required by batch normalization, we resize the input image to 256 with batch size as 64. ADAM optimizer [67] with learning rate 0.0001 (linear decay) is used and all experiments are trained for 1 million iterations. As mentioned in Section 3.3.3, modeling epistemic uncertainty requires using weight decay during training. To achieve a fair comparison, we use weight decay with the factor of 0.0001 for all experiments. For the Monte Carlo integration during inference, following [161], we set the number of samples as 50 (we will provide the analysis of this parameter in the following section).

Experimental Result and Analysis. We start with evaluating the use of uncertainty modeling and metadata injection (we consider the concatenation-based metadata injection first). Figure 3.25 shows the F1 scores with different off-nadir angles in the testing set. Compared with the baseline U-Net, with uncertainty modeling, there is a slight improvement across most of the off-nadir angles. Adding the metadata injection layer can further improve the performance, especially for the cases with larger off-nadir angles ($> 40°$) and negative off-nadir angles. As mentioned in [135], due to the data collection process, the images with large negative off-nadir angles have very differ-

ent lighting conditions and shadows. Since most of the images are collected from positive off-nadir angles, the baseline method will suffer from unbalanced data during training. With metadata injection and uncertainty modeling, the proposed method is able to deal with the changes of lighting and shadows.

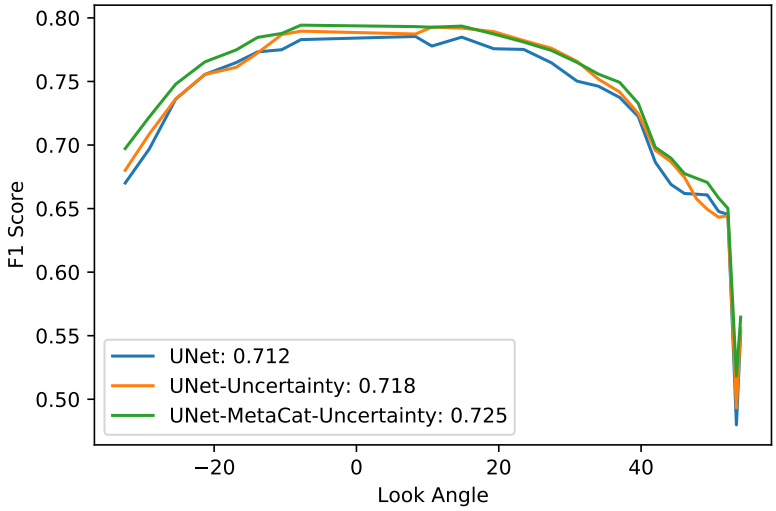


Figure 3.25. Testing F1 scores with different off-nadir angles. The average F1 scores of all off-nadir angles are shown in the legend.

Figure 3.26 shows three testing examples captured from the largest off-nadir angles to visualize the improvement of the proposed method compared to the baseline U-Net. Based on the ground truth, we can see that the proposed method is able to detect more accurate building area even under this high noise-level condition. For instance, in the first example, the baseline U-Net fails to differentiate the parking lot area and the building area in the top-left of the input image (highlighted by the red circle). The proposed method is able to segment the area correctly. From the epistemic uncertainty map, the proposed method raises higher uncertainty indicating the predictions from those class-ambiguous pixels are not reliable. Similar examples can be found in the highlighted areas in the second and third images. From the aleatoric uncertainty, we can also see that the input data has higher data noise around the forest region compared to the building region. This is due to the larger appearance variance of forests compared to buildings. Therefore, our model will focus more on the building region during training to avoid the adverse effect of the frequent appearance

changes from the forest region. Unlike aleatoric uncertainty, epistemic uncertainty focuses more around the buildings or other man-made structures (*e.g.*, roads). It highlights the area when the predictions are not reliable, such as the boundary of buildings due to the image blur and noise. Figure 3.20 shows the prediction difference of two images with same scene but different off-nadir angles. We can see that overall, aleatoric uncertainty has a significant increase from small to large off-nadir angles due to higher noise in the input image. Although there is less of an increase with epistemic uncertainty, the area where it highlights does get larger. Figure 3.27 shows the result with more off-nadir angles of the same scene for comparison.

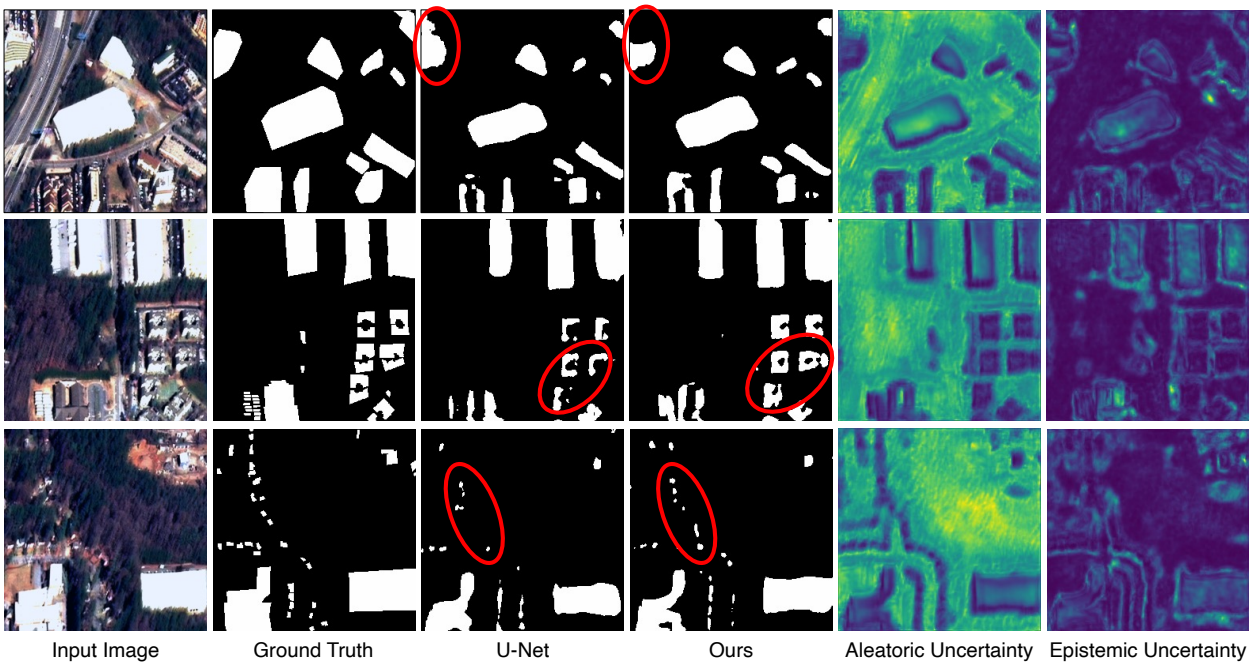


Figure 3.26. Result comparison of the baseline U-Net and the proposed method with uncertainty modeling and metadata injection. The input images are taken with the off-nadir angle as $54°$. The red circles highlight the improvement of the proposed method compared to the baseline U-Net.

