Multimodal Data Enhancement and Augmentation Methodologies for Developing Robust Artificial Intelligence Systems

A dissertation submitted by

Victor Oludare

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Tufts University

May 2022

© 2022, Victor Oludare, all rights reserved

Adviser:

Prof. Karen Panetta

ABSTRACT

One of the many advantages of artificial intelligence (AI) is the rapid technological advancements that have considerably enabled researchers to solve complex problems by discovering relationships between features and patterns not easily observable in data. Despite dramatic advancements in AI systems, the application of these technologies in tackling **humanitarian-related problems** is limited. Video and image analytics could, in theory, make significant contributions to a wide range of humanitarian challenges, including human trafficking, environmental monitoring, and animal conservation. The limiting factors include both the availability of such technologies and also the availability of data upon which the technologies might operate. And, even when the data are available, they are often challenging due to inherent data skewness that manifests when environmental and weather distortions impact the data – such as low-light, fog, or hazy conditions. The unavailability of robust curated and relevant datasets and the inadequacy of generalized algorithms solutions further limits AI applications to humanitarian-related challenges.

This dissertation seeks to develop robust, efficient, and effective artificial intelligence-inspired solutions that encompass data enhancement and augmentation strategies to address this knowledge gap. To accomplish this, this dissertation investigates three major research topics:

a) Developing **multimodal enhancement algorithms,** inspired by the human visual system, which utilize deep neural networks to dynamically adapt to several operating environments, coupled with reduced inference and re-training time. These algorithms also overcome challenges when faced with different distortions caused by location (e.g., underwater), lighting (e.g., low-light), and hazy (e.g., fog/smoke) environments.

b) **Developing adaptive data augmentation methodologies,** to compensate for inadequate training data encountered in various tasks. To achieve this, novel pseudo-labeled data generation techniques are developed from large unannotated datasets with discriminative features for augmenting the original datasets. These strategies thus provide additional data for training and validating models, which then functions as a **regularization strategy** to reduce overfitting.

c) **Developing adaptive and automated approaches** which consider individual uniqueness in building AI systems. AI has **a tendency for algorithmic bias**, especially in the development of systems for tools intended to aid individuals. A resultant factor of variability minimization in the data captured across different individuals.

Quantitative and qualitative analyses are employed in the comparison of formulations of these research topics with the current state-of-the-art. The resulting algorithms, methods, and datasets are presented in the context of addressing several humanitarian challenges, including wildlife conservation, healthy aging, and disaster relief. Further contributions to the existing body of knowledge include the introduction of new AI tools and methods for non-engineering disciplines in the utilization of AI-based methods in the data collection, annotation, training, and analysis of relevant data to their research disciplines.

To my father, mother, and siblings

ACKNOWLEDGEMENT

First, I would like to thank God for helping me through this journey and for being with me at my lowest and best moments. I would also like to convey my heartfelt gratitude to Dr. Karen Panetta for being an amazing adviser and giving me the opportunity to pursue research interests that I was passionate about, and providing every resource I needed to make the Ph.D. journey smoother.

I am also grateful to my co-advisor, Dr. Sos Agaian, for his guidance, unique research ideas, encouragement, and enthusiasm to find resources when studying a challenging topic. Thank you for the time you made available, often on short notice, to Skype and brainstorm ideas.

I want to thank Dr. Allen Rutberg, Dr. Susan Roberts, Ellen McDonald, and Kali Pereira for their advice, research ideas, and for taking the time to collect datasets in the field, which helped immensely speed up my research.

Special thanks go to Dr. James Intriligator for his significant research collaboration on the FHWA grant and Prof. Ron Lasser for being a member of my committee, and his invaluable comments and suggestions for improving my dissertation.

I would like to thank my colleague and close friend Landry Kezebou for his encouragement and immense collaborations throughout the Ph.D. program. I would like to thank my best friend, Victoria Omojeso, for always being there and believing in me.

I would like to thank my lovely parents, siblings, and mentor for their love and support throughout this Ph.D. journey. I couldn't have done this without them.

Lastly, I would like to thank Obafemi Jinadu and all members of Dr. Karen Panetta's Vision and Sensing Systems Lab, who have all contributed in no small way to my thesis work.

TABLE OF CONTENTS

ABSTRACT ..... **i**

ACKNOWLEDGEMENT ..... **iv**

TABLE OF CONTENTS ..... **v**

LIST OF TABLES ..... **x**

LIST OF FIGURES ..... **xiii**

LIST OF PUBLICATIONS ..... **xxiii**

Chapter 1. Introduction 1

1.1 Overview 1

1.2 Objectives 5

1.3 Dissertation Organization 7

1.4 Dissertation Contributions 9

PART 1: MULTIMEDIA DATA ENHANCEMENT **13**

Chapter 2. Generative Approach to Image Enhancement for Improved Visibility in Lowlight and Hazy Conditions 14

2.1 Overview 15

2.2 Proposed AG-CRN model 18

2.2.1 AG-CRN Network Architecture 18

2.2.2 Objective Function 20

2.3 Experiments and Results 22

2.4 Summary 27

Chapter 3. Improving the Performance of Object Trackers in Underwater Environments 28

3.1 Overview 29

3.2 Classical Methods for Underwater Image Enhancement 31

3.2.1 Atmospheric scattering model 32

3.2.2 Simplified Model 32

3.3 Deep Learning-based Methods 34

3.4 Proposed CRN-UIE Model 35

3.4.1 CRN-UIE Network architecture 35

3.4.2 Objective Function for Optimizing CRN-UIE 37

3.5 Experiments and Results 39

3.6 Proposed UOT100 Benchmark Dataset and Evaluation Metrics 41

3.6.1 Review of Existing Object Tracking Datasets 41

3.6.2 Proposed Underwater Object Tracking Benchmark Dataset – UOT100 43

3.6.3 Evaluation Metrics 45

3.6.4 Performance Benchmarking on Un-enhanced UOT100 dataset 46

3.6.5 Performance Benchmarking on CRN-UIE-enhanced UOT100 dataset 51

3.7 Summary 55

PART 2: DATA AUGMENTATION METHODOLOGIES FOR BUILDING ROBUST AI SYSTEMS 56

Chapter 4. Non-invasive Computer Vision Techniques for Object Detection and Tracking from Multimodal Camera Sensors 57

4.1 Overview 57

4.2 Feature-based Elephant Detection 59

4.2.1 Histogram of Oriented Gradients 59

4.2.2 Local Binary Patterns 60

4.2.3 Scale-Invariant Feature Transform 62

4.2.4 Flowchart for Feature-based Detection 65

4.2.5 Evaluation Metrics 67

4.3 Shadow-based Elephant Detection 69

4.4 Deep Learning-Based Elephant Detection 72
4.4.1 Experiments and Results 74
4.5 Kalman Filter-based Tracking of Elephants in Videos 77
4.5.1 Kalman Filter Algorithm 79
4.6 Summary 82
Chapter 5. A Unified Approach to Pose Estimation in Mammals using Noisy Labels 83
5.1 Overview 84
5.2 Diversified Training Dataset Consolidation and Standardization 88
5.2.1 AP-10K Dataset 88
5.2.2 Amur Tiger Dataset 89
5.2.3 Macaque Dataset 89
5.2.4 Elephant Dataset 90
5.2.5 Dataset Standardization 90
5.3 Proposed Unified Animal Pose Estimation Framework 91
5.3.1 Proposed framework 92
5.3.2 Objective Function 96
5.3.3 Implementation Details 97
5.4 Evaluation Metrics 98
5.5 Experiments and Results 99
5.6 Summary 104
Chapter 6. Effectiveness of Two-Stream Networks for Multi-task Learning 105
6.1 Overview 106
6.2 Proposed TS-HRNet Model 109
6.2.1 TS-HRNet Model Architecture 109
6.2.2 Objective Function 112

6.3 Experiments and Results 114
6.4 Summary 118
Chapter 7. Importance of Data Augmentation for Multi-Task Learning 119
7.1 Overview 119
7.2 Data Augmentation 121
7.2.1 Spatial-Aware CutMix Data Augmentation Strategy 121
7.3 ATS-HRNet Model Architecture 123
7.3.1 Attention-Guided HRNet 123
7.3.2 Two-Stream Attention-Guided HRNet 124
7.3.3 Pseudo-Label Generation 125
7.3.4 Objective Function 126
7.4 Evaluation Metrics 127
7.5 Experiments and Results 129
7.6 Summary 133
PART 3: BUILDING EFFICIENT REAL-TIME HEALTH MONITORING TECHNOLOGIES **134**
Chapter 8. Human Body Weight Estimation using Smart Shoe Insoles: Data Preprocessing, Analysis and Predictive Modeling 135
8.1 Overview 136
8.2 Methodology 140
8.2.1 Data Acquisition 140
8.2.2 Data Collection Details 141
8.2.3 Exploratory Data Analysis 144
8.3 Predictive Modeling 147
8.3.1 Dynamic Windowing-based Feature Extraction 147
8.3.2 Machine Learning-Based Predictive Modeling 149

8.4 Experiments and Results 150

8.5 Summary 155

Chapter 9. Concluding Remarks and Future Work 156

9.1 Conclusion 156

9.2 Future Work 160

References 163

LIST OF TABLES

Table 2-1: Quantitative comparison of AG-CRN to other state-of-the-art low-light enhancement algorithms across MEF, LIME, NPE, VV, DICM, and Dark Face datasets. Evaluation is performed using NIQE blind image quality measure in which a lower score indicates a more natural image. Metrics reported in this table represent the average score across all images in the corresponding dataset. The best scores are highlighted in green. 24

Table 2-2: Performance improvement of AG-CRN on face detection models in dark images. Results presented in this table are average scores on the 6000 images in the DARK FACE dataset [35]. The values give the model's average precision (AP) before and after enhancement. 26

Table 3-1: Quantitative comparison of the performance of CRN-UIE to other state-of-the-art GAN models using individual components UIQM including color (UICM), sharpness (UISM), and contrast (UIConM) and CCF no-reference underwater image quality measures. Green indicates the best score, and bolded values indicate the score for CRN-UIE. 40

Table 3-2: Comparative summary between existing object tracking datasets and the proposed UOT100 dataset. While most existing datasets focus on open-air data, the UOT100 and UOT32 datasets only contain underwater data with the ultimate goal of improving tracking accuracy and efficiency of state-of-the-art object tracking algorithms in underwater environments..... 42

Table 3-3: Area Under Curve (AUC) results for all trackers across several variations in visual quality of the underwater image. Results presented in this table represent the average scores across the UOT100 dataset. 51

Table 3-4: Comparative tracking benchmark results on distorted and enhanced underwater data. The Area Under Curve (AUC) presented in this table represents the average scores across the UOT100 dataset. 52

Table 4-1: Classification results for elephant vs non-elephant image patches. The results presented in this table are average scores across the test set. HOG gives the highest precision and accuracy scores while SIFT-based feature gives the best recall..... 68

Table 4-2: Average inference time for YOLOv4 and Faster RCNN running on an Nvidia GTX 1060 GPU..... 77

Table 5-1: Brief review on current state-of-the-art animal pose dataset. 85

Table 5-2: Results of several models on AP-10K validation set containing 1331 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across all threshold levels. 99

Table 5-3: Results of several models on Animal-Pose validation set containing 1020 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across most threshold levels. 99

Table 5-4: Results of several models on the Tiger validation set containing 515 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across most threshold levels. 100

Table 6-1: Joint Damage Scale (JDS) describes how the damage levels were determined. .. 108

Table 6-2: The weight values assigned to each class per task. This is empirically determined based on the dominance of each class in the training dataset. 112

Table 6-3: Size of xBD dataset..... 114

Table 6-4: Quantitative comparison of F1 scores with other methods. Results show that TS-HRNet + SSL outperforms the baseline method. 116

Table 7-1: Quantitative comparison of F1 scores with other methods. Results show that ATS-HRNet + SA-CutMix outperforms the baseline method. 130

Table 8-1: Evaluation of three machine learning models on all four test data samples. The fourth model is an ensemble of ExtraTreeRegressor, LightGBM, and CatBoost models.

The Ensemble-based approach significantly outperforms other models in three of the four experiments. 153

Table 8-2: Average Mean Absolute Error for data samples collected in the first and second experiment. The MAE score is lower for females because of the lower weight distribution while it is larger in males because of the wider weight distribution. Results are obtained using the Ensemble model. 155

LIST OF FIGURES

Figure 1-1: Overview of the dissertation contributions. (a) depicts the datasets created as part of this dissertation, (b) outlines the various algorithms and techniques implemented during the course of this dissertation, and (c) highlights the various areas of application explored in this dissertation. 3

Figure 1-2: Dissertation organization. This dissertation is divided into three main parts. Part 1 discusses the multimodal data enhancement methods, Part 2 presents data augmentation strategies developed and its direct areas of application, and Part 3 explores Weight monitoring technology using smart shoe insole sensors. 7

Figure 2-1: Visual illustration of the effect of histogram equalization on an image with poor contrast. The original image is on the left and the histogram equalized image is on the right. Corresponding histograms are shown below each image. It can be seen that histogram equalization spreads out the bins across the entire spectrum resulting in much better visual contrast. 16

Figure 3-2: AG-CRN's generator architecture. Conv denotes a 2D convolution, IN denotes Instance normalization and ReLU denotes a Rectified Linear Unit activation function. The Attention block is same as the Upsampling block with the tanh activation replaced with a sigmoid. The fusion operation sums the outputs from the multiplication operation to create the enhanced image..... 19

Figure 2-3: Network architecture of the PatchGAN discriminator. The PatchGAN discriminator is a rather shallow network that serves as an image classifier to distinguish original images from the generator's synthesized output image. The real and generated image pair are concatenated and fed to the network. The numbers under each convolutional block represent the number of filters for the corresponding layer. 20

Figure 2-4: Qualitative comparison of AG-CRN to other state-of-the-art low-light enhancement methods. The results show that AG-CRN is consistently able to bridge the gap between enhancement and preserving the 'naturalness' of the images across all the

benchmark datasets and with varying illumination levels and with multiple light sources in the background. 23

Figure 2-5: Performance of face detection and landmark localization in the dark. The red bounding boxes indicate regions where faces are detected, and the dots are the facial landmarks. AG-CRN significantly improves the performance of MobileNet and Resnet-based models with more detected faces. 25

Figure 2-6: Zoomed in visualization of the missing faces in the original low-light image compared to the face detection in the AG-CRN enhancement output. Images in the left column are original low-light input images, while images on the right are the output of the proposed AG-CRN model. It can be seen that enhancing the input low-light image using AG-CRN can significantly improve the face detection rate and accuracy of face recognition. 26

Figure 3-1: Sample images showing common distortions observed in underwater environments. Each column shows sample images of the corresponding distortions, including Blueish, Light green, dark green, and yellowish distortions. Varying levels of haze, blur, and fuzziness are also apparent in many of the images. 30

Figure 3-2: Network architecture of CRN-UIE generator. Conv denotes a 2D convolution layer, ReLU is the rectified linear unit activation function, InstanceNorm is the instance normalization layer. The blocks are color-coded, and the numbers below each combo block represent the number of filters, kernel size, and stride size, respectively. 35

Figure 3-3: Visual comparison of the sample results from the proposed CRN-UIE with other state-of-the-art generative models. CRN-UIE produces more realistic enhanced images from the original distorted underwater input images. It also yields better visual quality while preserving the color of objects underwater. 39

Figure 3-4: Sample tracking data from the UOT100 dataset showing various types of distortions. The red bounding box denotes the object of interest and the text below each column indicates the category of visual data. 43

Figure 3-5: An overview of the number of annotated frames and distortion type distribution across the dataset. (a) shows the number of annotated frames, and (b) is a pie chart distribution of various distortion types across the UOT100 dataset. 44

Figure 3-6: Precision and Success Rate plots on the UOT100 dataset. In the legend, the values in brackets after each tracker represent the area under curve (AUC) of the corresponding plot. Trackers in the legend are ranked from top to least performing using their AUC values. 48

Figure 3-7: OPE precision and success rate plots on Natural (first row) vs Artificial (second row) underwater visual data. Trackers in the legend are ranked from top to least performing using their AUC values and are each color-coded. 49

Figure 3-8: OPE precision and success rate benchmark plots on Clear (first row) vs Distorted (second row) underwater visual data. Using their AUC values, trackers are color-coded and ranked from top to least. 50

Figure 3-9: Comparative benchmark results on Original (first row) vs Enhanced dataset (second row). Trackers in legend are ranked from top to least performing using their AUC values. 52

Figure 3-10: Comparing the precision benchmark plots on original (first row) vs enhanced (second row) data (natural vs artificial underwater data). Trackers in legend are ranked from top to least performing using their AUC values. 53

Figure 3-11: Comparative benchmark results – Precision and Success rate plots on data enhanced using the proposed CRN-UIE (first row), FUnIE-GAN (second row), and UGAN (third row). Each row shows the corresponding enhancement method's precision and success rate plots. From the plots, it can be seen that most trackers perform better on underwater data enhanced using CRN-UIE. However, SiamRPN++ tends to do better on data enhanced with FUnIE-GAN and UGAN. 54

Figure 5-1: Sample visualizations of Histogram of Oriented Gradients (HOG) feature descriptor for elephants 60

Figure 4-2: Illustration of LBP with different neighborhoods settings. The circular (P=9, R=1), (P=16, R=2) and (P=8, R=2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel. 61

Figure 4-3: Visualization of LBP image for elephants. The first row shows an input image and its corresponding LBP image at a threshold of 0 and 3. The second row shows the histograms at a threshold of 0 and 3 respectively. 62

Figure 4-4: Illustration of a Gaussian pyramid [119] 63

Figure 4-5: Visualization of orientation assignment with the corresponding histogram of gradients [119] 64

Figure 4-6: Illustrative example of keypoint descriptor [119] 64

Figure 4-7: Illustrative chart and steps for training a Support Vector Machine Classifier using HOG, LBP, and SIFT-based features. Bag of Words is created by applying K-means clustering to the extracted features. This helps in grouping similar features together. 66

Figure 4-8: Sample training data used to train the binary classifier. The training data contains elephant images and random images which represent negative samples. Each image is resized to a size of 256x256 with elephants centered in the image. 67

Figure 4-9: Elephant detection using a combination of the sliding window approach and binary classification using HOG features 69

Figure 4-10: Flowchart for the shadow detection algorithm. meanA, meanB, and meanL are the mean of the pixels in the A, B, and L channels of the LAB color space. stdL is the standard deviation of pixels in the L channel and pixel value is a user defined value that helps filter out noisy detections. 70

Figure 4-11: Results of the shadow detection algorithm from drone footage of elephants. Images in the left column are original input images from drone footage, while images in the right column correspond to show detection from the image. Shadow detection is important because elephant shadows are more easily detectable from aerial views than the elephants themselves. 71

Figure 4-12: (a) The RPN module proposed by Ren et al. It is responsible for generating region proposals by serving as an attention module that guides the base model (Fast RCNN) on where to look for objects in an image [129]. (b) the detailed diagram of the Faster RCNN architecture [130]. 73

Figure 4-13: Simplified model architecture for YOLO single-stage object detector [130] 74

Figure 4-14: Performance benchmarking for YOLOv4 and Faster RCNN object detectors under varying environmental and weather conditions. This evaluation also provides insight on the effectiveness of single stage (YOLO) and two stage (faster RCNN) detectors. The percentages represent the accuracy of the detector, and it is a ratio of the number of localized elephants to the actual number of elephants in the scene. 76

Figure 4-15: Model underlying the Kalman filter. Squares represent matrices and ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors [142]. 79

Figure 5-16: Illustrative flowchart of the Kalman Filter-based elephant tracker implementation. 79

Figure 4-17: Elephant tracking results of the Kalman-filter-based tracker. The frames shown in this figure are extracted from a video sequence at multiple time stamps. The red bounding boxes are the ground-truth bounding boxes, while the light-blue boxes represent the predicted bounding boxes for the respective objects of interest. 81

Figure 5-1: Sample images and annotations from 5 animal pose datasets (a) Amur Tiger pose dataset (b) Horse pose dataset (c) Macaque pose dataset (d) Animal-Pose dataset (e) AP-10K dataset, and (f) our Elephant pose and landmarks dataset. 85

Figure 5-2: Visualization of key-points annotation from the AP-10k dataset [151]. The table on the left describes the numbered key-points which are apparent on the animal picture on the right. 88

Figure 5-3: Visualization of key-points annotation from Amur Tiger dataset [146]. The table on the left describes the numbered key-points which are apparent on the animal picture on the right. 89

Figure 5-4: Visualization of key-points annotation from Macaque dataset [148]. The table on the left describes the numbered key-points which are apparent on the animal picture on the right. 89

Figure 5-5: Visual overlay of key-points & other landmarks. The key-points on the animal on the right-hand side are the white dots, while the orange dots represent the landmarks. 90

Figure 5-6: Proposed training framework which involves training a teacher network $f\theta$ using ground-truth annotated datasets in Stage 1, generating high-confidence pseudo-labels in Stage 2 by using the learned weights from $f\theta$ and using a combination of the ground-truth labeled data and the pseudo-labeled data to then train a student network $g\emptyset$ in Stage 3. 92

Figure 5-7: Samples of pseudo-labels generated using $f\theta$ (a) images classified as 'easy' examples, and (b) images classified as 'hard' examples. The animal pose in the easy examples is often well-defined with most joints visible while hard examples have several joints that are either occluded or hard to predict. 94

Figure 5-8: Overview of the proposed approach for harnessing noisy pseudo-labels for animal pose estimation. The feature extractor is initialized using ImageNet pretrained

weights. L_{sup} and L_{pseudo} are the supervised and semi-supervised loss respectively. L_{joint} is a weighted combination of both. 95

Figure 5-9: Qualitative results from AP-10K validation set using HRNet-w48 + SSL model 101

Figure 5-10: Qualitative results from Animal Pose validation set using HRNet-w48 + SSL model 101

Figure 5-11: Qualitative results from Tiger Pose validation set using HRNet-w48 + SSL model 102

Figure 5-12: Qualitative results from Tiger Pose validation set using HRNet-w48 + SSL model 102

Figure 5-13: Sample failure cases of the proposed model. As can be seen in this image the failure occur as a result of occlusion, poor image quality, multiple animals present in the image, low-light condition, complex animal poses, etc. 103

Figure 6-1: Sample image pairs from the xBD dataset (a) and (b) are pre- and post-disaster images from tornadoes, (c) and (d) are pre- and post-disaster images from hurricane Harvey. 107

Figure 6-2: Network architecture for high resolution networks (HRNet) for image segmentation [166]. 110

Figure 6-3: Proposed TS-HRNet framework. A pair of pre- and post-disaster images are fed to each stream, and network localizes the buildings and classifies into damage types. The pseudo-label generation pipeline takes a pair of pre- and post-disaster images and generates pseudo-labels for each pair. 111

Figure 6-4: Localization and classification results for five disaster types. The buildings are colored based on the damage type and color mapping in Figure 6-5. The results show that the semi-supervised learning framework improves the classification performance of TS-HRNet as well as reduces class overlap. 117

Figure 7-1: Shows the chart of (a) the number of buildings for images (b) building damage class distribution in train and tier3 folders. 121

Figure 7-2: Proposed data augmentation pipeline using the spatial-aware CutMix technique 122

Figure 7-3: Proposed Attention-Guided High-Resolution network architecture. 123

Figure 7-4: Attention-guided Two-Stream HRNet for joint building localization and classification. This architecture design is motivated by the work of Koo et al. [200] (b) Criss-Cross Attention Module (CCAM), which helps in extracting contextual information of all pixels on its criss-cross path [199]. The multi-task learning framework helps in improving the data efficiency of the system by encouraging shared representations between building localization and damage classification. 124

Figure 7-5: Pseudo-label generation pipeline takes a pair of pre- and post-disaster images and generates pseudo-labels of each pair [174]. 126

Figure 7-6: Visualization of the image pairing using the proposed SA-CutMix technique. Images in this figure are illustrative examples of pre- and post-image pairing. (a) and (b) are two separate examples. The first two columns for each example represent the pre-and post-disaster images and the corresponding labels, while the third and fourth columns illustrate the post-disaster images and associated labels. 129

Figure 7-7: Localization and classification results for five disaster types. The buildings are colored based on the damage type and color mapping in figure 1. The results show that the ATS-HRNet still outperforms TS-HRNet trained with pseudo-labels presented in Chapter 6 especially when predicting less dominant classes. 131

Figure 7-8: Localization and classification results presenting the benefits of SA-CutMix as a data augmentation strategy. These results show that SA-CutMix helps reduce the amount of misclassified building damages and force finer label assignment. 132

Figure 8-1: Statistical data showing the progression of obesity rates among the US population. (a) and (b) show a trend in the obesity rate among adults and adolescents in the US, while (c) compares the severity level of obesity among adolescents [206]..... 137

Figure 8-2: Pressure sensor distribution on the Moticon shoe insole [211]..... 141

Figure 8-3: Data distribution for participants in experiment 1 (top row), and Data distribution for participants in experiment 2 (bottom row). 23 females and 30 males participated in experiment 1, while 5 females and 4 males participated in experiment 2..... 142

Figure 8-4: Shows images of (a) the Moticon show insoles, (b) the Withings scale and dumbbell weights for varying differences in the load carried by the participant, and (c) Mobile software interface for retrieving and visualizing pressure sensor data via Bluetooth communication..... 143

Figure 8-5: Plot of total left and right forces vs. ground truth weights measured using a Withings scale. The three rows correspond to data collected for three female candidates. The desired trend is a positively correlated graph between the x and y-axis; hence, for candidate 1, iteration 1 is discarded; for candidate 2, iteration 2 is discarded; and for candidate 3, iteration 3 is discarded. Participants holding different dumbbell weight sizes provided a means for varying the weight of individuals..... 145

Figure 8-6: Plot of total left and right forces vs. ground truth weights measured using a Withings scale. The three rows correspond to data collected for three male candidates. The desired trend is a positively correlated graph between the x and y-axis; hence, for candidate 1, iteration 1 is discarded; for candidate 2, iteration 1 is discarded; and for candidate 3, none is discarded..... 146

Figure 8-7: Process of sliding window at various time intervals and with some overlap..... 147

Figure 8-8: Training pipeline using Pycaret machine learning library. The preprocessed data extracted using our dynamic windowing approach serves as input to the model. Finetuning helps obtain the best hyperparameters for the individual, and finally, the model is tested on unseen data collected for a participant..... 149

Figure 8-9: Exploratory data analysis for a participant with 8 independent experiments conducted over several days. The plots show a graph of Force (N) vs. groundtruth weight (lbs). A positive correlation between the forces and measured weight is desired and gives a simple indication of the usability of a data sample for training the model. 151

Figure 8-10: Plot of the mean absolution error (MAE) vs. number of training instances (left) and the MAE vs. tree depth (right). This shows that increasing the number of training instances improves the model's predictive ability. A maximum tree depth of 8 is sufficient for this candidate to obtain the best results. 152

Figure 8-11: Plot of feature importance for participant 6. This ranks the features that contribute most to the performance of the model. The top 10 ranked features are represented in this figure and vary for each candidate. 152

Figure 8-12: Scatter plots of the predicted weights vs. ground truth weights. The minimum and maximum difference between the estimated and groundtruth weight is included because the regression line is not always accurate, as depicted in the plot for Experiment 5. 154

Figure 9-1: Overview of the dissertation contributions. (a) depicts the datasets created as part of this dissertation, (b) outlines the various algorithms and techniques implemented during the course of this dissertation, and (c) highlights the various areas of applications explored in this dissertation. 157

LIST OF PUBLICATIONS

Journal Papers

[1] K. Panetta, V. Oludare, L. Kezebou, O. Jinadu, and S. Agaian, "AnimPose: A Unified Approach for Pose Estimation in Mammals using Noisy Labels,".

[2] K. Panetta, L. Kezebou, V. Oludare, and S. Agaian, "Comprehensive Underwater Object Tracking Benchmark Dataset and Underwater Image Enhancement With GAN," in *IEEE Journal of Oceanic Engineering*, doi: 10.1109/JOE.2021.3086907.

[3] K. Panetta, L. Kezebou, V. Oludare, S. Again, and Z. Xia, "TMO-Net: A Parameter-Free Tone Mapping Operator Using Generative Adversarial Network, and Performance Benchmarking on Large Scale HDR Dataset," in IEEE Access**,** vol. 9, pp. 39500-39517, 2021, doi: 10.1109/ACCESS.2021.3064295.

[4] K. Panetta, L. Kezebou, V. Oludare, J. Intriligator, S. Agaian, "Artificial Intelligence for Text-based Vehicle Search, Recognition and Continuous Localization", in MPI-AI, 2021, AI volume 2, issue no. 4: 684704. https://doi.org/10.3390/ai2040041

Conference Papers

[5] Victor Oludare, Landry Kezebou, Karen Panetta, and Sos Agaian "Attention-guided cascaded networks for improved face detection and landmark localization under low-light conditions", Proc. SPIE 11399, Mobile Multimedia/Image Processing, Security, and Applications 2020, 113990J (21 April 2020); https://doi.org/10.1117/12.2558397

[6] Victor Oludare, Landry Kezebou, Karen Panetta, and Sos Agaian "Semi-supervised learning for improved post-disaster damage assessment from satellite imagery", Proc. SPIE 11734, Multimodal Image Exploitation and Learning 2021, 117340O (12 April 2021); https://doi.org/10.1117/12.2586232

[7] Victor Oludare, Landry Kezebou, Karen Panetta, and Sos Agaian "U-Tracker: Visual Tracking of Underwater Objects", Proc. SPIE, Multimodal Image Exploitation and Learning 2022, **(**Accepted**).**

[8] Landry Kezebou, Victor Oludare, Karen Panetta, and Sos Agaian "TR-GAN: thermal to RGB face synthesis with generative adversarial network for cross-modal face recognition", Proc. SPIE 11399, Mobile Multimedia/Image Processing, Security, and Applications 2020, 113990P (21 April 2020); https://doi.org/10.1117/12.2558166

[9] L. Kezebou, V. Oludare, K. Panetta and S. S. Agaian, "Underwater Object Tracking Benchmark and Dataset," *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2019, pp. 1-6, doi: 10.1109/HST47167.2019.9032954.

[10] Landry Kezebou, Victor Oludare, Karen Panetta, and Sos Agaian "A deep neural network approach for detecting wrong-way driving incidents on highway roads", Proc. SPIE 11734, Multimodal Image Exploitation and Learning 2021, 117340P (12 April 2021); https://doi.org/10.1117/12.2586039

[11] Landry Kezebou, Victor Oludare, Karen Panetta, and Sos Agaian "Joint image enhancement and localization framework for vehicle model recognition in the presence of non-uniform lighting conditions", Proc. SPIE 11734, Multimodal Image Exploitation and Learning 2021, 117340Q (12 April 2021); https://doi.org/10.1117/12.2586036

[12] L. Kezebou, V. Oludare, K. Panetta and S. S. Agaian, "Few-Shots Learning for Vehicle Recognition," *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2021.

[13] Landry Kezebou, Victor Oludare, Karen Panetta, and Sos Agaian "Highway incidents detection and classification from live traffic surveillance cameras : A comprehensive dataset ", Proc. SPIE 11734, Multimodal Image Exploitation and Learning 2022, April 2022.

Multimodal Data Enhancement and Augmentation Methodologies for Developing Robust Artificial Intelligence Systems

Chapter 1. Introduction

1.1 Overview

Artificial Intelligence (AI) is revolutionizing almost every industry; however, the cutting-edge technological advancements in AI, particularly computer vision applications, are yet to have a considerable impact on modern-day humanitarian-related problems. Research in wildlife conservation, man-made and natural disaster relief, and human health monitoring, is still limited, even though these hand-picked humanitarian-related applications present a huge potential for improving the lives of both humans and animals. AI may be able to help solve problems that arise when animals migrate into human settlements due to food shortage, deforestation, or the destruction of their natural habitats, thus causing human-wildlife conflicts, especially in Asia and Africa, as these animals graze on farmland crops. This often escalates into accidents where humans get violently attacked by the animals, causing communities to take drastic measures against these animals [1], [2]. The constant shrinkage of animal habitats and increased human-wildlife conflicts threaten the survival of these animals, with some on the brink of extinction. Clearly, there is a pressing need to develop AI technologies for non-invasive monitoring and population estimation.

AI tools can be developed to aid in solving man-made and natural disaster problems. According to the World Health Organization (WHO), about 160 million people are affected by natural disasters, with 90,000 killed annually [3]. This motivates the need to develop a means for rapid, efficient, and effective damage assessment made possible through very high-resolution (VHR) satellite imagery. VHR remote sensing data provides cost-effective and efficient ways of visualizing affected areas' structural and spatial characteristics while also providing sufficient information needed to run computer vision algorithms to extract features and automate post-

disaster damage assessment processes. These technologies can be harnessed to detect damaged buildings and to analyze the extent of damage during natural disasters like hurricanes, floods, tornadoes, and wildfires.

AI is also useful in monitoring human health as statistics have shown the need to develop efficient techniques for monitoring an individual's body mass index (BMI), particularly in older adults at higher risk of diseases due to obesity [4]. Recent research studies suggest that obesity is one of the major health issues affecting populations in the U.S and worldwide, often resulting in severe complications. Data recently published by the Center for Disease Control and Prevention (CDC) estimated that obesity, which is linked to chronic conditions such as type 2 diabetes, heart disease, and cancer [4], affects 42.4% of U.S adults [4]. Furthermore, changes in the body associated with aging, such as muscle and bone loss that limit mobility, make older adults more vulnerable to the risks associated with obesity [5]. More recently, obesity was shown to be a risk factor for severe COVID-19 infection. Data collected from 350,000 patients indicate that people with obesity were 113% more likely to be admitted to the hospital, 74% more likely to be admitted to an ICU, and 48% more likely to die after contracting SARS-CoV-2 [6].

The paucity of literature in the application of artificial intelligence to these prevailing issues motivates this dissertation with a specific focus on developing AI-driven engineering solutions that could help address the aforementioned humanitarian-related problems. The work presented in this thesis draws from understanding of how the Human Visual System works [7], [8] to design machine learning and deep learning architectures in the provision of practical solutions to real-world challenges. The algorithmic solutions proposed in this dissertation focus on i) **multimedia data enhancement,** ii) **multimedia data augmentation**, and iii) **1-D data analytics**. These key

aspects, and the corresponding breakdown of the algorithmic solutions, are detailed in the flow diagram shown in Figure 1-1.

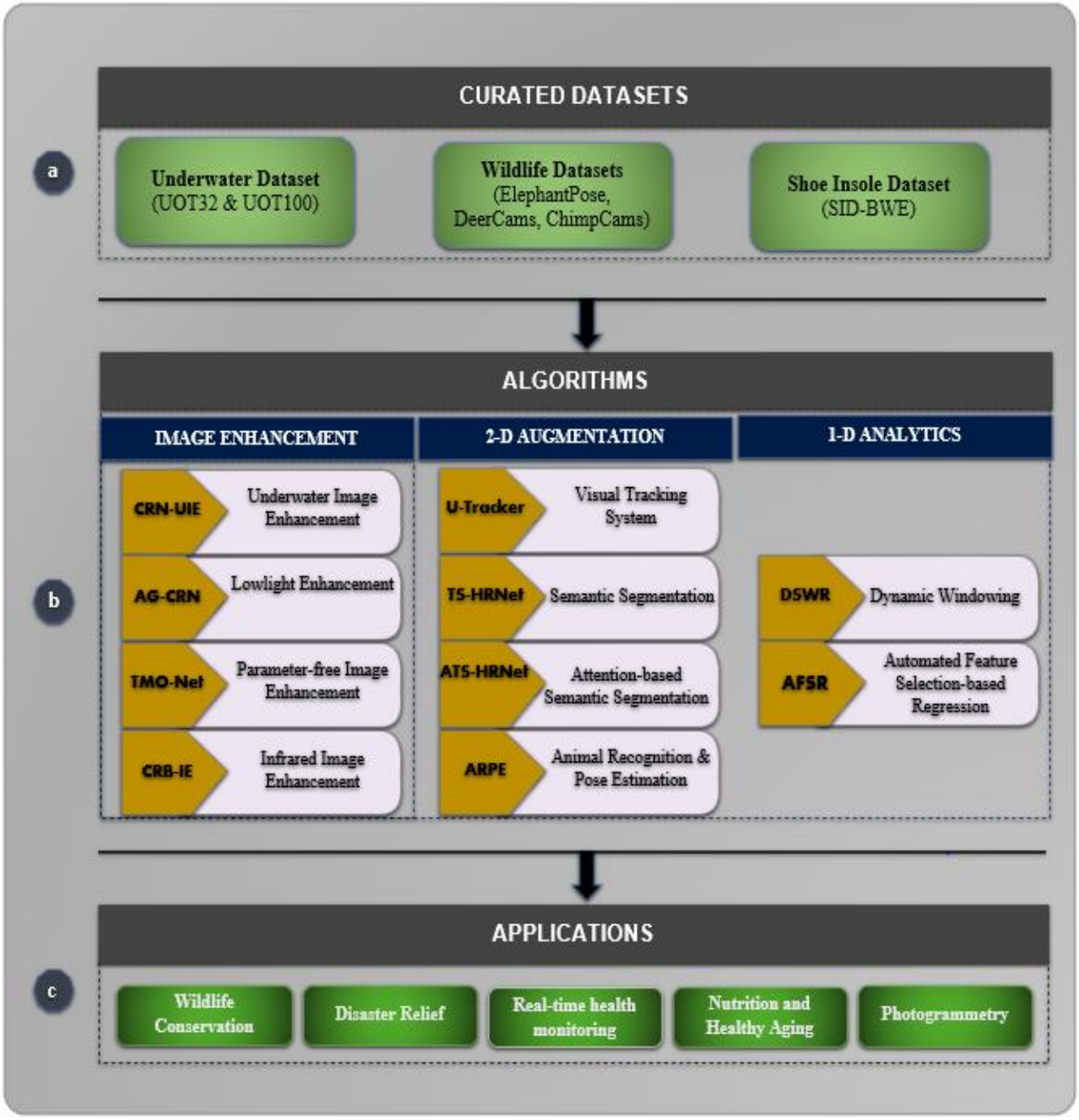


Figure 1-1: Overview of the dissertation contributions. (a) depicts the datasets created as part of this dissertation, (b) outlines the various algorithms and techniques implemented during the course of this dissertation, and (c) highlights the various areas of application explored in this dissertation.

a) **Dataset Curation, Annotation, and Augmentation**: The availability of accurately annotated datasets is crucial for building AI algorithms. Unfortunately, there is a lack of datasets with carefully annotated ground truth for underwater target tracking, wildlife conservation, and healthy aging. Thus, as part of this work, a set of novel **manually annotated datasets** was curated with corresponding annotations to aid in developing AI algorithms, supporting proposed AI models and framework solutions. AI is often data-hungry, and manually annotating large amounts of images and video datasets is tedious. Therefore, it is essential to develop new techniques for augmenting and training pipelines in harnessing large amounts of unlabeled data. These techniques would benefit the AI community in solving humanitarian challenges that were primarily neglected due to small datasets and the unavailability of needed datasets. Novel techniques for maximizing the learnable information from unannotated datasets to improve the performance of AI algorithms are presented herewith. To utilize this large amount of unlabeled data, we have developed **high-quality pseudo-labels** that can be used to improve the performance of AI algorithms. As part of this dissertation, these innovative strategies are applied to creating a dedicated underwater object tracker that outperforms other state-of-the-art open-air trackers in underwater environments and in satellite imagery to assess the extent of damage to buildings after a disaster.

b) **Multimedia Data Enhancement:** Novel multimedia enhancement algorithms are developed to improve the visual quality of the curated image and video datasets under diverse lighting conditions and in different image domains. This is achieved by developing a set of innovative convolutional neural network (CNN) based algorithms for feature extraction and adopting a generative approach for translating across various visual domains

such as low light, underexposed images, underwater images, and infrared images. This step is important because the performance of AI algorithms is largely dependent on the quality of their training data. A noisy or low-quality dataset will negatively affect even the most sophisticated algorithms; hence, improving the quality of datasets used for training these AI systems is vital for improving overall system performance.

c) **1-D Data Analytics**: This explores the role of wearable devices for real-time body weight estimation. This work is motivated by the severity of obesity to human health and the risks associated with obesity for older adults. This is achieved by i) developing a novel **feature extraction technique** for streaming pressure sensor data and ii) developing an end-to-end trainable model for personalized bodyweight predictions.