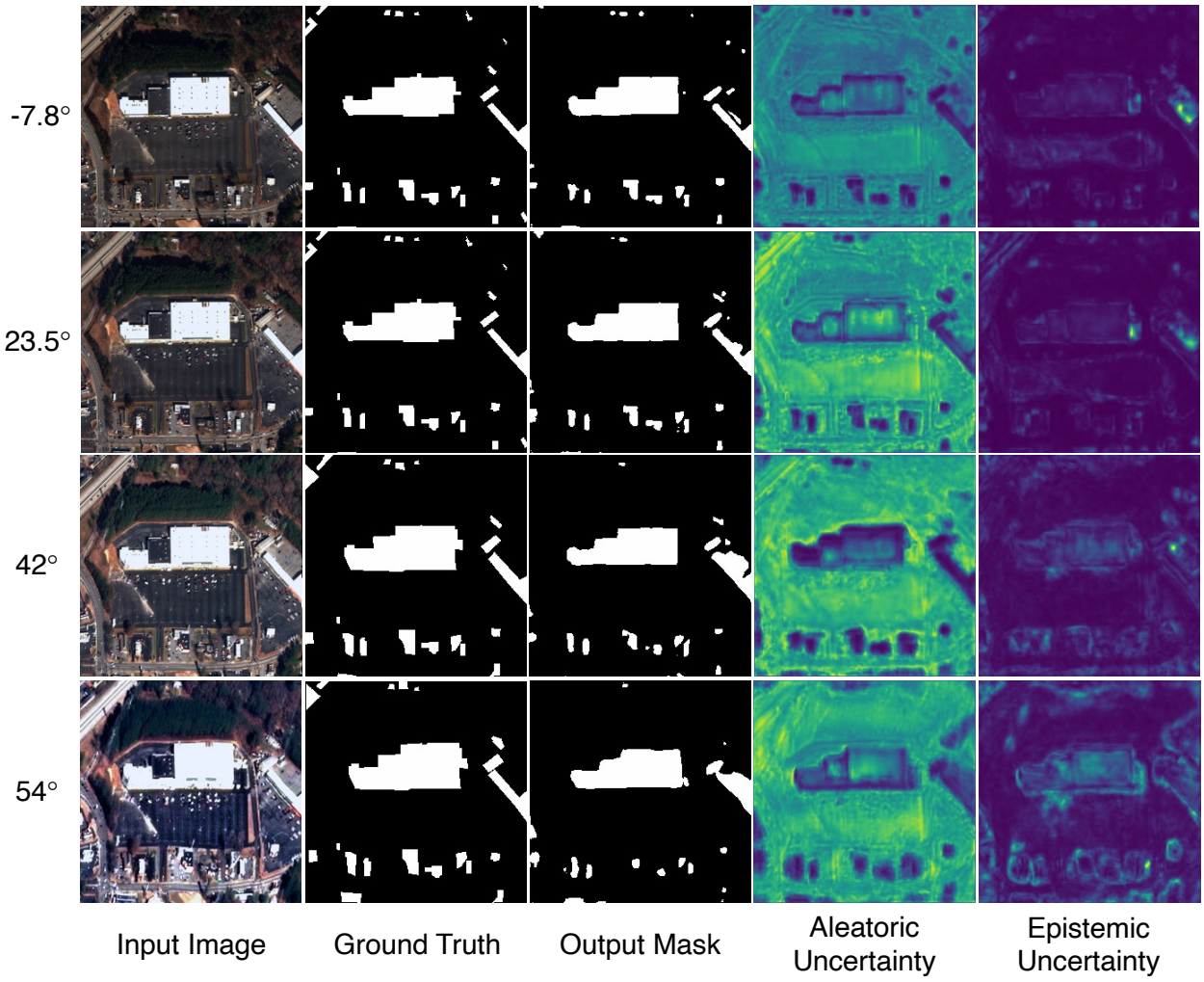


Figure 3.27. Results of the proposed method for the images taken from different off-nadir angles. The results are obtained from the model with uncertainty modeling and concatenation-based metadata injection.

We also provide the ablation study of uncertainty modeling to show that modeling both uncertainties does not necessarily yield the best result. Due to the close performance of the compared experiments, we group the F1 scores with different ranges of off-nadir angle in Table 3.10 to better visualize differences. As defined in [135], we group the images into three categories based on the off-nadir angles θ as following:

- *Nadir*: $0° \leq |\theta| \leq 25°$;
- *Off-Nadir*: $25° < |\theta| < 40°$;
- *Very Off-Nadir*: $40° \leq |\theta| < 90°$.

As shown in Table 3.10, the best performance from each category are not from the experiment with both uncertainties. Therefore, the effectiveness of uncertainty modeling could be different depending on the dataset and task. Furthermore, as shown in the highlighted cells, for the *Very Off-Nadir* category, all experiments with uncertainty modeling achieve much better performance than the method without uncertainty modeling. This confirms that using uncertainty modeling improves the model performance when larger data noise appears.

Table 3.10. F1 scores for the ablation study of uncertainty modeling. All of the listed experiments are based on U-Net with concatenation-based metadata injection. *None* means no uncertainty modeling.

Experiment	Nadir	Off-Nadir	Very Off-Nadir	Overall
None	0.7820	**0.7450**	0.6335	0.7219
Aleatoric	0.7822	0.7448	**0.6499**	**0.7275**
Epistemic	**0.7824**	0.7424	0.6380	0.7229
Both	0.7822	0.7429	0.6415	0.7249

Figure 3.28 shows the effectiveness of different number of samples in Monte Carlo integration obtained from our validation set. The *Regular Dropout* experiment uses dropout as a regularization method meaning that dropout is only used during training. The *No Dropout* experiment does not use dropout for both training and testing. From the overall F1 score plot (left) and the F1 score plot for the *Very Off-Nadir* category (right), we can see that the performance stops improving when the number of samples is over 40, which shows our choice of 50 samples is reasonable. Furthermore,

we also show that with Monte Carlo dropout, a better result can be achieved compared to regular dropout and no dropout experiments. Among the three experiments, regular dropout has the worst performance. This shows the same observation as mentioned in [156], since empirically adding dropout layer in CNN tends to have a deteriorated performance.

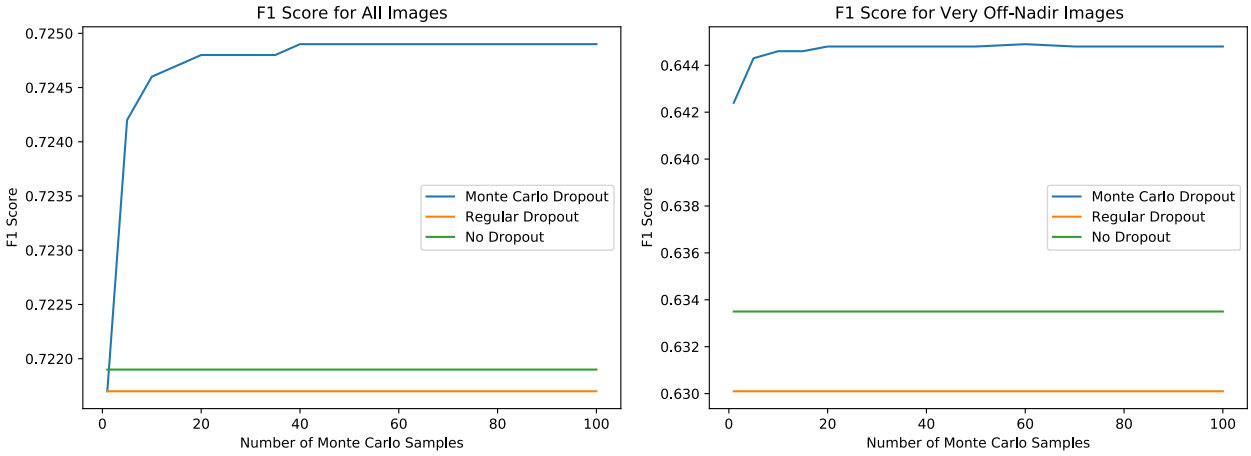


Figure 3.28. Ablation study of Monte Carlo dropout. F1 scores for different numbers of Monte Carlo samples are shown for all images from the validation set (left) and for the images in *Very Off-Nadir* category (right).

Table 3.11. F1 scores for ACM-based and concatenation-based metadata injection. All of the listed experiments are based on U-Net with uncertainty modeling of both aleatoric and epistemic uncertainties. *None* means no metadata injection.

Experiment	Nadir	Off-Nadir	Very Off-Nadir	Overall
None	0.7752	0.7359	0.6347	0.7180
MetaCat	**0.7822**	**0.7429**	0.6415	**0.7249**
MetaACM	0.7758	0.7382	**0.6419**	0.7197

We compare the ACM-based (MetaACM) and concatenation-based (MetaCat) metadata injection methods in Table 3.11. Overall, MetaCat achieves better performance than MetaACM. Compared with the method without metadata injection, MetaCat has significant improvement for all three off-nadir angle categories. Although MetaACM does not have a major improvement for the lower off-nadir angle images, it achieves the best performance under the *Very Off-Nadir* category.

Figure 3.29 shows the ACM feature maps obtained from the last decoder layer. Following [166], the visualization of these feature maps is obtained by computing the average along

the channel dimension. The results from the fourth column show the $\mathbf{h} \odot W(\mathbf{v})$ map based on Equation 3.14. As we discussed in Section 3.3.3, this feature map should highlight the metadata-relevant information, since it directly interacts with the metadata features (or the previous decoder features). The first row in Figure 3.29 shows the case with small off-nadir angle. Its $\mathbf{h} \odot W(\mathbf{v})$ map mainly addresses the entire building area, according to the inpainted result from the last column. However, when dealing with a large off-nadir angle, the $\mathbf{h} \odot W(\mathbf{v})$ map highlights the lower side of building area, as shown in the second row of Figure 3.29. With larger off-nadir angle, building facade becomes visible which increases the building area compared to the case with small off-nadir angle. ACM highlights the building facades (lower side of the building area) to improve the prediction on those regions. This confirms our observation in Table 3.11 that MetaACM is able to significantly improve the performance of the *Very Off-Nadir* category.