1.2 Objectives

The primary goal of this dissertation is to propose several data-centric approaches for building robust AI systems which are not severely affected by changes in environmental and weather conditions. To achieve this goal, novel algorithms for multimedia data enhancement and task-dependent data augmentation methodologies are developed. These algorithms enable AI systems to work efficiently in ideal scenarios and also in non-ideal conditions. Due to the vastness and capabilities of AI systems for providing solutions to complex problems, the algorithms and objectives of this dissertations center around providing solutions to three major humanitarian-related challenges, i) wildlife conservation, ii) disaster relief, and iii) health monitoring. Specifically, this dissertation aims at:

(1) Understanding the role of multimedia data quality on the efficacy of modern AI systems. Oftentimes, not enough time is taken to understand the distribution and quality of a dataset

used for training AI models. Hence, existing complex systems/models tend to underperform or become biased when tested on out-of-domain examples or test sets captured in a non-ideal environment. These non-ideal environments could result from changes in weather conditions, cross-modal data, or color distortions.

(2) Developing novel multimedia image enhancement algorithms for improving the performance of modern AI systems under various environmental conditions such as low light, and in various image modalities such as infrared, and in various image domains such as visible light and underwater.

(3) Introducing a set of objective functions that improve the stability of training for enhancement models based on the image contents and guided by the working principles of the human visual system.

(4) Creating a number of large-scale annotated datasets to facilitate research in under-explored fields and build robust AI models for custom applications. By benchmarking the performance of state-of-the-art algorithms on these datasets, the capabilities are better evaluated, and previously unidentified issues are brought to light, thus serving as a guide for subsequent research in these areas. Additionally, these datasets will help future researchers develop, test, and refine their models.

(5) Developing easy-to-use and task-dependent data augmentation methodologies that can be adopted when addressing the problem of insufficient training data, which is common with building AI systems that are data-hungry.

(6) Introducing a dynamic windowing strategy for working with streaming data from wearable devices and sensors, which can provide a dataset required for training supervised machine learning algorithms to solve predictive problems like personalized body weight estimation.

1.3 Dissertation Organization

The overview of this dissertation and its key contributions are shown in Figure 1-1. This dissertation consists of three parts, as listed in Figure 1-2.

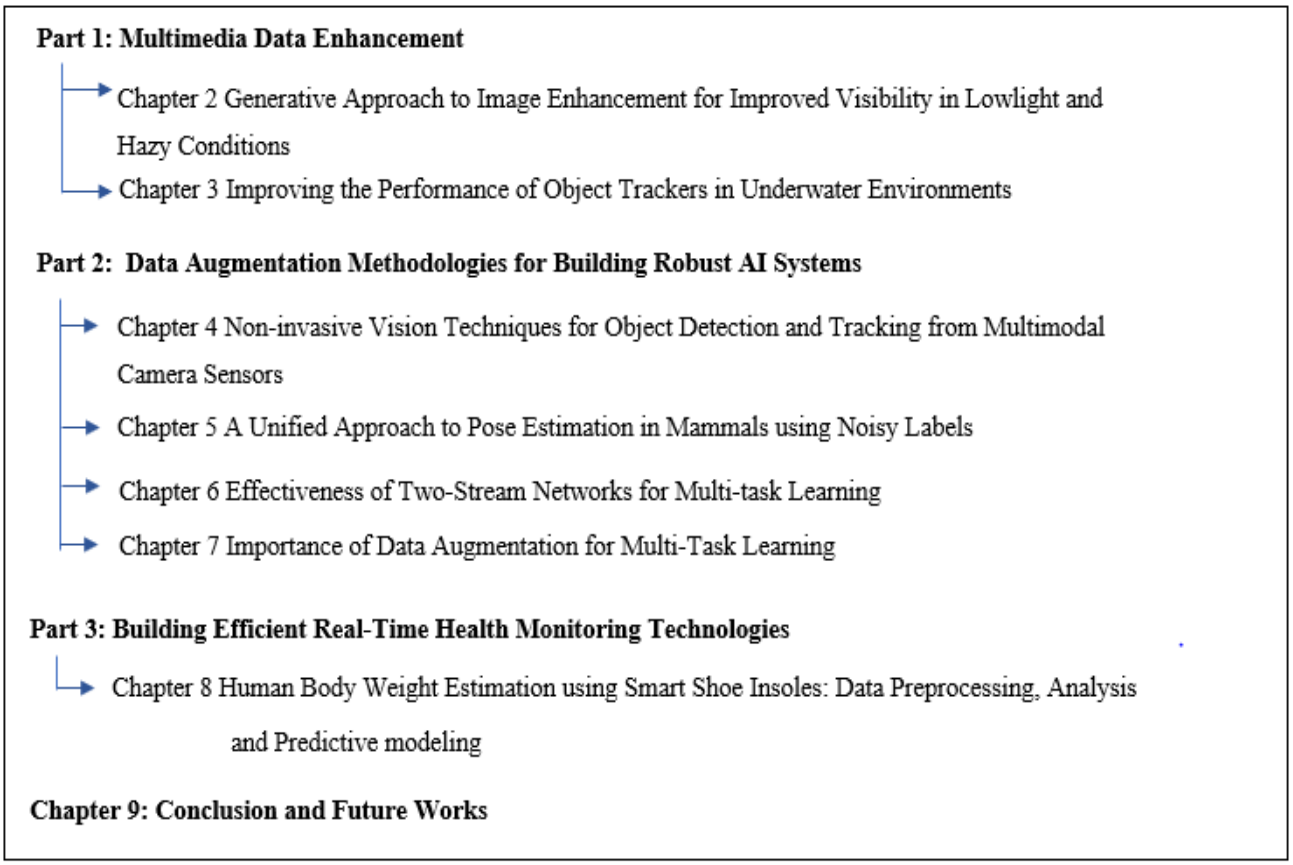


Figure 1-2: Dissertation organization. This dissertation is divided into three main parts. Part 1 discusses the multimodal data enhancement methods, Part 2 presents data augmentation strategies developed and its direct areas of application, and Part 3 explores Weight monitoring technology using smart shoe insole sensors.

Part 1 comprises chapters 2-3 and is titled “Multimedia Data Enhancement” presents generative approaches for automatically translating images from one domain to another. Chapter 2 introduces AG-CRN, a novel GAN-based algorithm for automatically translating images from a lowlight/under-exposed domain to a well-illuminated domain. Chapter 3 introduces CRN-UIE, a novel GAN-based method for automatically translating images from an underwater domain to an

open-air-like/enhanced domain. Performance evaluation of the proposed models is carried out using appropriate evaluation metrics commonly used for similar tasks. Furthermore, a large-scale dataset for underwater single object tracking is also introduced for benchmarking current state-of-the-art and future single object trackers.

Part 2 comprises chapters 4-7 and is titled "Data Augmentation Methodologies for Building Robust AI Systems" it presents several task-dependent data augmentation strategies for increasing the number of training examples for building robust models. Chapter 4 focuses the research and development efforts around humanitarian issues. Specifically, it introduces several methods for detecting elephants in the wild. A feature-based sliding window approach for elephant detection is presented and compared to a deep learning-based approach. A detailed benchmarking for deep learning-based elephant detection under various conditions and scenarios is presented to gain better insight into selecting object detectors. Chapter 5 introduces a unified approach for pose estimation in mammals using noisy pseudo-labels. Chapter 6 introduces TS-HRNet, a two-stream network for joint building localization and classification of building damages from satellite imagery. It also harnesses pseudo-labels generated from a large number of unlabeled datasets to improve the generalizability of the model. Chapter 7 introduces SA-CutMix, a spatial-aware modification of CutMix to solve the class imbalance issue evident in the training dataset. Also, an attention module is introduced to the TS-HRNet model to capture long-range feature interactions.

Part 3, comprised of chapter 8 and titled "Building Efficient Real-Time Health Monitoring Technologies," presents the components needed to perform real-time health monitoring for individuals. Chapter 8 introduces an end-to-end framework for personalized data collection, analysis, preprocessing, and model learning.

Finally, chapter 9 concludes the dissertation, identifies limitations of the introduced methodologies and provides suggestions for future research.

1.4 Dissertation Contributions

This dissertation seeks to contribute to the already existing body of knowledge in the area of multimedia data enhancement, image data augmentation, and 1-D predictive analytics. These contributions are detailed as follows:

- Develop a simple CNN-based architecture called cascaded residual blocks (CRBs) that outperforms the standard residual blocks for multimodal image-to-image translation. These CRBs are explored for various image enhancement problems, including:
    - AG-CRN, a generative model designed for translating lowlight images to well-illuminated images. It leverages the representative capacity of the cascaded residual blocks and the ability of attention modules to capture long-range interactions in the feature space to generate realistic, natural looking enhanced images given a lowlight pair while preventing artifacts that are often common in the GAN-based enhancement approach. Because of the difficulty associated with training GANs, a weighted combination of loss functions is designed for stabilizing the training process. These loss functions include the VGG loss, gradient profile loss, feature matching loss, and adversarial loss.
    - CRN-UIE, which is also a generative model, is designed for translating distorted underwater images to an enhanced image. It uses a similar architecture as AG-CRN with slight modifications to account for the type of distortions common in underwater environments. Unlike the AG-CRN framework in which a single

discriminator is used in the training process, CRN-UIE needs a multiscale discriminator to stabilize the training process and ensure that generated images do not contain distortions present in the original input underwater image.

- Develop a novel multi-task learning approach for joint building localization and damage classification from satellite imagery. The models developed include:
    - TS-HRNet is a two-stream high-resolution network for efficiently localizing buildings in a satellite image patch and classifying the extent of damage caused by natural or man-made disasters. These classes are ‘no damage’ and ‘minor damage ‘major damage,’ and ‘destroyed.’ The performance of this model is evaluated using standard metrics and compared to other two-stage approaches.
    - ATS-HRNet + SA-CutMix introduces a spatial-aware augmentation strategy suitable for satellite image processing. Also, an attention module is incorporated in the previously proposed TS-HRNet to boost the model’s feature representation capability, thus improving its performance.
- Creation of several annotated datasets to facilitate the advancement of research in under-explored fields and performance benchmarking for existing methodologies. These datasets include:
    - The UOT32 and UOT100 underwater object tracking dataset is the first single object tracking dataset for underwater environments. The database hopes to advance research in this area by developing dedicated tracking algorithms that overcome the challenges common with poor visibility underwater.

- LVZ-HDR dataset is a large-scale benchmark dataset developed to support the development of parameter-free tone mapping algorithms and performance benchmarking of existing tone mapping operators.
- ElephantCam, DeerCam, and ChimpCam are datasets collected from elephants, deer, and chimpanzees in their natural habitat. A first of its kind dedicated dataset for elephant pose estimation and landmark localization is created for the elephants. A dataset for detection, re-identification, and distance estimation is collected for the deer and chimpanzees. These datasets are created in order to address challenges faced by researchers when handling data collected from camera traps such as occluded animals, multi-domain animal identification, video summarization, etc.
- Insole Dataset, which is a dataset collected for estimating the bodyweight of individuals. It contains a pressure map of the individuals' feet, the force data applied on the insole, and the ground truth weight of the individual at any point in time. The pressure values associated with jumping, running, and standing activities are recorded at 50Hz and 100Hz sampling rates.

- Development of a unified framework for pose estimation in mammals. This is the first animal pose estimation approach that successfully consolidated multiple open-source datasets for individual animal species or a few species with a common anatomical structure, making the pose estimation task uniform across all mammals with similar skeletal forms. A novel framework for efficiently harnessing large amounts of unlabeled animal data is also introduced.
- An efficient framework for personalized data retrieval, analysis, pre-processing, model training, and predictions is developed for individual body weight estimation. By ensuring

the end-to-end system from data collection to prediction is personalized, data bias is avoided, ensuring a low error rate in estimated weight.

PART 1: MULTIMEDIA DATA ENHANCEMENT

Chapter 2. Generative Approach to Image Enhancement for Improved Visibility in Lowlight and Hazy Conditions

State-of-the-art techniques developed to solve various computer vision-related problems such as object detection and recognition, object tracking, and object segmentation exhibit suboptimal performance under poor illumination conditions. This chapter introduces a novel generative approach for translating an input image from a poorly illuminated domain to its approximated well-illuminated domain. The aim of this work is to improve the visual quality of image datasets and increase the visibility of objects of interest within an image.

The proposed Attention-Guided Cascaded Residual Network (AG-CRN) is able to efficiently synthesize enhanced and well-illuminated images given its corresponding low-light pairs with varying scene illumination without the need for parameter tuning or model re-training.

This chapter presents the architecture, training framework and optimization function of the proposed AG-CRN model and presents a comparative analysis with state-of-the-art methods for low-light image enhancement. Furthermore, this chapter presents the applicability of AG-CRN for improving imagery with other forms of distortion such as haziness. The benefits of applying AG-CRN as a preprocessing step for object detection tasks using relevant quantitative metrics is also presented.

The work presented here has been published in the proceedings of the 2020 SPIE conference for multimodal image exploitation and learning [9].

2.1 Overview

Images are often captured under proper illumination to capture detailed contents in an environment. However, constraints in traditional visible-light spectrum camera sensors introduce various artifacts in images captured at nighttime or under suboptimal conditions, such as the presence of multiple light sources in the background [10]. These suboptimal conditions can result in object shadows due to incorrect positioning of the object to a light source/backlight, or underexposure during image capture. Thus, this reduces visibility and can introduce high ISO noise and poor image contrast. The degradation in image quality not only makes it difficult for the human visual system to properly perceive and identify objects in images [11] but also leads to a significant decrease in the performance of state-of-the-art algorithms for various applications such as object classification and detection, face detection and recognition, scene understanding and autonomous driving.

Histogram equalization (HE) is one of the earlier methods for enhancing the contrast of an image. HE helps in distributing the pixel intensity over the entire dynamic range – hence adjusting the overall contrast within an image [12], [13]. Banik et al. [14] combined HE and illumination adjustment for contrast enhancement of low-light images by first performing HE on the V (brightness) channel from the HSV color space and then adjusting the pixel intensity in V. CLAHE [15] is based on an adaptive HE and converts the pixel intensity into a display range proportional to the intensity rank in the local intensity histogram. BPDHE [16] extends the conventional HE and tries to preserve the brightness of an input image by ensuring that the mean intensity of pixels in the output image is almost equal to the mean intensity of the input. Figure 3-1 shows a visual example of the effect of histogram equalization on the contrast of an image, including the artifacts introduced and the pixel distribution in the resulting image.

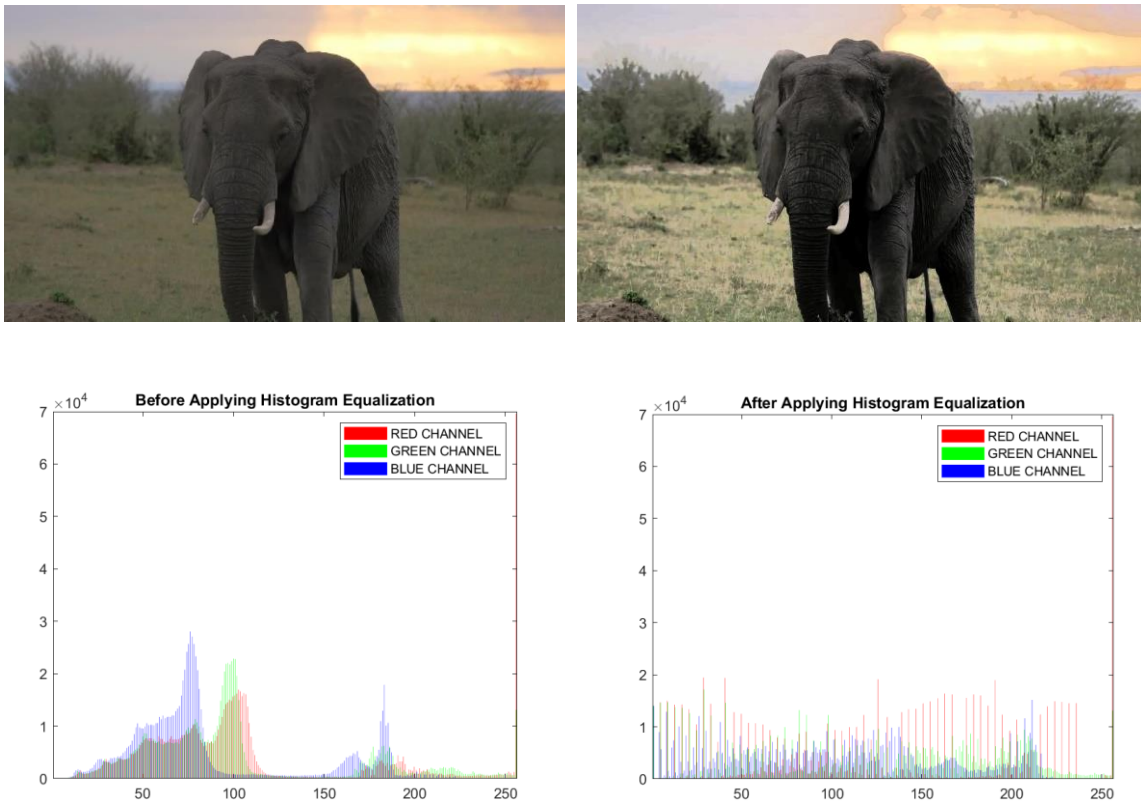


Figure 2-1: Visual illustration of the effect of histogram equalization on an image with poor contrast. The original image is on the left and the histogram equalized image is on the right. Corresponding histograms are shown below each image. It can be seen that histogram equalization spreads out the bins across the entire spectrum resulting in much better visual contrast.

Other conventional methods are based on the Retinex theory, which assumes that an image comprises illumination and reflectance. Lowlight enhancement is often formulated as an illumination estimation problem since the reflectance component can be assumed to be consistent under various lighting conditions. Single-Scale Retinex (SSR) approach [17] utilized a Gaussian filter to estimate the illumination map while the Multi-Scale Retinex (MSRCR) model [18] improves on SSR by adopting a multi-scale Gaussian filter and also introducing a method for color restoration. Jiang et al. [19] introduced an improved variation of the MSRCR to include auto-levels to help eliminate the effects of a small number of outliers on the image histogram and further enhance the image contrast. SRIE [20] proposed a weighted variational model to estimate both the

reflectance and illumination as well as suppress noise in an input image. LIME [21] estimates an illumination map and refines this map using a structure-aware smoothing method to improve the consistency of illumination. Although results from these approaches are quite promising, they are not robust to various lighting scenarios in real-world images.

These classical approaches appear to focus on certain assumptions about pixel statistics and distribution and visual mechanisms that do not often apply to all real-world scenarios. They also do not consider the effects of artifacts introduced in dark regions in an image and image noise, which is a result of insufficient light exposure during image capture. Finally, the performance of these algorithms is largely dependent on the careful tuning of hyperparameters for manipulation of the handcrafted illumination maps [22].

To address these shortcomings, several deep learning-based methods have been proposed. MSR-Net [23] and LightenNet [24] use the retinex theory to design CNN models to estimate the illumination maps and obtain an enhanced image. LLNet [25] proposed a stacked-sparse denoising autoencoder for joint low-light image enhancement and denoising without over-amplifying/saturating the lighter parts in the image. Cai et al. [26] first introduced a large-scale multi-exposure image dataset and then utilized this dataset for training a neural network to perform joint luminance and detail enhancement. MBLLEN [27] proposed a multi-branch network that applies enhancement via multiple subnets and produces an output image via multi-branch fusion. SID [28] converts raw short-exposure sensor images to RGB via a U-Net-based denoising approach followed by histogram stretching. Zero-DCE [29] proposed a method to estimate pixel-wise and high-order curves for dynamic range adjustment for an image. Generative approaches for translating images across multiple domains have also been explored for image enhancement. EnlightenGAN [30] proposed an unsupervised generative adversarial network that can be trained

without low-light/normal-light image pairs. Although these methods outperform classical methods on several benchmark datasets, they often fail to adapt to varying illumination levels and sometimes introduce artifacts in the enhanced image.

2.2 Proposed AG-CRN model

2.2.1 AG-CRN Network Architecture

The proposed AG-CRN model seeks to address the shortcomings identified in previous methodologies, discussed above, by harnessing the effectiveness of attention modules for capturing long-range feature interactions and improving representation capability in convolutional neural networks.

Figure 3-2 shows the model architecture of AG-CRN, its building blocks, and internal working are further discussed here. Given a low-light image X, AG-CRN learns a function that maps an image input X to a target domain Y which is a properly illuminated image. The network consists of a generator G and a discriminator D.

The generator network consists of a fully convolutional downsampling/encoding block, a set of cascaded residual blocks, and an upsampling/decoding block. An attention module is incorporated into the decoding block to improve the generative task. The attention module helps regulate the distribution of light, especially when there are multiple light sources in a dark image, such as the headlights from cars and roadside stores, therefore, serving the purpose of a normalizer for preventing under-exposure/over-exposure in the synthesized image.

The downsampling block is made up of 3 convolutional blocks. Each block consists of 3x3 convolutional layers followed by an instance normalization layer and a ReLU activation function. The cascaded block is made up of 2 residual blocks cascaded together to improve the features

learned and transmitted to the upsampling block, which is made up of a bilinear upsampling layer, a 3x3 convolutional layer, an instance normalization layer, and a ReLU activation layer to reduce the prevalence of checkerboard artifacts present in images.

The discriminator is a Markovian discriminator (PatchGAN) proposed by [31], which tries to classify if an N x N patch in an image is real or fake. By modeling the image as a Markov random field, the discriminator assumes the independence of pixels beyond the patch-size N, thereby discriminating based on only the patch-level information. By discriminating at smaller scales rather than the entire image, high-frequency features such as local texture and style can be captured as well as reducing the learnable parameters making it faster for large images. The PatchGAN comprises 5 convolutional layers which are used to transform a 6x256x256 input (including the real and generated image pair) to a 1x35x35 output that represents the average validity responses of the discriminator. Figure 3-3 shows the visual illustration of the discriminator architecture.

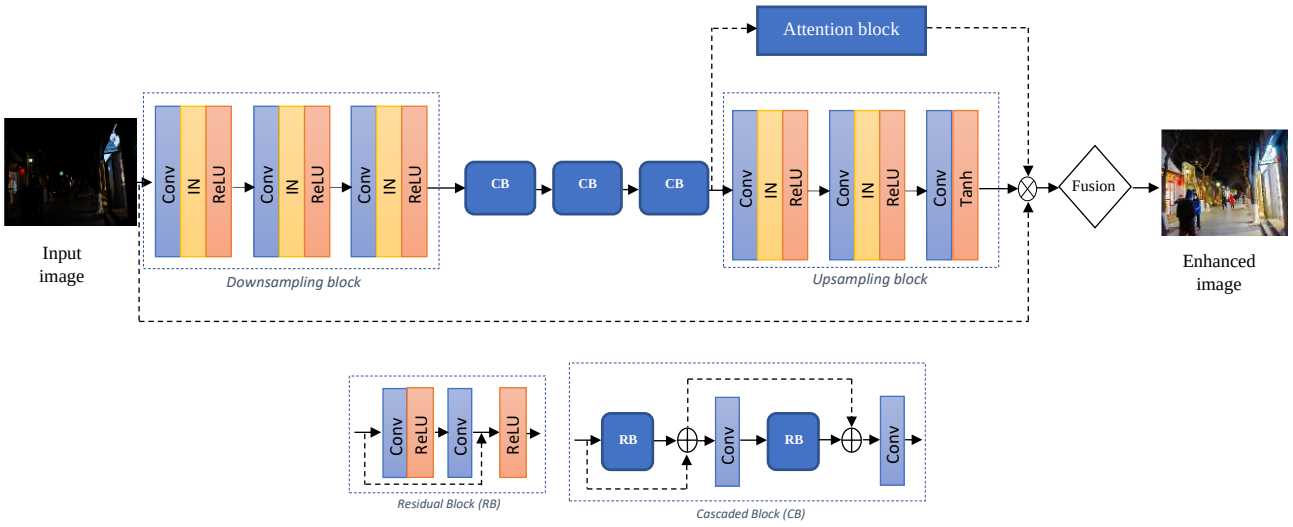


Figure 2-2: AG-CRN's generator architecture. Conv denotes a 2D convolution, IN denotes Instance normalization and ReLU denotes a Rectified Linear Unit activation function. The Attention block is same as the Upsampling block with the tanh activation replaced with a sigmoid. The fusion operation sums the outputs from the multiplication operation to create the enhanced image.

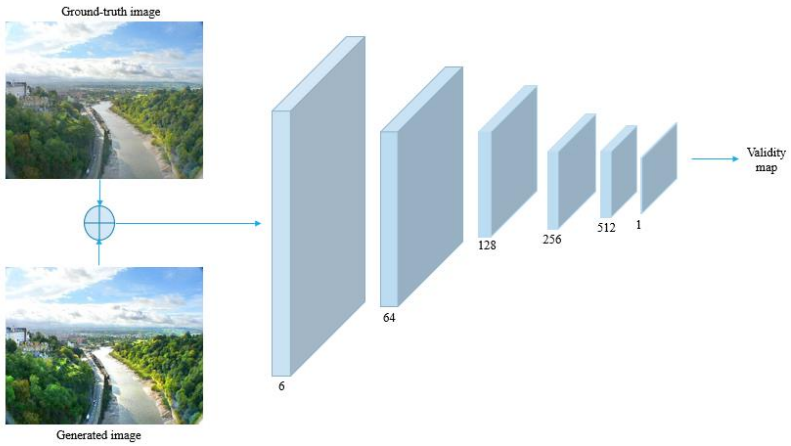


Figure 2-3: Network architecture of the PatchGAN discriminator. The PatchGAN discriminator is a rather shallow network that serves as an image classifier to distinguish original images from the generator's synthesized output image. The real and generated image pair are concatenated and fed to the network. The numbers under each convolutional block represent the number of filters for the corresponding layer.

2.2.2 Objective Function

The loss/objective function is a weighted combination of the adversarial loss, feature matching loss, perceptual loss, and gradient profile loss functions as shown in equation 2.1. Each of these loss functions considers different properties in the images to efficiently optimize the training objective. Therefore, the objective function for AG-CRN is given as

$$L_{AG-CRN} = L_{GAN} + \lambda_1 L_{FM} + \lambda_2 L_{VGG} + \lambda_3 L_{GPL} \qquad 2.1$$

$\lambda_1, \lambda_2, \lambda_3$ are weighting constants and are set to 10.0, 1.0, and 0.8, respectively. These values were experimentally determined to obtain the best results.

Adversarial loss: To model the conditional distribution of enhanced images given the input low-light images, an adversarial loss of GAN [31] is adopted

$$L_{GAN}(G, D) = \mathbb{E}_X[log\ D(X)] + \mathbb{E}_X\big[log(1 - D(G(\hat{X}))\big] \qquad 2.2$$

Where generator G seeks to minimize L_{GAN} and discriminator D tries to maximize it. X and $\hat{X}$ represent the low-light image and its enhanced ground truth, respectively. G($\hat{X}$) is the output of the generator network.

Feature matching loss: The adversarial loss is improved by incorporating a feature matching loss based on the discriminator. The features from multiple layers in the discriminator are extracted and matching intermediate feature maps between real and fake images is learned. This helps stabilize the training process and ensures that the generator produces reasonable natural statistical information at multiple scales [32].

$$L_{FM}(G,D) = \mathbb{E}_{X,Y} \sum_{i=1}^{T} \frac{1}{N_i} \left[\left\| D^{(i)}(X) - D^{(i)}(G(\hat{X})) \right\|_1 \right] \quad 2.3$$

where T is the total number of layers, N_i is the number of elements in each layer and $D^{(i)}$ is the *i*th-layer feature extractor of the discriminator.

Perceptual loss: This is used to evaluate the high-level perceptual and semantic differences between the low-light image and the enhanced pair [32]. The activations of the j-th layers of a pretrained VGG network for image classification are denoted by $\phi_j()$ are extracted. Pixel-wise distance is used to calculate the difference between the perceptual features of the low-light and enhanced image.

$$L_{VGG}^{\phi,j} = \frac{1}{C_j H_j W_j} \left\| \phi_j(\hat{Y}) - \phi_j(X) \right\|_1 \quad 2.4$$

where $\hat{Y}$ is the generated image, H_j and W_j are the height and width of the j-th feature map and C_j indicates the channel.

Gradient profile loss: This measures the difference between the edge information of the generated enhanced image and the ground-truth [33]. This is computed as

$$L_{GPL}(X,\hat{Y}) = \sum_{c}\left(\frac{1}{H}trace(\nabla G(\hat{Y})_c \cdot \nabla X_c^{\tau}) + \frac{1}{W}trace(\nabla G(\hat{Y})_c^{\tau} \cdot \nabla X_c)\right) \qquad 2.5$$

where $(\cdot)^{\tau}$ represents transpose, H and W are the height and width of the image, X is the ground-truth image and $\hat{Y}$ is the generated enhanced image.

2.3 Experiments and Results

Several open-source datasets were curated for training the AG-CRN network. This included 789 images from the DARK FACE dataset [34] , 485 images from the LOLdataset and 1000 synthetic images provided by [22]. To validate the novelty of the proposed AG-CRN model, a thorough evaluation is performed on commonly used benchmark datasets for low-light image enhancement, including test images in MEF [35] (17 images), LIME [36] (10 images), NPE [37] (8 images), VV [38] (24 images), DICM [39] (64 images), as well as the entire DARK FACE dataset [34] (6000 images).

Figure 2-4 shows the visual comparison of AG-CRN to other state-of-the-art approaches for low-light image enhancement of these benchmark datasets. As shown in the results, AG-CRN can successfully enhance dark regions in images while preserving its "naturalness" by avoiding artifacts due to over-exposure and maintaining the textural details of the original image.

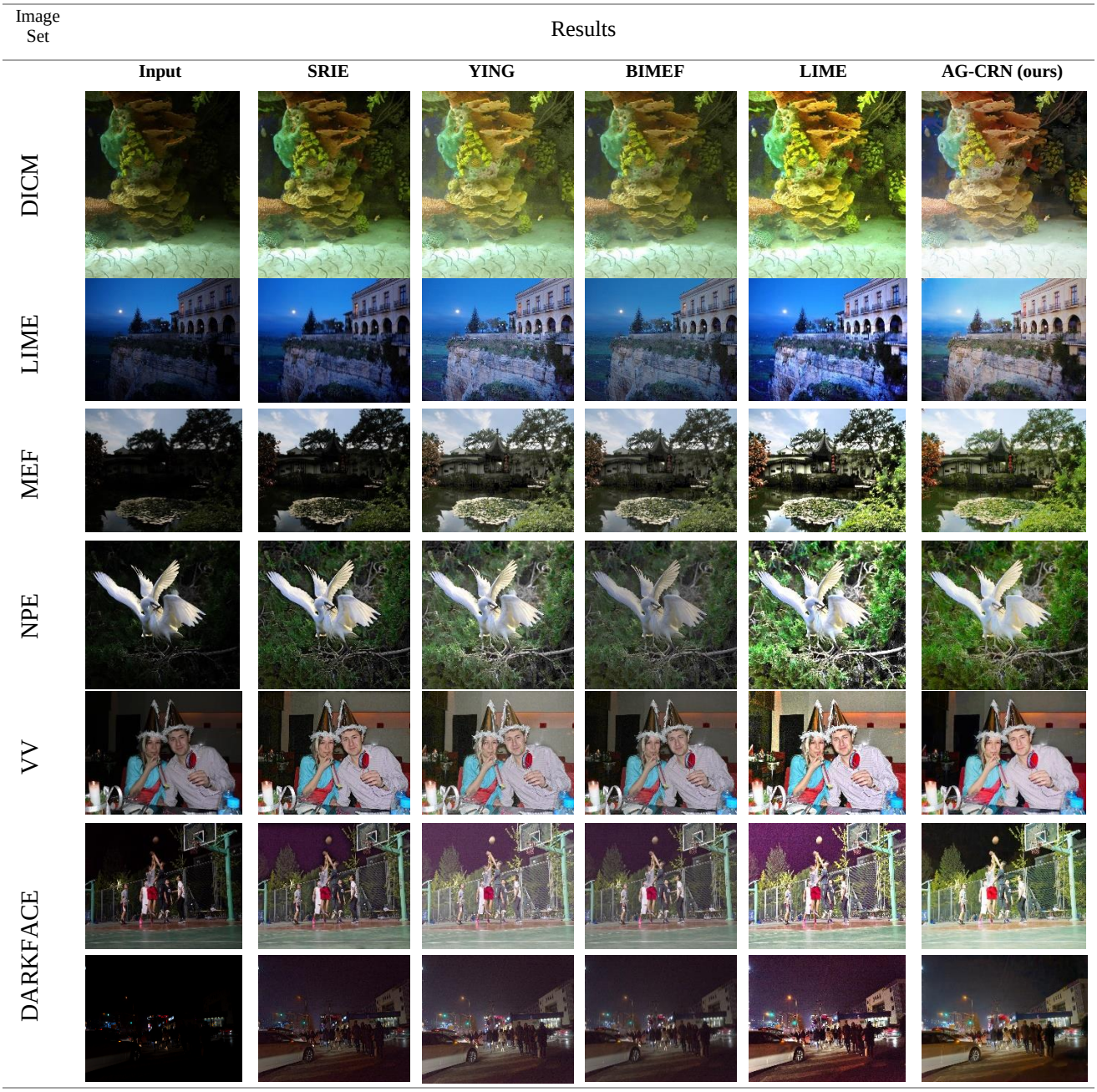


Figure 2-4: Qualitative comparison of AG-CRN to other state-of-the-art low-light enhancement methods. The results show that AG-CRN is consistently able to bridge the gap between enhancement and preserving the 'naturalness' of the images across all the benchmark datasets and with varying illumination levels and with multiple light sources in the background.

Quantitative evaluation is also performed on all the benchmark datasets using the Natural Image Quality Evaluator (NIQE), a no-reference image quality metric for evaluating the performance of

image restoration algorithms [40]. It uses components of an image such as contrast, color, resolution, and exposure level to compute scores based on the natural scene statistical model. A low NIQE score indicates a more natural-looking image, hence lower is better for evaluating an algorithm's performance.

Table 2-1: Quantitative comparison of AG-CRN to other state-of-the-art low-light enhancement algorithms across MEF, LIME, NPE, VV, DICM, and Dark Face datasets. Evaluation is performed using NIQE blind image quality measure in which a lower score indicates a more natural image. Metrics reported in this table represent the average score across all images in the corresponding dataset. The best scores are highlighted in green.

	Test datasets					
Methods	MEF	LIME	NPE	VV (*resized*)	DICM	Dark Face
Original input	4.264	4.358	4.318	4.395	4.256	4.447
LIME [21]	3.702	4.155	4.263	4.372	3.859	3.807
SRIE [20]	3.474	3.786	3.980	4.135	3.899	3.511
BIMEF [41]	3.320	3.817	4.196	4.146	3.846	3.617
YING [42]	3.239	3.855	4.323	4.190	**2.805**	3.633
EnlightenGAN [30]	3.232	3.719	4.113	-	3.570	-
Zero-DCE [29]	3.406	4.151	4.105	**3.238**	3.553	-
AG-CRN	**3.069**	**3.505**	**3.826**	3.849	3.547	**2.779**

Table 2-1 shows that AG-CRN outperforms other state-of-the-art methods on most test images across several benchmark datasets. The ability of AG-CRN to improve the illumination in an image while preserving the naturalness as well as prevent artifacts such as the halo artifact introduced as a result of over-enhancing regions with light sources makes it preferable albeit with a slight compromise in speed. AG-CRN runs at ≈ 30 fps, compared to other methods, which could run at speeds greater than 100 fps.

To further test the effectiveness of AG-CRN for common computer vision tasks, it is applied as a preprocessing step for face detection and landmark localization in current frameworks. RetinaFace [43] model trained on the WIDER FACE dataset [44] is used as the baseline model.

Figure 2-5: Performance of face detection and landmark localization in the dark. The red bounding boxes indicate regions where faces are detected, and the dots are the facial landmarks. AG-CRN significantly improves the performance of MobileNet and Resnet-based models with more detected faces.

Figure 2-5 shows the effects of first enhancing the images using AG-CRN on MobileNet and ResNet-based models. The results indicate that adding an enhancement model to the preprocessing step improves the capacity of lightweight models to extract meaningful features needed for face detection and landmark localization. The enhanced images are also natural looking with minimal artifacts introduced due to multiple light sources. Figure 2-6 presents a zoomed-in visualization emphasizing the benefits of AG-CRN.

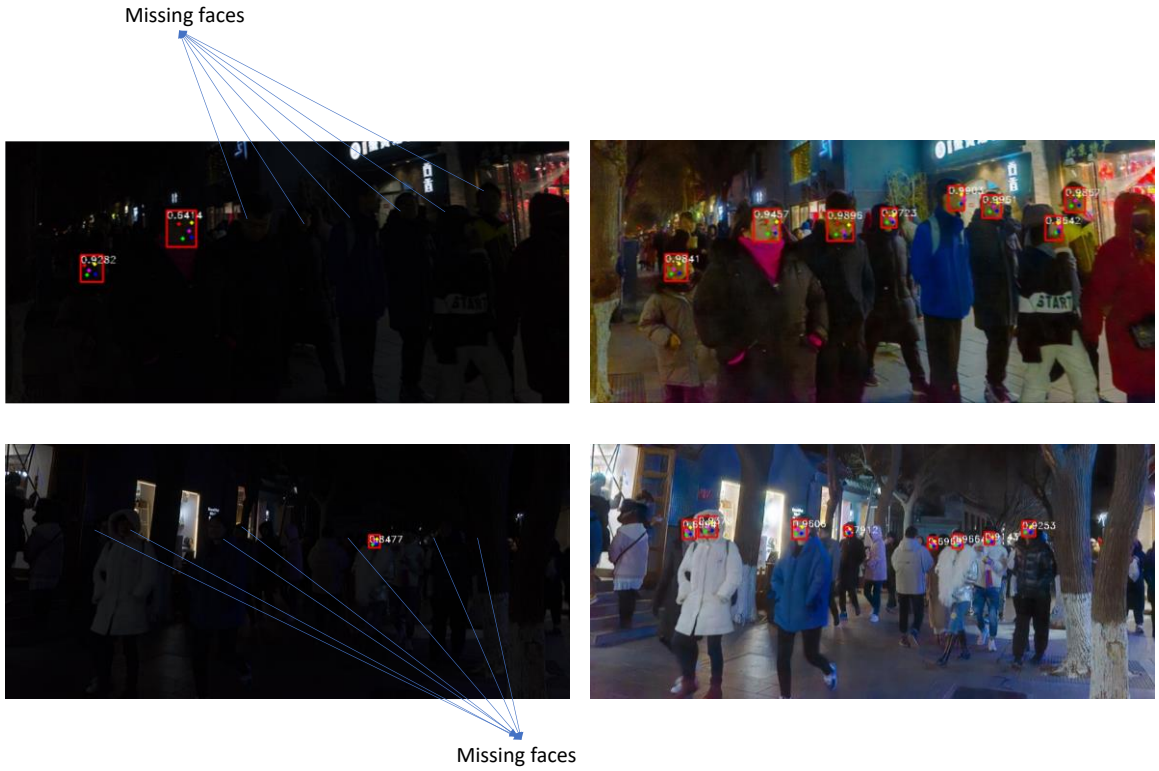


Figure 2-6: Zoomed in visualization of the missing faces in the original low-light image compared to the face detection in the AG-CRN enhancement output. Images in the left column are original low-light input images, while images on the right are the output of the proposed AG-CRN model. It can be seen that enhancing the input low-light image using AG-CRN can significantly improve the face detection rate and accuracy of face recognition.

All 6000 images from the DARK FACE dataset [34] are enhanced using the trained AG-CRN model, and the average precision is computed using the evaluation tool provided. Table 2-1 shows that enhancing the images leads to a significant improvement in the performance of the baseline detection and localization models.

Table 2-2: Performance improvement of AG-CRN on face detection models in dark images. Results presented in this table are average scores on the 6000 images in the DARK FACE dataset [35]. The values give the model's average precision (AP) before and after enhancement.