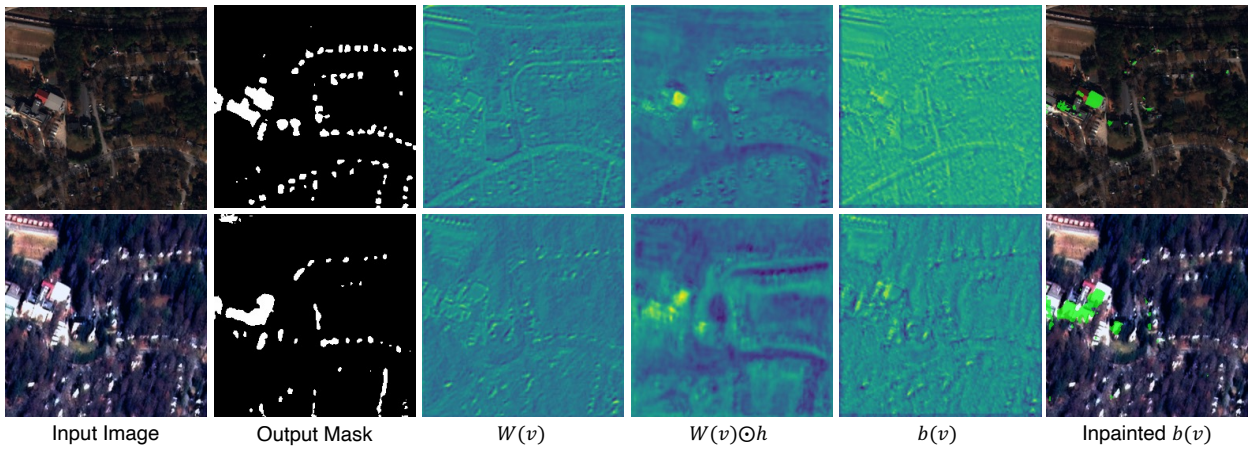


Figure 3.29. Illustration of ACM feature maps obtained from the last decoder layer. The inpainted results are obtained by thresholding the normalized ACM product map (green region) with threshold value as 0.5. The first row shows the case with off-nadir angle as $-7.8°$. The second row shows the result of the same scene but with off-nadir angle as $54°$.

Figure 3.30 shows the ACM $\mathbf{h} \odot W(\mathbf{v})$ map from different decoder layers. We can see that the feature maps from different decoder layers address different part of the image. The design of our MetaACM enables the model to locate different areas for different feature resolutions. This is important for metadata injection, since if we only modify the image features using metadata features in the lowest resolution (*e.g.*, MetaCat), these modifications will affect a large area in the final full-resolution result. For example, in our case, the bottleneck layer has the resolution

of 8×8 and the final result has the resolution of 256×256. If we only consider the effect of upsampling operators (without considering the change of receptive field caused by convolution), any modifications of the features from the bottleneck layer will affect at least 32×32 area in the final result. These modifications are not accurate enough for the buildings that are much smaller than 32×32 pixels. Therefore, injecting metadata features for the image features with different resolutions is important for the refinement of small buildings.

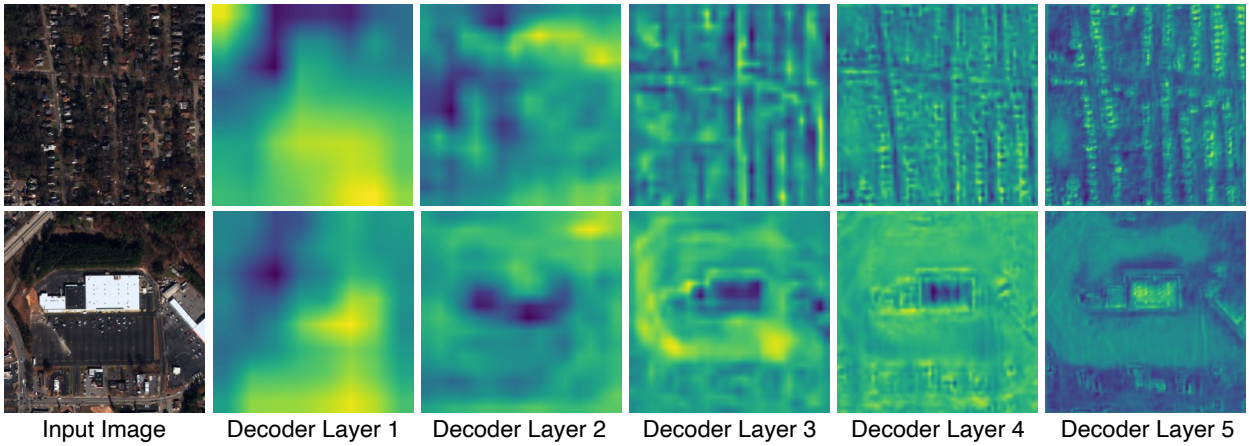


Figure 3.30. Resized ACM $W(v) \odot h$ map for different decoder layers. The resolution of ACM map from decoder layer 1 is 8×8 and increases with the factor of 2 after each decoder layer.

To show the effectiveness of the proposed ACM-based metadata injection method, we also evaluate it on a different backbone model, U^2-Net [167]. The proposed uncertainty modeling and metadata injection methods can be extended to other backbone models. As shown in Figure 3.31, we can apply the proposed methods for U^2-Net [167], which is a modified version of the original U-Net. It is able to utilize a two-level nested U-structure to enlarge the receptive field in each encoder/decoder block. Moreover, deep supervision [114] (*i.e.*, output multiple masks for different decoder blocks) is used to enforce the integration of multi-level deep features to further improve the performance. Please check the original paper [167] for the detailed design of U^2-Net. Similar to the U-Net backbone, we use the epistemic uncertainty modeling (*i.e.*, Monte Carlo dropout layers) in the first three decoder blocks in U^2-Net. Then we split the final layer into two branches to learn the aleatoric uncertainty map. Figure 3.31 shows the model with concatenation-based metadata

injection. The ACM-based metadata injection method can be obtained in a manner similar to the block diagram shown in Figure 3.23.

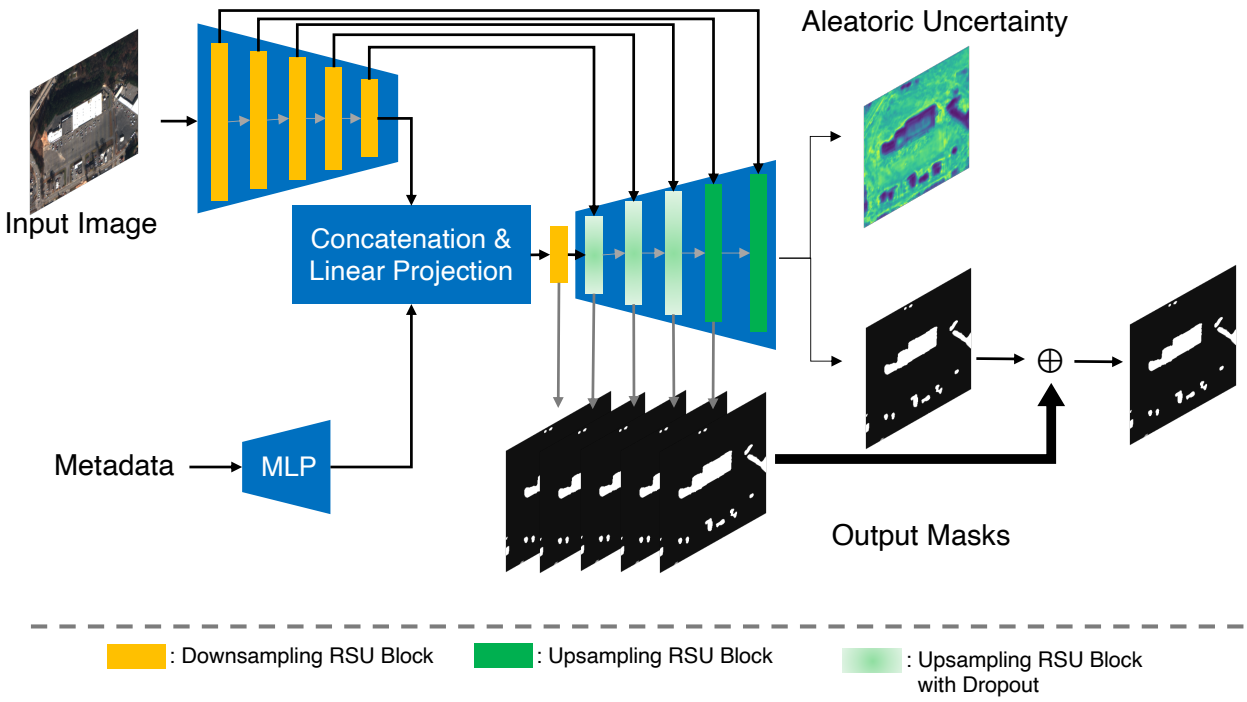


Figure 3.31. The block diagram of the proposed U^2-Net [167] with uncertainty modeling and concatenation-based metadata injection.

Table 3.12 shows the results of U^2-Net with uncertainty modeling and the different metadata injection approaches. Using the proposed uncertainty modeling and metadata injection methods can improve the original U^2-Net model, especially for the cases with large off-nadir angles (except the *Very Off-Nadir* case from the concatenation-based metadata injection experiment). The experiment with both uncertainty modeling and the ACM-based metadata injection method achieves the best performance. It achieves the best performance for all off-nadir angle cases, which confirms the benefit of using the multi-level features in metadata injection. Therefore, from the aforementioned experiments, we show that the proposed uncertainty modeling and metadata injection methods can improve the performance of both U-Net and U^2-Net.