Model	**Raw image**	**Enhanced image**	**% Improvement**
MobileNet [45]	0.069	0.135	**95.652%**
ResNet50 [46]	0.162	0.198	**22.222%**

2.4 Summary

This chapter introduces a generative approach to low-light image enhancement for face detection and landmark localization in the dark. The proposed attention-guided cascaded residual network (AG-CRN) can translate a low-light image to a more visually pleasing and properly illuminated image while preserving its naturalness and with minimal artifacts introduced. Quantitative and qualitative results show that AG-CRN outperforms multiple state-of-the-art approaches across several test datasets and with varying illumination levels. Furthermore, we demonstrated the potential of our model to enhance face detection and landmark localization in naturally dark images using the DARK FACE dataset. Experimental results show a significant improvement in the performance of baseline models on AG-CRN-enhanced images compared to that of the original images. Future work will involve merging the enhancement, detection, and landmark localization tasks as a one-stage network. The potentials of AG-CRN for real-time applications will also be explored by making slight modifications to model architecture to address the limitations in speed.

Chapter 3. Improving the Performance of Object Trackers in Underwater Environments

This chapter introduces a novel approach for improving the visual quality of images captured in underwater environments. The model developed in this work is able to remove distortions caused by various factors such as diffraction, polarization, absorption, scattering, color loss, and attenuation of light at different depths underwater [47]. The work presented here has been published in the proceedings of the 2019 HST conference [48] as well as in the 2021 Journal of Oceanic Engineering [49].

The proposed Cascaded Residual Network for Underwater Image Enhancement (CRN-UIE) model addresses shortcomings of existing generative and deep learning-based approaches for removing distortions from underwater images.

While there has been tremendous advancement in object tracking for open-air visual data, less work has been explicitly done for underwater object tracking. Current trackers tend to work efficiently when tracking objects in an open-air environment facilitated by the availability of multiple benchmark datasets captured in a similar environment. However, no such datasets and benchmarks exist for underwater target tracking, and this lack of data has hindered the development of dedicated underwater tracking algorithms.

This chapter expounds on the design of the CRN-UIE model, including the building blocks, model architecture, and mathematical formulations for optimizing the network. Additionally, a novel comprehensive Underwater Object Tracking benchmark dataset (UOT100) is introduced to facilitate the development of tracking algorithms well-suited for underwater environments.

Finally, the quantitative and qualitative performance of CRN-UIE is analyzed in comparison to other state-of-the-art generative models on a large test benchmark.

3.1 Overview

Underwater object tracking is pivotal in applications such as underwater search and rescue operations, homeland and maritime security, deep ocean exploration, underwater robot navigation, and sea life monitoring. These applications require efficient and accurate vision-based underwater sea analytics, including image enhancement, image quality assessment, and target tracking methods. On the other hand, high-noise and low-light situations pose enormous challenges for understanding marine image/video analytics. Further exacerbating these issues are the inherent underwater distortions, including absorption and scattering of light causing low contrast, non-uniform illumination, diminished colors, and fuzz. This makes computer vision tasks for detection, recognition, and tracking in underwater environments much more challenging than in open-air. Figure 3-1 shows examples of common distortions observed in images captured underwater.

Several methods have been proposed to overcome the inherent distortions in underwater data by enhancing the visual quality of the images, including color restoration algorithms, contrast enhancement, quality metrics, and deblurring. Generative adversarial networks have recently been used for automatic underwater image correction and enhancement. While other methods work well for correcting one type of distortion, GAN-based methods attempt to improve visual quality by translating visual data from the underwater domain to its “enhanced” domain. Although GAN-based models have shown promising results on such translation tasks, more work needs to be done to improve the existing approach, including addressing the weaknesses exhibited in either color, contrast, or sharpness correction tasks.

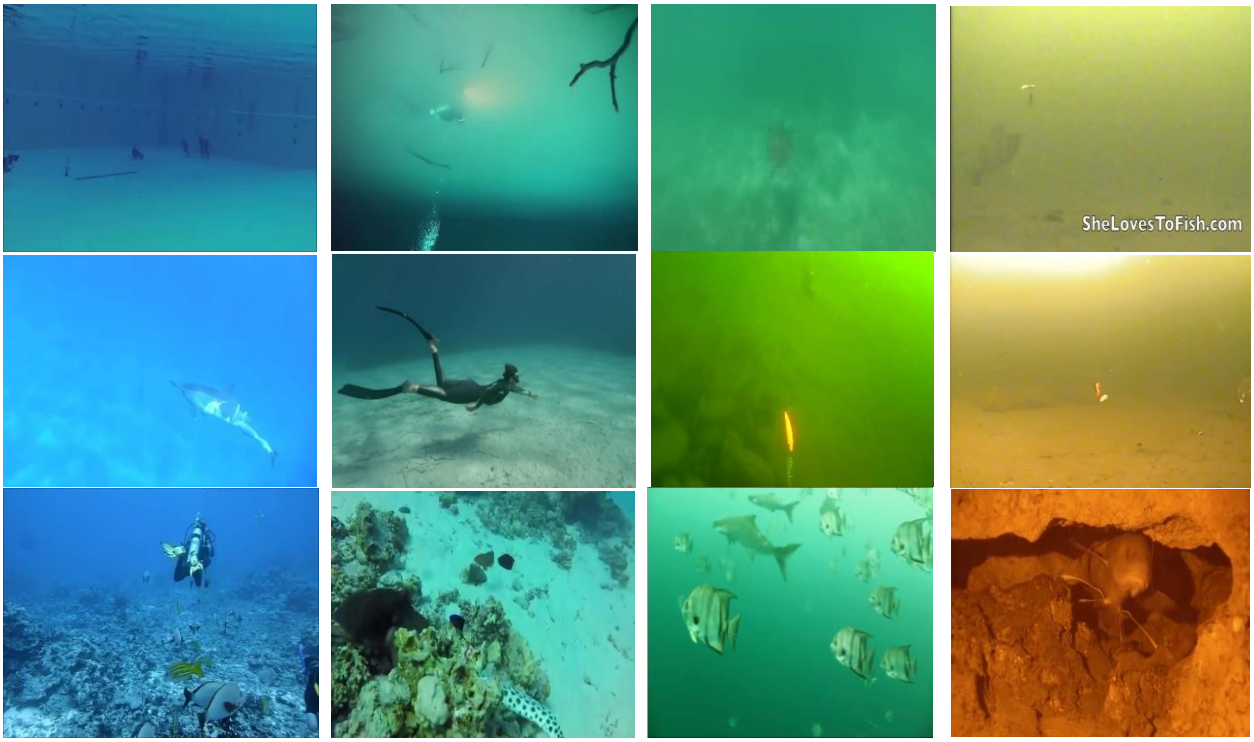


Figure 3-1: Sample images showing common distortions observed in underwater environments. Each column shows sample images of the corresponding distortions, including Blueish, Light green, dark green, and yellowish distortions. Varying levels of haze, blur, and fuzziness are also apparent in many of the images.

The recent success in object tracking has been facilitated by dedicated benchmarking datasets such as Object Tracking Benchmark (OTB) [50][51], Visual Object Tracking (VOT) [52], and Multiple Object Tracking (MOT) [52]–[55]. Although several object tracking methods have been proposed over the years [56]–[63], the bias of the publicly available benchmark datasets which mostly focus on open-air visual data, has greatly skewed the strength of these tracking algorithms to open-air environments. This is because the visual data these trackers are trained and tested on are not representative of underwater scenarios. As such, they each degrade in performance when tested on underwater scenarios, as demonstrated in previous exploratory work [64]. This motivates the necessity to develop a comprehensive underwater database and benchmark to foster the

development of tracking algorithms that will achieve comparatively high performance in both underwater and open-air environments.

The contributions outlined in this chapter are as follow:

a) Introduction of a novel Cascaded Residual Network for Underwater Image Enhancement (CRN-UIE), a GAN-based enhancement method for improving the trackers' performance on underwater data by translating visual data from a naturalistic underwater domain to an enhanced/clear underwater domain is presented.

b) The first comprehensive and diversified underwater benchmarking dataset (UOT100) was created, consisting of 104 video sequences and over 74000 annotated frames. The distortions in the dataset are well distributed, including both artificial underwater and natural underwater visual data. The distortions are also summarized into visual categories that describe the type of water, including blueish, greenish, and yellowish water visual data.

c) Performance benchmarking and detailed analysis of modern object tracking algorithms on the proposed UOT100 dataset.

d) Analysis of the impact of CRN-UIE on the performance of modern trackers.

3.2 Classical Methods for Underwater Image Enhancement

Classical methods for underwater image enhancement commonly adopt physical models such as the atmospheric scattering model or the simplified underwater formation model [65], [66].

3.2.1 Atmospheric scattering model

The atmospheric scattering model proposed by McCartney [67], is often used to describe the degradation of underwater images due to the absorption and scattering effects when light falls on an underwater medium. This model is expressed as

$$\boldsymbol{I}(x) = \boldsymbol{t}(x)\boldsymbol{J}(x) + (1 - \boldsymbol{t}(x)) \cdot \boldsymbol{A} \qquad 3.1$$

Where I is the observed image, x is the pixel location of the image, t(x) is the medium transmission map, J(x) ∈ [0,1] is the scene radiance reaching the camera, and A is the atmospheric light vector in the RGB domain. When the atmospheric condition is homogenous, t(x) can be expressed as

$$\boldsymbol{t}(x) = e^{-\beta d(x)} \qquad 3.2$$

Where β is the atmospheric attenuation coefficient and *d(x)* is the depth between the object and the camera. Increasing the distance between the camera and the objects leads to a decrease in **t**(x), denoting an increased sensitivity of the image to changes in atmospheric light.

3.2.2 Simplified Model

Chiang et al. [68] proposed a simplified underwater image formulation model to compensate for the disparity in wavelength attenuation along the underwater propagation path and account for artificial light's influence on a camera. Assuming the image captured underwater is $\boldsymbol{U}_\lambda(x)$, $\boldsymbol{J}_\lambda(x)$ is the scene radiance, $\boldsymbol{t}_\lambda(x)$ is the medium energy ratio which is the percentage of the scene radiance that is captured by the camera, and $\boldsymbol{B}_\lambda$ is the homogenous background light. The degradation model is represented as

$$\boldsymbol{U}_\lambda(x) = \boldsymbol{t}_\lambda(x)\boldsymbol{J}_\lambda(x) + (1 - \boldsymbol{t}_\lambda(x)) \cdot \boldsymbol{B}_\lambda, \qquad \lambda \in \{R, G, B\}) \qquad 3.3$$

$\boldsymbol{t}_\lambda(x)$ can be expressed as the energy of a light beam with wavelength λ before and after traveling distance *d*(x) within the water and is shown as follows:

$$\boldsymbol{t}_{\lambda}(x) = \frac{E_{\lambda}(x, d(x))}{E_{\lambda}(x, 0)} = 10^{-\beta(\lambda)d(x)} = N_{\lambda}(d(x)) \qquad 3.4$$

Where $E_{\lambda}(\mathrm{x}, d(\mathrm{x}))$ is the strength of light after absorption by the transmission medium, $E_{\lambda}(\mathrm{x}, 0)$ is the energy of light from the submerged scene before it passes through the medium and $\beta(\lambda)$ is the medium attenuation coefficient. N_{λ} is the ratio of the residual energy to the initial energy per unit of distance and can be affected by the wavelength of the light transmitted, water salinity, and the concentration of phytoplankton. Hence, a blue-ish colored underwater image is formed due to the fast attenuation of red wavelengths in open water compared to blue or green wavelengths.

These models help guide the development of underwater enhancement algorithms as they obey the principles defined by the equations above.

Filtering-based approach for underwater enhancement [69]–[71] use the transmission properties of light underwater to reduce underwater perturbations by suppressing noise, correcting for non-uniform illumination, enhancing contrast, and color adjustment in successive steps.

Color correction-based approaches [72]–[75] seek to remove the non-uniform color cast which is typical in underwater images. Chambah et al. [73] use a human visual system-inspired color equalization algorithm for achieving lightness and color constancy. Ghani and Isa [74][75] proposed two methods for color correction based on Rayleigh stretching, which stretches the blue histogram of the Rayleigh distribution to a low grayscale level and the red histogram to a high grayscale level.

One major drawback of these classical methods is that the effectiveness is often dependent on the hyperparameters selected and does not generalize well to the diversity of distortions common in underwater images.

3.3 Deep Learning-based Methods

Deep learning-based methods for underwater image enhancement have been proposed to improve the models' generalizability by learning from a wide variety of underwater distortions. Anwar et al. [76] proposed UWCNN, an end-to-end trainable convolutional neural network-based model for underwater image enhancement. The UWCNN model is trained on a synthetically created underwater image database using the attenuation properties of different water types. Li et al. [77] developed an underwater image enhancement benchmark dataset including 950 real-world underwater images and proposed a gated fusion CNN model called Water-Net. A multi-stage approach is adopted in training Water-Net by creating three inputs generated by sequentially applying white balancing, histogram equalizing, and gamma correction algorithms on the original underwater image. The generated images are then used for training the multiscale, fully connected CNN layers that make up the Water-Net architecture. Hou et al. [78] proposed a residual convolutional neural network, URCNN, which aims to learn the transmission map of the underwater images.

Recently, generative adversarial networks (GANs) have been used for automatic underwater image correction and enhancement. While other methods work well for correcting one type of distortion, GAN-based methods attempt to improve visual quality by translating visual data from the underwater domain to its equivalent “enhanced” domain. Existing GAN-based models have shown promising results on such translation tasks. However, more work needs to be done to improve existing underwater GAN models, including addressing the weaknesses exhibited in either color, contrast, or sharpness correction tasks. In this work, an efficient GAN model is developed to improve the performance of existing trackers on distorted underwater data by efficiently translating the distorted data to their non-distorted/enhanced or clear underwater versions. The

proposed model also addresses the above-mentioned weaknesses in existing GAN models on underwater image enhancement tasks as demonstrated in subsequent computer simulation tasks.

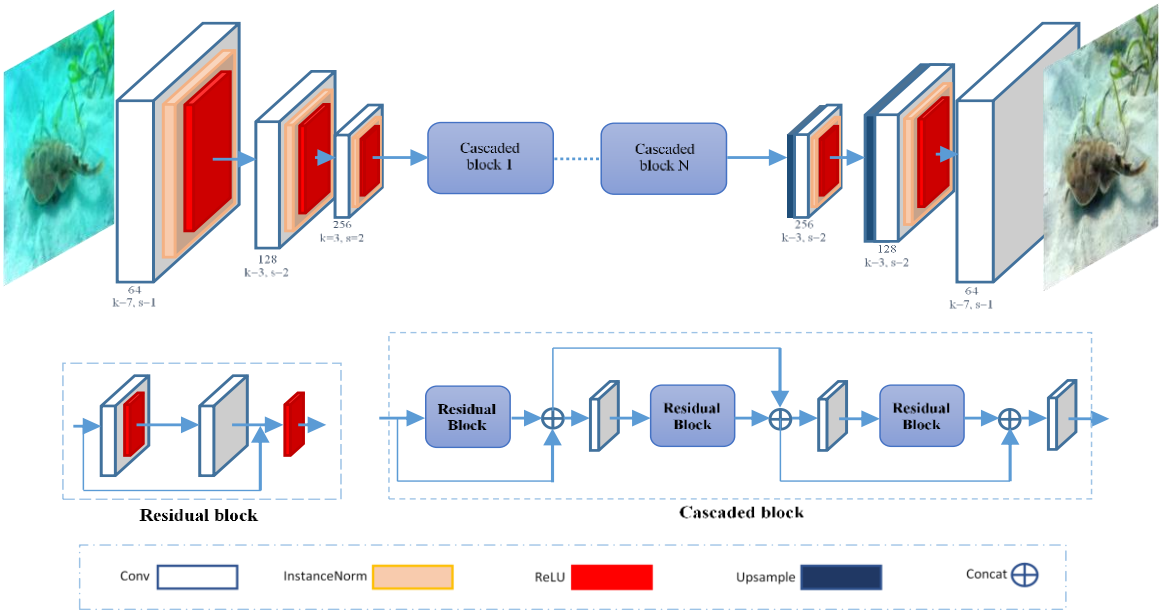


Figure 3-2: Network architecture of CRN-UIE generator. Conv denotes a 2D convolution layer, ReLU is the rectified linear unit activation function, InstanceNorm is the instance normalization layer. The blocks are color-coded, and the numbers below each combo block represent the number of filters, kernel size, and stride size, respectively.

3.4 Proposed CRN-UIE Model

Given a distorted underwater image X, the goal of CRN-UIE is to learn a mapping function that translates X to a target non-distorted/enhanced domain Y.

3.4.1 CRN-UIE Network architecture

Inspired by the work of Wang et al. [79], similar image generator and discriminator architectures are adopted in CRN-UIE for synthesizing high-resolution photo-realistic images using conditional generative adversarial networks. The residual blocks used in the Pix2PixHD model [79] are replaced with cascaded residual blocks originally designed for image super-resolution tasks [81]

to extract rich local features via the densely connected convolutional layers. CRN-UIE follows the encoder-decoder pattern of the U-Net architecture [80]. The encoder makes up the first half of the architecture, and it encodes the input image into feature representations across multiple downsampled layers while the decoder semantically projects the discriminative features learned by the encoder onto an upsampled higher resolution space. Skip connections are also added to the network to ensure maximum information propagation between the corresponding encoder and decoder layers.

The generator network consists of 3 down-sampling blocks in the encoder and three up-sampling blocks in the decoder. Each consists of 2D convolutional layers combined with an instance normalization layer and a rectified linear unit (ReLU) activation layer. Each cascaded block (CB) consists of 3 residual blocks and a single output convolution layer. Residual blocks, in turn, are also made up of 2 convolutional layers with a filter size of 3x3, each followed by a ReLU activation layer, with skip connections. The cascaded block ensures that the generator mapping function learns the image's finest textural and structural characteristics and passes them on to the up-sampling blocks. It still allows for the correct transmission of more global elements learned in earlier layers. To ensure the eradication of any checkerboard artifacts, the decoder network employs bilinear sampling in conjunction with instance normalization and ReLU activation. The details of the architecture are shown in Figure 3-2.

CRN-UIE also uses a multiscale discriminator similar to that of pix2pixHD [79]. This consists of two identical discriminators D_1 and D_2. The output from the generator is fed to D_1, downsampled by a factor of 2, then fed to D_2. This process guides the generator to generate globally consistent images and produce fine details.

3.4.2 Objective Function for Optimizing CRN-UIE

The objective function formulated for optimizing the proposed CRN-UIE model is a weighted combination of the adversarial, perceptual, feature matching and gradient profile loss. Each component of the loss helps in addressing a specific component of the generating image.

i. Adversarial loss [81]**:** The generator tries to learn a mapping function $G: X \rightarrow Y$ and fool discriminators D_1 and D_2. The adversarial loss is expressed as

$$L_{ADV} = \min_{G}\left[\max_{D_1,D_2}\sum_{k=1,2}\left[E_{(X)}[log\, D_k(X)] + E_{(X)}\left[log(1 - D_k(G(\hat{X}))\right]\right]\right] \qquad 3.5$$

X and $\hat{X}$ denote a distorted and undistorted underwater image respectively, $G(\hat{X})$ is the output of the generator network, and k distinguishes between discriminators 1 and 2.

ii. Feature matching loss [79]**:** This improves the adversarial loss by stabilizing training and ensuring that the generator produces reasonable statistical information at multiple scales. To do this, we learn to match intermediate feature maps between the real and generated image.

$$L_{FM} = \min_{G}\left[\sum_{k=1,2} E_{(X)}\sum_{i=1}^{T}\frac{1}{N_i}\left[\left\|D_k^{(i)}(X) - D_k^{(i)}(G(\hat{X}))\right\|_1\right]\right] \qquad 3.6$$

where T is the total number of layers, N_i is the number of elements in each layer and $D_k^{(i)}$ is the *i*th-layer feature extractor of discriminator D_k.

iii. **Perceptual loss** [32]**:** This is used to measure the high-level perceptual and semantic differences (such as color and gradient orientation) between images. The activations of the j-th layers of a pretrained VGG network for image classification denoted by $\phi_j()$ is extracted. Pixel-wise distance is used to measure the difference between the perceptual features of the distorted and enhanced underwater image.

$$L_{VGG}^{\phi,j} = \frac{1}{C_j H_j W_j} \left\| \phi_j(\hat{Y}) - \phi_j(X) \right\|_1 \quad 3.7$$

where $\hat{Y}$ is the generated image, H_j and W_j are the height and width of the j-th feature map and C_j indicates the channel.

iv. Gradient profile loss [33]**:** To measure the difference between the edge information of the generated and target images, we compute the gradient profile loss proposed by [33].

$$L_{GPL}(X, \hat{Y}) = \sum_c \left(\frac{1}{H} trace(\nabla G(\hat{Y})_c \cdot \nabla X_c^\tau) + \frac{1}{W} trace(\nabla G(\hat{Y})_c^\tau \cdot \nabla X_c) \right) \quad 3.8$$

where $(\cdot)^\tau$ represents transpose, H and W are the height and width of the image, X is the target image and $\hat{Y}$ is the generated image.

Therefore, the total loss is the sum of the adversarial loss L_{ADV}, feature matching loss L_{FM}, perceptual loss $L_{\text{VGG,}}$ and gradient profile loss L_{GPL}.

$$\boldsymbol{L_{CRN-UIE} = L_{ADV} + \lambda_1 L_{FM} + \lambda_2 L_{VGG} + \lambda_3 L_{GPL}} \quad 3.9$$

Where λ_1, λ_2 and λ_3 are constants.

We set N = 2, where N is the number of cascaded blocks used. An Adam optimizer [82] with an initial learning rate of 2e-4, beta1 = 0.5, and beta2 = 0.999 is used to optimize the loss function. The values were determined empirically. An intuitive and empirical hyperparameter search through extensive computer simulations indicated that λ_1 = 10, λ_2 = 1, and λ_3 = 1 represented the best parameter combination for the proposed model yielding the desired outputs. A *step decay* annealing strategy is used with a decay constant of 100 to decrease the learning rate after 100 epochs.

3.5 Experiments and Results

Figure 3-3 shows the comparative visual outputs of the CRN-UIE model compared with several other state-of-the-art models. Subjectively, CRN-UIE produces a better visual output on distorted to enhanced underwater image translation tasks. These results are further quantitatively substantiated in Table 3-1. All four models being compared here, including the proposed CRN-UIE, were trained on the EUVP (Enhancing Underwater Visual Perception) dataset prepared by the Interactive Robotics and Vision Lab from the University of Minnesota.

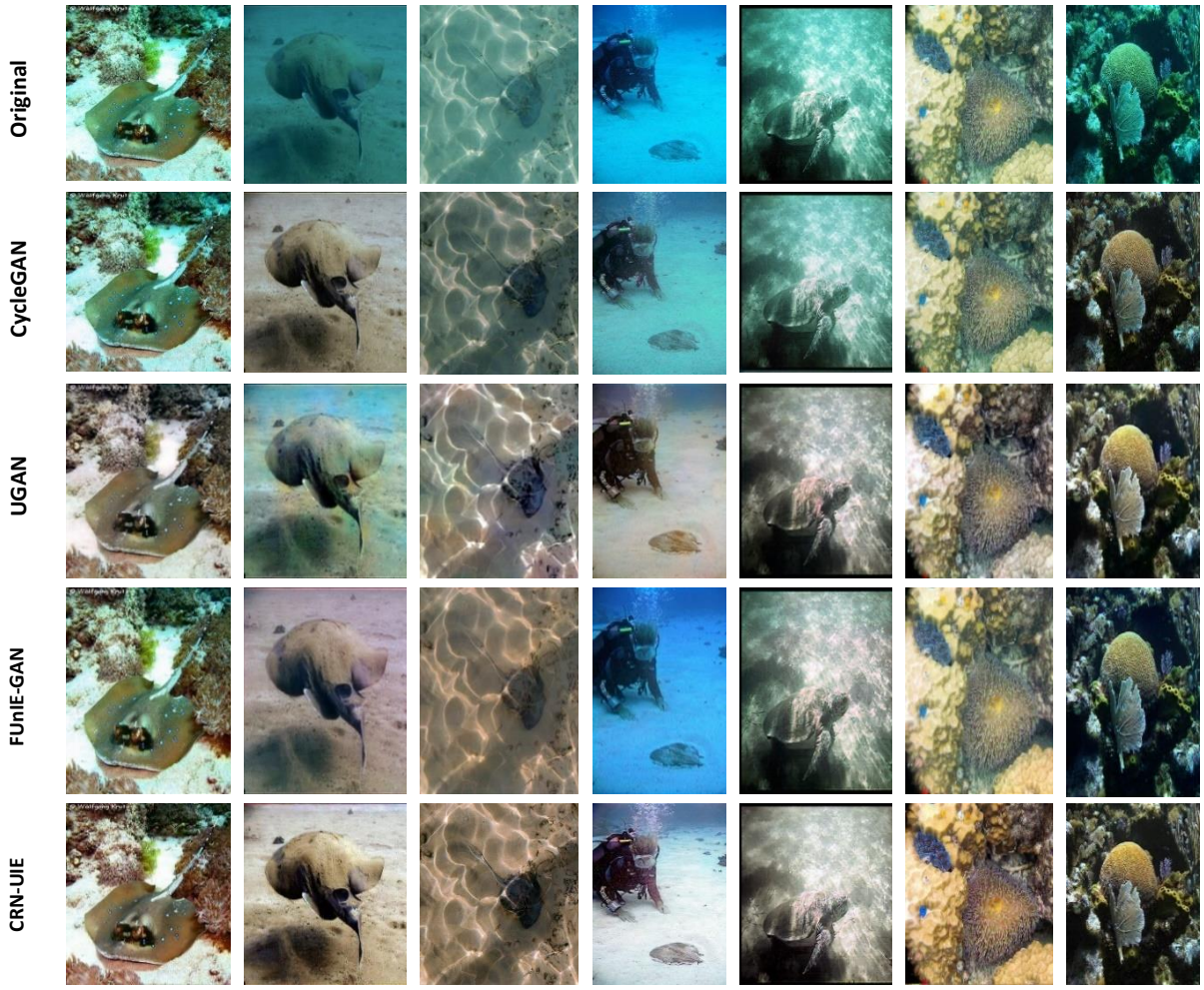


Figure 3-3: Visual comparison of the sample results from the proposed CRN-UIE with other state-of-the-art generative models. CRN-UIE produces more realistic enhanced images from the original distorted underwater input images. It also yields better visual quality while preserving the color of objects underwater.

The dataset was introduced in the FUnIE-GAN (Fast Underwater Image Enhancement) paper [83] and contained paired and unpaired image samples of clear and distorted underwater visual quality. FUnIE-GAN used the procedure in [84] to generate paired dataset [85]. Here, for CRN-UIE, we use paired data from the EUVP dataset for training.

A quantitative evaluation of the performance of these models using no-reference measures including Underwater Image Quality Measure (UIQM) [86] and the recently proposed CCF metric [87], which is a weighted summation of the colorfulness, contrast, and fog density index, was performed on 1381 images from the test dataset provided in the EUVP dataset.

Table 3-1: Quantitative comparison of the performance of CRN-UIE to other state-of-the-art GAN models using individual components UIQM including color (UICM), sharpness (UISM), and contrast (UIConM) and CCF no-reference underwater image quality measures. Green indicates the best score, and bolded values indicate the score for CRN-UIE.

Method	**UICM**	**UISM**	**UIConM**	**UIQM**	**CCF**
Original	-59.6120	5.6595	0.7320	2.6074	28.6571
CycleGAN [84]	-25.5348	6.4704	0.8401	4.1942	22.9768
FUnIE-GAN [83]	-14.9349	7.2558	0.8623	4.8044	26.0130
UGAN [85]	-3.1183	7.1064	0.8497	5.0485	27.6772
CRN-UIE	**2.5228**	**6.5798**	**0.7842**	**4.8177**	**35.5643**

The quantitative results presented in Table 4-1 indicate that CRN-UIE can effectively remove artifacts introduced as a result of the effect of light attenuation and refraction underwater, as well as improve the colorfulness in the enhanced underwater output images to other methods. Outputs from FUnIE-GAN tend to have better sharpness and better contrast as measured by the UISM and UIConM components of the UIQM metric. UGAN on the other hand has better overall UIQM

score, whereas CRN-UIE has a much higher CCF score which measures the overall quality of the underwater color image. The higher the value for each metric, the better the image's visual quality.

3.6 Proposed UOT100 Benchmark Dataset and Evaluation Metrics

3.6.1 Review of Existing Object Tracking Datasets

Several object tracking benchmarks have been proposed in the past and are still being updated to include more challenging test data. The most popular benchmark datasets are competition datasets. OTB [50], [51] is one of the most popular single object tracking challenge datasets. The earlier version of OTB, OTB50 consisted of 51 video sequences but was later updated to include up to 98 video sequences (OTB100) with more challenging tracking data. The sequences are classified into nine different attributes or categories, each representing a challenging aspect of visual tracking, including: partially occluded targets, motion blur, deformation, fast motion, background clutter, out of view, in-plane rotation, scale variation and illumination variation. A well-maintained online toolkit is available for the user to run their trackers on the dataset and compare their performance to the most recent state-of-art tracking algorithms. OTB challenge uses success rate and precision plots in One Pass Evaluation (OPE) to evaluate the performance of trackers. This will be used as the evaluation metric for the presented benchmarking dataset.

Since 2013, the VOT challenge [50]–[52], [88] has published top-performing tracking algorithms on their competition dataset. The Thermal Infrared challenge of the VOT (VOT-TIR2015) encouraged the research community to develop state-of-art trackers with optimal performance on thermal images.

The MOT [53]–[55] benchmark dataset is aimed at advancing research in the development of efficient multiple object tracking algorithms. Like VOT, MOT provides a yearly list of top-performing trackers on their 2D and 3D datasets. Evaluation is based on two metrics: accuracy,

and robustness. Accuracy measures the distance between tracker and ground truth bounding box centroids, while robustness measures the frequency of tracking failures [89].

Table 3-2: Comparative summary between existing object tracking datasets and the proposed UOT100 dataset. While most existing datasets focus on open-air data, the UOT100 and UOT32 datasets only contain underwater data with the ultimate goal of improving tracking accuracy and efficiency of state-of-the-art object tracking algorithms in underwater environments.

Benchmark Dataset	**Open-air Visual**	**Underwater Visual data**	**#Video sequences**	**#Annotated frames**	**Avg # frames / video**
OTB50	✓		51	29,491	578
OTB100	✓		98	58,610	598
MOT16	✓		14	182,326	13,023
MOT17	✓		21	564,228	845
VOT (13, 14, 15, 16, 17, 18, 19)	✓		60	21,455	357
UAV123	✓		123	110,000	894
NUS-PRO	✓		365	135,305	370
TLP (Long Term)	✓		50	676,000	13,520
Long-Term Tracking in the Wild	✓		337	1.55M	4,599
UOT32		✓	**32**	**24,241**	**757**
UOT100		✓	**104**	**74,042**	**698**

The success of these competitions and other challenge benchmark datasets has led to numerous research publications in object tracking. Although existing benchmarks have strived to incorporate as many challenging aspects of visual tracking as possible, the inherent tracking challenges in the underwater visual domain are yet to be explored. Our goal is to make the dataset available and generate traction that will enable researchers to develop more sophisticated trackers for underwater environments. A brief comparison of UOT100 with some prominent challenge datasets is provided in Table 3-2.

With the availability of these large annotated datasets, numerous sophisticated tracking algorithms have been proposed [56]–[63], [90], [60]–[63], [91]–[96]. The methods used in most of these trackers can be classified into various categories depending on how features are extracted and used

for tracking. Trackers such as KCF [56], HCF [95] , DCF [93], [95] , CFNet [97], STRCF [61], BACF [60], and fDSST [98] are considered kernelized correlation filters. The CNN feature-based approach has been predominantly used in state-of-the-art trackers such as MDNet [62], SiamFC [58], CCOT [96], and ECO [57]. Older trackers such as Struck [99] are based on local and global feature extraction.

3.6.2 Proposed Underwater Object Tracking Benchmark Dataset – UOT100

A novel dataset called UOT100 is curated to further research in the tracking of objects of interest in underwater environments. It consists of 104 underwater video sequences sourced and segmented from different YouTube videos.

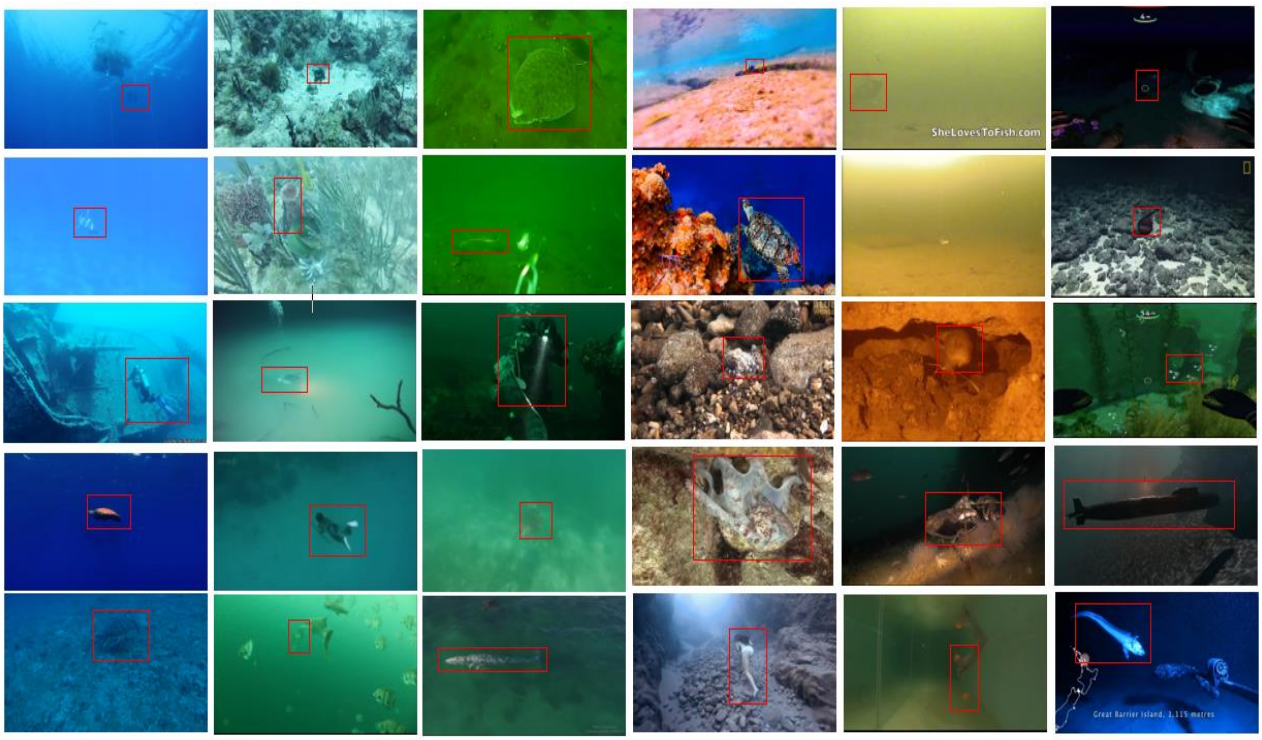
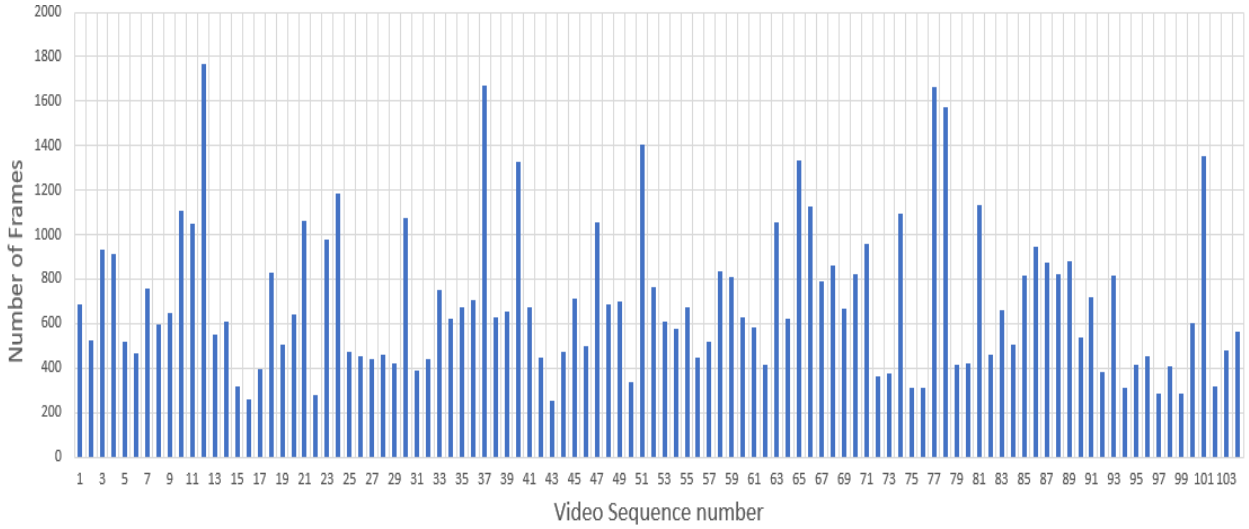
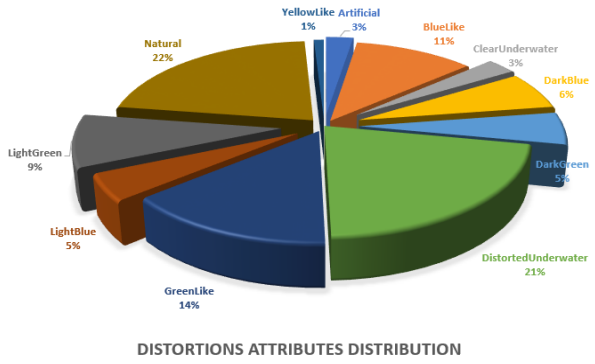


Figure 3-4: Sample tracking data from the UOT100 dataset showing various types of distortions. The red bounding box denotes the object of interest and the text below each column indicates the category of visual data.



(a)



(b)

Figure 3-5: An overview of the number of annotated frames and distortion type distribution across the dataset. (a) shows the number of annotated frames, and (b) is a pie chart distribution of various distortion types across the UOT100 dataset.

These samples are diverse and vary by camera type, imaging conditions (viewing distance, viewing angle, background, and water quality), and target objects. The dataset also contains four videos of artificial underwater images generated from the **Subnautica** game [100]. This is to introduce more diversity and distortions in the dataset and test the performance of trackers in simulated underwater environments. The dataset contains a total of 74,042 annotated frames, with an average of 698

annotated frames and 26.2 seconds per video, and captures a wide variety of underwater distortions. Samples of the underwater tracking data extracted from the UOT100 dataset are shown in Figure 3-4, with each column showing variations of similar types of visual data. The UOT100 benchmark dataset is a substantially improved version over an earlier dataset, UOT32 [48]. UOT100 is a much larger dataset and includes many more underwater distortion categories. Here, a comprehensive analysis and a new enhancement algorithm that results in drastic performance increases of tracking algorithms are presented.

The dataset is organized in folders and subfolders. The root folder "*UOT100*" contains all the sequences in the dataset, and each sequence is stored in a separate folder. Each sequence folder contains an "*img*" folder which includes all frames in the sequence; an mp4 video file; a ground truth text file which contains the ground truth annotations for each frame in the sequence; and a description file listing the distortion categories that the sequence is assigned to. Figure 3-5 shows the distribution of the number of frames across the dataset. This work provides a comprehensive dataset for benchmarking the performance of state-of-the-art single object trackers in underwater scenarios.

3.6.3 Evaluation Metrics

The One Pass Evaluation (OPE) protocol of the OTB benchmark proposed by Wu et al. [50] will be adopted in this work for evaluating trackers' performance. The average precision and the success rate were computed to benchmark the performance of tracking algorithms on the UOT100 dataset.