Table 3.12. F1 scores of U^2-Net with uncertainty modeling and metadata injection. *None* means no metadata injection and no uncertainty modeling. The experiments with *Uncertainty* use both aleatoric and epistemic uncertainties.

Experiment	Nadir	Off-Nadir	Very Off-Nadir	Overall
None	0.8019	0.7447	0.6185	0.7259
Uncertainty	0.8081	0.7588	0.6305	0.7356
Uncertainty + MetaCat	0.8080	0.7580	0.6137	0.7304
Uncertainty + MetaACM	**0.8163**	**0.7700**	**0.6348**	**0.7426**

4. SUMMARY AND FUTURE WORK

4.1 Utility Preserving Face Redaction

We develop a deep learning-based model,UP-GAN, that is able to generate synthetic faces that preserve utility information while also removing identifiable information from the original faces. By utility-preserving we mean preserving facial features that do not reveal identity. By swapping the generated face back on the original image, we can produce an effective obscuration that not only removes personal identifiable information, but also retains the information that does not reveal identity, such as expression, age, gender and skin tone. Moreover, we also provide a comprehensive robustness analysis of face obscuration techniques. We analyze eight obscuration methods: Gaussian blurring, median blurring, pixelation, k-same, k-same-net, UP-GAN (Ours), P3, and scrambling. We examine the robustness of these methods under different attacking scenarios including identification, verification, and reconstruction with two widely used deep learning models, VGG19 and ResNet50. Threat modeling is also considered to evaluate the obscuration methods under different strength of attacks. Methods such as Gaussian blurring, median blurring, P3, and scrambling fail to provide an effective obscuration under the designed attackers, although they successfully defeat human perception. We also show that the k-same based methods, especially the proposed UP-GAN model can provide a secured privacy protection. To improve the generated face quality obtained from UP-GAN, we design a system to use face reenactment method for generating photo-realistic synthetic faces with target facial expression and head pose. More specifically, we propose a one-shot face reenactment model, FaR-GAN, that is able to transform a face image to the target expression given one image from any identity. We evaluate our method using the VoxCeleb1 dataset and show that the proposed model is able to generate face images with better visual quality than the compared methods.

Although the results from FaR-GAN achieve a high visual quality, in some cases, when the identity that provides the target landmarks has a large appearance difference from the source identity, such as different genders or face sizes, there is still a visible identity gap between the input source identity and the reenacted face. In future work, we can improve our model to bridge this identity gap, such as using an additional finetuning step to explicitly direct the model to reduce the identity changes, as proposed from [56], [57]. Furthermore, in the current model setting, we

do not consider the pupil movement in our landmark representation. As proposed by [58], we can add the gaze information in the landmark mask to make the reenacted face contain more realistic facial movement. Although the proposed method achieves a good performance in terms of FID, compared with the unconditional face generation methods (ProGAN [5], StyleGAN [4], StyleGAN2 [54]), our generated images are still qualitatively poorer. To further improve our method, we can adopt the progressive training approach from the aforementioned methods. We first train a small portion of the model to produce a good quality image in a small resolution, and then gradually add the rest of the model to produce higher resolution images. By doing so, we can stabilize the training process to produce images with better visual quality with higher resolution.

4.2 Change Detection For Satellite Imagery

We develop a Siam-U-Net-Attn model with attention technique that accurately classifies damage levels of buildings in satellite imagery. The proposed technique compares a pair of images captured before and after disasters to produce segmentation masks that indicate damage scale classifications and building locations. We use the self-attention module to enhance damage scale classification by considering information from the entire image. Results show that the proposed model accomplishes both damage classification and building segmentation more accurately than other approaches. Furthermore, to detect the changes of building height, we present a method for building height estimation using building and shadow instance detection and satellite image metadata. We propose a multi-stage instance detection method to achieve accurate instance detection with limited data annotation. We show that the multi-stage instance detection method achieves better performance than the compared approach. Given the previous instance detection results, we propose a method to estimate building height with satellite image metadata, including ground sample distance, solar angles, and satellite angles. We show that the proposed height estimation method achieves good performance on the proposed dataset as well as a new dataset even without finetuning. In order to provide an accurate building segmentation as required in the previously proposed methods, we present a method that can provide accurate building segmentation despite the data noise that is caused by large off-nadir angles. We use uncertainty modeling and satellite imagery metadata to achieve accurate building segmentation for the noisy images that are taken

from large off-nadir angles. By conducting the experimental analysis and ablation study, we show that the proposed method is able to achieve a clear improvement compared to the baseline method, especially for the noisy images taken from large off-nadir angles.

Although we show that the proposed Siam-U-Net-Attn model can provide accurate damage scale classification based on the feature difference between the two images taken before and after a disaster, dealing with temporal image sequence that contains more than two images is not a trivial task that can be extended from our current approach. Such multi-temporal change detection task requires the model accumulating the difference through multiple frames taken from different times, which needs more complex temporal-aware model, such as long short-term memory (LSTM) [104] or transformer [168]. Therefore, more work is required to extend the proposed method for the multi-temporal change detection task. Furthermore, instance segmentation is a potential method to improve the damage scale classification accuracy. Since the current model can only produce the semantic segmentation mask of buildings, it is not able to differentiate each building instance, especially when buildings are close to each other. Therefore, like the method mentioned in Section 3.2.3, instance segmentation method is able to assign a consistent label for each building. However, based on our experiments, simply applying the the instance segmentation-based methods, such as Mask R-CNN [108] to our damage scale classification task cannot achieve better performance than the proposed Siam-U-Net-Attn model. Because of the large amount of small buildings in the xView2 dataset, many inaccurate proposals (including both false positive and false negative cases) are generated from the region proposal network (RPN) in the Mask R-CNN model, which causes such poor performance. As a future work, the non-RPN based instance detection methods (*e.g.*, DETR [169]) can be potentially useful to improve the damage scale classification performance. Lastly, in this thesis, we analyze the use of uncertainty modeling and metadata injection to improve the task of building segmentation. Since using uncertainty modeling and metadata injection can enable the model being less sensitive to the noise from input data, they are able to improve other vision tasks (*e.g.*, object detection and object classification) as well. Therefore, as a future work, additional experimental analysis is needed to evaluate the improvement of the proposed method for other vision tasks.

4.3 Contributions Of This Thesis

The main contributions of the thesis are listed as follows:

- Utility-Preserving Face Redaction

    1. A performance analysis of face obscuration approaches is proposed.
    2. The analysis is based on three attack scenarios: obscured face identification, verification, and reconstruction.
    3. We analyze these attacks based on two widely used deep learning models, VGG19 [8] and ResNet50 [9] in different threat model conditions.
    4. We show that the traditional obscuration methods, such as blurring and pixelation can not guarantee privacy protection.
    5. To provide a more secured privacy protection, we propose two novel obscuration methods that are based on the generative adversarial networks.
    6. With qualitative and quantitative analysis, we show that the proposed methods can not only remove the identifiable information, but also preserve the non-identifiable facial information, such as facial expression, age, skin tone and gender.

- Change Detection For Satellite Imagery

    1. We develop a multi-class deep learning model with attention technique that accurately classifies damage levels of buildings based on 2D appearance changes in satellite imagery.
    2. We demonstrate that the proposed model achieves better results for building damage scale classification than other methods while simultaneously achieving accurate building segmentation results.
    3. To detect the changes from 3D building height, we propose a building height estimation model.
    4. The proposed method can estimate building height based on building shadows and solar angles without relying on height annotations.

5. We qualitatively and quantitatively show that the proposed method achieves accurate building height estimation.

6. To provide a more reliable building segmentation method as required in the previously proposed change detection methods, we present a model that can provide accurate building segmentation even for the low quality satellite images captured from a large range of off-nadir angles.

7. Both uncertainty modeling and satellite imagery metadata are used in the proposed method to achieve a good building segmentation performance, especially for the noisy images taken from large off-nadir angles.

4.4 Publications Resulting From This Thesis

Conference Papers

- **Hanxiang Hao**, David Güera, Amy R. Reibman, Edward J. Delp, "A Utility-Preserving GAN for Face Obscuration", Proceedings of the International Conference on Machine Learning, Synthetic Realities: Deep Learning for Detecting AudioVisual Fakes Workshop, June 2019, Long Beach, CA.

- **Hanxiang Hao**, David Güera, János Horváth, Amy R. Reibman, Edward J. Delp, "Robustness Analysis of Face Obscuration", Proceedings of the International Conference on Automatic Face and Gesture Recognition, November 2020, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Amy R. Reibman, Edward J. Delp, "FaR-GAN for One-Shot Face Reenactment", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, AI for Content Creation Workshop, May 2020, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Emily Bartusiak, Latisha Konz, Kevin LaTourette, Michael Gribbons, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "An Attention-Based System for Damage Assessment Using Satellite Imagery", Proceedings of International Geoscience and Remote Sensing Symposium, July 2021, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Emily Bartusiak, Mridul Gupta, Kevin LaTourette, Latisha Konz, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "Building Height Estimation via Satellite Metadata and Shadow Instance Detection", Proceedings of SPIE 11729, Automatic Target Recognition XXXI, April 2021, Virtual Conference.