The precision is calculated as the distance error between the center pixel of the ground truth bounding box and the predicted bounding box. The prediction of the tracker for frame k is considered accurate if the pixel distance from the predicted to the ground truth is less than a given

pixel threshold. The precision plots shown in subsequent figures are computed by calculating the precision for each tracker at increasing threshold values between 0 and 20 pixels.

$$P_k = \begin{cases} 1, & if \ \sqrt{(C^{gt} - C^{tk})^2} \leq Thres \\ 0 & otherwise \end{cases} \quad 3.10$$

The average precision of a tracker over a sequence is calculated as the ratio of the total number of accurate predictions to the total number of frames in the sequence.

$$P_{seq} = \frac{1}{N}\sum_{1}^{N} P_k \quad 3.11$$

Precision alone is not enough to compare the performance of multiple trackers on a given sequence or dataset. This is because precision does not consider how accurate the predicted bounding box is with respect to the ground truth bounding box in terms of size and how much area of the target object is covered in the predicted bounding box. The precision metric is therefore complemented by the success rate, which measures the Intersection over Union (IoU), representing the overlap ratio between the ground truth bounding boxes and the predicted bounding boxes, where a **0** ratio means no overlap (complete miss) and a **1** means 100 percent overlap (perfect match). The IoU is defined in Equation 3.12, and success rate plots are obtained by calculating IoU at various threshold values between **0** and **1**.

$$S = \frac{|GT_{bbox} \cap TK_{bbox}|}{|GT_{bbox} \cup TK_{bbox}|} \quad \begin{cases} GT_{bbox} : ground\ truth\ bounding\ box \\ TK_{bbox} : Tracker\ \ bounding\ box \\ \cap \ \ and \ \cup : intersection\ and\ union\ respectively \end{cases} \quad 3.12$$

3.6.4 Performance Benchmarking on Un-enhanced UOT100 dataset

The performance of 20 state-of-the-art tracking algorithms using the UOT100 dataset was obtained. Some of the trackers are from the OTB benchmark, and others are more recent deep learning-based tracking algorithms including: BACF [60], BOOSTING [101], CCOT [96], CFWCR [102], CSR-DCF [103] , DCF [93], ECO [57] , fDSST [98], KCF [56], MDNet [49], MEDIANFLOW [104], MIL [103], MOSSE [93], SiamMask [105], SiamRPN++ [106], STAPLE [90], STRCF [63], TLD [107] and VITAL [94].

Next, the performance of the selected trackers across the entire UOT100 dataset was investigated and reported using the One Pass Evaluation (OPE) precision and success rates. The Area Under Curve (AUC) is used to further rank the performance of the trackers listed. Figure 3-6 shows the OPE Precision and Success rate benchmark plots across the entire dataset. The deep learning method, SiamRPN++, achieves the highest performance on the UOT100 dataset, in both precision and success rate, as shown by the red curve. However, while SiamRPN++ did well on natural open-air datasets like OTB, it exhibited significant performance degradation on underwater data, achieving only a precision AUC of **0.230** and a success rate AUC of **0.511** on the UOT100, as opposed to a precision and success rate AUC of **0.914** and **0.696** respectively on the OTB.

COT, CFWCR, STRCF, CSR-DCF, ECO, BACF and STAPLE all exhibited low performance on the UOT100 with precision of **0.131, 0.126, 0.120, 0.119, 0.111, 0.117,** and **0.101,** and success rate AUC of **0.511, 0.407, 0.391, 0.465, 0.399, 0.397,** and **0.361,** respectively. Other Trackers never reached a precision AUC above **0.1**, demonstrating the difficulty of tracking in an underwater environment.

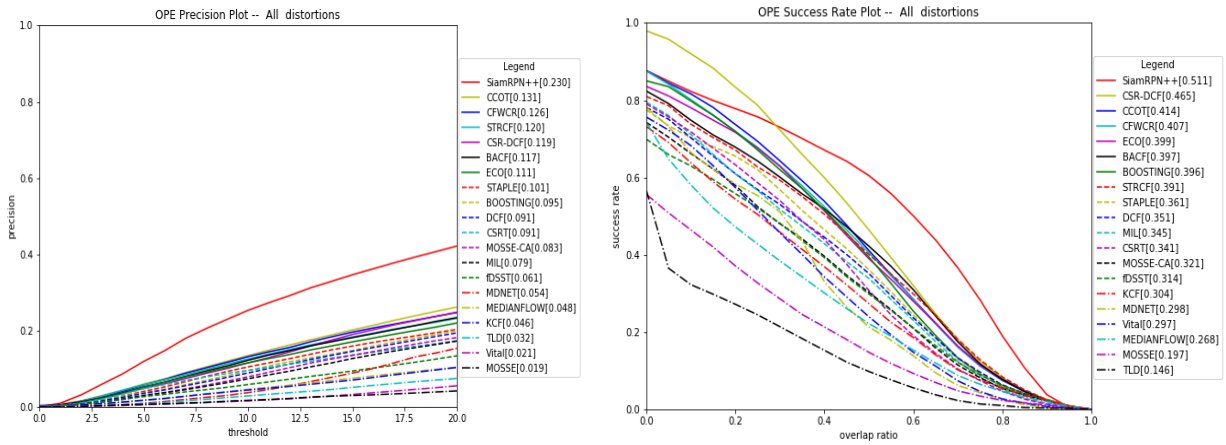


Figure 3-6: Precision and Success Rate plots on the UOT100 dataset. In the legend, the values in brackets after each tracker represent the area under curve (AUC) of the corresponding plot. Trackers in the legend are ranked from top to least performing using their AUC values.

Analyzing further, the trackers' performance across the different types of underwater visual qualities including Natural vs Artificial, and Clear vs Distorted can be seen in Figure 3-7 and Figure 3-8 compares the benchmark performance of the trackers on Natural vs Artificial underwater visual data, while comparison of trackers performance on Clear Underwater vs Distorted Underwater visual data is shown in Figure 3-8. The trackers tend to exhibit slightly better precision AUC performance on artificial underwater visual data than natural. However, the corresponding success rate plots for artificial data are much less stable. This is because artificially generated underwater data either under-simulate or over-simulate distortions and often do not capture them all.

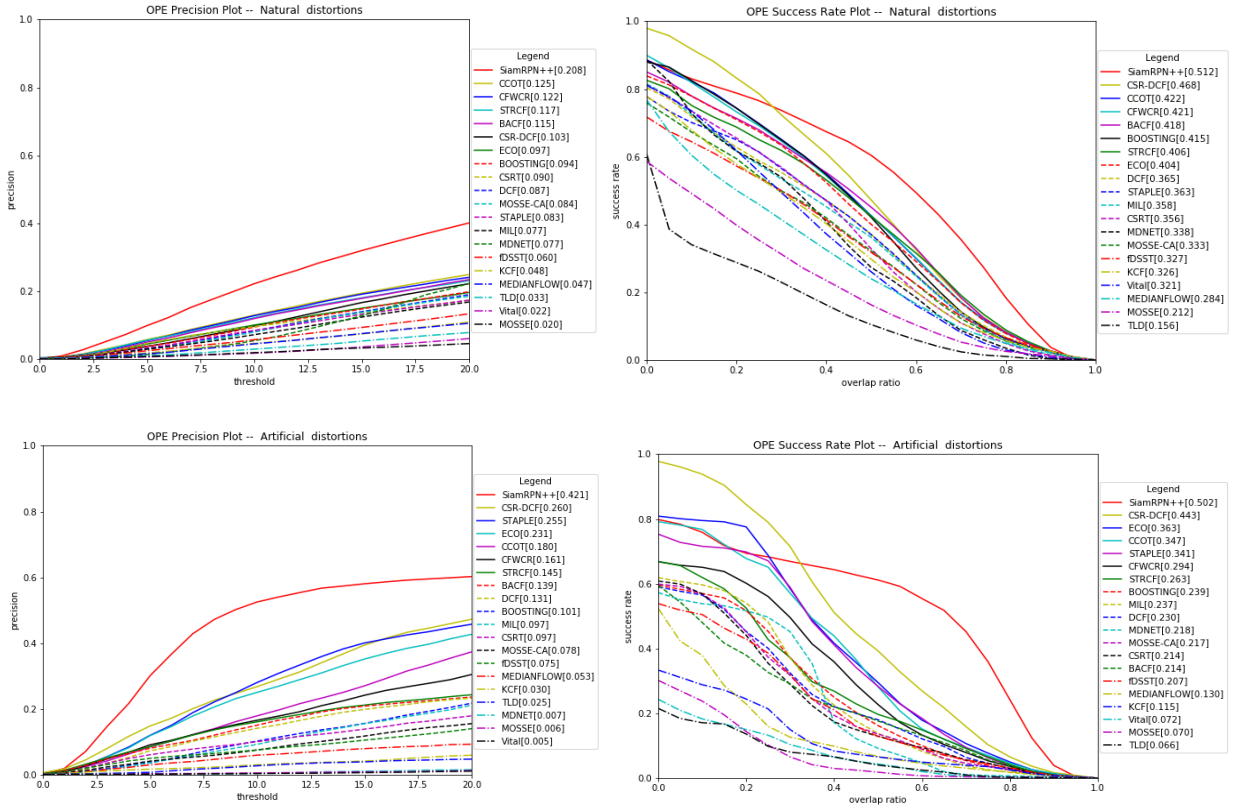


Figure 3-7: OPE precision and success rate plots on Natural (first row) vs Artificial (second row) underwater visual data. Trackers in the legend are ranked from top to least performing using their AUC values and are each color-coded.

Further analysis based on different visual distortion categories is included in the subsequent section. The trackers also tend to perform better on clear underwater than distorted underwater visual data, especially looking at the AUC results on the OPE success rate plots.

SiamRPN++, CFWCR, CCOT, ECO, and STRCF trackers consistently ranked as the best trackers on the overall dataset and across simulations on various types of underwater visual data and distortions.

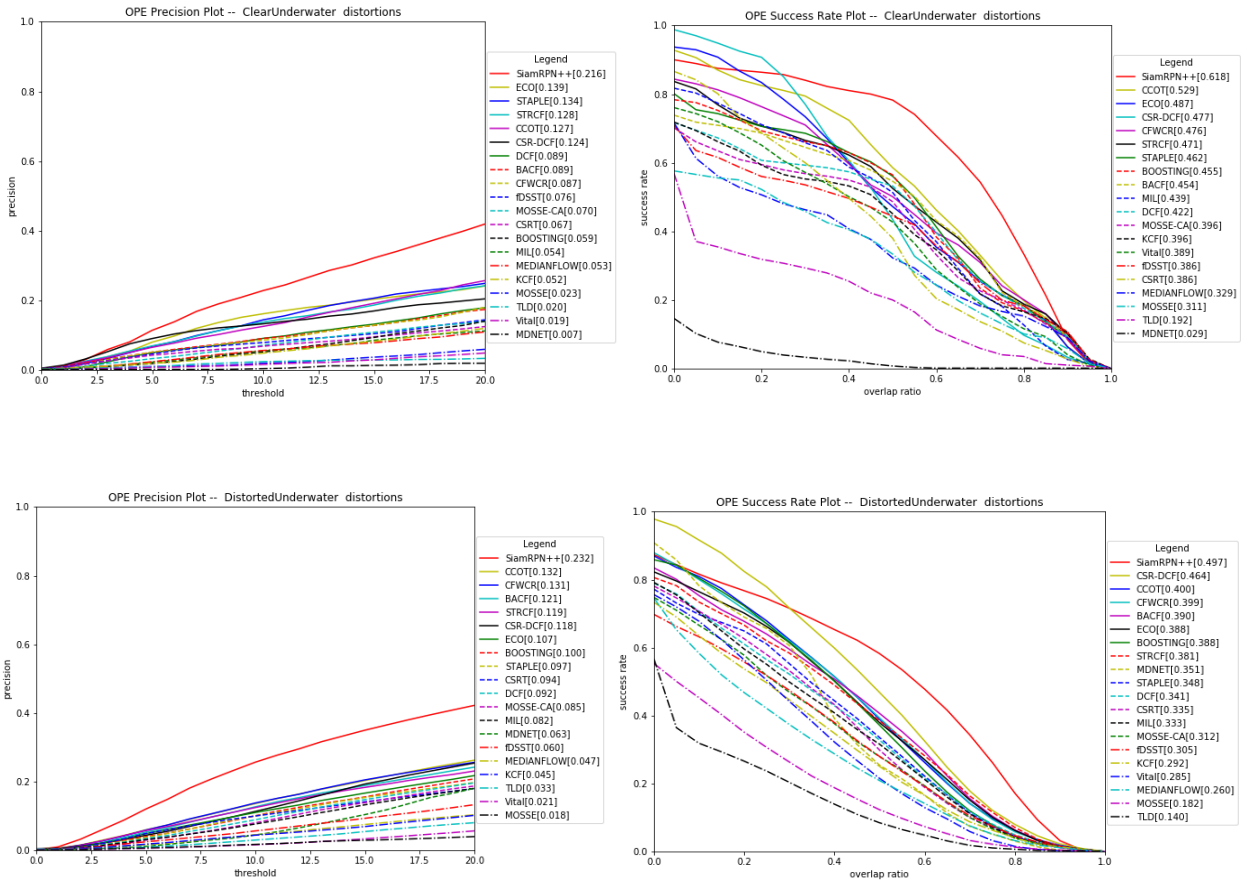


Figure 3-8: OPE precision and success rate benchmark plots on Clear (first row) vs Distorted (second row) underwater visual data. Using their AUC values, trackers are color-coded and ranked from top to least.

Table 3-3 reports the AUC performance of the trackers across the UOT100 dataset and under various types of underwater distortions and visual quality. This represents the data used in generating the benchmark plots. The highlighted blue values indicate the best performance among all benchmarked trackers for each evaluation set.

Table 3-3: Area Under Curve (AUC) results for all trackers across several variations in visual quality of the underwater image. Results presented in this table represent the average scores across the UOT100 dataset.

Precision AUC (Area Under Curve)																				
	BACF	BOOSTING	CCOT	CFWCR	CSR-DCF	CSRT	DCF	ECO	fDSST	KCF	MDNET	MEDIANFLOW	MIL	MOSSE	MOSSE-CA	SiamRPN++	STAPLE	STRCF	TLD	Vital
All	0.117	0.095	0.131	0.126	0.119	0.091	0.091	0.111	0.061	0.046	0.054	0.048	0.079	0.019	0.083	0.23	0.101	0.12	0.032	0.021
Natural	0.115	0.094	0.125	0.122	0.103	0.09	0.087	0.097	0.06	0.048	0.077	0.047	0.077	0.02	0.084	0.208	0.083	0.117	0.033	0.022
Artificial	0.139	0.101	0.18	0.161	0.26	0.097	0.131	0.231	0.075	0.03	0.007	0.053	0.097	0.006	0.078	0.421	0.255	0.145	0.025	0.005
ClearUnderwater	0.089	0.059	0.127	0.087	0.124	0.067	0.089	0.139	0.076	0.052	0.007	0.053	0.054	0.023	0.07	0.216	0.134	0.128	0.02	0.019
DistortedUnderwa	0.121	0.1	0.132	0.131	0.118	0.094	0.092	0.107	0.06	0.045	0.063	0.047	0.082	0.018	0.085	0.232	0.097	0.119	0.033	0.021
BlueLike	0.156	0.12	0.174	0.178	0.146	0.137	0.112	0.154	0.082	0.046	0.074	0.06	0.117	0.026	0.099	0.279	0.123	0.167	0.048	0.025
DarkBlue	0.129	0.083	0.131	0.14	0.115	0.09	0.08	0.083	0.076	0.034	0.008	0.051	0.082	0.011	0.08	0.245	0.072	0.126	0.042	0.03
LightBlue	0.192	0.17	0.233	0.231	0.192	0.204	0.155	0.253	0.089	0.062	0.088	0.072	0.165	0.048	0.124	0.327	0.193	0.226	0.057	0.017
GreenLike	0.13	0.111	0.145	0.146	0.127	0.104	0.104	0.131	0.058	0.052	0.063	0.045	0.086	0.023	0.092	0.253	0.121	0.13	0.032	0.021
DarkGreen	0.152	0.089	0.152	0.138	0.138	0.069	0.113	0.108	0.07	0.045	0.061	0.058	0.059	0.011	0.083	0.26	0.109	0.13	0.033	0.023
LightGreen	0.116	0.123	0.141	0.151	0.119	0.126	0.099	0.146	0.05	0.056	0.063	0.037	0.102	0.03	0.097	0.248	0.129	0.131	0.031	0.019
YellowLike	0.067	0.079	0.102	0.146	0.19	0.02	0.166	0.114	0.016	0.011		0.027	0.035	0.047	0.138	0.248	0.013	0.024	0.008	0.009
Success rate AUC (Area Under Curve)																				
All	0.397	0.396	0.414	0.407	0.465	0.341	0.351	0.399	0.314	0.304	0.298	0.268	0.345	0.197	0.321	0.511	0.361	0.391	0.146	0.297
Natural	0.418	0.415	0.422	0.421	0.468	0.356	0.365	0.404	0.327	0.326	0.338	0.284	0.358	0.212	0.333	0.512	0.363	0.406	0.156	0.321
Artificial	0.214	0.239	0.347	0.294	0.443	0.214	0.23	0.363	0.207	0.115	0.218	0.13	0.237	0.07	0.217	0.502	0.341	0.263	0.066	0.072
ClearUnderwater	0.454	0.455	0.529	0.476	0.477	0.386	0.422	0.487	0.386	0.396	0.029	0.329	0.439	0.311	0.396	0.618	0.462	0.471	0.192	0.389
DistortedUnderwa	0.39	0.388	0.4	0.399	0.464	0.335	0.341	0.388	0.305	0.292	0.351	0.26	0.333	0.182	0.312	0.497	0.348	0.381	0.14	0.285
BlueLike	0.391	0.37	0.397	0.399	0.453	0.326	0.337	0.403	0.309	0.267	0.194	0.257	0.332	0.169	0.283	0.504	0.341	0.389	0.143	0.268
DarkBlue	0.394	0.352	0.377	0.383	0.469	0.302	0.311	0.348	0.3	0.261	0.181	0.244	0.311	0.144	0.25	0.48	0.289	0.372	0.108	0.294
LightBlue	0.387	0.396	0.425	0.422	0.43	0.36	0.373	0.481	0.321	0.276	0.196	0.273	0.361	0.207	0.327	0.537	0.412	0.414	0.19	0.232
GreenLike	0.382	0.396	0.404	0.404	0.457	0.342	0.351	0.405	0.295	0.293	0.335	0.242	0.338	0.193	0.331	0.515	0.362	0.386	0.142	0.282
DarkGreen	0.407	0.358	0.395	0.375	0.496	0.342	0.337	0.391	0.295	0.264	0.527	0.236	0.284	0.134	0.292	0.502	0.354	0.387	0.14	0.245
LightGreen	0.367	0.418	0.41	0.421	0.433	0.341	0.36	0.413	0.295	0.311	0.252	0.246	0.369	0.228	0.354	0.523	0.367	0.386	0.143	0.304
YellowLike	0.385	0.47	0.472	0.509	0.54	0.296	0.466	0.389	0.249	0.3		0.31	0.406	0.332	0.585	0.55	0.305	0.28	0.147	0.259

3.6.5 Performance Benchmarking on CRN-UIE-enhanced UOT100 dataset

To improve the tracking accuracy and precision of the trackers, we use an image enhancement approach to improve the quality of the underwater visual data by eliminating some inherent distortions. Given that trackers are found to have better performance in open-air environments than in underwater, we synthesize clear underwater images using the proposed CRN-UIE. We enhanced the UOT100 dataset to generate UOT100_enhanced and benchmark the performance of the same trackers on the enhanced visual data.

The OPE precision and success rate plots for the enhanced dataset are shown in Figure 3-9. It can be seen that the proposed CRN-UIE GAN model can be used to enhance the performance of tracking algorithms on underwater visual data. A significant and consistent improvement in the precision as well as success rate of the selected trackers on the enhanced visual data can be observed. Figure 3-9 compares benchmark performance on the entire enhanced vs original dataset, while Figure 3-10 compares the benchmark performance on natural vs artificial underwater data.

Consistent and significant increase in the accuracy of all trackers on the enhanced dataset is observed as corroborated by the quantitative AUC results in Table 3-4.

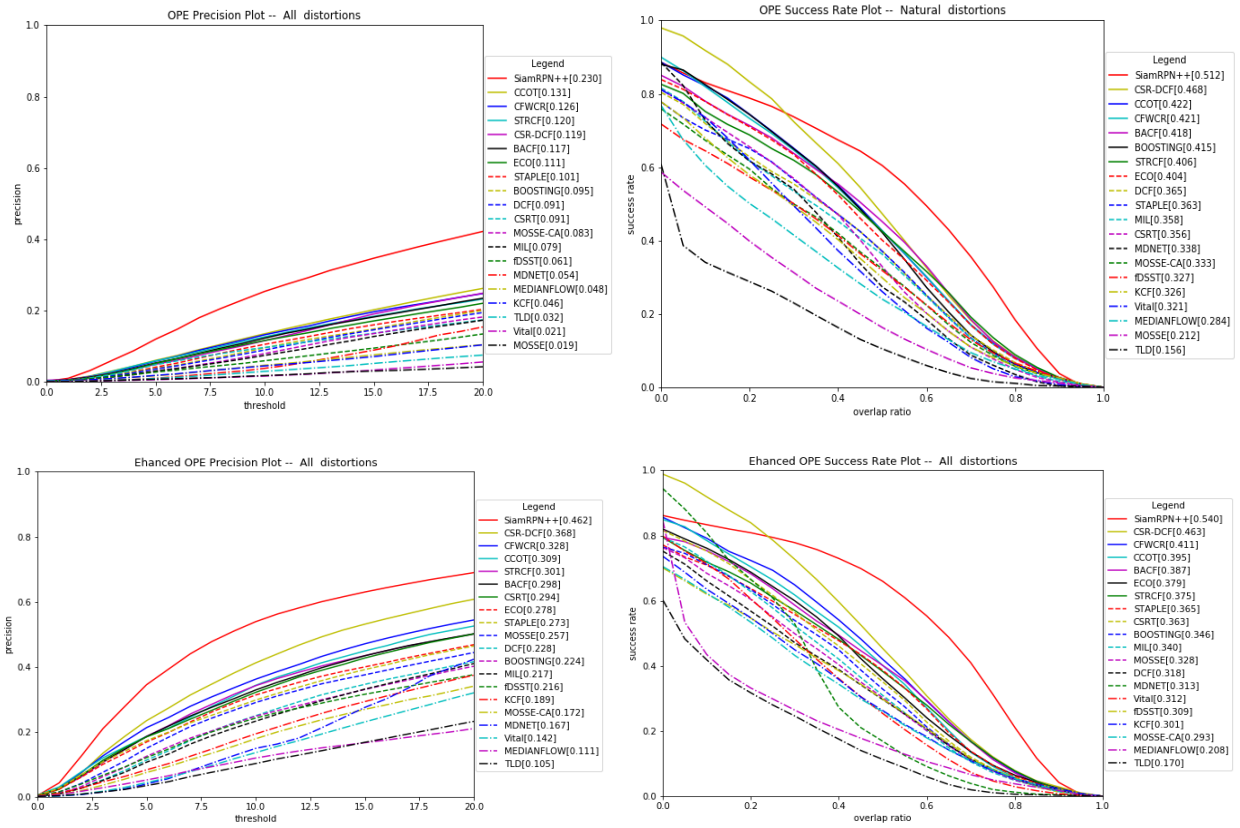


Figure 3-9: Comparative benchmark results on Original (first row) vs Enhanced dataset (second row). Trackers in legend are ranked from top to least performing using their AUC values.

Table 3-4: Comparative tracking benchmark results on distorted and enhanced underwater data. The Area Under Curve (AUC) presented in this table represents the average scores across the UOT100 dataset.

	Precision AUC (Area Under Curve)																			
		BACF	BOOSTING	CCOT	CFWCR	CSR-DCF	CSRT	DCF	ECO	fDSST	KCF	MDNET	MEDIANFLOW	MIL	MOSSE	MOSSE-CA	SiamRPN++	STAPLE	STRCF	TLD	Vital
Non-enhanced	All	0.117	0.095	0.131	0.126	0.119	0.091	0.091	0.111	0.061	0.046	0.054	0.048	0.079	0.019	0.083	0.23	0.101	0.12	0.032	0.021
	Natural	0.115	0.094	0.125	0.122	0.103	0.09	0.087	0.097	0.06	0.048	0.077	0.047	0.077	0.02	0.084	0.208	0.083	0.117	0.033	0.022
	Artificial	0.139	0.101	0.18	0.161	0.26	0.097	0.131	0.231	0.075	0.03	0.007	0.053	0.097	0.006	0.078	0.421	0.255	0.145	0.025	0.005
After GAN enhancement	All	**0.298**	**0.224**	**0.309**	**0.328**	**0.368**	**0.294**	**0.228**	**0.278**	**0.216**	**0.189**	**0.167**	**0.111**	**0.217**	**0.257**	**0.172**	**0.462**	**0.273**	**0.301**	**0.105**	**0.142**
	Natural	**0.308**	**0.232**	**0.328**	**0.328**	**0.34**	**0.28**	**0.232**	**0.275**	**0.207**	**0.196**	**0.207**	**0.112**	**0.221**	**0.246**	**0.179**	**0.468**	**0.262**	**0.303**	**0.116**	**0.148**
	Artificial	**0.22**	**0.151**	**0.142**	**0.326**	**0.616**	**0.414**	**0.189**	**0.301**	**0.291**	**0.127**	**0.088**	**0.098**	**0.183**	**0.353**	**0.114**	**0.408**	**0.366**	**0.289**	**0.013**	**0.036**

	Success rate AUC (Area Under Curve)																			
Non-enhanced	All	**0.397**	**0.396**	**0.414**	0.407	**0.465**	0.341	**0.351**	**0.399**	**0.314**	**0.304**	0.298	**0.268**	0.345	0.197	**0.321**	0.511	0.361	**0.391**	0.146	0.297
	Natural	**0.418**	**0.415**	**0.422**	0.421	0.468	0.356	**0.365**	**0.404**	**0.327**	0.326	0.338	**0.284**	0.358	0.212	**0.333**	0.512	0.363	0.406	0.156	0.321
	Artificial	**0.214**	**0.239**	**0.347**	0.294	**0.443**	0.214	**0.23**	**0.363**	**0.207**	**0.115**	0.218	0.13	**0.237**	0.07	**0.217**	**0.502**	**0.341**	**0.263**	**0.066**	0.072
After GAN enhancement	All	0.387	0.346	0.395	**0.411**	0.463	**0.363**	0.318	0.379	0.309	0.301	**0.313**	0.208	**0.346**	**0.328**	0.293	**0.54**	**0.365**	0.375	**0.17**	**0.312**
	Natural	0.412	0.371	0.417	**0.43**	**0.471**	**0.373**	0.34	0.396	0.322	**0.326**	**0.267**	0.219	**0.363**	**0.341**	0.319	**0.565**	**0.376**	**0.41**	**0.188**	**0.322**
	Artificial	0.176	0.13	0.194	0.25	0.39	**0.279**	0.13	0.227	0.191	0.103	**0.404**	**0.116**	0.138	**0.221**	0.074	0.326	0.262	0.164	0.017	**0.158**

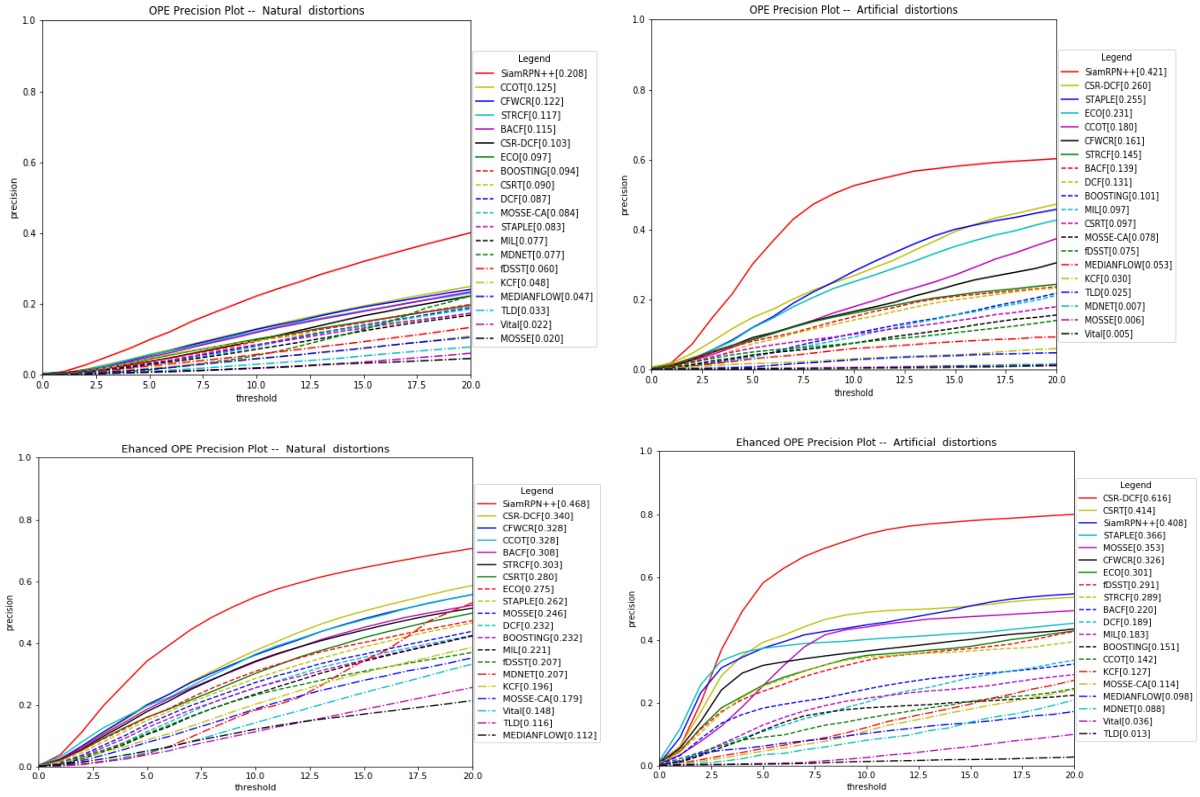


Figure 3-10: Comparing the precision benchmark plots on original (first row) vs enhanced (second row) data (natural vs artificial underwater data). Trackers in legend are ranked from top to least performing using their AUC values.

Further comparative analysis is performed for evaluating the tracking accuracy of CRN-UIE enhanced datasets to enhanced datasets obtained by applying FUnIE-GAN [83] and UGAN [85] to the original UOT100 sequences. Overall, the plots in Figure 3-11 show that enhancing the quality of the underwater visual data using these methods also improves trackers' precision and success rate in underwater environments. However, these results also demonstrate that most trackers perform slightly better on UOT100_enhanced using the proposed CRN-UIE enhancement method, than with FUnIE-GAN and UGAN. SiamRPN++, Vital and TLD tend to perform slightly better on the data enhanced by both FUnIE-GAN and UGAN than CRN-UIE. We observe this trend in benchmark plots for all distortion types. It is equally important to point out that the

performance ranking of the benchmarked trackers remains almost consistent across the various methods.

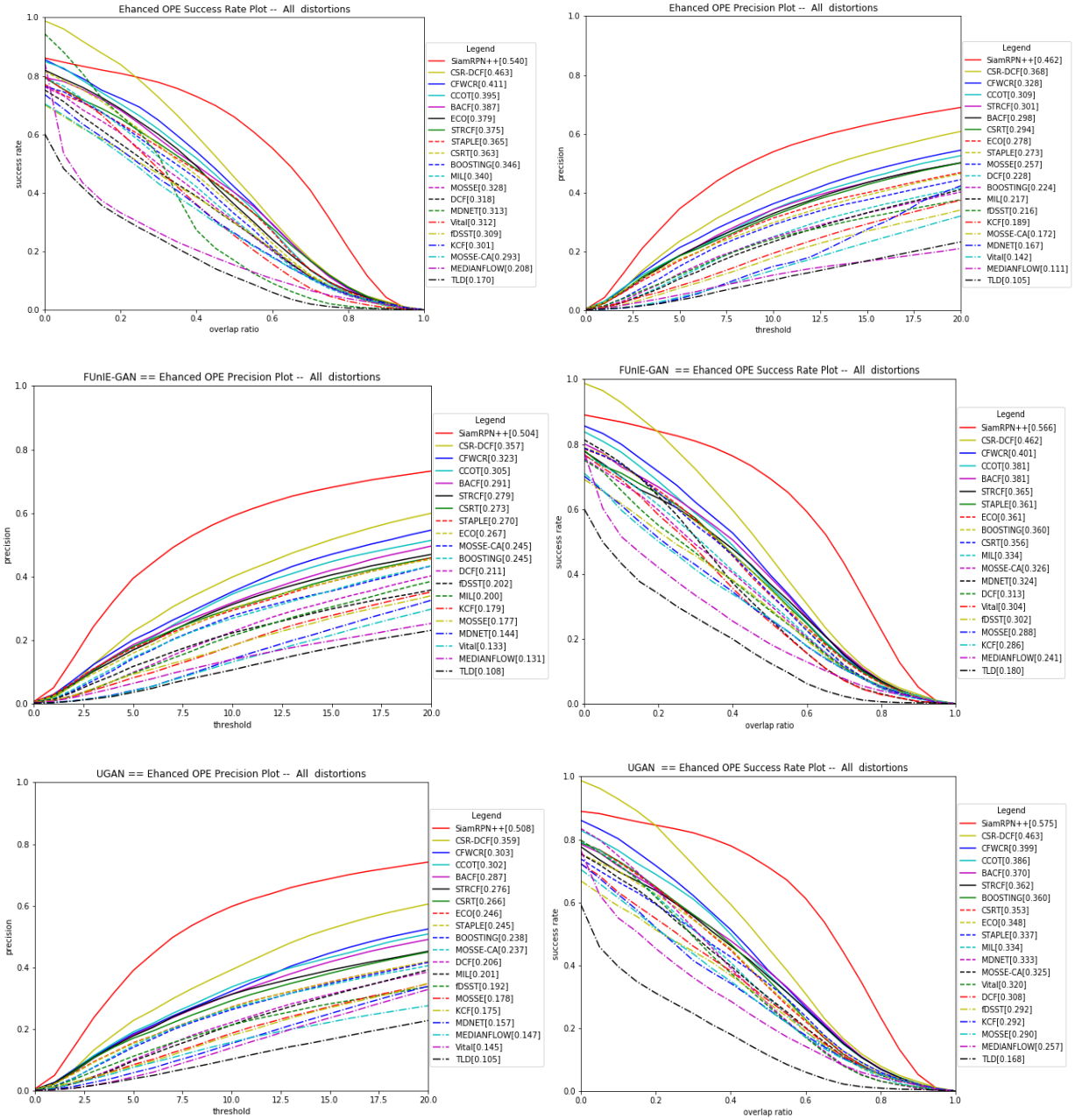


Figure 3-11: Comparative benchmark results – Precision and Success rate plots on data enhanced using the proposed CRN-UIE (first row), FUnIE-GAN (second row), and UGAN (third row). Each row shows the corresponding enhancement method's precision and success rate plots. From the plots, it can be seen that most trackers perform better on underwater data enhanced using CRN-UIE. However, SiamRPN++ tends to do better on data enhanced with FUnIE-GAN and UGAN.

3.7 Summary

In this chapter, the performance of state-of-the-art object tracking algorithms was shown to degrade considerably when tested on underwater environments as opposed to the open-air environments due to the inherent distortions that affect the quality of underwater visual data. To help address this problem, we created a comprehensive underwater object tracking and benchmarking dataset to foster the development of dedicated trackers, well suited for underwater environments and robust to the various inherent distortions. We further proposed a new, improved Generative Adversarial Network model, the CRN-UIE model for underwater image enhancement. Subsequent analysis shows that correcting the underwater distortions by translating the visual data to an enhanced/clear domain using our model significantly improves tracking accuracy in underwater environments. The work done in this paper will benefit the sophistication of applications such as underwater search and rescue operations, homeland and maritime security, deep ocean exploration, underwater robot navigation, and sea life monitoring.

PART 2: DATA AUGMENTATION METHODOLOGIES FOR BUILDING ROBUST AI SYSTEMS

Chapter 4. Non-invasive Computer Vision Techniques for Object Detection and Tracking from Multimodal Camera Sensors

Current methods explored to help with elephant protection and conservation consist of a tracking tag and redirecting animals away from farm areas. The sheer process of tagging animals can cause stress and incur accidental death. The tags have a limited lifetime that could require battery replacement from time to time. More importantly, these tags do not tell if and when the endangered animals are heading towards danger, i.e. if poachers are in the area or not.

This chapter introduces methodologies based on non-invasive computer vision techniques to detect, recognize and track wildlife animals from different camera sensors. It follows three guiding principles: (1) the tracking technique or data collection strategy must not inflict pain or cause discomfort for the animals, and (2) the data collection strategy should not be created in such a way that it tends to change the animal's natural behavioral response to danger.

4.1 Overview

The rise in human-related activities such as poaching and land use has led to the death of over 144,000 elephants, especially in African countries. Therefore, there is an urgent need to use the advantages that accrue from the exponential improvement in technology capabilities to develop new ideas and means to curb this recent surge.

Results obtained from a survey published in [108], estimate a population of 352,271 savannah elephants on study sites in 18 countries. These elephants represent approximately 93% of all savannah elephants in those countries. The elephant population in areas surveyed decreased by an estimated 144,000 from 2007 to 2014 [108]. The primary cause for the massive decline in population is poaching activities that have steadily increased in the last decade. Among the worst-

hit African countries is Mozambique. Surveys led by the Government of Mozambique in partnership with the Wildlife Conservation Society showed an estimated 48% decline in the elephant numbers in the last five years from about 20,000 to 10,300 in 2015. This decline was due to rampant elephant poaching in the country's largest elephant populations [108].

Several methods have been proposed to help address the rising threat to elephants in sub-Saharan Africa and Asia. Some of these methods include: 1) implementing new laws, especially the ban of the ivory trade, 2) coordination with environmental police and rangers to monitor parks, and 3) building bee and electric fences to secure farmlands and residential areas. These are more traditional methods that are still not effective when used alone. Technological advancements in image, vibration, and sound analysis have also been harnessed to provide a paradigm shift in monitoring elephant populations.

Patrick et al. [109] presented a system to automatically classify African elephant vocalizations using a hidden Markov model (HMM). HMMs are good choices for this task since they can model the temporal and spectral differences between similar vocalizations and are the most popular model used in speech processing. Payne et al. [110] used the information from the calling patterns as indicators of group size and composition as a basis for creating an acoustic monitoring system.

Wood et al. [111], rather than use sounds, proposed a technique to detect and count elephants using recordings from seismic sensors. An accuracy of 82% was obtained when differentiating between species using spectral and temporal clues between species. These approaches are very prone to noise, and elephants have to be close to the seismic sensor to obtain reasonable results, hence the need to develop more efficient and robust solutions to overcome these challenges.

This chapter presents new contributions for addressing the issues identified in existing methods and are as follows:

a) Introduce a feature-based approach for elephant localization in images.
b) Introduce a shadow detection algorithm for segmenting elephant shadows from drone footage.
c) Present a deep learning approach for elephant detection in visible and infrared images.
d) Present a baseline elephant tracking algorithm using simple Kalman filters.

4.2 Feature-based Elephant Detection

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps and, in some cases, leading to better human interpretations [112].

Image descriptors provide a compact representation that summarizes the contents of an image region. Three descriptors are explored for our binary classification problem, this includes i) histogram of oriented gradients (HOG), ii), local binary patterns (LBP), and iii) scale-invariant feature transform (SIFT).

4.2.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) descriptor attempts to construct a more detailed characterization of the spatial structure with a small image window. The HOG descriptor consists of a collection of normalized histograms computed over spatially offset patches; the result is a descriptor that captures coarse spatial structure but is invariant to small local deformations [113]. The process of computing HOG descriptor involves 4 basic steps. First, the image gradients' orientation and amplitude are computed at every pixel in a 64 x 128 window [114]. The orientation is quantized into 9 bins spread over the range $0\text{-}180^0$. The 64 x 128 detector region is divided into a regular grid of overlapping 6x6 cells. A 9D orientation histogram is computed within each cell,

where the contribution to the histogram is weighted by the gradient amplitude and the distance from the center of the cell so that more central pixels contribute more. The descriptors are concatenated and normalized for each 3x3 block of cells to form a block descriptor. These block descriptors are then stacked together to form the HOG descriptor. HOG descriptors are invariant to illumination (after normalization), the spatial and orientation sampling densities can be tuned for different applications and are also rotation and translation invariant [114].

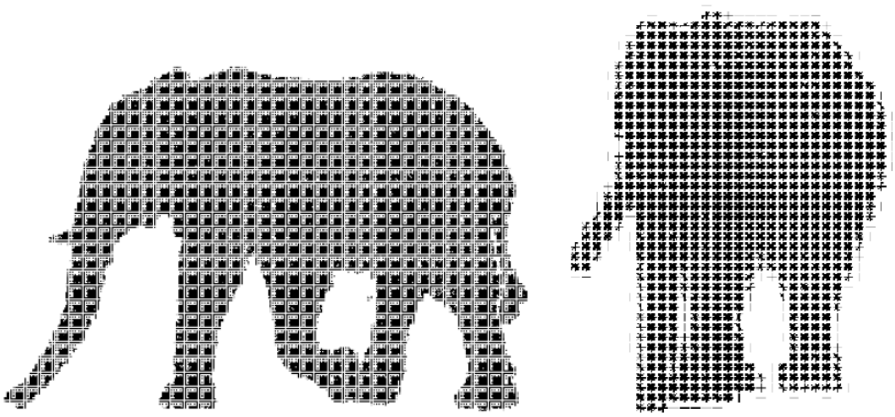


Figure 4-1: Sample visualizations of Histogram of Oriented Gradients (HOG) feature descriptor for elephants

4.2.2 Local Binary Patterns

LBP is a type of descriptor that is based on image texture. Texture is an important characteristic of many kinds of images that range from multispectral remotely sensed data to microscopic images [115]. An image's texture may provide information about the physical properties of objects such as smoothness or roughness or differences in surface reflectance such as color [116]. The local binary pattern descriptor was introduced by Ojata et al. [117]. It was based on the assumption that texture has locally two complementary aspects, a pattern, and its strength. LBP labels the pixels of an image by considering the properties of the neighborhood surrounding each pixel. The LBP operator can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis [115]. It has become one of the most widely used descriptors because

of its resistance to lighting changes, its invariance against monotonic gray level changes, low computational complexity, and ability to code fine details.

The operator works in a 3x3 pixel block of an image. The pixels in this block are thresholded by its center pixel value, multiplied by the powers of two, and then summed to obtain a label for the center pixel.

An example of computing LBP in a 3x3 neighborhood is given below

6	5	2
7	6	1
9	8	7

1	0	0
1		0
1	1	1

1	2	4
128		8
64	32	16

Pattern = 11110001

LBP = 1 + 16 + 32 + 64 + 128 = 241

As the neighborhood consists of 8 pixels, a total of 2^8 = 256 different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood [118]. An example of an LBP image and histogram is shown if Figure 3

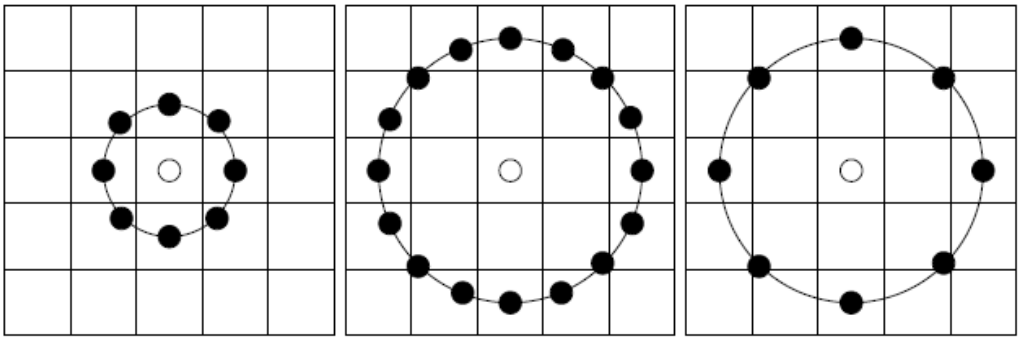


Figure 4-2: Illustration of LBP with different neighborhoods settings. The circular (P=9, R=1), (P=16, R=2) and (P=8, R=2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel.

Figure 2: The circular (P = 8, R =1), (P = 16, R = 2) and (P = 8, R = 2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel

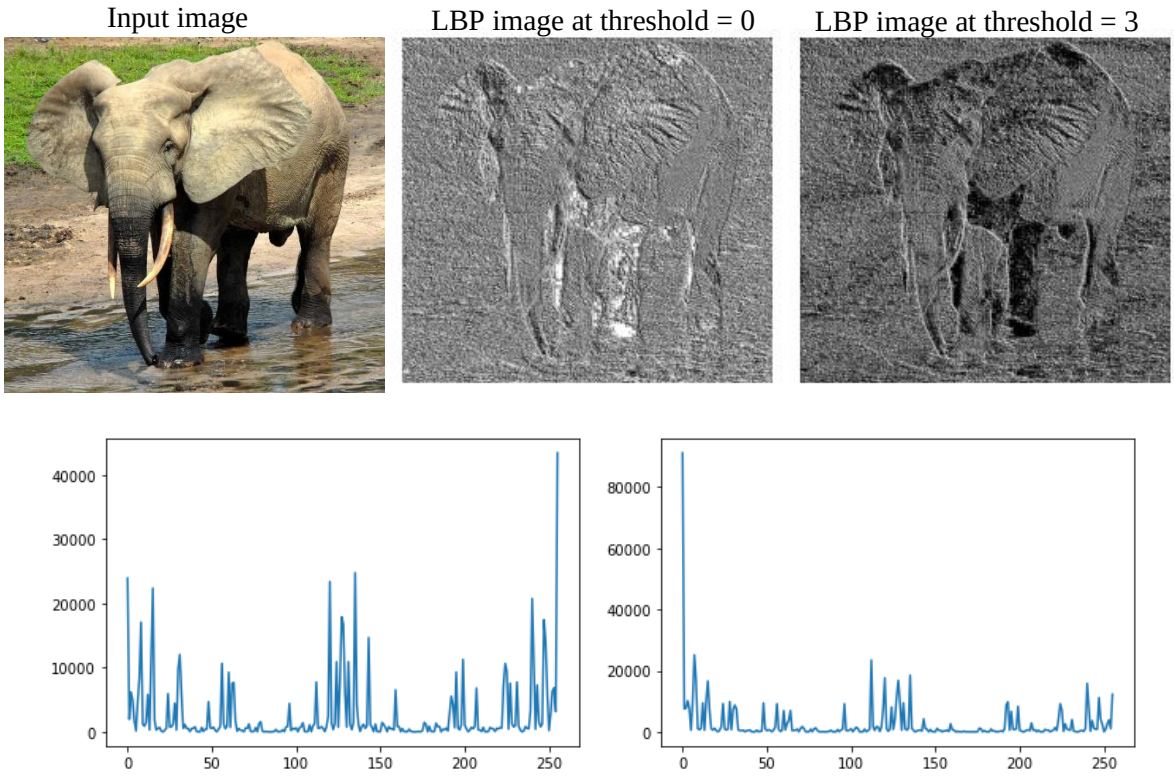


Figure 4-3: Visualization of LBP image for elephants. The first row shows an input image and its corresponding LBP image at a threshold of 0 and 3. The second row shows the histograms at a threshold of 0 and 3 respectively.

4.2.3 Scale-Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) characterizes the image region around a given point. The SIFT feature descriptor is invariant to uniform scaling, orientation, illumination changes and partially invariant to affine distortion. It can robustly identify objects even among clutter and partial occlusion, making it very suitable for applications involving image matching, object recognition, image stitching, robotic mapping, navigation and video tracking.

SIFT keypoints of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors [119]. To compute the SIFT descriptors, the following steps are involved:

Scale-space Extrema Detection: Laplacian of Gaussians (LoG) are found for the image with various σ values. LoG acts as a blob detector which detects blobs of various sizes due to changes in sigma. The local maxima across different scale and space is computed to form a list of (x,y, σ) values which indicates a potential keypoint at (x,y) at σ scale [119] [120].

Because LoG is computationally intensive, SIFT algorithm uses Difference of Gaussians which is an approximation of LoG. An image is blurred, and the difference of Gaussian is computed with two different σ. This process is done for different octaves of the image in a Gaussian Pyramid represented below

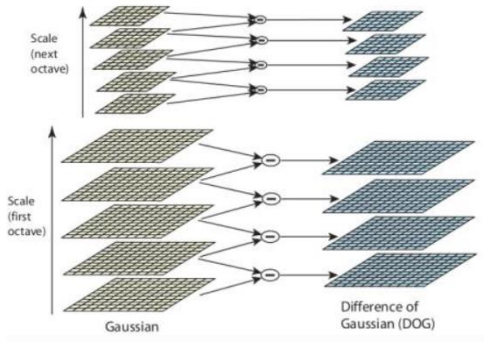


Figure 4-4: Illustration of a Gaussian pyramid [119]

After computing the Difference of Gaussian (DoG), images are searched for local extrema over scale and space.

Keypoint Localization: After computing potential keypoint locations, these points are then fitted to the nearby data to find the accurate location, scale, and ratio of principal curvatures (computed using a 2x2 Hessian matrix). This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. Taylor series expansions of scale space are used to obtain more accurate locations of extrema and if the intensity at this extrema is less than a threshold value, it is rejected [119], [121].

Orientation Assignment: Each keypoint is assigned one or more orientations based on local image gradient directions. This step helps in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

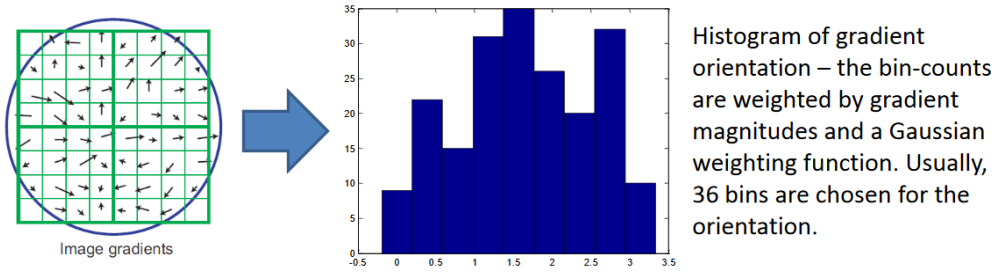


Figure 4-5: Visualization of orientation assignment with the corresponding histogram of gradients [119]

Keypoint Descriptor: A descriptor vector for each keypoint is computed such that the descriptor is highly distinctive and partially invariant to the remaining variations (other than location, scale and rotation) such as illumination, 3D viewpoint etc. This is performed on the image closest in scale to the keypoint's scale. A 16x16 neighborhood around each keypoint is taken. It is then divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histograms are created which gives a total of 128 bin values [120], [121]. This is represented as a vector to form the keypoint descriptor. Keypoints between two images are matched by identifying their nearest neighbors.

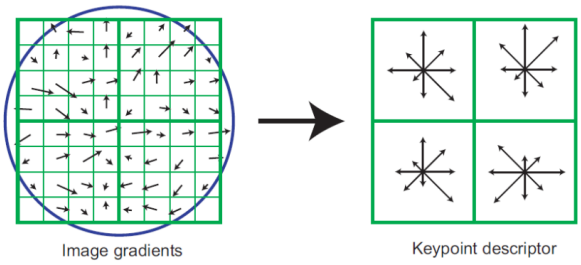


Figure 4-6: Illustrative example of keypoint descriptor [119]

4.2.4 Flowchart for Feature-based Detection

Figure 4-7 illustrates the steps followed in training a Support Vector Machine (SVM) [122] classifier for elephant detection using the HOG, SIFT, and LBP-based features. The result for each feature is computed, and performance is compared using different kernels and hyperparameters for finetuning the SVM classifier.

A total of 4400 elephant images (positive samples) and 11,698 non-elephant images (negative samples) were used for the training set. The training images used were obtained from the google search engine and the ImageNet database [123]. Figure 4-8 shows a few sample images from the dataset.

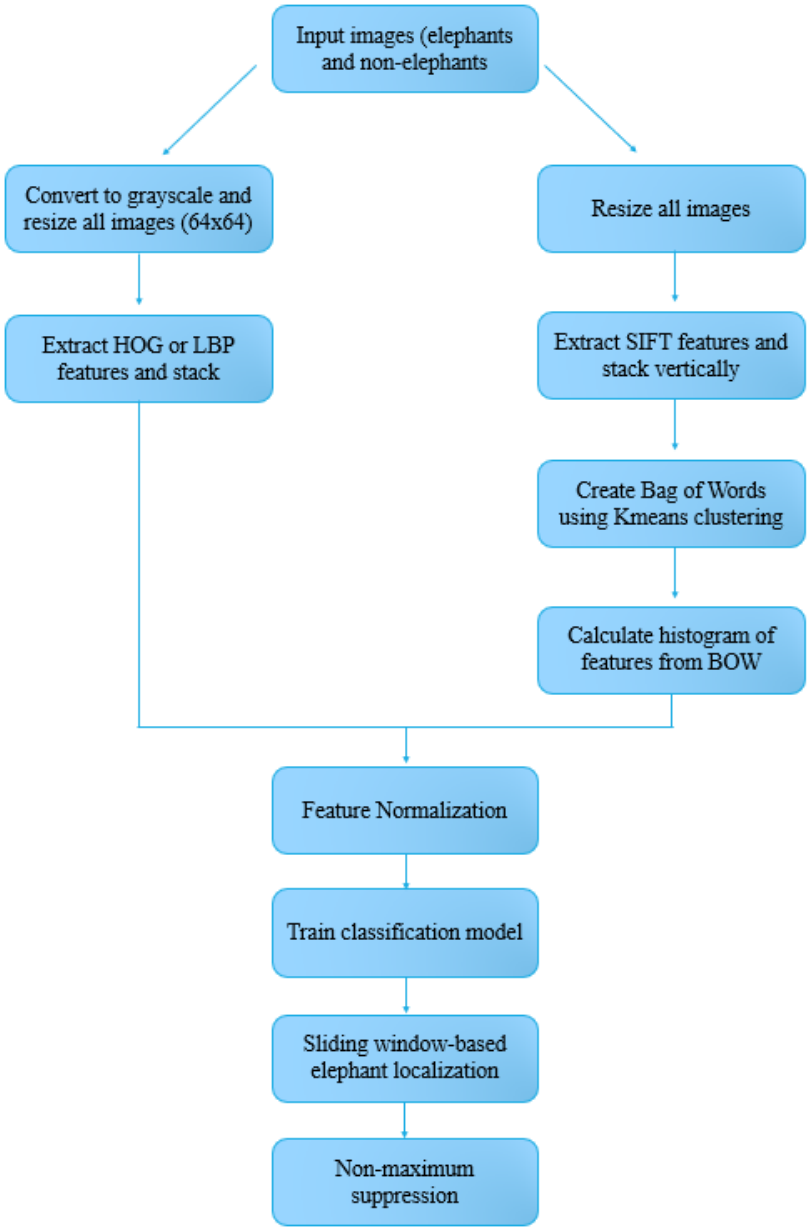


Figure 4-7: Illustrative chart and steps for training a Support Vector Machine Classifier using HOG, LBP, and SIFT-based features. Bag of Words is created by applying K-means clustering to the extracted features. This helps in grouping similar features together.

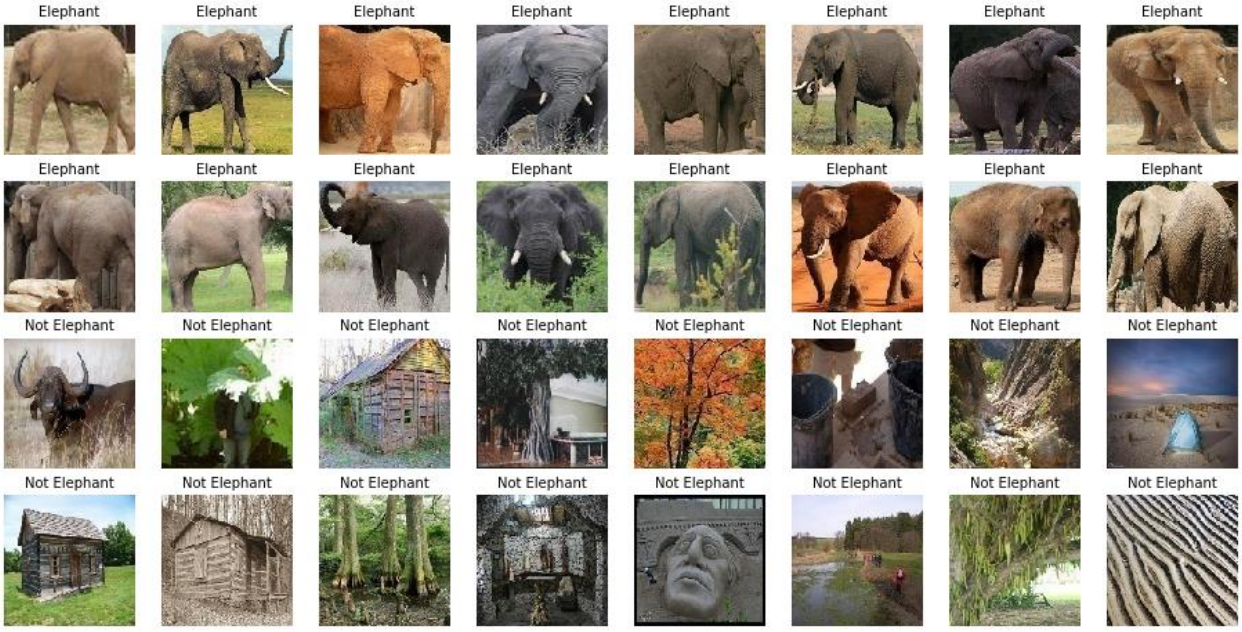


Figure 4-8: Sample training data used to train the binary classifier. The training data contains elephant images and random images which represent negative samples. Each image is resized to a size of 256x256 with elephants centered in the image.

4.2.5 Evaluation Metrics

Evaluating a model is one of the most important tasks in machine learning which delineates how good the predictions are. Various methods can be used to evaluate the performance of a model. Some measures include:

- Accuracy is the ratio of correctly predicted observations to the total observations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad 4.1$$

- Precision, also called positive predictive value, is the ratio of correctly predicted positive observations to the predicted positive observations. High precision relates to the low false-positive rate.

$$Precision = \frac{TP}{TP + FP} \quad 4.2$$

- Recall: Also referred to as sensitivity is the ratio of correctly predicted positive observations to all observations in the actual class - true (elephants).

$$Recall = \frac{TP}{TP + FN} \qquad 4.3$$

Where,

True Positives (TP): These are correctly predicted positive values, which means that the value of the actual class is elephants, and the value of the predicted class is also elephants.

True Negatives (TN): It measures the proportion of negatives that are correctly identified as such. They are the correctly predicted negative values, which means that the actual class value is false (not elephant), and the value of the predicted class is false (not elephant). It is also referred to as specificity.

False Positives (FP): Commonly referred to as false alarm, it is a result that indicates a given condition exists when it does not. This occurs when the actual class is false (not elephant), but the predicted class is true (elephant).

Table 4-1: Classification results for elephant vs non-elephant image patches. The results presented in this table are average scores across the test set. HOG gives the highest precision and accuracy scores while SIFT-based feature gives the best recall.

Method	Precision (%)	Recall (%)	Accuracy (%)
HOG	**95.64**	89.25	**95.93**
SIFT	83.92	**94.83**	86.00
LBP	86.04	88.52	82.46

Table 4-1 shows the robustness of structural-based features (HOG) for classification tasks. Typically, a model with high precision and recall scores is preferable with a smaller difference between both scores. Although SIFT features provide the best recall, its precision is significantly lower, making it less preferred.

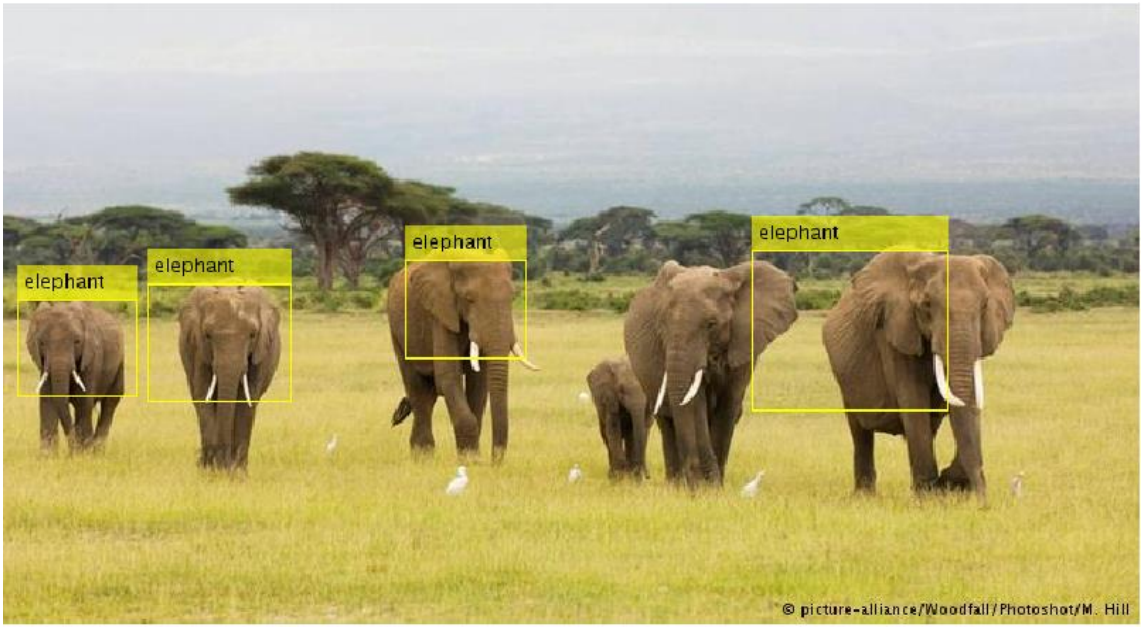


Figure 4-9: Elephant detection using a combination of the sliding window approach and binary classification using HOG features

4.3 Shadow-based Elephant Detection

One drawback of feature-based elephant detection is its lack of robustness to partial or heavy occlusions. Shadow detection provides a means for detecting elephants when there is a light source that casts the shadow of the elephants. Shadow detection has been used to estimate the positions of objects in videos and images. Wang et al. [123] proposed to simultaneously use appearance, shadow, and motion information for pedestrian detection. Russell et al. [124] extracted shadow regions using the pixel-gradient magnitude of intensity measurements from sub-divided image blocks for real-time human shadow detection. Xiang et al. [125] proposed a normalized local intensity ratio via a gaussian mixture model to eliminate the influence of illumination changes on detected shadows.

To achieve elephant shadow detection from drone footage, chromaticity feature in the LAB color space is considered. Chromacity is an objective specification of the quality of color regardless of

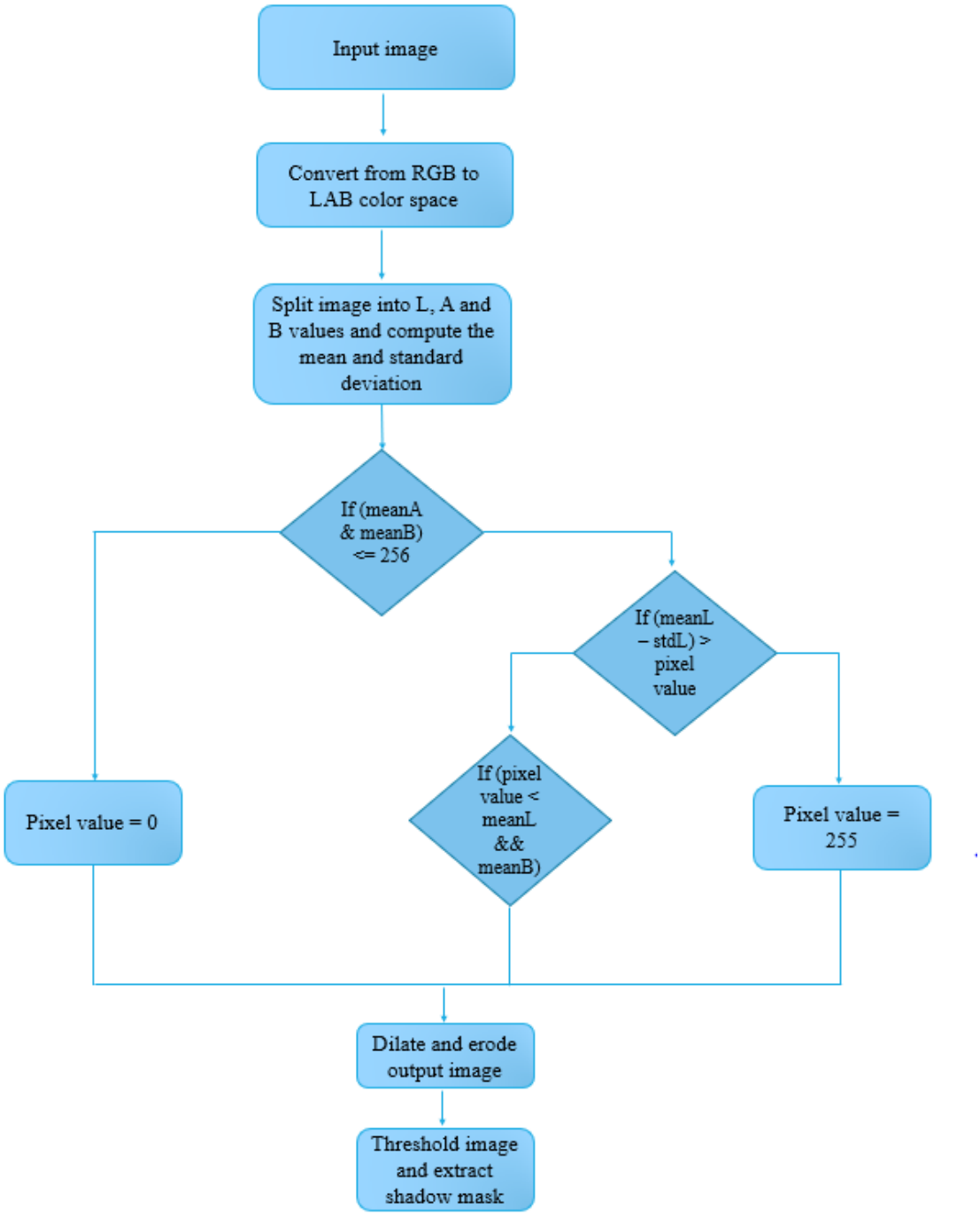


Figure 4-10: Flowchart for the shadow detection algorithm. meanA, meanB, and meanL are the mean of the pixels in the A, B, and L channels of the LAB color space. stdL is the standard deviation of pixels in the L channel and pixel value is a user defined value that helps filter out noisy detections.

its luminance. It consists of two independent parameters, hue and colorfulness [126]. Color features are used based on the assumption that regions under a shadow become darker but retain their chromacity e.g. a green pixel covered by shadow becomes dark-green which is darker than green but has the same chromacity [126]–[128]. Figure 10 shows the overall flowchart for shadow detection and Figure 11 shows some of the results obtained using our algorithm.

Original images Predicted shadow masks

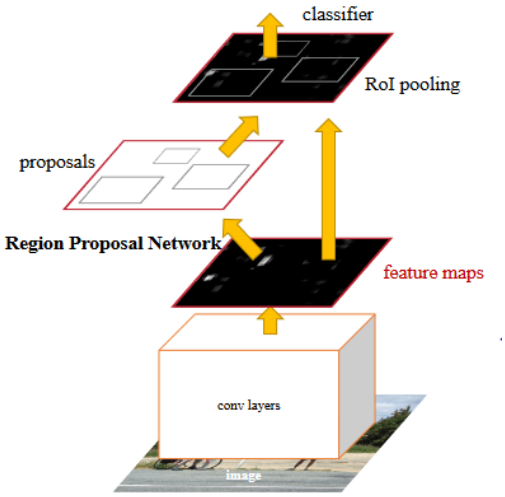
Figure 4-11: Results of the shadow detection algorithm from drone footage of elephants. Images in the left column are original input images from drone footage, while images in the right column correspond to show detection from the image. Shadow detection is important because elephant shadows are more easily detectable from aerial views than the elephants themselves.

4.4 Deep Learning-Based Elephant Detection

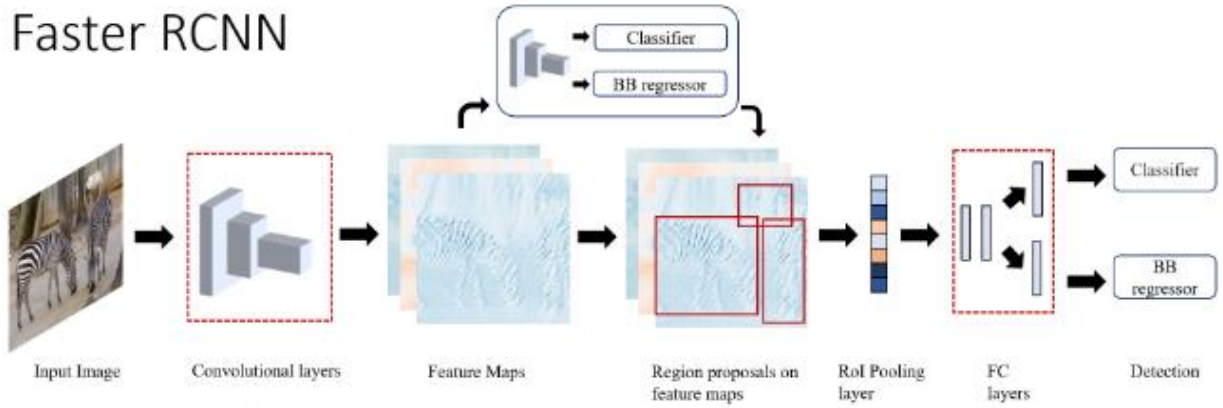
Sliding window-based object detection methods exhibit inherent challenges including high computation requirements, very slow inference time, non-robustness to partial or severe occlusion, and they are also not scale-invariant. Shadow-based detection only works in daytime with a single light source casting the shadows. It also doesn't work for different camera positions or camera sensors e.g., infrared sensors. Deep learning-based object detectors harness convolutional neural networks for dynamic feature extraction. Current architectures for object detection have achieved state-of-the-art performance for hundreds of object categories hence these architectures are benchmarked on a sample dataset for elephant detection to address the challenges observed in the traditional approaches explored.

Faster RCNN: Ren et al. [129] proposed a fully convolutional network as a regional proposal network (RPN) that takes an input image and automatically outputs a set of candidate windows with various scales and aspect ratios. Each window is associated with an *objectness* score which determines the probability of an object within that windowed region. To account for differences in object sizes, Ren proposed anchor boxes that are multiple reference bounding boxes of different aspect ratios. These anchor boxes are regressed over the image to localize an object.

Figure 4-12 shows the architecture of Faster RCNN. The input image is passed through a CNN feature extractor to obtain a set of feature maps. These maps are forwarded to the RPN layer to predict a set of bounding boxes and classes. The selected proposals are mapped back to the feature maps obtained from the CNN layer in the ROI pooling layer and fed to a fully connected layer to output the final bounding boxes and corresponding classes.



(a)



(b)

Figure 4-12: (a) The RPN module proposed by Ren et al. It is responsible for generating region proposals by serving as an attention module that guides the base model (Fast RCNN) on where to look for objects in an image [129]. *(b) the detailed diagram of the Faster RCNN architecture* [130].

Yolov4: Bochkovskiy et al. [131] proposed a significant number of improvements to the You Only Look Once (YOLO) model architecture proposed by Redmond et al. [132] for single-stage object detection. These improvements include utilizing the concept of i) 'Bag of Freebies,' i.e., methods that only increase training time and do not affect inference time, and they are data augmentation techniques, regularization methods, class label smoothing, CIoU-loss [133], cross mini-batch

normalization (CmBN), self-adversarial training, cosine annealing scheduler [134] and other tricks for improving model training, and ii) 'Bag of Specials' which are added to the network to improve its representational capacity. This includes adding Mish activation [135], cross-stage partial connections (CSP) [136], SPP-Block [137], PAN path aggregated block [138], multi-input weighted residual connections (Mi-WRC) etc. A genetic algorithm is also used for hyperparameter search.

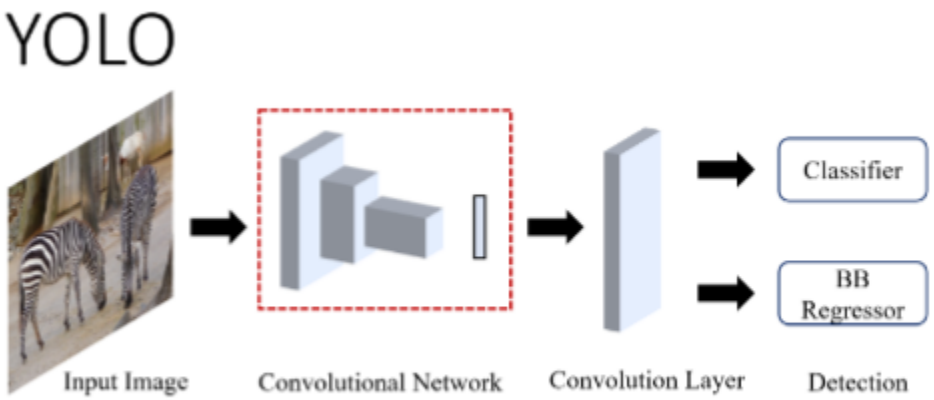


Figure 4-13: Simplified model architecture for YOLO single-stage object detector [130]

4.4.1 Experiments and Results

The performance of YOLOv4 and Faster RCNN were evaluated on a manually annotated dataset with 6 categories of images. The accuracy and speed of predictions are computed, and the robustness of each detector is tested on various types of weather conditions potentially encountered in a wildlife environment. The confidence score is set to a threshold of 0.5, and the IoU threshold was also set to 0.5. These high thresholds help to reduce false positives in the predicted bounding boxes. Performance evaluation for both models was carried out on an Nvidia GTX 1060 6G desktop computer.

YOLOv4
100%
100%
100%
37.5%
Faster RCNN
80%
80%
75%
50%
Night-time
YOLOv4
69%
22%
89%
28%
Faster RCNN
100%
68%
89%
76%
Crowded scene
YOLOv4
100%
100%
94%
100%
Faster RCNN
90%
100%
94%
100%
At far distances

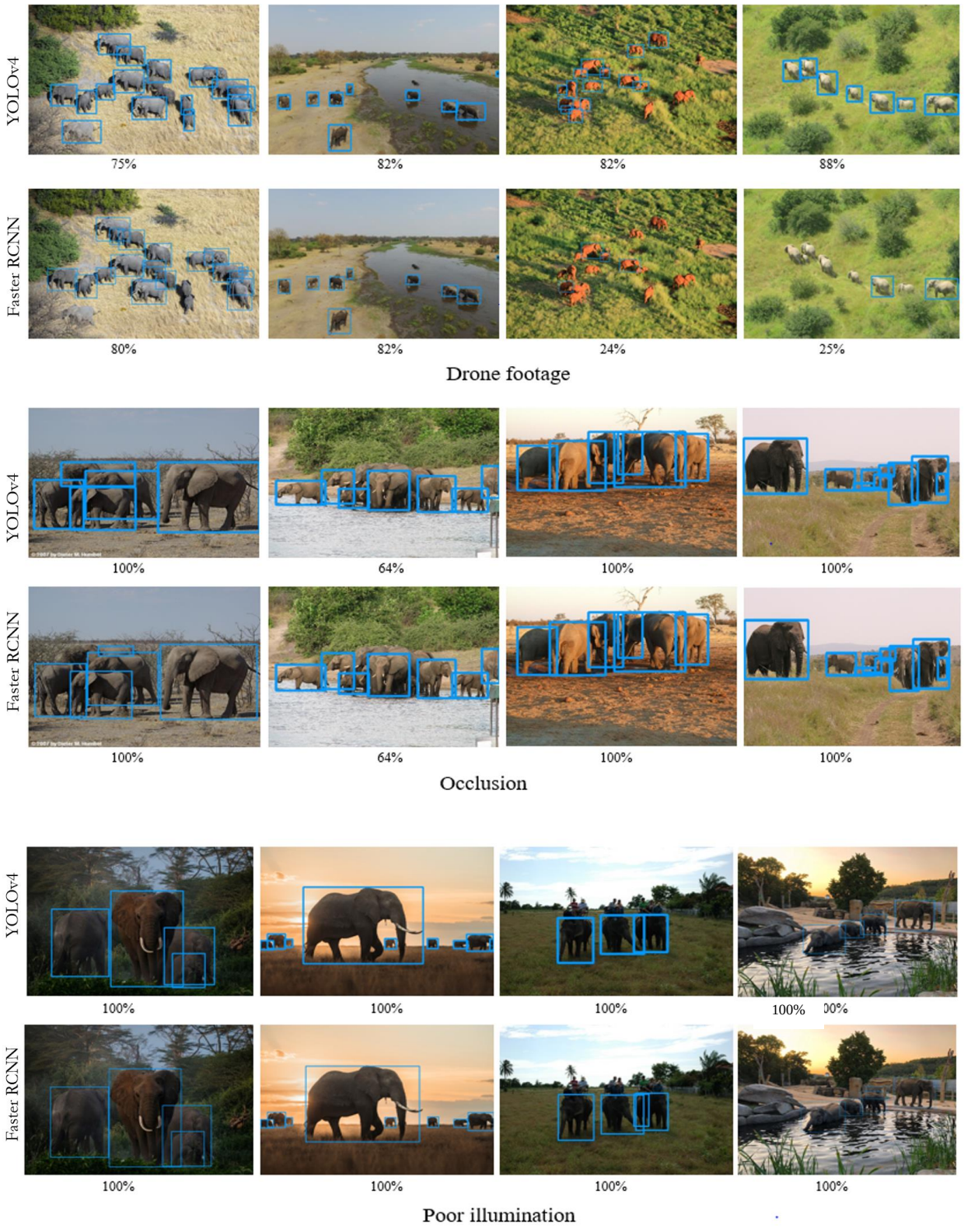


Figure 4-14: Performance benchmarking for YOLOv4 and Faster RCNN object detectors under varying environmental and weather conditions. This evaluation also provides insight on the effectiveness of single stage (YOLO) and two stage (faster RCNN) detectors. The percentages represent the accuracy of the detector, and it is a ratio of the number of localized elephants to the actual number of elephants in the scene.

Figure 4-14 presents some sample results for both object detectors under varying conditions. The performance of YOLOv4 and Faster-RCNN are comparable under most conditions, but the former seems to significantly degrade when the scene is crowded with little to no space between each candidate proposal. This shows that Faster RCNN's RPN architecture for automatically generating region proposals is more effective than anchor boxes used in YOLO for crowded scenes. However, the number of false positives seems to be higher than YOLO. Table 4-2 shows the runtime for both models. YOLOv4 is ~3.5x faster than Faster RCNN, making it preferable to run on simpler devices, like a Raspberry Pi, while achieving competitive results.

Table 4-2: Average inference time for YOLOv4 and Faster RCNN running on an Nvidia GTX 1060 GPU.

Model	**Average inference time**
YOLOv4	34ms
Faster RCNN	119ms

4.5 Kalman Filter-based Tracking of Elephants in Videos

Kalman filtering is a linear quadratic estimation algorithm which uses a series of measurements observed over time, containing statistical noise and other inaccuracies and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement, by estimating a joint probability distribution over the variables for each timeframe [139]. If all noise is Gaussian, the Kalman filter minimizes the mean square error of the estimated parameters [140]. The following assumptions are made when designing our Kalman filter tracker [141]:

- The noise is not time-correlated.
- The modeled system is linear. This means that the system's state at time k can be modelled in the form of the state of the system at the previous time k-1 pre-multiplied by a matrix.

- The system noise is Gaussian in nature i.e., the noise amplitude can be modeled using mean and covariance.