- **Hanxiang Hao**, Sriram Baireddy, Kevin LaTourette, Latisha Konz, Moses W. Chan, Mary L. Comer, and Edward J. Delp, "Improving Building Segmentation Using Uncertainty Modeling and Metadata Injection", Proceedings of ACM SIGSPATIAL: International Conference on Advances in Geographic Information Systems, November 2021, Virtual Conference.

4.5 Other Publications Not Related to This Thesis

Book Chapters

- **Hanxiang Hao**, Emily R. Bartusiak, David Güera, Daniel M. Montserrat, Sriram Baireddy, Ziyue Xiang, Sri K. Yarlagadda, Ruiting Shao, János Horváth, Justin Yang, Fengqing M. Zhu, Edward J. Delp, "Handbook of Digital Face Manipulation and Detection - From DeepFakes to Morphing Attacks", Advances in Computer Vision and Pattern Recognition, Springer, 2022 (To Be Published)

Conference Papers

- Daniel M. Montserrat, **Hanxiang Hao**, Sri K. Yarlagadda, Sriram Baireddy, Ruiting Shao, János Horváth, Justin Yang, Emily R. Bartusiak, David Güera, Fengqing M. Zhu, and Edward J. Delp, "Deepfakes Detection with Automatic Face Weighting", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2020, Virtual Conference.

- János Horváth, Daniel M. Montserrat, **Hanxiang Hao**, and Edward J. Delp, "Manipulation Detection in Satellite Images Using Deep Belief Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2020, Virtual Conference.

- János Horváth, Sriram Baireddy, **Hanxiang Hao**, Daniel M. Montserrat, and Edward J. Delp, “Manipulation Detection in Satellite Images Using Vision Transformer”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Media Forensics, June 2021, Virtual Conference.

- Emily R. Bartusiak, **Hanxiang Hao**, Michael Jacobs, Nhat X. Nguyen, Moses W. Chan, Mary L. Comer, and Edward J. Delp, “A Stochastic Grammar Approach to Predict Flight Phases of a Hypersonic Glide Vehicle”, Proceedings of the IEEE Aerospace Conference, March 2022, Montana, USA

REFERENCES

[1] F. Dufaux and T. Ebrahimi, "A framework for the validation of privacy protection solutions in video surveillance," *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 66–71, Jul. 2010, Singapore, Singapore. DOI: https://doi.org/10.1109/ICME.2010.5583552.

[2] R. McPherson, R. Shokri, and V. Shmatikov, "Defeating image obfuscation with deep learning," *arXiv:1609.00408v2*, Sep. 2016. [Online]. Available: https://arxiv.org/abs/1609.00408.

[3] S. Sah, A. Shringi, R. Ptucha, A. M. Burry, and R. P. Loce, "Video redaction: A survey and comparison of enabling technologies," *Journal of Electronic Imaging*, vol. 26, no. 5, pp. 1–14, Jul. 2017. [Online]. Available: https://doi.org/10.1117/1.JEI.26.5.051406.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Jun. 2019. [Online]. Available: https://doi.org/10.1109/CVPR.2019.00453.

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *Proceedings of the IEEE International Conference on Learning Representations*, Apr. 2018. [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb.

[6] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xbd: A dataset for assessing building damage from satellite imagery," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2019. [Online]. Available: https://arxiv.org/abs/1911.09296.

[7] G. Christie, R. R. R. Munoz Abujder, K. Foster, S. Hagstrom, G. D. Hager, and M. Z. Brown, "Learning geocentric object pose in oblique monocular images," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14 500–14 508, Jun. 2020, Seattle, WA. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01452.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proceedings of the International Conference on Learning Representations*, May 2015, San Diego, CA. [Online]. Available: http://arxiv.org/abs/1409.1556.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Jun. 2016, Las Vegas, NV. DOI: 10.1109/CVPR.2016.90.

[10] L. Du, M. Yi, E. Blasch, and H. Ling, "Garp-Face: Balancing privacy protection and utility preservation in face de-identification," *Proceedings of the IEEE International Joint Conference on Biometrics*, pp. 1–8, Sep. 2014, Clearwater, FL. [Online]. Available: https://doi.org/10.1109/BTAS.2014.6996249.

[11] R. Gross, E. Airoldo, B. Malin, and L. Sweeney, "Integrating utility into face de-identification," *Proceedings of the International Workshop on Privacy Enhancing Technologies*, pp. 227–242, May 2005, Cavtat, Croatia. [Online]. Available: https://doi.org/10.1007/11767831_15.

[12] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," *Technical Report*, 1998, Harvard Data Privacy Laboratory. [Online]. Available: https://epic.org/privacy/reidentification/Samarati_Sweeney_paper.pdf.

[13] E. M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE Transactions on Knowledge and Data Engineering*, pp. 232–243, Feb. 2005. [Online]. Available: https://doi.org/10.1109/TKDE.2005.32.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 484–498, Jun. 2001. [Online]. Available: https://doi.org/10.1109/34.927467.

[15] B. Meden, Ž. Emeršič, V. Štruc, and P. Peer, "K-same-net: K-anonymity with generative deep neural networks for face deidentification," *Entropy*, vol. 20, no. 1, Jan. 2018. [Online]. Available: https://doi.org/10.3390/e20010060.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of Advances in Neural Information Processing Systems*, pp. 2672–2680, Dec. 2014, Montréal, Canada. [Online]. Available: https://arxiv.org/abs/1406.2661.

[17] Y. Wu, F. Yang, Y. Xu, and H. Ling, "Privacy-protective-gan for privacy preserving face de-identification," *Journal of Computer Science and Technology*, pp. 47–60, Jan. 2019, Beijing, China. [Online]. Available: https://doi.org/10.1007/s11390-019-1898-8.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784v1*, Nov. 2014. [Online]. Available: https://arxiv.org/abs/1411.1784.

[19] Q. Sun, L. Ma, S. Joon Oh, L. V. Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5050–5059, Jun. 2018, Salt Lake City, UT. [Online]. Available: https://doi.org/10.1109/CVPR.2018.00530.

[20] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2089–2093, Sep. 2017, Beijing, China. [Online]. Available: https://doi.org/10.1109/ICIP.2017.8296650.

[21] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Attribute-guided face generation using conditional cyclegan," *arXiv:1705.09966v2*, Nov. 2018. [Online]. Available: https://arxiv.org/abs/1705.09966.

[22] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, Nov. 2018, Auckland, New Zealand. [Online]. Available: https://doi.org/10.1109/AVSS.2018.8639163.

[23] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4352–4360, Jul. 2017, Hawaii, HI. [Online]. Available: https://doi.org/10.1109/CVPR.2017.463.

[24] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882269.

[25] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: Automatically replacing faces in photographs," *ACM Transactions on Graphics*, vol. 27, no. 3, 39:1–39:8, 2008. [Online]. Available: https://doi.org/10.1145/1360612.1360638.

[26] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3697–3705, Oct. 2017, Venice, Italy. [Online]. Available: https://doi.org/10.1109/ICCV.2017.397.

[27] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, "Learning to generate chairs, tables and cars with convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 692–705, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2567384.

[28] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *European Conference on Computer Vision*, pp. 694–711, 2016, Amsterdam, Netherlands. [Online]. Available: https://doi.org/10.1007/978-3-319-46475-6_43.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y.

[30] H. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," *Proceedings of the IEEE International Conference on Image Processing*, pp. 343–347, Oct. 2014, Paris, France. [Online]. Available: https://doi.org/10.1109/ICIP.2014.7025068.

[31] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1755843.

[32] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, vol. 2, pp. 958–963, Aug. 2003, Edinburgh, Scotland. [Online]. Available: https://doi.org/10.1109/ICDAR.2003.1227801.

[33] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in Neural Information Processing Systems*, pp. 6629–6640, 2017, Long Beach, CA. [Online]. Available: http://arxiv.org/abs/1706.08500.

[34] H. Hao, D. Güera, A. R. Reibman, and E. J. Delp, "A utility-preserving gan for face obscuration," *Proceedings of the International Conference on Machine Learning, Synthetic Realities: Deep Learning for Detecting AudioVisual Fakes Workshop*, Jun. 2019, Long Beach, CA. [Online]. Available: https://arxiv.org/abs/1906.11979.

[35] M.-R. Ra, R. Govindan, and A. Ortega, "P3: Toward privacy-preserving photo sharing," *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, pp. 515–528, Apr. 2013, Lombard, IL. [Online]. Available: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/ra.

[36] C. Pares-Pulido and I. Agudo, "Lockpic: Privacy preserving photo sharing in social networks," *Proceedings of the International Workshop on Data Privacy Management, and Security Assurance*, pp. 281–290, Sep. 2016, Heraklion, Crete. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-29883-2_21.

[37] L. Yuan, P. Korshunov, and T. Ebrahimi, "Privacy-preserving photo sharing based on a secure jpeg," *Proceedings of the IEEE Conference on Computer Communications Workshops*, pp. 185–190, Apr. 2015, Hong Kong, China. [Online]. Available: https://doi.org/10.1109/INFCOMW.2015.7179382.

[38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, Jun. 2016, Las Vegas, NV. [Online]. Available: https://arxiv.org/abs/1611.07004.

[39] P. Samarati and L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1142/S0218488502001648.

[40] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pp. 138–142, Dec. 1994, Sarasota, USA. [Online]. Available: https://doi.org/10.1109/ACV.1s994.341300.

[41] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, "Faceless person recognition: Privacy implications in social media," *Proceedings of European Conference on Computer Vision*, pp. 19–35, Jan. 2016, Amsterdam, The Netherlands. [Online]. Available: https://doi.org/10.1007/978-3-319-46487-9_2.

[42] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2019, Long Beach, CA. [Online]. Available: https://arxiv.org/pdf/1801.07698.

[43] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VggFace2: A dataset for recognising faces across pose and age," *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, May 2018, Xi'an, China. [Online]. Available: https://arxiv.org/abs/1710.08092.

[44] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2892–2900, Jun. 2015, Boston, MA. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298907.

[45] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," *Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems*, pp. 1–8, Sep. 2016, Buffalo, New York. [Online]. Available: https://arxiv.org/abs/1604.05417.

[46] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6738–6746, Jul. 2017, Hawaii, HI. [Online]. Available: https://arxiv.org/abs/1704.08063.

[47] H. J. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, Jun. 2018, Salt Lake City, UT. [Online]. Available: https://arxiv.org/abs/1801.09414.

[48] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," *Technical Report*, Oct. 2007, University of Massachusetts, Amherst. [Online]. Available: http://vis-www.cs.umass.edu/lfw/lfw.pdf.

[49] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nieundefinedner, "Face2Face: Real-time face capture and reenactment of RGB videos," *Communications of the ACM*, vol. 62, no. 1, pp. 96–104, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3292039.

[50] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," *Conference on Computer Graphics and Interactive Techniques*, pp. 187–194, Aug. 1999, Los Angeles, CA. [Online]. Available: https://doi.org/10.1145/311535.311556.

[51] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," *IEEE International Conference on Computer Vision*, pp. 7183–7192, Oct. 2019, Seoul, Korea. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00728.

[52] J. Thies, M. Zollhöfer, and M. Nieundefinedner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics*, vol. 38, no. 4, 2019. [Online]. Available: https://doi.org/10.1145/3306346.3323035.

[53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, Jun. 2015. [Online]. Available: https://arxiv.org/abs/1505.04597.

[54] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *arXiv:1912.04958*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1912.04958.

[55] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, "Reenactgan: Learning to reenact faces via boundary transfer," *Proceedings of the European Conference on Computer Vision*, Sep. 2018. [Online]. Available: https://arxiv.org/abs/1807.11079.

[56] O. Wiles, A. Koepke, and A. Zisserman, "X2face: A network for controlling face generation by using images, audio, and pose codes," *Proceedings of the European Conference on Computer Vision*, Sep. 2018. [Online]. Available: https://arxiv.org/abs/1807.10550.

[57] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2019. [Online]. Available: https://arxiv.org/abs/1905.08233.

[58] Y. Zhang, S. Zhang, Y. He, C. Li, C. C. Lo, and Z. Liu, "One-shot face reenactment," *Proceedings of the British Machine Vision Conference*, Sep. 2019. [Online]. Available: https://arxiv.org/abs/1908.03251.

[59] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," *IEEE Conference on Computer Vision and Pattern Recognitionn*, pp. 4295–4304, Jun. 2015, Boston, MA. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7299058.

[60] T. Park, M. Liu, T. Wang, and J. Zhu, "Semantic image synthesis with spatially-adaptive normalization," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2332–2341, Jun. 2019. [Online]. Available: https://doi.org/10.1109/CVPR.2019.00244.

[61] S. Ioffe and C. Szegedy, "Semantic image synthesis with spatially-adaptive normalization," *International Conference on Machine Learning*, pp. 448–456, Jul. 2015. [Online]. Available: https://doi.org/10.5555/3045118.3045167.

[62] X. Liu, G. Yin, J. Shao, X. Wang, and h. Li, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," *Proceedings of Advances in Neural Information Processing Systems*, pp. 570–580, Dec. 2019, Vancouver, Canada. [Online]. Available: https://arxiv.org/abs/1910.06809.

[63] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *IEEE International Conference on Computer Vision*, pp. 2813–2821, Oct. 2017. [Online]. Available: https://doi.org/10.1109/ICCV.2017.304.

[64] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," *International Conference on Neural Information Processing Systems*, pp. 5769–5779, Dec. 2017. [Online]. Available: https://doi.org/10.5555/3295222.3295327.

[65] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *International Conference on Learning Representations*, Apr. 2018. [Online]. Available: https://arxiv.org/abs/1802.05957.

[66] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *Proceedings of the IEEE International Conference on Machine Learning*, vol. 97, pp. 7354–7363, Jun. 2019. [Online]. Available: http://proceedings.mlr.press/v97/zhang19d.html.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, May 2015, San Diego, CA. [Online]. Available: http://arxiv.org/abs/1412.6980.

[68] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Conference of the International Speech Communication Association*, Sep. 2017. [Online]. Available: https://arxiv.org/abs/1706.08612.

[69] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. [Online]. Available: https://doi.org/10.1109/TIP.2003.819861.

[70] W. H. O. (WHO), *Environmental health in emergencies*. [Online]. Available: https://www.who.int/environmental_health_emergencies/natural_events/en.

[71] L. Boustan, M. Kahn, P. Rhode, and M. Yanguas, "The effect of natural disasters on economic activity in us counties: A century of data," *National Bureau of Economic Research, Inc.*, vol. NBER Working Papers 23410, Jun. 2019. [Online]. Available: https://www.nber.org/papers/w23410.pdf.

[72] K. Amadeo, *Haiti earthquake facts, its damage, and effects on the economy*. [Online]. Available: https://www.thebalance.com/haiti-earthquake-facts-damage-effects-on-economy-3305660.

[73] N. N. C. for Environmental Information (NCEI), *U.s. billion-dollar weather and climate disasters (2020)*. [Online]. Available: https://www.ncdc.noaa.gov/billions/.

[74] M. V. Aalst, "The impacts of climate change on the risk of natural disasters," *Disasters*, Mar. 2006. [Online]. Available: https://doi.org/10.1111/j.1467-9523.2006.00303.x.

[75] "Digitalglobe search and discovery." (2019), [Online]. Available: https://discover.digitalglobe.com.

[76] A. Asokan and A. Jude, "Change detection techniques for remote sensing applications: A survey," *Earth Science Informatics*, vol. 12, no. 2, pp. 1–18, Mar. 2019. [Online]. Available: https://doi.org/10.1007/s12145-019-00380-5.

[77] L. Ke, Y. Lin, Z. Zheng, L. Zhang, and L. Meng, "Adaptive change detection with significance test," *IEEE Access*, vol. 6, pp. 27 442–27 450, Feb. 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2807380.

[78] Z. Qi, A. G.-O. Yeh, X. Li, and X. Zhang, "A three-component method for timely detection of land cover changes using polarimetric sar images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 107, pp. 3–21, Sep. 2015. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2015.02.004.

[79] V. Sadeghi, F. F. Ahmadi, and H. Ebadi, "Design and implementation of an expert system for updating thematic maps using satellite imagery (case study: Changes of lake urmia," *Arabian Journal of Geosciences*, vol. 9, no. 257, Mar. 2016. [Online]. Available: https://doi.org/10.1007/s12517-015-2301-x.

[80] L. Gueguen and R. Hamid, "Large-scale damage detection using satellite imagery," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1321–1328, Jun. 2015, Boston, MA. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298737.

[81] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000. [Online]. Available: https://doi.org/10.1109/83.841532.

[82] N. Chehata, C. Orny, S. Boukir, D. Guyon, and J. Wigneron, "Object-based change detection in wind storm-damaged forest using high-resolution multispectral images," *IEEE Transactions on Image Processing*, vol. 35, no. 13, pp. 4758–4777, 2014. [Online]. Available: https://doi.org/10.1080/01431161.2014.930199.

[83] J. Xu, W. Lu, Z. Li, P. Khaitan, and V. Zaytseva, "Building damage detection in satellite imagery using convolutional neural networks," *arXiv:1910.06444*, Oct. 2019. [Online]. Available: https://arxiv.org/abs/1910.06444.

[84] A. Fujita, K. Sakurada, T. Imaizumi, R. Ito, S. Hikosaka, and R. Nakamura, "Damage detection from aerial images via convolutional neural networks," *Proceedings of the IAPR International Conference on Machine Vision Applications*, May 2017. [Online]. Available: https://doi.org/10.23919/MVA.2017.7986759.