The Kalman filter addresses the general problem of trying to estimate the state $x \in \mathcal{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Fx_{k-1} + Bu_k + w_{k-1} \qquad 4.4$$

with a measurement $z \in \mathcal{R}^n$ to model the relationship between the current state and actual measurements.

$$z_k = Hx_k + v_k \qquad 4.5$$

F is an n x n matrix known as the state transition matrix that relates the state at the previous time step (k-1) to the state at the current step (k) in the absence of process noise.

B is an n x l matrix known as the control input transition matrix that relates the optional control input $u \in \mathcal{R}^l$ to the state x.

H is an n x m matrix known as the measurement transition matrix that relates the state to the measurement zk, where zk is a measurement vector that considers the estimated state and measurement noise.

w_k and v_k represent the process (from the irregular movement of the target, which is in this case, the elephant) and measurement noise (from camera or detection algorithm), respectively. They are assumed to be independent of each other, with normal probability distributions

$$p(w) \approx N(0, Q) \; and \; p(v) \approx N(0, R) \qquad 4.6$$

Where Q and R are the process noise covariance and measurement noise covariance, respectively. The values of Q and R may randomly change in each timestamp of the process i.e. between time k-1 and k.

It is important to note that the better the estimate of the noise parameters, the better the predictions obtained

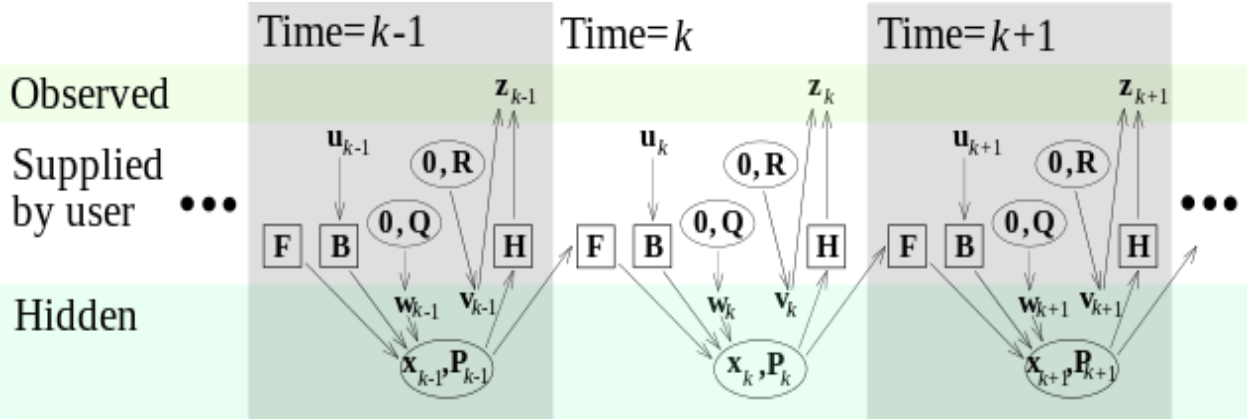


Figure 4-15: Model underlying the Kalman filter. Squares represent matrices and ellipses represent multivariate normal distributions (with the mean and covariance matrix enclosed). Unenclosed values are vectors [142].

4.5.1 Kalman Filter Algorithm

The flowchart below illustrates the computation of the Kalman filtering algorithm

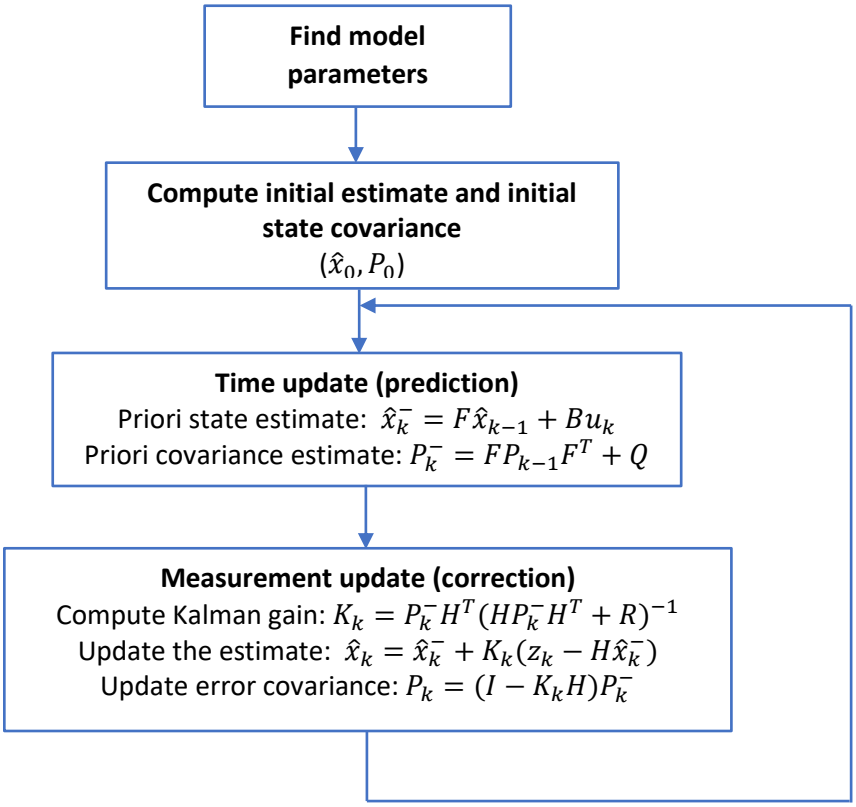
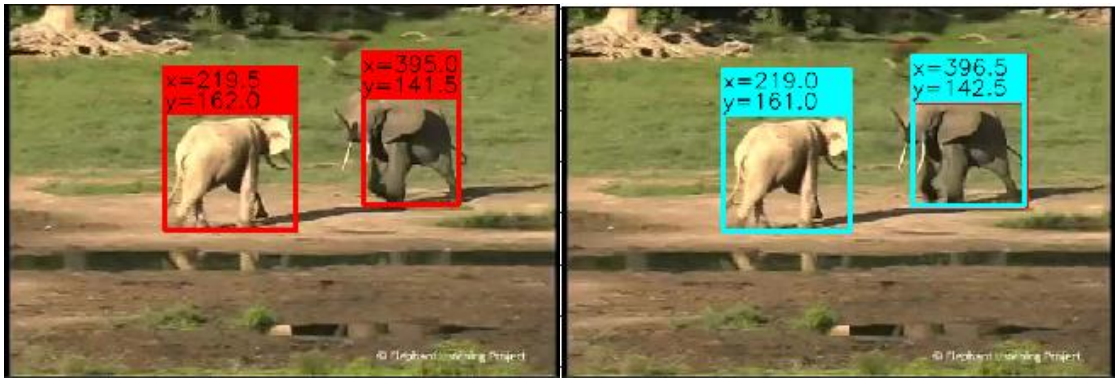
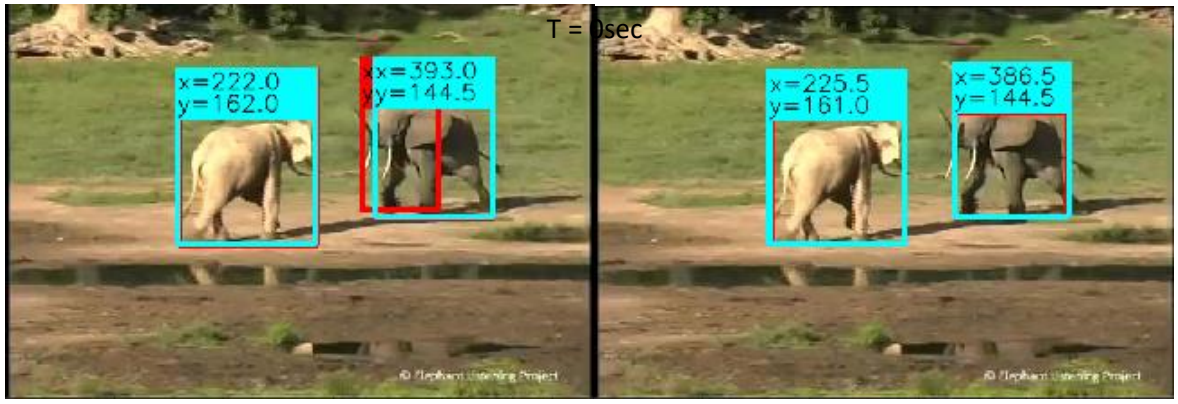


Figure 4-16: Illustrative flowchart of the Kalman Filter-based elephant tracker implementation.

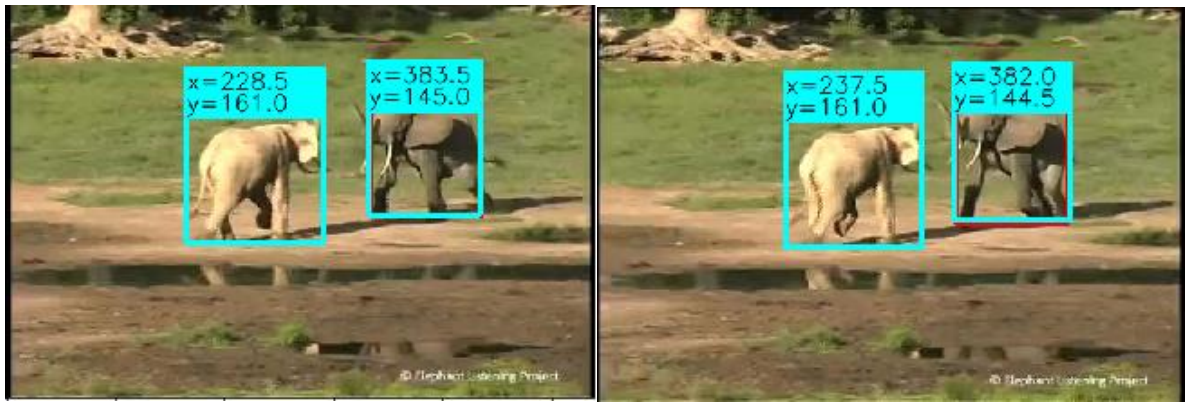
Where F = $\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$; $B = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$; $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$



T = 0sec



T = 2sec



T = 4sec

T = 6sec

Figure 4-17: Elephant tracking results of the Kalman-filter-based tracker. The frames shown in this figure are extracted from a video sequence at multiple time stamps. The red bounding boxes are the ground-truth bounding boxes, while the light-blue boxes represent the predicted bounding boxes for the respective objects of interest.

Figure 4-17 shows some sample frames with predicted and ground truth boxes overlayed. It is evident that Kalman filters are susceptible to noise and often fail when the target's movement is erratic. This is depicted in the failed predictions at T=6 seconds hence part of the future work is to build robust tracking systems for wildlife in their natural habitats. The algorithms should efficiently model the pattern of movements observed in several animals in the design of architecture and training frameworks.

4.6 Summary

In this chapter, feature-based descriptors were used for classification of elephants and non-elephant images in a curated dataset. The accuracy, recall, and precision scores computed shows that structural-based feature descriptor like the Histogram of Oriented Gradients (HOG) outperformed texture-based LBP and SIFT feature-descriptors. A deep learning-based approach for elephant detection was also explored and the effectiveness of each detector under various environmental conditions was benchmarked. Furthermore, the bounding boxes obtained from the sliding window detection algorithm serves as input points to our Kalman filter-based tracking algorithm to track multiple elephants over successive video frames. The results obtained for both the detection and tracking tasks provides the needed baseline on which to build upon and provides various factors to be considered in developing more robust approaches.

Chapter 5. A Unified Approach to Pose Estimation in Mammals using Noisy Labels

While several standardized datasets and state-of-the-art algorithms have been proposed to improve the human pose estimation task, animal pose estimation is still a growing field with several challenges such as lack of large-scale annotated datasets, the wide range of pose variation animals exhibit, and the diversity of animal species. Publicly available benchmark datasets for pose estimation in mammals often have different defined skeletal structures hence limiting their generalization ability to other mammals outside the distribution of the training and test datasets.

This chapter explores innovative solutions to these challenges through the following contributions: 1) Standardization of the skeletal structure across publicly available mammals pose estimation datasets, 2) performance comparison of transformer networks to other state-of-the-art approaches for animal pose estimation, and 3) introduction of a consistency-dependent pseudo-labeling strategy for improving model generalizability to diverse animal species. Furthermore, a novel large-scale manually annotated elephant landmarks dataset is introduced for evaluating the performance of proposed models on a complex landmark prediction task. This dataset contains over 1300 labeled images with 33 landmarks, including the elephant trunk, tusks, and ears. Experimental simulations show model learning from both partly and fully annotated data improves the prediction accuracy. Additionally, training with carefully generated pseudo-labels boosts the generalizability of models to unseen animal species with slightly different skeletal structures.

5.1 Overview

The field of computer vision has tremendously benefited from advances in machine learning. State-of-the-art methods for object detection [129], recognition [46], segmentation [80] and pose estimation [143] adopt deep convolutional neural network for feature extraction. Pose estimation is the problem of localizing anatomical key-points, e.g., elbows, wrists [143]. Practical applications often require pose estimation as a fundamental task – for example, behavior understanding, human-object interaction, and activity recognition. The availability of large-scale datasets such as COCO [144] and MPII [145] dataset for human pose estimation has paved the way for tremendous progress to be made in this field. However, this has not been the case for animal pose estimation. One major reason for this setback is the unavailability of a large-scale labeled benchmark dataset with diverse species for animal pose estimation.

A few researchers have, in recent times, explored the development of mammal specific pose estimation datasets which are labeled according to the animal's anatomical structure, including the tigers dataset[146], horses dataset [147], macaque dataset [148], and zebra dataset [149]. Other more improved datasets such as Animal-Pose [150] and AP-10K [151] capture several animal species. While the public availability of these datasets has improved the ability of wildlife researchers and conservationists to better understand animal behavior and activity recognition, the lack of standardization in key-point annotation style across datasets, as well as the number of points defining the animal's skeletal structure, remains an important challenge and a limiting factor for generalizable animal pose estimation models. Figure 1 shows sample images from multiple datasets and the varied anatomical structures annotations.

Table 5-1: Brief review on current state-of-the-art animal pose dataset.

Dataset	Species	Number of key-points
Animal-Pose Dataset [150]	5	20
Horses-10 [147]	1	22
Tiger [146]	1	15
Macaque [148]	1	17
Ap-10K [151]	54	17
Elephant-Pose (ours)	**1**	**33**

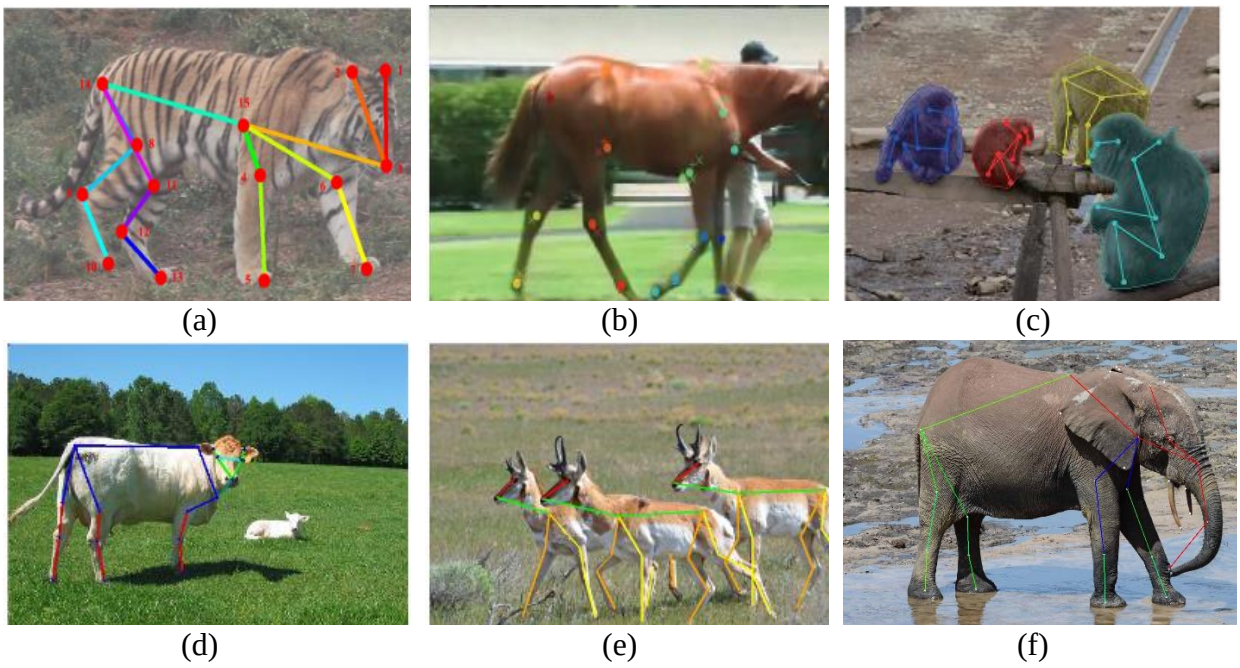


Figure 5-1: Sample images and annotations from 5 animal pose datasets (a) Amur Tiger pose dataset (b) Horse pose dataset (c) Macaque pose dataset (d) Animal-Pose dataset (e) AP-10K dataset, and (f) our Elephant pose and landmarks dataset.

Figure 5-1 shows how different skeletal structures, naming conventions, and keypoint locations vary from one dataset to another. Mammals share similar anatomical structures, and as such, it is possible to consolidate the images and labels from all five datasets into a single dataset to benefit from shared properties across mammals. Also, these anatomical similarities can be extended for out-of-domain classes to improve the model's generalizability to unseen animals.

The problem of pose estimation involves predicting joints in the body of objects which could be whole body joints e.g human pose [144], or body parts such as the hands [152], [153] or the face [154], [155]. Although human pose estimation is a very challenging problem, great progress has been made, which can be attributed to the availability of large-scale datasets such as MPII [145], COCO [144] and OCHuman [156] needed to train deep convolutional neural networks. Existing approaches for human pose estimation can be categorized into top-down approaches [157] and bottom-up approaches [158][159]. The top-down approach typically involves using a human detector model such as Faster-RCNN [129] to obtain a set of coordinate locations for each person detected and subsequently performing single-person pose estimation. While the bottom-up approach doesn't require a separate detector to find a person, but instead directly predicts the keypoints at once and then puts the points together into full poses for all persons [160]. Several methods for human pose estimation have been proposed based on these two approaches. Hao et al. [161] proposed a regional multi-person pose estimation framework to improve the robustness of top-down approaches in the presence of inaccurate human bounding boxes by refining the detection and pose estimation results. Sun et al. [162] proposed a high-resolution network with repeated multi-scale fusions for learning rich high-resolution representations throughout the network. Because of the parallel and feature-sharing design of the network, it is capable of predicting more accurate and spatially precise keypoint heatmaps. Pishchulin et al. [163] proposed a bottom-up approach called deepcut which jointly labels candidate joints and assigns them to individual persons by integer linear programming. Cao et al. [159] proposed a more computationally efficient approach by first predicting keypoint heatmaps and corresponding part affinity fields of all persons and then grouping the candidate joints into each person by a greedy algorithm. Newell et al. [164] proposed associative embedding for end-to-end learning for

simultaneous joint detection and grouping. Recently, Yang et al. [165] proposed a keypoint localization network using transformers. The attention layers in the transformer enable the model to capture long-range relationships more efficiently and reveal salient dependencies that predicted keypoints rely on.

Animal pose estimation has also seen tremendous progress over the years with the increased availability of datasets for computer vision tasks such as animal classification, detection, re-identification and tracking. Cao et al. [150] proposed a cross-domain approach for adapting prior learned knowledge from a large-scale human pose dataset to animal pose estimation. Their proposed network consists of a feature extractor, a domain discriminator for distinguishing between human and animal domains, and a keypoint estimator for predicting keypoints. Mathis et al. [147] proposed a pre-training step for boosting the performance and robustness of MobilenetV2, ResNet, and Efficientnet architecture on out-of-domain examples when trained on a limited dataset. Although these methods have provided great results for animal pose estimation, they often focus on either a single animal species, or the datasets are limited in the diversity of poses, textures, habitats, and animal species hence underperforming for several out-of-domain animal species. This paper seeks to harness the benefits of five of the datasets publicly available for animal pose estimation to build a more robust model that is effective for out-of-domain examples.

In summary, the main contributions in this work are as follows:

a) Introduction of a novel manually labeled dataset for elephant pose estimation and landmark detection.

b) Standardization of mammal pose estimation tasks across multiple datasets by creating a uniform anatomical structure for mammals.

c) Performance evaluation of transformer networks with other state-of-the-art pose estimation models.

d) Introduction of a novel pseudo-label generation and utilization strategy for improving model generalizability to unseen animal species using shared anatomical similarities.

5.2 Diversified Training Dataset Consolidation and Standardization

Five animal pose estimation datasets are consolidated and standardize for training and testing the proposed models. A detailed description of each dataset is presented in this section.

5.2.1 AP-10K Dataset

Yu et al. [151] curated the AP-10K dataset, a large-scale manually annotated data for animal pose estimation. It consists of 10,015 images and over 13 thousand labeled instances with 23 animal families and 54 species making it the largest dataset for pose estimation in mammals. It also contains 49,643 images with animal families and species labeled but without annotated key-points. This will be harnessed in our pseudo-label generation pipeline. Similar to the key-point annotation convention in a human pose, 17 key-points are defined to represent the poses of the animals. **Error! Reference source not found.** shows the visual location of the points on the animal and the c orresponding description.

Keypoint	Definition	Key-point	Definition
1	Left eye	**10**	Right elbow
2	Right eye	**11**	Right front paw
3	Nose	**12**	Left hip
4	Neck	**13**	Left knee
5	Root of tail	**14**	Left back paw
6	Left shoulder	**15**	Right hip
7	Left elbow	**16**	Right knee
8	Left front paw	**17**	Right back paw
9	Right shoulder		

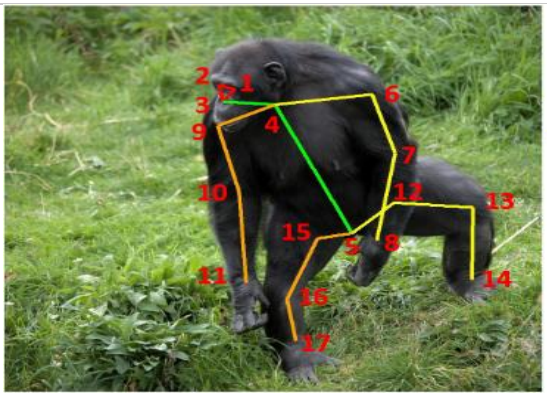


Figure 5-2: Visualization of key-points annotation from the AP-10k dataset [151]*. The table on the left describes the numbered key-points which are apparent on the animal picture on the right.*

5.2.2 Amur Tiger Dataset

Li et al. [146] proposed a large-scale Amur Tiger dataset for re-identification and pose estimation in the wild. It contains 8,076 images and 9,496 labeled instances with 15 key-points.

Keypoint	Definition	Key-point	Definition
1	Left ear	**9**	Right knee
2	Right ear	**10**	Right back paw
3	Nose	**11**	Left hip
4	Right shoulder	**12**	Left knee
5	Right front paw	**13**	Left back paw
6	Left shoulder	**14**	Root of tail
7	Left front paw	**15**	Center, midpoint of 3 & 14
8	Right hip		

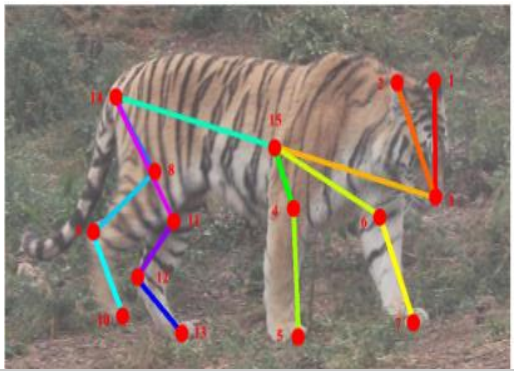


Figure 5-3: Visualization of key-points annotation from Amur Tiger dataset [146]*. The table on the left describes the numbered key-points which are apparent on the animal picture on the right.*

5.2.3 Macaque Dataset

Labuguen et al. [148] proposed a large-scale dataset for pose estimation of Macaque monkeys in the wild. It contains over 13000 images with manually labeled body key-points for macaques in naturalistic scenes. A total of 17 key-points are defined for each monkey even when some key-points are occluded.

Keypoint	Definition	Key-point	Definition
1	Nose	**10**	Left wrist
2	Left eye	**11**	Right wrist
3	Right eye	**12**	Left hip
4	Left ear	**13**	Right hip
5	Right ear	**14**	Left knee
6	Left shoulder	**15**	Right knee
7	Right shoulder	**16**	Left ankle
8	Left elbow	**17**	Right ankle
9	Right elbow		

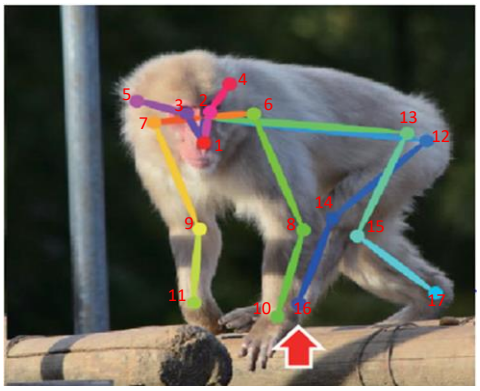


Figure 5-4: Visualization of key-points annotation from Macaque dataset [148]*. The table on the left describes the numbered key-points which are apparent on the animal picture on the right.*

5.2.4 Elephant Dataset

Due to the limited number of annotated images of elephant poses, part of this work involved curating a first large-scale dataset specifically for elephant pose estimation. It contains over 1300 labeled images and 5000 unlabeled images of elephants. 20 key-points define the anatomical structure of the elephant, and an additional 13 key-points are labeled to provide vital landmarks that could be needed for elephant behavior understanding and re-identification tasks.

Keypoint	Definition	Key-point	Definition
1	Bottom trunk	**17**	Top left ear
2	Mid trunk	**18**	Top right tip ear
3	Top trunk	**19**	Top left tip ear
4	Bottom right tusk	**20**	Withers
5	Bottom left tusk	**21**	Tail
6	Top right tusk	**22**	Right front elbow
7	Top left tusk	**23**	Left front elbow
8	Right eye	**24**	Right back elbow
9	Left eye	**25**	Left back elbow
10	Right bottom ear	**26**	Right front knee
11	Left bottom ear	**27**	Left front knee
12	Right bottom tip ear	**28**	Right back knee
13	Left bottom tip ear	**29**	Left back knee
14	Right side tip ear	**30**	Right front foot
15	Left side tip ear	**31**	Left front foot
16	Top right ear	**32**	Right back foot
		33	Left back foot

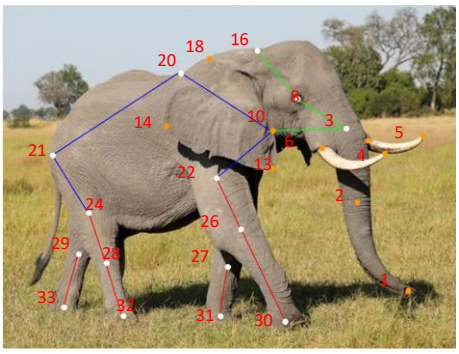


Figure 5-5: Visual overlay of key-points & other landmarks. The key-points on the animal on the right-hand side are the white dots, while the orange dots represent the landmarks.

5.2.5 Dataset Standardization

Due to the difference in the number of key-points, labeling style, and key-point location, a standard form is developed across all the model training datasets. The anatomical structure proposed by Cao et al. [150] as shown in **Error! Reference source not found.** , is adopted since it accounts f

or an additional number of useful keypoints like the ears and withers and better covers the human bones as defined in the COCO-format key-point annotations. The keypoints not annotated in other datasets are given a ‘missing’ tag and left unannotated. Animal Pose, AP-10K, and Tiger datasets are used for training our models. For the AP-10K dataset, the annotation for ‘neck‘ (key-point 4) is removed since the location is somewhat different from the Animal-Pose datasets, and ‘midpoint’ (key-point 15) is removed from the Tiger dataset.

5.3 Proposed Unified Animal Pose Estimation Framework

This section introduces the details of the training framework and network architectures. A transfer learning approach is adopted for training our animal pose estimator with limited data. High-resolution networks (HRNet) [166] are adopted as the primary backbone of our training framework because it has proven to outperform other ResNet-based models for pose estimation tasks [151], [162].

5.3.1 Proposed framework

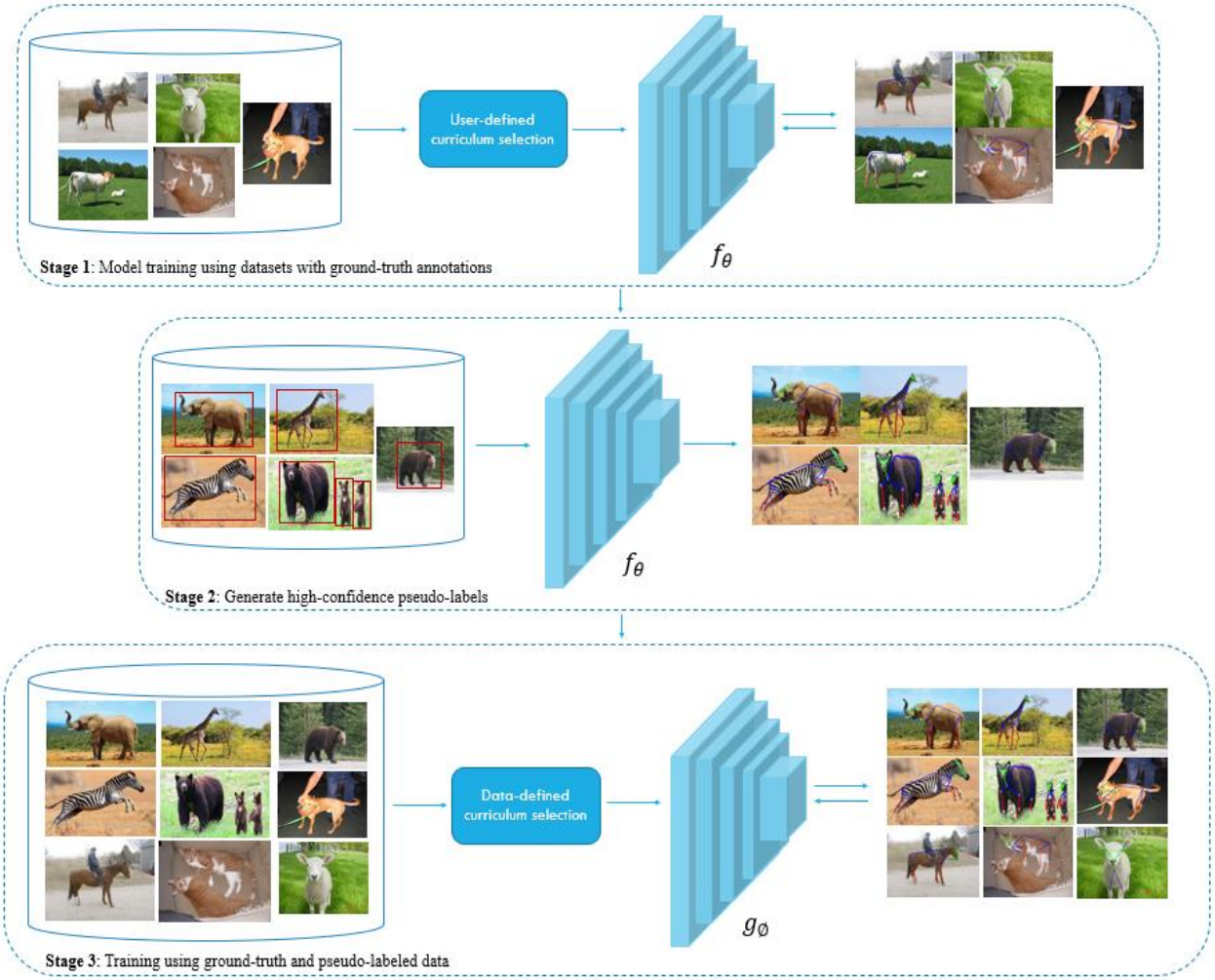


Figure 5-6: Proposed training framework which involves training a teacher network f_θ using ground-truth annotated datasets in Stage 1, generating high-confidence pseudo-labels in Stage 2 by using the learned weights from f_θ and using a combination of the ground-truth labeled data and the pseudo-labeled data to then train a student network $g_\emptyset$ in Stage 3.

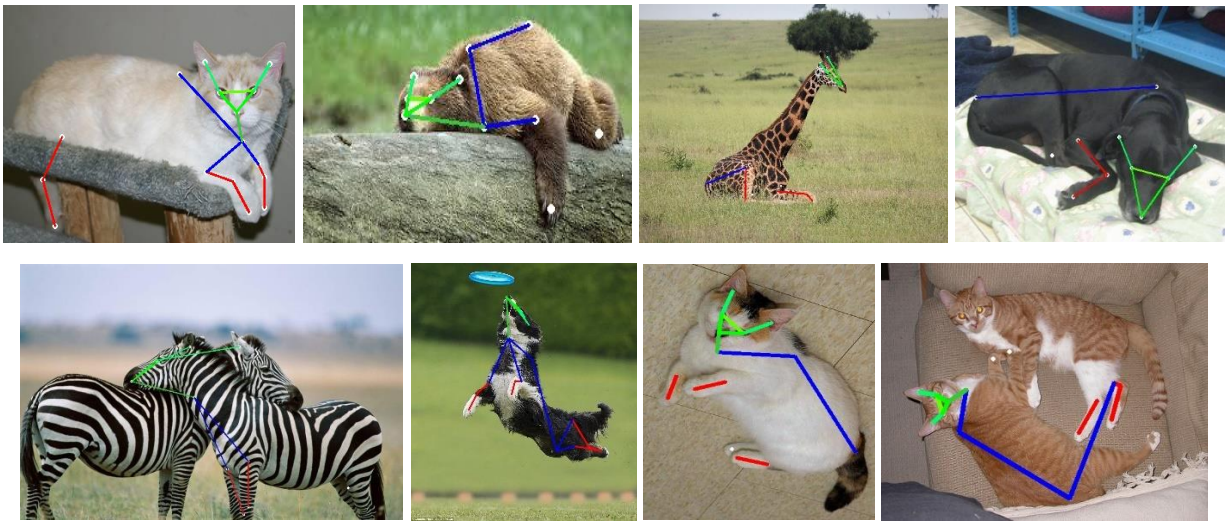
A progressive learning approach is adopted for training the pose estimation network. As shown in Figure 5-6, there are three stages in our training framework. Curriculum learning is a method used to gradually increase the complexity of data samples in the training process [167]. A simple curriculum learning strategy that involves introducing data samples from easy to hard examples is adopted in training the teacher (f_θ) and student ($g_\emptyset$) networks. For the first stage, training is done

on three animal datasets: the animal pose, ap-10k and tiger datasets. A simple curriculum that depends on the number of annotated key-points helps determine which datasets are used at multiple intervals during training. This helps the model progressively learn from data distributions that provide the most information to others containing the least information. Hence, for our training purpose, the curriculum is designed to assign a higher priority to data from the animal pose dataset with 20 standard key-points, then the ap-10k dataset with 16 standard key-points, and finally, the tiger dataset with 14 standard key-points.

In the second stage, pseudo-labels are generated for a large set of unlabeled data for unseen animal species using the learned weights from a f_θ. Each image sample is assigned a class that could be either 'easy' or 'hard', indicating the complexity of the animal pose. The class assigned is dependent on the number of high confidence key-points detected in each localized animal. This is based on the hypothesis that the complete anatomical structure of an animal tends to be visible and the model can easily detect the pose of the animal if it is well-defined. By ensuring the threshold is high, more unreliable labels can be filtered. Finally, missing annotations from the standardized datasets used to train the teacher model in stage 1 are generated and used to train the student network. Figure 5-7 shows some samples from the pseudo-labeled data.

In the third stage, the ground-truth annotated dataset and the pseudo-labeled data are used for training a student network, $g_\emptyset$. Figure 5-8 shows an expanded diagram of this semi-supervised approach via parameter sharing. Sharing the weights allows the model to learn more robust feature representations across diverse animal species hence improving the model's generalizability. The loss function and training scheduling process is discussed in detail in the subsequent section.

(a)



(b)

Figure 5-7: Samples of pseudo-labels generated using f_θ (a) images classified as 'easy' examples, and (b) images classified as 'hard' examples. The animal pose in the easy examples is often well-defined with most joints visible while hard examples have several joints that are either occluded or hard to predict.

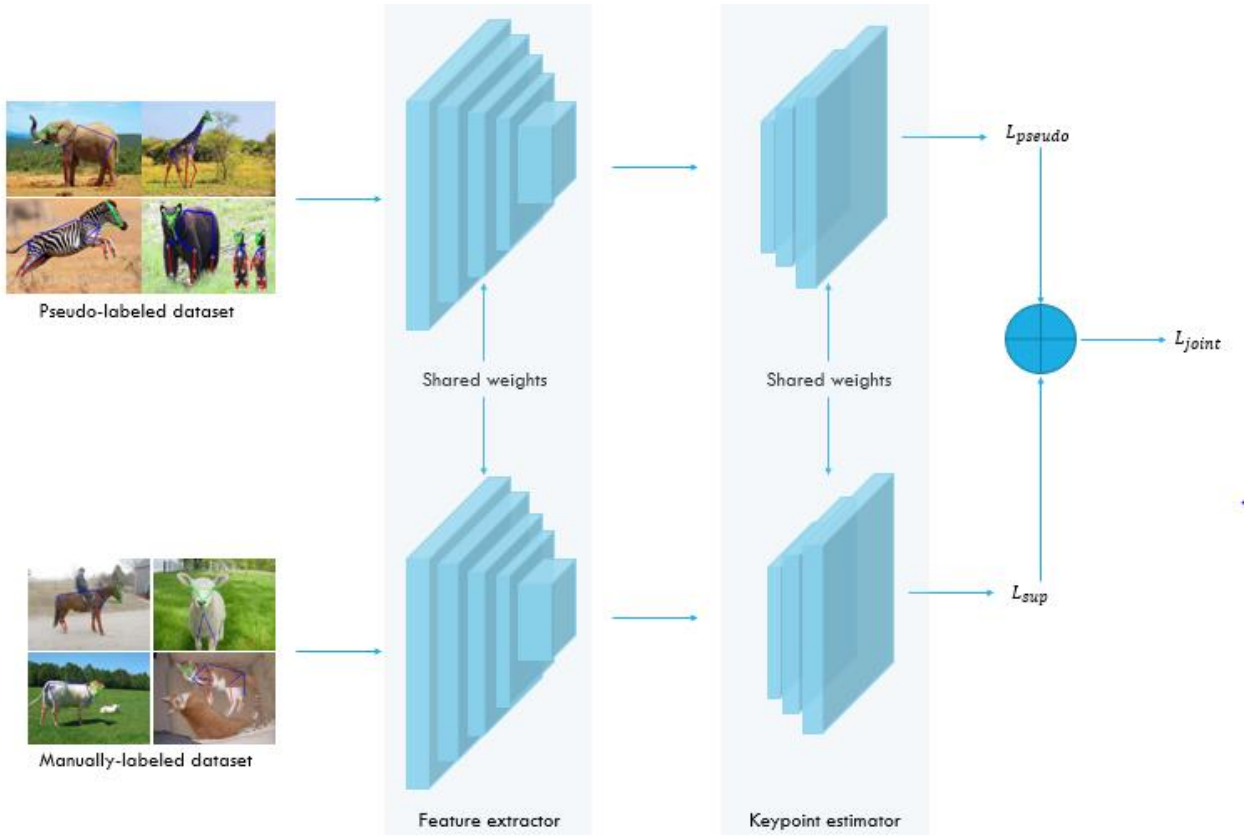


Figure 5-8: Overview of the proposed approach for harnessing noisy pseudo-labels for animal pose estimation. The feature extractor is initialized using ImageNet pretrained weights. L_{sup} and L_{pseudo} are the supervised and semi-supervised loss respectively. L_{joint} is a weighted combination of both.