[85] F. Nex, D. Duarte, F. Tonolo, and N. Kerle, "Structural building damage detection with deep learning: Assessment of a state-of-the-art cnn in operational conditions," *Remote Sensing*, vol. 11, p. 2765, Nov. 2019. [Online]. Available: https://doi.org/10.3390/rs11232765.

[86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, Jul. 2017, Honolulu, HI. DOI: 10.1109/CVPR.2017.243.

[87] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv*, vol. 1511.07122, Apr. 2016. [Online]. Available: https://arxiv.org/abs/1511.07122.

[88] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924–935, Feb. 2019. [Online]. Available: https://doi.org/10.1109/TGRS.2018.2863224.

[89] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sensing*, vol. 8, no. 6, 2016. [Online]. Available: https://doi.org/10.3390/rs8060506.

[90] C. Connors and R. R. Vatsavai, "Semi-supervised deep generative models for change detection in very high resolution imagery," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 1063–1066, Jul. 2017. [Online]. Available: https://doi.org/10.1109/IGARSS.2017.8127139.

[91] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 3581–3589, Dec. 2014. [Online]. Available: https://doi.org/10.1109/IGARSS.2017.8127139.

[92] H. Liu, Z. Wang, F. Shang, M. Zhang, M. Gong, F. Ge, and L. Jiao, "A novel deep framework for change detection of multi-source heterogeneous images," *International Conference on Data Mining Workshops*, pp. 165–171, Nov. 2019. [Online]. Available: https://doi.org/10.1109/ICDMW.2019.00034.

[93] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, Dec. 2017. [Online]. Available: https://doi.org/10.1109/TCSVT.2016.2589879.

[94] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv:1711.04043*, Feb. 2018. [Online]. Available: https://arxiv.org/abs/1711.04043.

[95] J. Sublime and E. Kalinicheva, "Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the tohoku tsunami," *Remote Sensing*, vol. 11, no. 9, p. 1123, 2019. [Online]. Available: https://doi.org/10.3390/rs11091123.

[96] J. Doshi, S. Basu, and G. Pang, "From satellite imagery to disaster insights," *Proceedings of the Conference on Neural Information Processing Systems at AI for Social Good Workshop*, Dec. 2018, Montreal, Canada. [Online]. Available: https://arxiv.org/abs/1812.07033.

[97] K. L. de Jong and A. S. Bosman, "A fast learning algorithm for deep belief nets," *Proceedings of the International Joint Conference on Neural Networks)*, Jul. 2019, Budapest, Hungary. [Online]. Available: https://arxiv.org/abs/1812.05815.

[98] J. Liu, M. Gong, K. Qin, and P. Zhang, "A deep convolutional coupling network for change detection based on heterogeneous optical and radar images," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 545–559, Mar. 2018. [Online]. Available: https://doi.org/10.1109/TNNLS.2016.2636227.

[99] B. Demir, F. Bovolo, and L. Bruzzone, "Updating land-cover maps by classification of image time series: A novel change-detection-driven transfer learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 300–312, Jan. 2013. [Online]. Available: https://doi.org/10.1109/TGRS.2012.2195727.

[100] T. Rudner, M. Rubwurm, J. Fil, R. Pelich, B. Bischke, V. Kopackova, and P. Bilinski, "Multi3net: Segmenting flooded buildings via fusion of multiresolution, multisensor, and multitemporal satellite imagery," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 702–709, Jul. 2019, Honolulu, Hawaii. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.3301702.

[101] Y. Chu, G. Cao, and H. Hayat, "Change detection of remote sensing image based on deep neural networks," *Proceedings of the International Conference on Artificial Intelligence and Industrial Engineering*, Nov. 2016, Beijing, China. [Online]. Available: https://doi.org/10.2991/aiie-16.2016.61.

[102] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computing*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006. [Online]. Available: https://doi.org/10.1162/neco.2006.18.7.1527.

[103] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzalos, "Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2019, Yokohama, Japan. [Online]. Available: https://arxiv.org/abs/1910.07778.

[104] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[105] R. C. Daudt, B. L. Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," *Proceedings of the IEEE International Conference on Image Processing*, pp. 4063–4067, Oct. 2018, Athens, Greece. [Online]. Available: https://doi.org/10.1109/ICIP.2018.8451652.

[106] R. C. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "Multitask learning for large-scale semantic change detection," *Computer Vision and Image Understanding*, vol. 187, p. 102 783, Oct. 2019. [Online]. Available: https://doi.org/10.1016/j.cviu.2019.07.003.

[107] E. Weber and H. Kan, "Building disaster damage assessment in satellite imagery with multi-temporal fusion," *Proceedings of the IEEE International Conference on Learning Representations: AI For Earth Sciences Workshop*, Apr. 2020. [Online]. Available: https://doi.org/10.1109/CVPR.2018.00813.

[108] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2017. [Online]. Available: https://doi.org/10.1109/ICCV.2017.322.

[109] M. Dunnhofer, M. Antico, F. Sasazawa, Y. Takeda, S. Camps, N. Martinel, C. Micheloni, G. Carneiro, and D. Fontanarosa, "Siam-u-net: Encoder-decoder siamese network for knee cartilage tracking in ultrasound images," *Medical Image Analysis*, vol. 60, p. 101 631, 2020. [Online]. Available: https://doi.org/10.1016/j.media.2019.101631.

[110] D. Kwon, J. Ahn, J. Kim, I. Choi, S. Jeong, Y.-S. Lee, J. Park, and M. Lee, "Siamese u-net with healthy template for accurate segmentation of intracranial hemorrhage," *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 848–855, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-32248-9_94.

[111] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, Salt Lake City, UT. [Online]. Available: https://doi.org/10.1109/CVPR.2018.00813.

[112] V. Durnov, *1st place solution for xview2: Assess building damage challenge*. [Online]. Available: https://github.com/DIUx-xView/xView2_first_place.

[113] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2018, Salt Lake City, UT. [Online]. Available: https://arxiv.org/abs/1709.01507.

[114] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, Salt Lake City, UT. [Online]. Available: https://arxiv.org/abs/1611.05431.

[115] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Jun. 2009, Miami, FL. DOI: 10.1109/CVPR.2009.5206848.

[116] L. Mou and X. X. Zhu, "IM2HEIGHT: height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," *arXiv:1802.10249*, Aug. 2018. [Online]. Available: https://arxiv.org/abs/1802.10249.

[117] S. Srivastava, M. Volpi, and D. Tuia, "Joint height estimation and semantic labeling of monocular aerial images with cnns," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, pp. 5173–5176, Jul. 2017, Fort Worth, TX. [Online]. Available: https://doi.org/10.1109/IGARSS.2017.8128167.

[118] P. Ghamisi and N. Yokoya, "Img2dsm: Height simulation from single imagery using conditional generative adversarial net," *Proceedings of the IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 794–798, 2018. [Online]. Available: https://doi.org/10.1109/LGRS.2018.2806945.

[119] Y. Shao, G. Taff, and S. Walsh, "Shadow detection and building-height estimation using ikonos data," *International Journal of Remote Sensing*, pp. 6929–6944, Nov. 2011. [Online]. Available: https://doi.org/10.1080/01431161.2010.517226.

[120] M. Izadi and P. Saeedi, "Three-dimensional polygonal building model estimation from single satellite images," *IEEE Transactions of Geoscience and Remote Sensing*, vol. 50, no. 6, pp. 2254–2272, 2012. [Online]. Available: https://doi.org/10.1109/TGRS.2011.2172995.

[121] R. Gouiaa and J. Meunier, "3d reconstruction by fusioning shadow and silhouette information," *Proceedings of the Canadian Conference on Computer and Robot Vision*, pp. 378–384, May 2014, Montreal, Canada. [Online]. Available: https://doi.org/10.1109/CRV.2014.58.

[122] R. Gouiaa and J. Meunier, "Shadow analysis technique for extraction of building height using high resolution satellite single image and accuracy assessment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. XL-8, pp. 1185–1192, Nov. 2014. DOI: 10.5194/isprsarchives-XL-8-1185-2014.

[123] G. Liasis and S. Stavrou, "Satellite images analysis for shadow detection and building height estimation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 19, pp. 437–450, 2016. DOI: 10.1016/j.isprsjprs.2016.07.006.

[124] N. Kadhim and M. Mourshed, "3d reconstruction by fusioning shadow and silhouette information," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 1, pp. 8–12, 2018. [Online]. Available: https://doi.org/10.1109/LGRS.2017.2762424.

[125] A. Trekin, V. Ignatiev, and P. Yakubovskiy, "Deep neural networks for determining the parameters of buildings from single-shot satellite imagery," *Computer and Systems Sciences Internationa*, vol. 59, pp. 755–767, 2020. DOI: 10.1134/S106423072005007X.

[126] T. Wang, X. Hu, Q. Wang, P. .-. Heng, and C. .-. Fu, "Instance shadow detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1877–1886, Jun. 2020, Seattle, WA. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.00195.