Figure 5-8 shows the network design used in stage 3 of the training framework for manually labeled and pseudo-labeled datasets. The 'manually-annotated' dataset contains the original labels. The predicted 'missing' labels are not accounted for due to the standardization of the mammals' skeletal structure, which is different from that used by the original authors. The feature extractor are the convolutional layers of the HRNet model, and the keypoint estimator are the aggregated features of all layers of the HRNet which are regressed to a heatmap from which the final joint coordinates are obtained in the original image space. In this case, a distribution-aware coordinate representation (DARK) proposed by Zhang et al. [168] is used to decode the joint coordinates' location.

5.3.2 Objective Function

The mean squared error is used to evaluate the Euclidian distance between the predicted and ground-truth keypoints. The objective function, L_{joint} for training the network using a set of labeled and pseudo-labeled datasets, is given as:

$$L_{joint} = L_{sup} + c\alpha L_{pseudo} \quad 5.1$$

$$c = \begin{cases} 1, if\ using\ pseudo-labels \\ 0, otherwise \end{cases}$$

Where, L_{sup} is the mean squared error on the labeled data and L_{pseudo} is the mean squared error on the pseudo-labeled examples

The value of α is dynamic and changes at pre-defined steps

$$\alpha_n = \alpha_{init} + (\beta) * (n) * d \quad 5.2$$

$$d = \frac{\alpha_{max} - \alpha_{init}}{number\ of\ epochs} \quad 5.3$$

Where α_n is the weight at the n-th update step, β is the interval before each update, and d is a constant pre-computed using α_{max} $(maximum\ value\ for\ \alpha)$.

We set $\boldsymbol{\alpha_{init}}$ **= 0.1,** $\boldsymbol{\alpha_{max}}$ **= 0.8,** $\boldsymbol{\beta}$ **= 10**, and the number of **epochs = 250** when training with pseudo-labeled data. These values were experimentally determined.

Ensuring that the contribution of the pseudo-labels to the loss function is controlled by gradually changing the value of α rather than using a preset value ensures that our approach doesn't reduce the performance of the model to unseen animals because of the noisy/unreliable labels.

L_{pseudo} is similar to equation (8) in the work proposed by Cao et al. [150] for self-paced selection of pseudo-labels. It involves training the network with pseudo-labels for easier examples before harder ones. This also helps prevent model degradation due to the aggressive use of pseudo-labels.

$$L_{pseudo} = \sum_{j=0}^{N} w_j^{\emptyset} L_{MSE}(I_j^T, m(I_j^T|\emptyset)) \quad 5.4$$

Where, $m(I_j^T|\emptyset)$ is the output by the model of current weights $\emptyset$ on input image I_j^T. $w_j^{\emptyset}$ denotes whether the pose prediction on I_j^T is a hard or easy example

$$w_j^{\emptyset} = \begin{cases} 1.0, if\ C(m(I_j^T|\emptyset)) > \mu \\ 0.2, otherwise \end{cases} \quad 5.5$$

Where $C(m(I_j^T|\emptyset))$ denotes the output confidence score on I_j^T by the current model and μ is the threshold to filter unreliable outputs. $\boldsymbol{\mu}$ **is set to 0.9** for our task to select only high confidence pseudo-labels.

5.3.3 Implementation Details

HRNet-w48 [169] trained on the ImageNet dataset is the primary backbone for training our benchmark networks. Training, validation, and testing were carried out on two Nvidia Tesla V100 GPUs with 32GB memory. An Adam optimizer [82] with an initial learning rate of 0.001 is used for model optimization. The batch size is set to 24 and 16 for training and testing, respectively. Progressive finetuning is carried out by feeding the network with different datasets at different intervals. The primary dataset, which is the Animal Pose dataset, is trained for 150 epochs, and an extra epoch of 200 is needed for training on the AP-10K and Tiger datasets. However, model evaluation is performed on all five datasets discussed in section 2.1. Overall, training is carried out

in 2 steps: (i) Train the teacher network on all three labeled datasets (ii) Train student network with all three labeled datasets and pseudo-labeled data. The algorithm is implemented using the Pytorch deep learning framework [170].

5.4 Evaluation Metrics

Two metrics are used for evaluation. The percentage of correct keypoints (PCK) which is a precision metric that determines if the predicted keypoint and the true joint are within a given distance threshold. The PCK is typically set in relation to the subject's scale, which is contained within the bounding box [171]. The threshold can either be:

- PCKh@0.5 is when the threshold >= 50% of the head bone link
- PCKh@0.2 is when the distance between predicted and ground-truth < 0.2 * torso diameter.

The Object Keypoint Similarity (OKS) calculates the distance between predicted keypoints, and ground-truth points normalized by the object's scale [172]. It is used to calculate the mean average precision (MAP) which varies from 0 to 1.

$$\frac{\sum_i exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \quad 5.6$$

Where,

- d_i is the Euclidean distance between the ground-truth and predicted key-point
- s is the square root of the object segment area
- k is the per-key-point constant that controls fall off
- v_i is the visibility flag that can be 0,1,2 for not labeled, labeled but not visible, and visible and labeled respectively.

5.5 Experiments and Results

Evaluation is performed on the validation sets available in the tiger dataset with 515 images, the animal pose dataset with 1020 images, and the AP-10K dataset with 1331 images. To be fair, the computations are done on the original annotations for each dataset.

Table 5-2: Results of several models on AP-10K validation set containing 1331 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across all threshold levels.

Model	***AP***	***$AP_{.5}$***	***$AP_{.75}$***	***AP_M***	***AP_L***	***PCKh@0.5***
TransPose-H [165]	0.764	0.946	0.829	0.692	0.766	-
HRNet-w32 [162]	0.783	0.939	0.853	0.643	0.787	0.770
HRNet-w48 [162]	0.803	0.959	0.874	0.680	0.806	0.789
TransPose-H + SSL	0.772	0.946	0.831	0.687	0.777	0.754
HRNet-w32 + SSL	0.833	0.960	0.898	0.744	0.836	0.812
HRNet-w48 + SSL	0.842	0.959	0.901	0.766	0.845	0.826
HRNet-w48 + PSA + SSL	**0.850**	**0.960**	**0.911**	**0.792**	**0.852**	**0.830**

Table 5-3: Results of several models on Animal-Pose validation set containing 1020 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across most threshold levels.

Model	***AP***	***$AP_{.5}$***	***$AP_{.75}$***	***AP_M***	***AP_L***	***PCKh@0.5***
TransPose-H [165]	0.742	0.957	0.840	0.602	0.775	-
HRNet-w32 [162]	0.751	0.958	0.841	0.606	0.789	0.778
HRNet-w48 [162]	0.771	0.968	0.862	0.624	0.802	0.789
TransPose-H + SSL	0.745	0.947	0.842	0.594	0.782	0.759
HRNet-w32 + SSL	0.785	0.968	0.874	0.665	0.814	0.797
HRNet-w48 + SSL	0.790	0.968	**0.888**	**0.675**	0.822	0.800
HRNet_w48 + PSA + SSL	**0.792**	**0.968**	0.884	0.667	**0.824**	**0.803**

Table 5-4: Results of several models on the Tiger validation set containing 515 images. The values for the average precision shows that HRNet-w48 + PSA + SSL gives the best performance across most threshold levels.

Model	***AP***	***$AP_{.5}$***	***$AP_{.75}$***	***AP_M***	***AP_L***	***PCKh@0.5***
TransPose-H [165]	0.833	0.977	0.951	0.794	0.834	-
HRNet-w32 [162]	0.894	0.979	0.952	0.824	0.894	0.929
HRNet-w48 [162]	0.898	0.979	0.965	0.828	0.898	0.932
TransPose-H + SSL	0.840	0.977	0.951	0.806	0.873	0.877
HRNet-w32 + SSL	**0.920**	0.977	0.961	0.836	**0.920**	**0.944**
HRNet-w48 + SSL	0.914	0.976	0.959	0.844	0.914	0.943
HRNet-w48 + PSA + SSL	0.895	**0.980**	**0.966**	**0.844**	0.895	0.920

Table 5-2, Table 5-3, and Table 5-4 show the performance of several model architectures on benchmark datasets. AP signifies the average precision across all IoU thresholds, $AP_{.5}$ indicates performance at IoU $\geq$0.5, etc. PCKh@0.5 measures the distance between the predicted keypoint and true joint. Overall, the result across all datasets shows that harnessing noisy (pseudo-generated) labels significantly improves the performance of state-of-the-art models with an average percentage boost in AP of ~**3.8%** for the AP-10K dataset, ~**2.3%** for the Animal Pose dataset, and ~**1.8%** for the Tiger dataset. The average improvement in PCK is about **5.1%, 1.9%**, **1.4%** for AP-10K, Animal Pose, and Tiger datasets, respectively. The results also show that incorporating polarized self-attention (PSA) modules [173] improves the representation capacity of HRNet and thus outperforms the standard HRNet for most of the datasets.

Figure 5-9: Qualitative results from AP-10K validation set using HRNet-w48 + SSL model

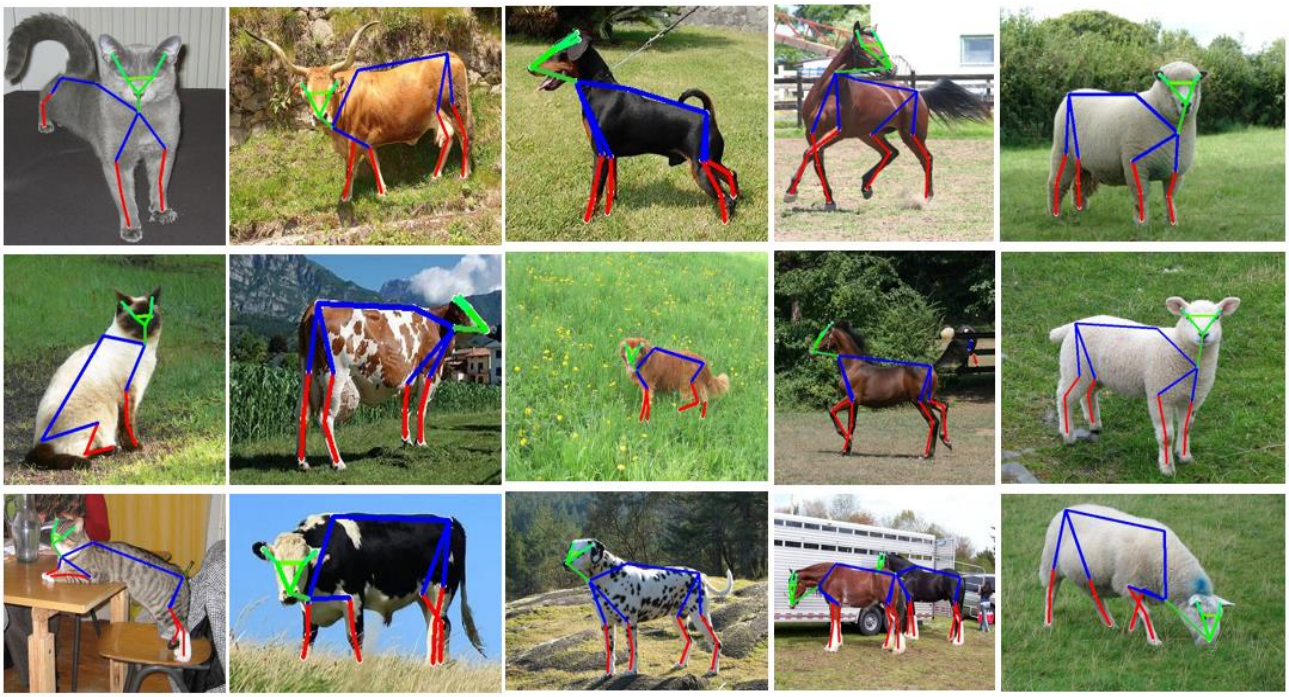


Figure 5-10: Qualitative results from Animal Pose validation set using HRNet-w48 + SSL model

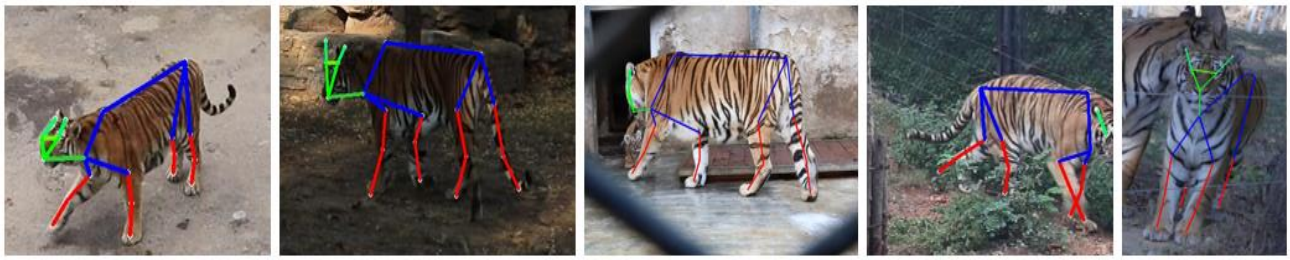


Figure 5-11: Qualitative results from Tiger Pose validation set using HRNet-w48 + SSL model

Figure 5-12: Qualitative results from Tiger Pose validation set using HRNet-w48 + SSL model

Figure 5-9 to Figure 5-12 shows samples results obtained using our HRNet-w48 + SSL model. These results demonstrate the effectiveness of our training framework for predicting missing keypoint annotations and adapting to unseen animal categories. It shows the framework's robustness to partly occluded joints, pose variations, and animal sizes/length. The model leverages the shared anatomical structures of mammals to efficiently predict the location of each keypoint with context as seen in the visual results for the giraffe and elephant.

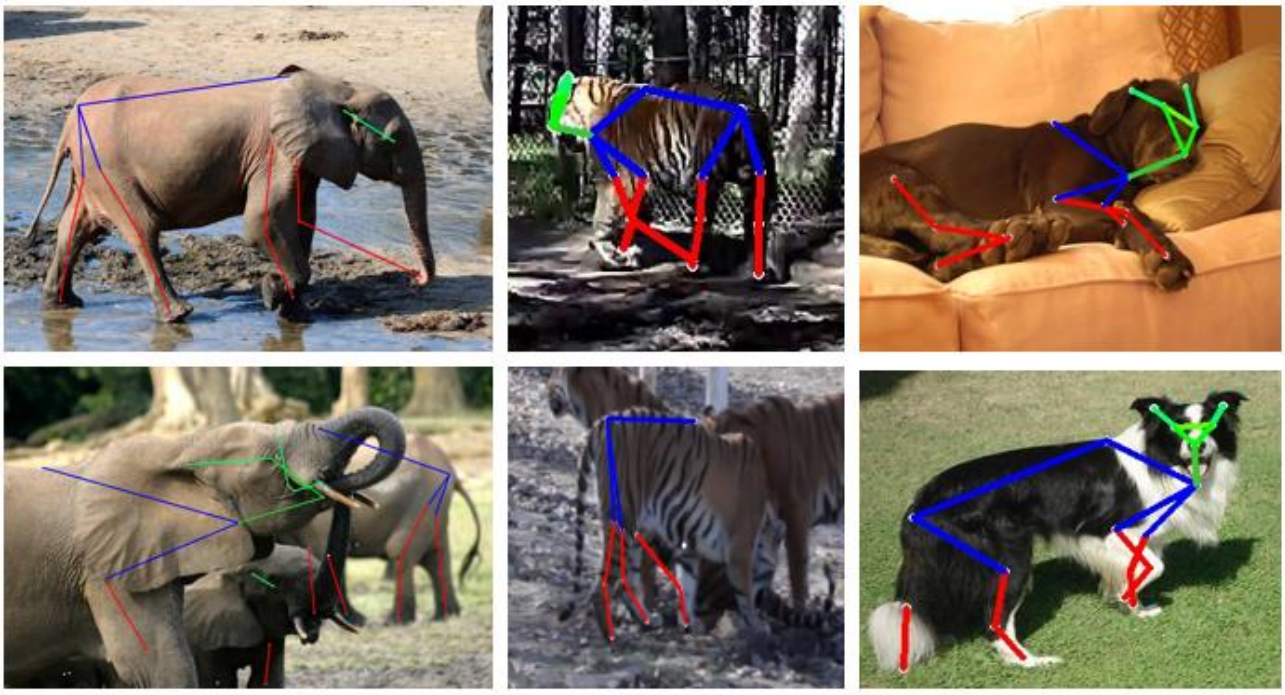


Figure 5-13: Sample failure cases of the proposed model. As can be seen in this image the failure occur as a result of occlusion, poor image quality, multiple animals present in the image, low-light condition, complex animal poses, etc.

Figure 5-13 shows some fail cases from our model. Animals exhibit diverse poses making it difficult for the model to account for every possible case. Poor image quality also severely degrades the model's ability to correctly identify and assign labels to predicted keypoint. For animals that often move as a herd, such as elephants, predicting the pose of each animal becomes increasingly difficult due to the increased possibility of an occluded part or mis-assignment of keypoints corresponding to a particular animal.

5.6 Summary

This chapter introduces a unified standard for pose estimation in mammals and a simple but efficient framework for effectively utilizing noisy labels to improve a model's generalizability to unseen animal breeds and species. The work also conducts a detailed performance evaluation of standard convolutional neural networks compared to transformer networks for animal pose estimation. Results show that traditional CNN-based models, specifically, high-resolution networks are best for handling large pose variations common in animals. Finally, a large-scale dataset for elephant pose estimation and landmark localization is presented.

Acknowledgments

The Humane Society of America partly provided funding through the Tufts Veterinary School and the Tufts Elephant Conservation Alliance (TECA). We thank Professor Allen Rutberg and Kali Pereira for their help while working on this research.

Chapter 6. Effectiveness of Two-Stream Networks for Multi-task Learning

Assessing the extent of damages incurred by a region in the aftermath of a natural disaster is challenging and often prone to human-in-the-loop errors when carried out manually. Evaluating the level of damage on buildings provides better insight into the number of homes affected as well as serving as a quick and effective means for planning search and rescue operations. Most existing methods for building damage estimation either focus on the binary classification problem where the classes are either damaged vs non-damaged or they use a multi-stage approach in which one model performs building localization and another model takes the coordinates of the localized buildings and classifies them into the damage levels. Although the latter approach provides a more practical solution, it is time consuming and inefficient.

In this work, a two-stream network (TS-HRNet) is proposed for joint building detection and damage classification at the same time. Additionally, a semi-supervised framework for improving the generalizability of current methods to other housing styles is presented. This framework includes novel contributions; 1) a pseudo-label generator that uses a TS-HRNet model trained on a labeled dataset for assigning labels to an unlabeled dataset, and (2) a refinement module for filtering out noisy labels. By harnessing information from a large amount of unlabeled data and aerial imagery, TS-HRNet can outperform other baseline models.

Experimental results on the xView2 dataset demonstrate that combining a semi-supervised approach significantly improves the performance of the proposed TS-HRNet model for unseen satellite images depicting a scene before and after a disaster.

The work presented in this chapter has been published in the proceedings of the SPIE conference, 2020 [174].

6.1 Overview

Very High Resolution (VHR) remote sensing imagery provides a means of analyzing the impacts of a disaster in both the pre- and post-disaster events [175]. It provides a view of the landscape, state of the buildings and other information that could be relevant in providing the needed aid to affected populations. With the increased availability of satellite and aerial imagery, artificial intelligence and computer vision algorithms can be developed to harness these imageries in order to automate the task of assessing damages before, during, and after a disaster. Several datasets have been introduced to help with these tasks such as the xBD dataset [176], and Inria aerial imagery [177]. The xBD dataset covers a diverse set of disasters across multiple geographical locations, providing a standard benchmark for training and testing machine learning methodologies.

Several methodologies have been developed for building damage assessment from satellite imagery. Cooner et al. [178] evaluated the effectiveness of two neural network variants and random forests for detecting damages to buildings after an earthquake with the lowest error rate of 40%. Ji et al. [179] utilized a convolutional neural network (CNN) to identify collapsed buildings from post-earthquake events using a case study of Haiti. Duarte et al. [180] proposed a CNN framework using residual connections and dilated convolutions which was trained on a combination of drone and satellite images to classify building damage. These methods framed the building assessment task as a binary pixel classification problem and hence are ineffective for assessing various levels of damage to buildings in a multi-class setting.

Recent methods for semantic segmentation of buildings often depend on the U-Net model [181] as a baseline or used as a backbone. Other encoder-decoder networks similar to the U-Net model have also been developed for improved segmentation. Zhou et al. [182] improved upon the performance of UNet for medical segmentation tasks and named it UNet++ by connecting the encoder and decoder sub-networks through a series of nested and dense skip pathways. Chaurasia et al. [184] proposed LinkNet, replaced the skip connections in U-Net's encoder with residual connections. Iglovikov et al. [183] developed TernausNetv2, which is a modified U-Net for extracting objects from high-resolution satellite imagery. It extends networks applied for semantic segmentation for instance segmentation tasks by predicting instance boundaries along with pixel-wise masks.

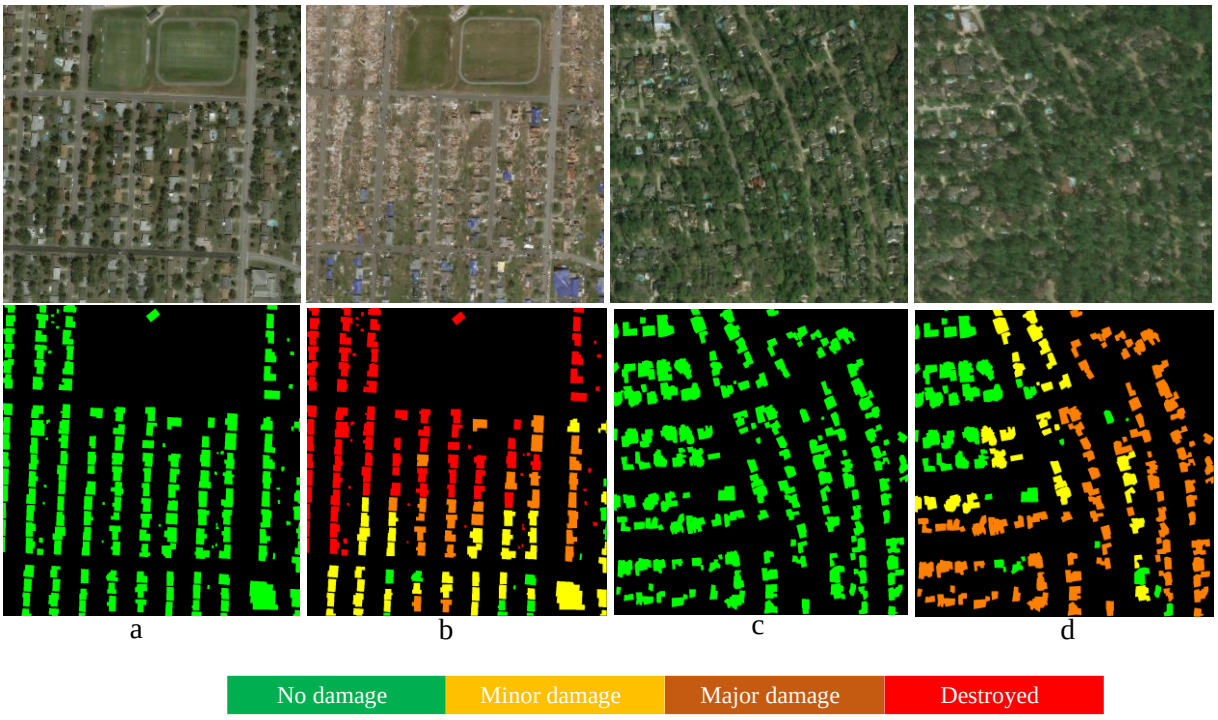


Figure 6-1: Sample image pairs from the xBD dataset (a) and (b) are pre- and post-disaster images from tornadoes, (c) and (d) are pre- and post-disaster images from hurricane Harvey.

The xBD dataset provides a large-scale public dataset to advance humanitarian assistance and disaster recovery research. It comprises of a large number of annotated image pairs for change detection and building damage assessment. It comprises 850,736 annotated buildings covering 45,362 km^2 of imagery and covers a diverse set of disasters across multiple geographical locations. Figure 1 shows image samples from the xView2 dataset, and Table 1 presents the original paper's explanation of the Joint Damage Scale (JDS) for categorizing building damage for the xBD dataset.

Table 6-1: Joint Damage Scale (JDS) describes how the damage levels were determined.

Damage level	**Structure Description**
0 (No damage)	Undisturbed. No sign of water, structural or shingle damage, or burn marks.
1 (Minor damage)	Building partially burnt, water surrounding structure, volcanic flow nearby, roof elements missing or visible cracks
2 (Major damage)	Partial wall or roof collapse, encroaching volcanic flow, or surrounded by water/mud
3 (Destroyed)	Scorched, completely collapsed, partially/completely covered with water/mud, or otherwise no longer present

Although these methods perform significantly better than conventional methods, their performance degrades across other datasets. Therefore, there's a need to improve the model's performance and generalizability on other building styles and sizes. Our proposed method adopts the Weber's approach [183] but uses High-Resolution Networks (HRNet) [169] for the feature extraction stage. These networks have proven to outperform well-known series-connected networks such as ResNet and VGG-Net for position-sensitive (spatially precise) tasks like pose estimation, semantic segmentation, and object detection.

The main contributions in this thesis chapter are as follows:

a) A two-stream high-resolution network (TS-HRNet) for localizing buildings and classifying the type of damage incurred by the buildings before and after disasters.
b) A simple consistency-based pipeline for generating high confidence pseudo-labels on unseen satellite imagery while discarding low confidence labels.
c) A training pipeline to effectively harness pseudo-labels to improve model performance.

6.2 Proposed TS-HRNet Model

A pair of image patches of pre- and post-disaster images are separately fed as input to each stream. The first stream localizes the building, while the second stream classifies it into damage categories. For each network, HRNet is exploited because of its ability to maintain high-resolution representations throughout each stream, resulting in spatially more precise and semantically stronger representations, e.g., color and structure content. Therefore, given a pair of input data (X_i^{pre}, X_i^{post}), where X_i is an RGB satellite imagery, our network is able to learn a set of network parameters θ that predicts the location of buildings in the image y_i^{loc} and predict the disaster level of these buildings y_i^{cls}. In the test phase, the network is able to localize and classify buildings given a single image i.e, $f_\theta(X_j^{test}) = (\hat{y}_j^{loc}, \hat{y}_j^{cls})$.

6.2.1 TS-HRNet Model Architecture

The HRNetV2 architecture for object segmentation is adopted. It starts with a stem consisting of 3x3 convolutions and decreases by a factor of 1/4. The first stage contains 4 residual units while the second, third, and fourth stages contain 1, 4, and 3 multiresolution blocks with convolution widths of C, 2C, 4C, and 8C, respectively, where C is the width of feature maps. By connecting high-to-low resolution convolutions in parallel and performing repeated multiscale fusions across

these convolutions, HRNet is able to maintain high-resolution representation across the network hence making them suitable for spatially precise applications like ours. Figure 2 shows the two-stream HRNet used for localization of buildings as well as damage level classification where C is set to 48. Pairs of pre-disaster and post-disaster images are fed into the network. Feature extraction is done separately for different streams and weight sharing across both streams helps improve the performance of each task. For classification, feature fusion is performed to improve the performance of the pixel-wise classification. This is done by concatenating the features of both streams across their width. The fused features are fed to a convolution block before classification.

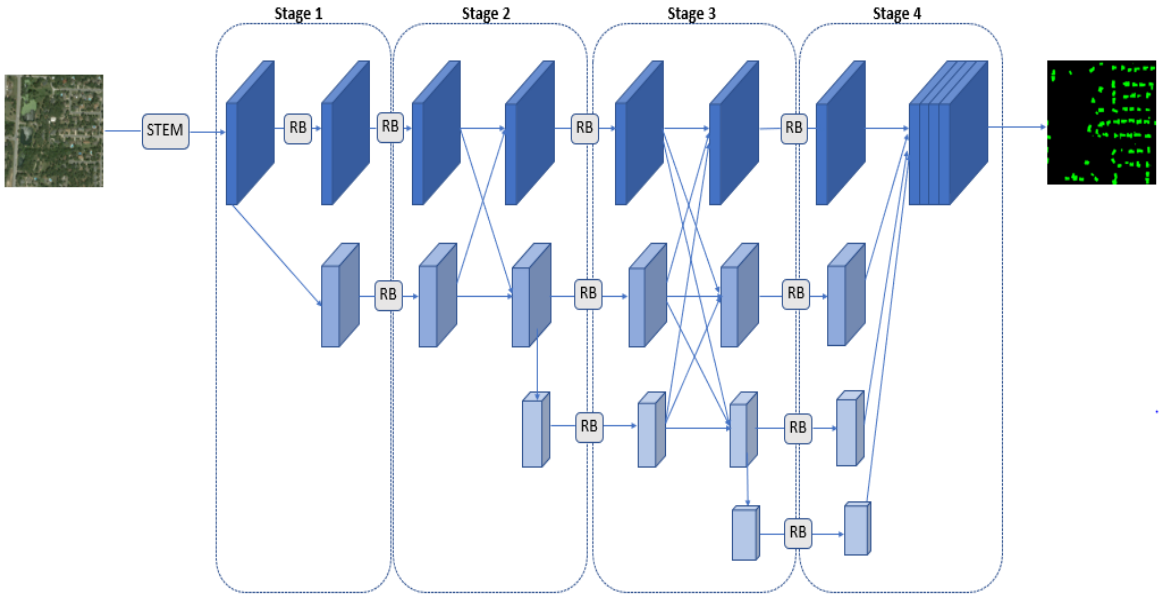


Figure 6-2: Network architecture for high resolution networks (HRNet) for image segmentation [166].

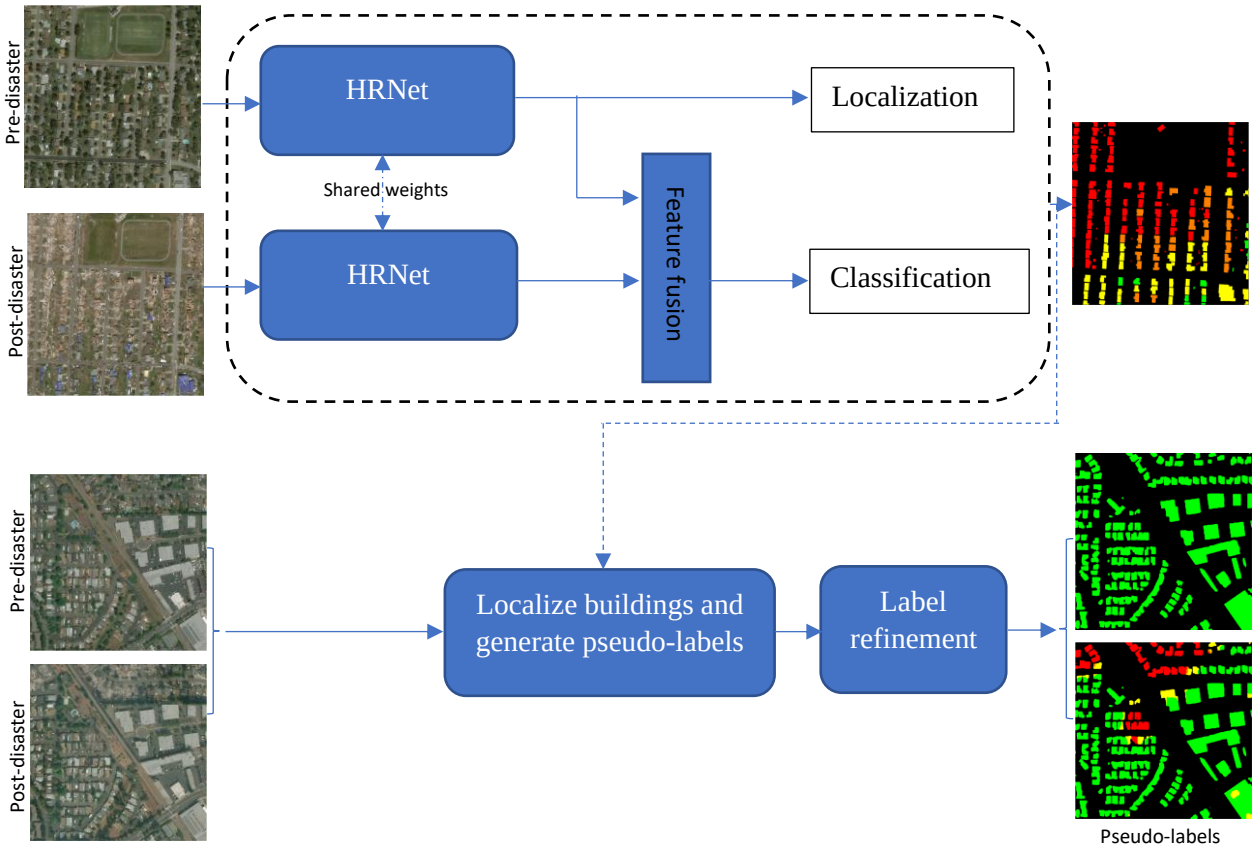


Figure 6-3: Proposed TS-HRNet framework. A pair of pre- and post-disaster images are fed to each stream, and network localizes the buildings and classifies into damage types. The pseudo-label generation pipeline takes a pair of pre- and post-disaster images and generates pseudo-labels for each pair.

Semi-supervised learning provides a means of harnessing large amounts of unseen data across other regions and thus improves the generalizability of existing models to other building types. For our task, a simple pipeline for generating pseudo-labels is explored. A trained teacher network is used to generate pseudo-labels on unseen data. These labels include the location of the buildings and the associated label according to the JDS standard given in Table 6-1. Label refinement is performed to remove noisy labels as well as improve the boundaries of the segmentation masks. A consistency-based approach to label refinement is adopted. Four spatial level transformations, including horizontal flip, vertical flip, transpose, and rotation, are applied to the input images, and

the trained model is used to localize and classify buildings in images from each transformation. The inverse transform is applied, and the most frequent class from all five outputs, including the untransformed image results, is computed to obtain the final output. Results show that incorporating a simple semi-supervised pipeline helps in improving the performance of our model on the labeled and unlabeled data.

6.2.2 Objective Function

Because the dataset contains highly imbalanced classes, the choice of the loss function is not trivial. We experimented with two variants of the loss function including the class-weighted cross-entropy loss and a weighted combination of focal loss [184] and dice loss [185]. A larger weight is assigned to classes with fewer examples, while lower weights are assigned to classes with a larger number of examples. Table 6-2 shows the class weights for classification and localization. For building localization using pre-disaster images only, buildings are assigned a class value of 1 and others 0. For the classification task, the largest weight is assigned to the 2 (minor damage), and class 1 (no damage) has the lowest score of 0.5.

Table 6-2: The weight values assigned to each class per task. This is empirically determined based on the dominance of each class in the training dataset.

Task	**Label**				
	0	*1*	*2*	*3*	*4*
Localization	0.5	1.5	-	-	-
Classification	-	0.5	1.5	1.0	1.0

Focal and dice loss are region-based loss functions [186]. Focal loss is an improvement of the standard cross-entropy loss. The loss reduces the contribution of easy examples and enables the

model to focus more on learning hard examples. This helps reduce the class imbalance. For imbalanced classes, the cross-entropy loss is scaled with the scaling factors decaying to zero as the confidence in the correct classes increases. The focal loss defines the estimated probability of class as:

$$p_t = \begin{cases} p, & if\ y = 1 \\ 1 - p, & otherwise \end{cases} \quad 6.1$$

The focal loss is given as

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma}\log(p_t) \quad 6.2$$

α ranges from [0,1]. For our task, it is set to 1. If γ = 1, the focal loss works like a cross-entropy loss function.

Dice loss is obtained by calculating the dice coefficient function.

$$DL(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad 6.3$$

Where X is the ground truth building localization label and Y is the predicted label. The intersection term is computed as the product of the ground-truth and prediction. The Dice Coefficient (DC) is usually used to assess the level and statistical significance of overlap between reference and candidate image object pairs.

Therefore, the final training objective function is given below

$$L_{total} = \lambda_1 L_{fl} + \lambda_2 L_{dl} \quad 6.4$$

where λ_1 and λ_2 are empirically set to 2.0 and 0.5 respectively.

6.3 Experiments and Results

The xBD dataset is used for training, validation, and testing. xBD is the recent large-scale dataset for the advancement of change detection and building damage assessment for humanitarian assistance and disaster recovery research. It contains 850,736 building annotations across 45,362 km^2 of imagery. The dataset also consists of geo-registered pairs of pre- and post-disaster images of size 1024x1024 pixels with building polygons and 4-class building damage levels outlined in table 1. Table 6-3 shows the number of images in the dataset.

Table 6-3: Size of xBD dataset

Folder	**Image pairs**
Train	2,799
Tier3	6,369
Held-out	933

The data from the train and tier3 folders are combined and split into 97% train and 3% validation sets. The held-out data served as our test set. The input resolution to our network is a randomly cropped 512x512 image. Data augmentation is also adopted and includes random horizontal and vertical flipping, random scaling (between 0.75 and 1.5), random rotation (between -30 and 30 degrees).

To improve the model's performance and especially the building localization task, the Inria Aerial Image labeling dataset is also included in the training phase. The dataset includes satellite imagery of urban settlements over the United States and Austria and is labeled as building and not building classes. It contains 180 colored images of size 5000x5000 pixels with building polygons.

For training, the dataset is cropped into size 1024x1024 as the xBD dataset. Because there are no pre- and post-disaster pairs, we assumed no damage for the cropped images and created the pairs.

Stochastic Gradient Descent (SGD) optimizer with a base learning rate of 0.01, a momentum of 0.9, and weight decay of 0.0005 are used. The model is trained on a single Nvidia Tesla V100 GPU with a batch size of 16. The two HRNetV2 (C=48) are initialized with ImageNet pretrained weights and the convolution layers in the feature fusion block are randomly initialized.

The xBD test set containing 933 pairs of pre- and post-disaster images is used for the pseudo-label generation pipeline.

The final model is evaluated on the held-out data of the xBD dataset, and simulation results are computed using the Xview2 Challenge metric [187]. The F1 score (also known as balanced F-score or F-measure) is most appropriate to evaluate performance on this task because it presents several benefits: (i) it evaluates the weighted average precision and recall, (ii) it takes into account both false positives and false negatives (iii) it is more valuable than accuracy, especially if you have an uneven class distribution and, (iv) it reaches its best value at 1 and worst score at 0. The evaluation metric F1 localization and classification tasks are defined as:

$$F1 = 2\,x\,\frac{precision\,x\,recall}{precision + recall} \tag{6.5}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{6.6}$$

Where TP, FP, and FN are the number of true positive, false-positive, and false-negative pixels for the entire test set.

Figure 3 shows the results for localization and classification of building damage on a small random subset on the held-out data. The images selected are from different regions, with different disaster types, and type of building damage. Also, Table 4 shows the quantitative comparison to the xBD baseline results. Our method significantly outperforms the baseline as well as improves the performance of the model in classifying most damage types over other methods.

Table 6-4: Quantitative comparison of F1 scores with other methods. Results show that TS-HRNet + SSL outperforms the baseline method.