[127] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 27, Dec. 2014, Montreal CANADA. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[128] S. O. Mehdi Mirza, "Conditional generative adversarial nets," *arxiv:1411.1784*, Nov. 2014. [Online]. Available: https://arxiv.org/abs/1411.1784.

[129] M. Nagao, T. Matsuyama, and Y. Ikeda, "Region extraction and shape analysis in aerial photographs," *Computer Graphics and Image Processing*, vol. 10, no. 3, pp. 195–223, Jul. 1979. [Online]. Available: https://doi.org/10.1016/0146-664X(79)90001-7.

[130] V. J. D. Tsai, "A comparative study on shadow compensation of color aerial images in invariant color models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1661–1671, Jun. 2006. [Online]. Available: https://doi.org/10.1109/TGRS.2006.869980.

[131] G. Liasis and S. Stavrou, "Optimizing level set initialization for satellite image segmentation," *Proceedings of the 20th IEEE International Conference on Telecommunications*, pp. 1–5, May 2013, Casablanca, Morocco. [Online]. Available: https://doi.org/10.1109/ICTEL.2013.6632078.

[132] A. Manno-Kovacs and T. Sziranyi, "Orientation-selective building detection in aerial images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 94–112, Oct. 2015. [Online]. Available: https://doi.org/10.1016/j.isprsjprs.2015.06.007.

[133] R. B. Irvin and D. M. McKeown, "Methods for exploiting the relationship between buildings and their shadows in aerial imagery," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 1564–1575, 6 Nov. 1989. [Online]. Available: https://doi.org/10.1109/21.44071.

[134] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, Oct. 2017, Venice, Italy. [Online]. Available: https://doi.org/10.1109/ICCV.2017.322.

[135] N. Weir, D. Lindenbaum, A. Bastidas, A. Etten, V. Kumar, S. Mcpherson, J. Shermeyer, and H. Tang, "Spacenet mvoi: A multi-view overhead imagery dataset," *IEEE/CVF International Conference on Computer Vision*, pp. 992–1001, Oct. 2019, Seoul, Korea. DOI: 10.1109/ICCV.2019.00108.

[136] A. V. Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv*, vol. 1807.01232, Aug. 2028. [Online]. Available: http://arxiv.org/abs/1807.01232.

[137] J. Yap, *2018 open ai tanzania building footprint segmentation challenge*. [Online]. Available: https://competitions.codalab.org/competitions/20100#learn_the_details-overview.

[138] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 32, 2019, Vancouver, Canada. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/d0f4dae80c3d0277922f8371d5827292-Paper.pdf.

[139] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2577031.

[140] R. P. Brent, *Algorithms for Minimization without Derivatives*. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

[141] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: https://doi.org/10.1038/s41592-019-0686-2.

[142] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. G. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose, D. Wilson, A. Tudor, N. Hovakimyan, T. S. Huang, and H. Shi, "Agriculture-vision: A large aerial image database for agricultural pattern analysis," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2825–2835, Jun. 2020, Seoul, Korea. DOI: 10.1109/CVPR42600.2020.00290.

[143] A. V. Etten, D. Hogan, J. Martinez-Manso, J. Shermeyer, N. Weir, and R. Lewis, "The multi-temporal urban development spacenet dataset," *arXiv*, vol. 2102.04420, Feb. 2021. [Online]. Available: https://arxiv.org/abs/2102.04420.

[144] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *European Conference on Computer Vision*, pp. 740–755, Sep. 2014, Zurich, Switzerland. [Online]. Available: https://arxiv.org/abs/1405.0312.

[145] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, Jun. 2016, Las Vegas, NV. DOI: 10.1109/CVPR.2016.350.

[146] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 936–944, Jul. 2017, Honolulu, HI. DOI: 10.1109/CVPR.2017.106.

[147] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Jan. 2017. DOI: 10.1109/TPAMI.2016.2644615.

[148] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, Jul. 2017, Honolulu, HI. DOI: 10.1109/CVPR.2017.632.

[149] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," *arXiv*, vol. 1801.05746, Aug. 2018. [Online]. Available: http://arxiv.org/abs/1801.05746.

[150] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018. DOI: 10.1109/TPAMI.2017.2699184.

[151] Z. Zhang and Y. Wang, "Jointnet: A common neural network for road and building extraction," *Remote Sensing*, vol. 11, no. 6, p. 696, Mar. 2019. DOI: 10.3390/rs11060696.

[152] W. Liu, J. Xu, Z. Guo, E. Li, X. Li, L. Zhang, and W. Liu, "Building footprint extraction from unmanned aerial vehicle images via pru-net: Application to change detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2236–2248, Jan. 2021. DOI: 10.1109/JSTARS.2021.3052495.

[153] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6230–6239, Jul. 2017, Honolulu, HI. DOI: 10.1109/CVPR.2017.660.

[154] H. Jing, X. Sun, Z. Wang, K. Chen, W. Diao, and K. Fu, "Fine building segmentation in high-resolution sar images via selective pyramid dilated network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–1, Apr. 2021. DOI: 10.1109/JSTARS.2021.3076085.

[155] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, vol. 1511.07122, pp. 680–688, Jun. 2016, Las Vegas, NV. DOI: 10.1109/CVPRW.2016.90.

[156] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *International Conference on Learning Representations*, vol. 48, pp. 1050–1059, Jun. 2016, New York City, NY. [Online]. Available: http://proceedings.mlr.press/v48/gal16.html.

[157] B. Bischke, P. Helber, D. Borth, and A. Dengel, "Segmentation of imbalanced classes in satellite imagery using adaptive uncertainty weighted class loss," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 6191–6194, Jul. 2018, Valencia, Spain. DOI: 10.1109/IGARSS.2018.8517836.

[158] M. Pritt and G. Chern, "Satellite image classification with deep learning," *IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1–7, Jul. 2017, Washington, DC. DOI: 10.1109/AIPR.2017.8457969.

[159] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6172–6180, Jun. 2018, Salt Lake City, UT. DOI: 10.1109/CVPR.2018.00646.

[160] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *International Conference on Machine Learning*, May 2016, San Juan, Puerto Rico. [Online]. Available: https://arxiv.org/abs/1506.02158.

[161] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Conference on Neural Information Processing Systems*, vol. 30, Dec. 2017, Long Beach, CA. [Online]. Available: https://arxiv.org/abs/1703.04977.

[162] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *British Machine Vision Conference*, pp. 57.1–57.12, Sep. 2017, London, United Kingdom. DOI: 10.5244/C.31.57.

[163] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," *Conference on Neural Information Processing Systems*, pp. 950–957, Dec. 1991, Denver, Colorado. [Online]. Available: https://dl.acm.org/doi/10.5555/2986916.2987033.

[164] R. Gouiaa and J. Meunier, "3d reconstruction by fusioning shadow and silhouette information," *IEEE Canadian Conference on Computer and Robot Vision*, pp. 378–384, May 2014, Montreal, Canada. DOI: 10.1109/CRV.2014.58.

[165] H. Hao, S. Baireddy, E. Bartusiak, M. Gupta, K. LaTourette, L. Konz, M. Chan, M. L. Comer, and E. J. Delp, "Building height estimation via satellite metadata and shadow instance detection," *Automatic Target Recognition XXXI*, vol. 11729, pp. 175–190, Apr. 2021. DOI: 10.1117/12.2585012.

[166] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, vol. 30, pp. 7877–7886, Jun. 2020, Seattle, WA. DOI: 10.1109/CVPR42600.2020.00790.

[167] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. Zaiane, and M. Jagersand, "U^2-net: Going deeper with nested u-structure for salient object detection," *Pattern Recognition*, vol. 106, p. 107 404, 2020. DOI: 10.1016/j.patcog.2020.107404.

[168] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017, Long Beach, CA. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[169] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *European Conference on Computer Vision*, 2020. [Online]. Available: https://arxiv.org/abs/2005.12872.

VITA

Hanxiang Hao was born in Beijing, China on March 23, 1993. He received the Bachelor of Science degree in Electrical Engineering and Automation from Beijing Institute of Technology, Beijing, China. He obtained his Master of Engineering degree in Electrical and Computer Engineering from Cornell University, Ithaca, NY. He worked at Softwear Automation Inc., Atlanta, GA as a computer vision engineer for one year. Mr. Hao then joined the Ph.D. program at the School of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana in August 2018. He worked at the Video and Image Processing Laboratory (VIPER) under the supervision of Professor Edward J. Delp. While pursuing his Ph.D. at Purdue, he primarily worked on projects sponsored by the Department of Homeland Security and Lockheed Martin Space. During his studies, Mr. Hao also worked at Apple Inc. as a research intern during the summer of 2020.

His research interests are image processing, computer vision, and machine learning. He is a student member of the IEEE, the IEEE Computer Society, and the IEEE Signal Processing Society. He has served as the reviewer of the IEEE Transactions on Information Forensics and Security, the IEEE CVPR Workshop on Media Forensics, the IEEE CVPR Workshop on AI for Content Creation, and the IEEE Winter Conference on Applications of Computer Vision.