	Overall score	Localization	No damage	Minor damage	Major damage	Destroyed
xBD baseline	0.265	-	0.663	0.144	0.009	0.466
Weber and Kane [183]	0.741	0.835	0.906	0.493	0.722	**0.837**
TS-HRNet	0.645	0.836	0.857	0.370	0.664	0.774
TS-HRNet + SSL	**0.745**	**0.849**	**0.910**	**0.528**	**0.751**	0.790

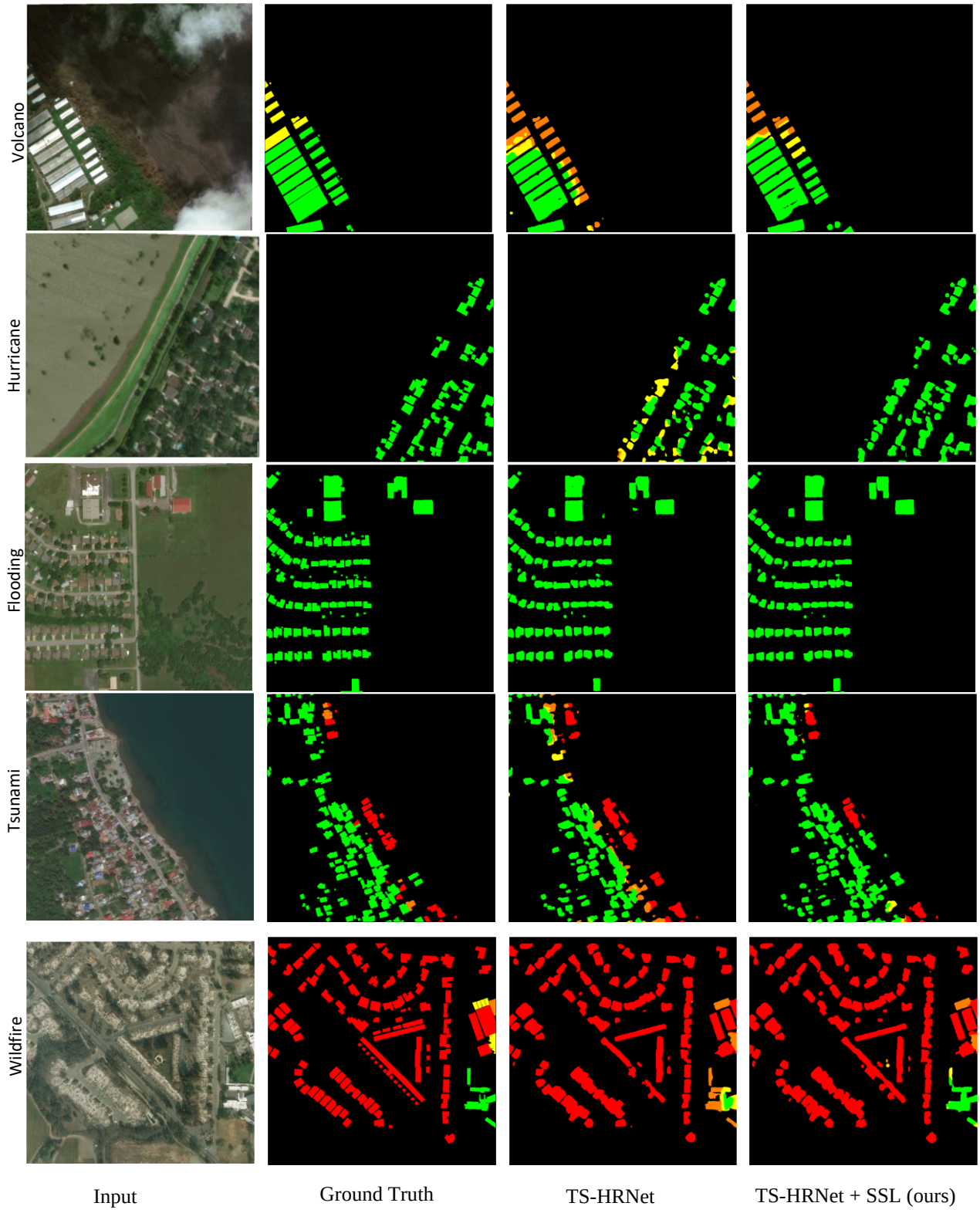


Figure 6-4: Localization and classification results for five disaster types. The buildings are colored based on the damage type and color mapping in Figure 6-5. The results show that the semi-supervised learning framework improves the classification performance of TS-HRNet as well as reduces class overlap.

6.4 Summary

This chapter presents a two-stream high-resolution network (TS-HRNet) for joint building localization and damage classification, and a semi-supervised framework for improving the generalizability of current methods for building damage localization and classification to unseen datasets that include other housing types. Qualitative, quantitative, and visual results show that this framework improves the performance of models when applied to data samples not present in the original data distributions. Models trained with an SSL pipeline performs better than supervised methods for building types from other regions not accounted for in the training dataset. Furthermore, we demonstrated the potential of our model on the xBD held-out dataset. Experimental results show a significant improvement in the performance of our model over the original baseline model. The work done in this paper will benefit applications such as disaster relief, search and rescue, and data analytics for satellite imagery.

Chapter 7. Importance of Data Augmentation for Multi-Task Learning

Deep convolutional neural networks are heavily dependent on big data to train good discriminative features to perform computer vision tasks such as object detection, recognition, segmentation, and tracking [188]. The availability of large amounts of diverse data prevents the model from overfitting to in-domain examples only and improves its ability to generalize to out-of-domain examples. Unfortunately, obtaining large, annotated datasets is not feasible for several application domains, one of which is satellite imagery. To help address this research problem, this chapter presents a novel technique for augmenting sparsely annotated satellite image datasets. Incorporating this augmentation step is shown to outperform the baseline model trained using other simple augmentation techniques such as geometric transformation, color jittering, or random erasing.

Additionally, an attention module is incorporated into the two-stream high-resolution network (ATS-HRNet) presented in Chapter 6 to exploit inter-spatial and inter-channel relationships of features in each stage of the network. Experimental results show the improvement of ATS-HRNet over TS-HRNet and other state-of-the-art techniques for building damage assessment from satellite imagery.

7.1 Overview

Data augmentation tackles overfitting and improves generalization by starting with the source of the problem; the training dataset. This is done based on the hypothesis that augmentations will extract more information from the original dataset [188]. Several methods have been proposed to solve the overfitting problem common in neural networks trained on small datasets. Srivastava et

al. [189] proposed a dropout regularization strategy that zeros the activation values of randomly selected neurons during training. Spatial dropout was proposed by Tompson et al. [190] to extend dropout to convolutional networks in which entire feature maps are dropped out rather than neurons. Batch normalization regularization strategy was proposed by Sergey et al. [191], and it involves normalizing the set of activations in a layer.

Because image datasets tend to share low-level spatial characteristics, transfer learning [192] can also provide a way to minimize model overfitting to limited data. It involves training a network on a large dataset such as ImageNet [193] and then using the learned weights to initialize the weights of the convolutional and fully-connected layers used in a different task. Erhan et al. [194] presented the concept of pre-training as a more flexible option than transfer learning. The learned weights from training on big datasets can be used to initialize a network's weights that are slightly different from the original network architecture design.

These methods focus on the model weights or feature level transformations for preventing overfitting. In this chapter, the focus is placed on the training dataset itself. Assembling large amounts of data with manually annotated ground-truth labels is a challenging task hence the need to develop novel strategies for maximizing information from limited data. To address this issue when working with satellite imagery, this thesis chapter introduces the following contributions:

a) Introduce a Spatial-Aware CutMix [195] augmentation strategy and a class-based oversampling technique to resolve the severe class imbalance that exists in the xBD dataset. This augmentation strategy is developed specifically for the task of building damage assessment from satellite imagery.

b) Incorporate an attention mechanism in the HRNet blocks in the two-stream network to improve the network's representation capacity for improved building detection and damage classification.

7.2 Data Augmentation

7.2.1 Spatial-Aware CutMix Data Augmentation Strategy

Figure 1 shows the distribution of the classes for the post-disaster images in the xBD dataset. The plot shows a severe class imbalance with the combined number of buildings with no damages being 10x more than any other class hence the need for a data preprocessing step to reduce the effect of this imbalance on the model's performance. This would prevent the network from being biased towards dominant classes.

Yun et al. [195] proposed the CutMix regularization strategy; patches are cut and pasted among training images where the ground-truth labels are also mixed proportionally to the area of the patches. This strategy has consistently outperformed other augmentation strategies for image classification and localization tasks.

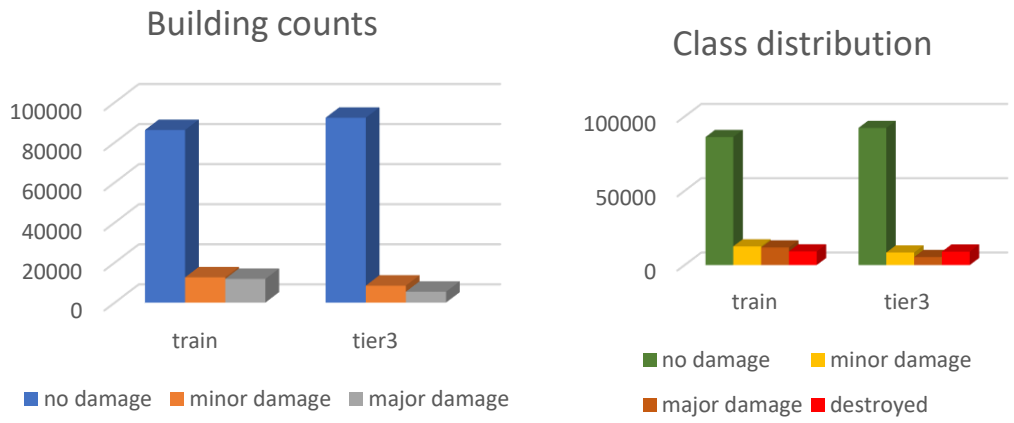


Figure 7-1: Shows the chart of (a) the number of buildings for images (b) building damage class distribution in train and tier3 folders.

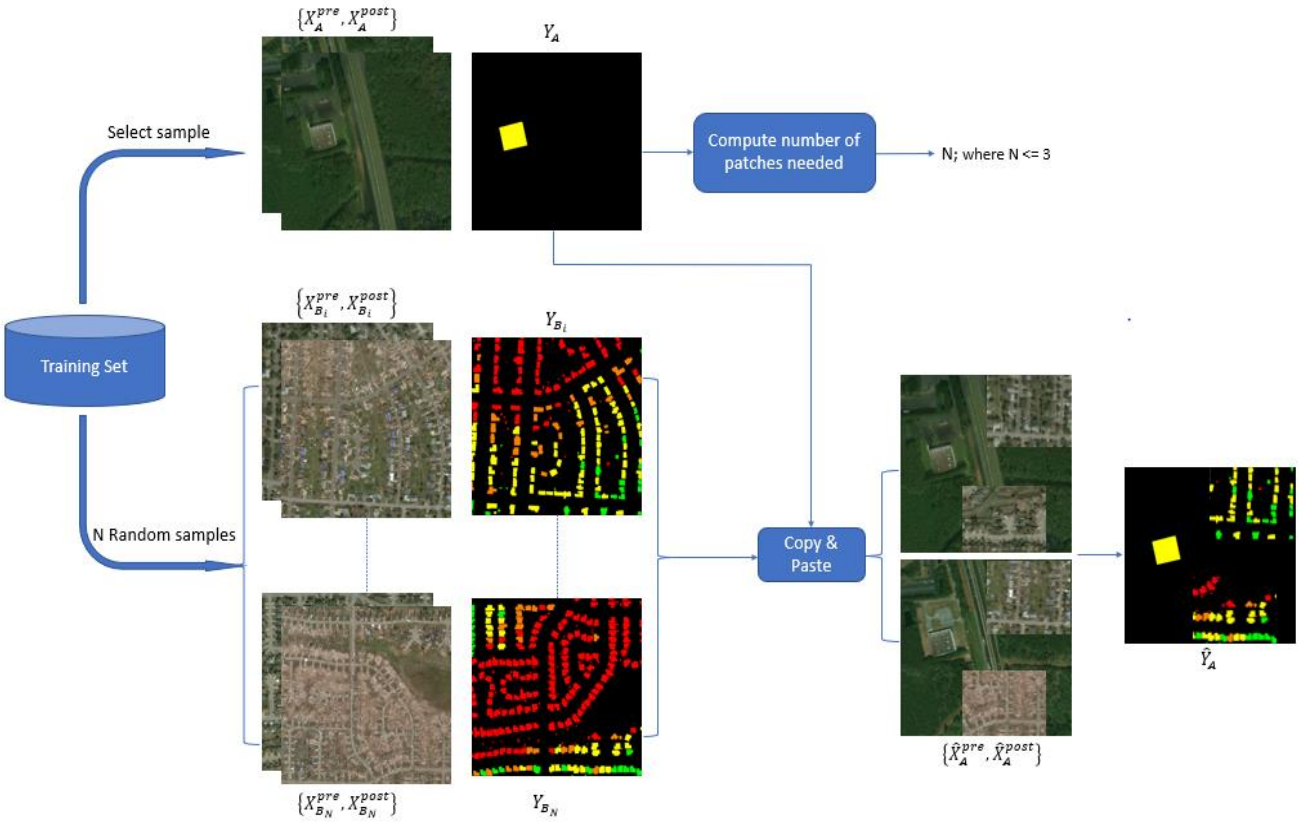


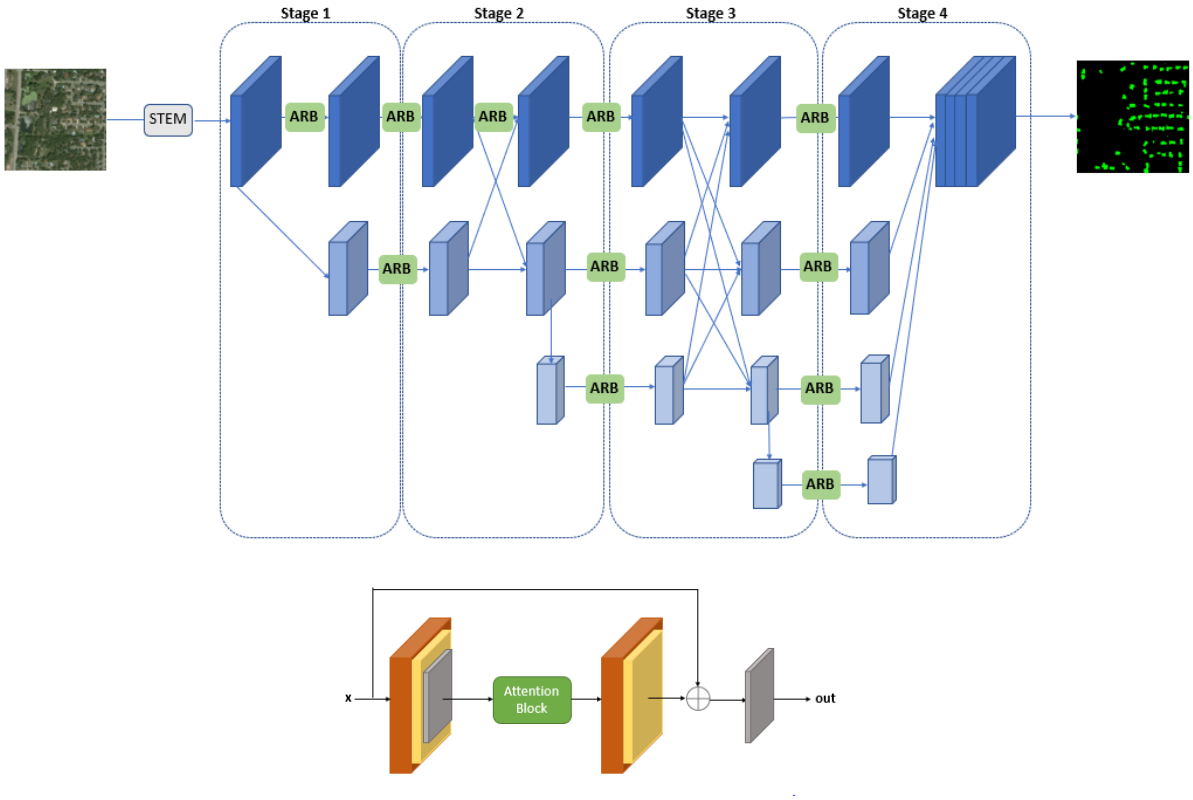
Figure 7-2: Proposed data augmentation pipeline using the spatial-aware CutMix technique

Shen et al. [196] adopted CutMix for data augmentation by performing the cut and paste strategy on random samples from the training set. Although the same strategy is adopted, a few modifications are made to ensure the strategy doesn't adversely impact models' performance. The spatial-aware CutMix data augmentation strategy is illustrated in Figure 2. Images with few buildings and annotations are selected from the training set, and this includes the pre- and post-image pairs X_A^{pre}, X_A^{post} and its corresponding label, Y_A. The number of patches needed, N is computed, and N random samples of densely annotated image pairs are selected for extracting N patches which are used to create the new image pairs $\hat{X}_A^{pre}, \hat{X}_A^{post}$ and label $\hat{Y}_A$. The stochastic approach to selecting the patches and pasting in areas of the image with no information present ensures that desirable information is not lost in the training process.

7.3 ATS-HRNet Model Architecture

7.3.1 Attention-Guided HRNet

The proposed AG-HRNet model replaces the residual blocks in the original HRNet [169] with attention-based residual blocks. The visual attention modules exploit inter-spatial and inter-channel relationships of features from each stage to extract spatial and channel maps. The channel attention maps extract features that are considered "meaningful" while spatial attention maps extract features that are considered as "informative" [197] [198].

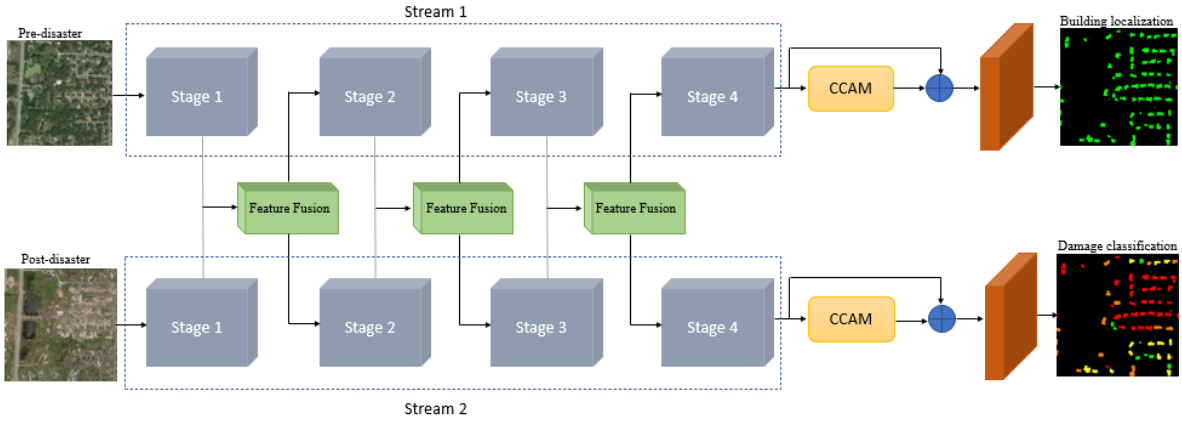


Attention-based Residual Blocks (ARBs)

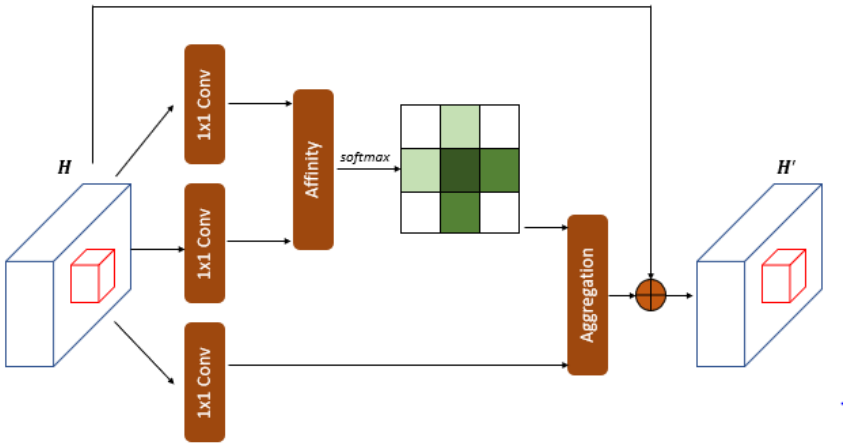
Figure 7-3: Proposed Attention-Guided High-Resolution network architecture.

7.3.2 Two-Stream Attention-Guided HRNet

Our network comprises of two AG-HRNet models, the first stream takes as input a pre-disaster image and localizes all buildings in the satellite image, while the second stream takes as input a post-disaster image pair and classifies the extent of damages on the localized buildings. The criss-cross attention module (CCAM) proposed by Huang et al. [199] is incorporated in each network stream for extracting contextual information from the images.



(a)



(b)

Figure 7-4: Attention-guided Two-Stream HRNet for joint building localization and classification. This architecture design is motivated by the work of Koo et al. [200] *(b) Criss-Cross Attention Module (CCAM), which helps in extracting contextual information of all pixels on its criss-cross path* [199]*. The multi-task learning framework helps in improving the data efficiency of the system by encouraging shared representations between building localization and damage classification.*

7.3.3 Pseudo-Label Generation

Similar to prior work presented in Chapter 6 for building classification [174], a semi-supervised approach is explored for harnessing information from unlabeled data. The number of unlabeled samples is frequently bigger than the labeled data. Also, building types differ substantially between areas and continents due to the climate or the culture of the people who reside there. Semi-supervised learning allows for the collection of vast volumes of previously unknown data from various places, enhancing the generalizability of existing models to different building types. According to the JDS standard, these labels comprise the location of the structures and the accompanying label. To remove noisy labels and improve the limits of the segmentation masks, label refinement is used. Label refining is done using a consistency-based method. The input images are subjected to four spatial level transformations: horizontal flip, vertical flip, transposition, and rotation, with the trained model utilized to localize and categorize buildings in each transformation. The inverse transform is applied to obtain the final output, and the most frequent class from all five outputs, including the findings from the untransformed picture, is determined. The results show that using a simple semi-supervised pipeline improves our model's performance on labeled and unlabeled data. A simple process for creating pseudo-labels is investigated for our task. The pseudo labels are generated using an ATS-HRNet model trained on the annotated datasets. Figure 10 shows the overall framework for extracting pseudo-labels.

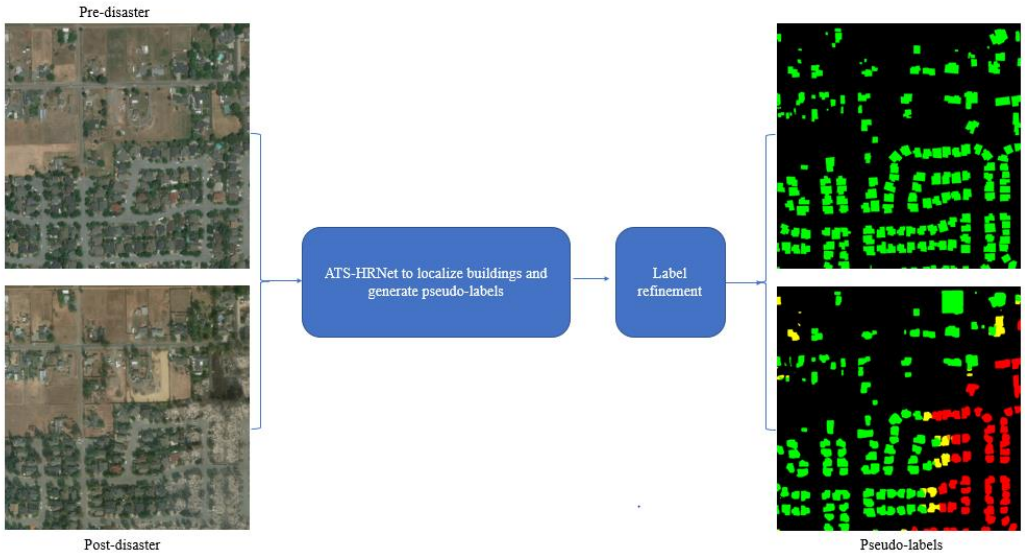


Figure 7-5: Pseudo-label generation pipeline takes a pair of pre- and post-disaster images and generates pseudo-labels of each pair [174].

7.3.4 Objective Function

The objective function adopted is a weighted combination of a supervised loss $L_{S\text{-}CLS}$ and a semi-supervised loss $L_{SSL\text{-}CLS}$. For the localization step, a weighted binary cross-entropy loss (L_{loc}) is adopted for training the network. This provides a weighting function for the two-class problem, i.e., building vs. nonbuilding.

$$L_{loc} = -(\alpha * ylog(\hat{y}) + (1 - y)log(1 - \hat{y})) \quad 7.1$$

α is set to 1.5 to assign higher weights for positive classes.

The classification loss is a weighted combination of the multi-label cross-entropy loss (L_{mce}) and the dice loss. The dice loss (L_{DL}) assesses the overlap between two regions and improves the ability of the model to generate finer boundaries for the segmentation task. As shown in equation 3, the numerator evaluates the overlap between the two regions at a local scale. At the same time, the denominator considers the total number of boundary pixels on a global scale [201].

$$L_{mce} = -\sum_{c=1}^{4} \alpha_c y(c) \log p(c) \quad 7.2$$

$$L_{DL} = \frac{2|X \cap Y|}{|X| + |Y|} \quad 7.3$$

Where,

X and Y are the ground truth and predicted label, respectively,

α_c = 0.5, 1.5, 1.0, 1.0 for class labels no damage, minor damage, major damage, destroyed.

y(c) and p(c) are the ground truth and predicted labels, respectively.

The classification loss is, therefore,

$$L_{S-CLS} = L_{mce} + \beta L_{DL} \quad 7.4$$

The total loss when training with unlabeled data

$$L_{total} = L_{S-CLS} + \gamma L_{SSL-CLS} \quad 7.5$$

$L_{SSL-CLS}$ is same as L_{S-CLS} but computed separately on the pseudo-generated labels.

7.4 Evaluation Metrics

Evaluation is performed on the 933 pre- and post-image pairs on the held-out data from the xBD dataset. Two metrics are used for evaluation: (i) the F1 score and (ii) the boundary intersection over union (IoU) [202]. The F1 score is adopted for evaluating the model's performance for localization and classification tasks. It takes into account four important values; the true positive value (TP) represents the number of pixel classes that are correctly classified, true negative value (TN) represents the number of negative classes correctly classified, false-positive value (FP) is the number of negative classes misclassified as a positive and false negative value (FN) is the number

of positive classes misclassified as negative. These metrics are shown in equations 7.6 – 7.9. Equation 7.10, $F1_{cls}$, is the harmonic mean of class-wise damage classification F1.

$$precision = \frac{TP}{TP + FP} \quad 7.6$$

$$recall = \frac{TP}{TP + FN} \quad 7.7$$

$$F1_{loc} = 2\,x\,\frac{precision\,x\,recall}{precision + recall} \quad 7.8$$

$$F1_{loc} = \frac{2TP}{2TP + FP + FN} \quad 7.9$$

$$F1_{cls} = \frac{n}{\sum_{i=1}^{n}\frac{1}{F1_{C_i}}} \quad 7.10$$

$\frac{1}{F1_{C_i}}$ denotes the *F1* score of each damage level (C_i) for damage assessment. Due to the tendency of the *F1* score to heavily penalize over-represented classes, the overall score, $F1_s$ provides a more comprehensive evaluation metric for building segmentation and damage assessment [176].

$$F1_s = 0.3\,x\,F1_{loc} + 0.7\,x\,F1_{cls} \quad 7.11$$

The boundary IoU measures the quality of the segmented boundaries. This is an additional metric used to evaluate the model's performance on the building localization task.

$$Boundary\ IoU = \frac{|(G_d \cap G) \cap (P_d \cap P)|}{|(G_d \cap G) \cup (P_d \cap P)|} \quad 7.12$$

Where:

G is the ground truth mask

P is the predicted mask

G_d, P_d are the set of pixels in the boundary region of the mask

d is the pixel width of the boundary region

7.5 Experiments and Results

The experimental setup is same as Chapter 6. However, SA-CutMix provided an additional set of training dataset with 1093 pairs of pre- and post-disaster satellite images. Figure 7-6 shows image samples from SA-CutMix-ed data.

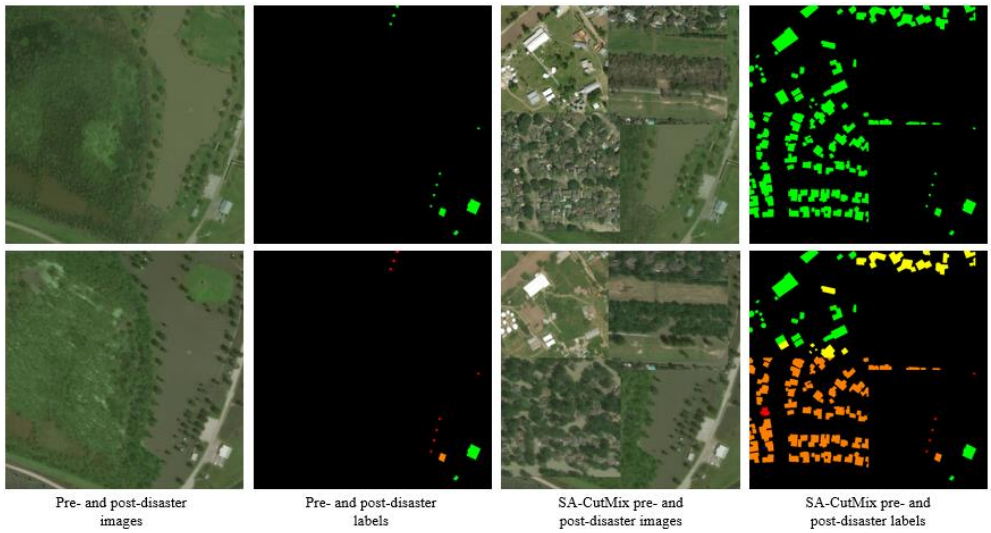


Figure 7-6: Visualization of the image pairing using the proposed SA-CutMix technique. Images in this figure are illustrative examples of pre- and post-image pairing. (a) and (b) are two separate examples. The first two columns for each example represent the pre-and post-disaster images and the corresponding labels, while the third and fourth columns illustrate the post-disaster images and associated labels.

As shown in Figure 7-6, SA-CutMix is able to preserve most of the information contained in the original images while randomizing the number of image patches needed for copy-pasting.

Table 7-1 shows the quantitative evaluation of the performance of ATS-HRNet to other state-of-the-art approaches. Figure 7-7 and Figure 7-8 show the results for localization and classification of building damage on a subset of the held-out data containing groundtruth annotations. Images shown in Figure 7-7 are selected from different regions, with different disaster types, building density, and class of building damage. ATS-HRNet + proposed SA-CutMix significantly outperforms the baseline model as well as other joint localization and classification models.

Table 7-1: Quantitative comparison of F1 scores with other methods. Results show that ATS-HRNet + SA-CutMix outperforms the baseline method.

	Overall score	**Localization**	**No damage**	**Minor damage**	**Major damage**	**Destroyed**
xBD baseline	0.265	-	0.663	0.144	0.009	0.466
Weber and Kane [183]	0.741	0.835	0.906	0.493	0.722	0.837
TS-HRNet	0.645	0.836	0.857	0.370	0.664	0.774
TS-HRNet + SSL	0.745	0.849	0.910	0.528	0.751	0.790
ATS-HRNet	0.756	0.880	0.920	0.629	0.782	0.864
ATS-HRNet + SA-CutMix	**0.778**	**0.892**	**0.928**	**0.640**	**0.796**	**0.875**

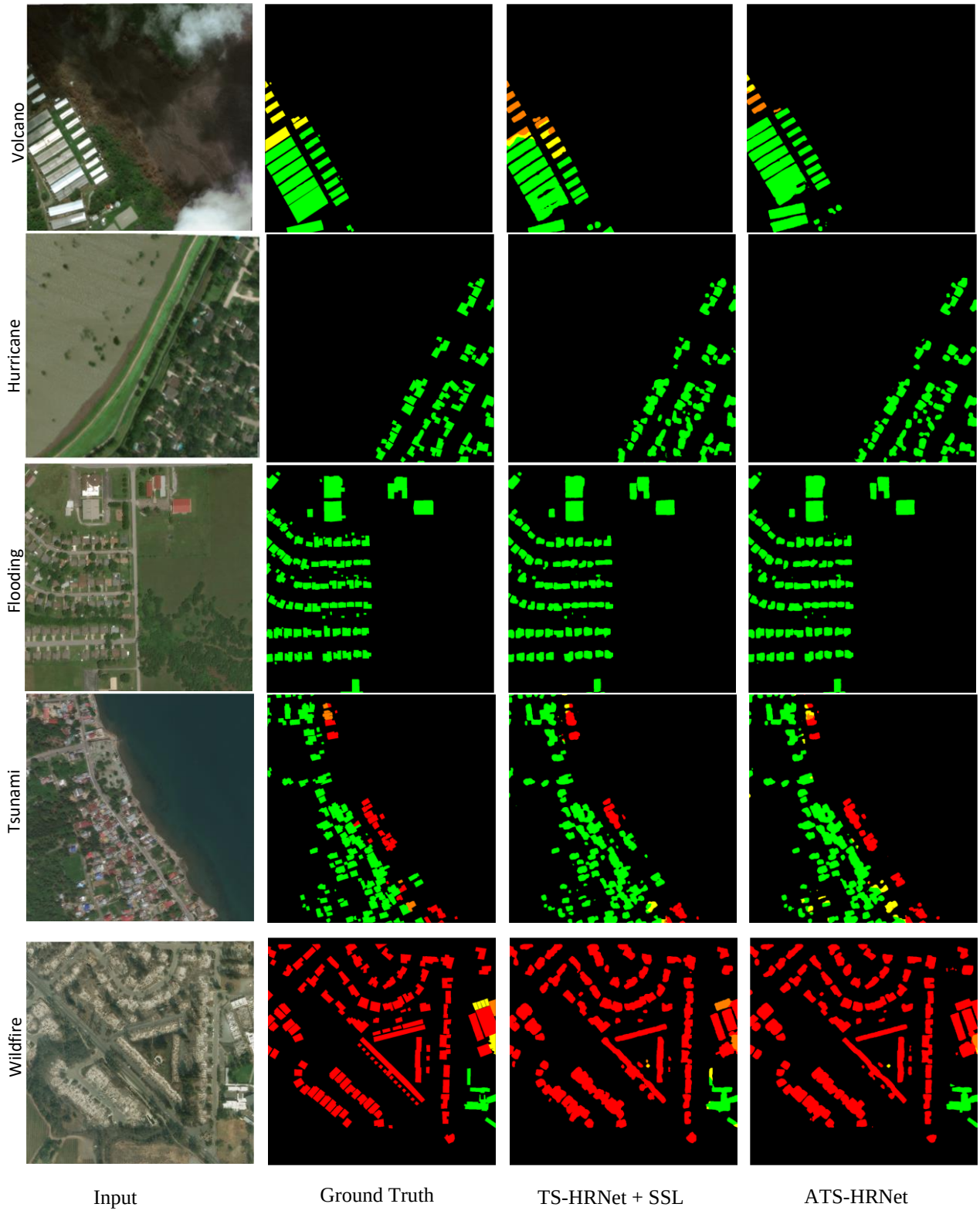


Figure 7-7: Localization and classification results for five disaster types. The buildings are colored based on the damage type and color mapping in figure 1. The results show that the ATS-HRNet still outperforms TS-HRNet trained with pseudo-labels presented in Chapter 6 especially when predicting less dominant classes.

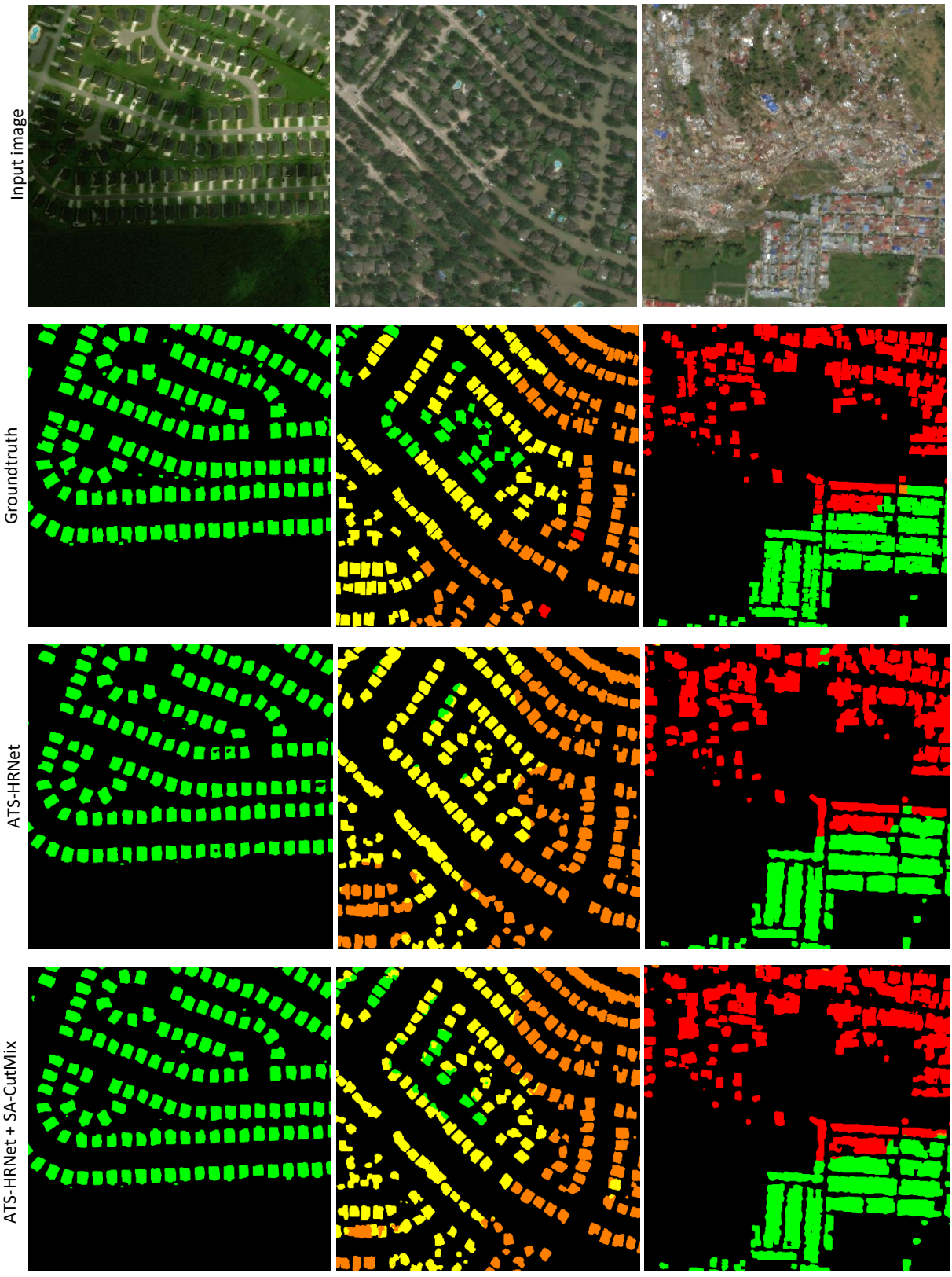


Figure 7-8: Localization and classification results presenting the benefits of SA-CutMix as a data augmentation strategy. These results show that SA-CutMix helps reduce the amount of misclassified building damages and force finer label assignment.

7.6 Summary

This chapter introduces a spatial-aware CutMix strategy for effective data augmentation and an attention-guided high-resolution network (ATS-HRNet) for joint building localization and damage classification. Furthermore, a semi-supervised approach for improving the model's performance for the localization and classification tasks using a set of unannotated image pairs is also presented. Our ablation experiments on the xBD and Inria datasets demonstrate the effectiveness of the augmentation strategy and ATS-HRNet for effectively localizing buildings of various types and damage classification. The model's performance is also compared to current state-of-the-art approaches, and the results show that ATS-HRNet + SA-CutMix significantly outperforms other methods across all evaluation metrics adopted.

Model evaluation for building localization on the Inria dataset shows some limitations of our strategy. Building styles vary significantly across regions and cultures; hence, it is impractical to account for the vast variability in building structures. This explains the drop in F1 score on the Inria dataset compared to the xBD dataset. As part of the future work, the framework would be trained on more datasets targeted towards building localization from satellite imagery across more regions, such as the African building dataset provided by Sirko et al. [203], to fully evaluate the potential of the proposed approach.

PART 3: BUILDING EFFICIENT REAL-TIME HEALTH MONITORING TECHNOLOGIES

Chapter 8. Human Body Weight Estimation using Smart Shoe Insoles: Data Preprocessing, Analysis and Predictive Modeling

The lack of real-time monitoring systems for accurately estimating changes in weight has resulted in inadequate measures being taken to mitigate the effects of lifting heavy loads over time, due to improper dietary and lifestyle patterns. Wearable devices have been developed to provide a convenient and efficient way to obtain valuable data points for several human-related applications. Smart shoe insoles equipped with pressure sensors provide an efficient method for performing gait analysis which may be necessary in analyzing the locomotive pattern of medical patients and detecting potential issues.

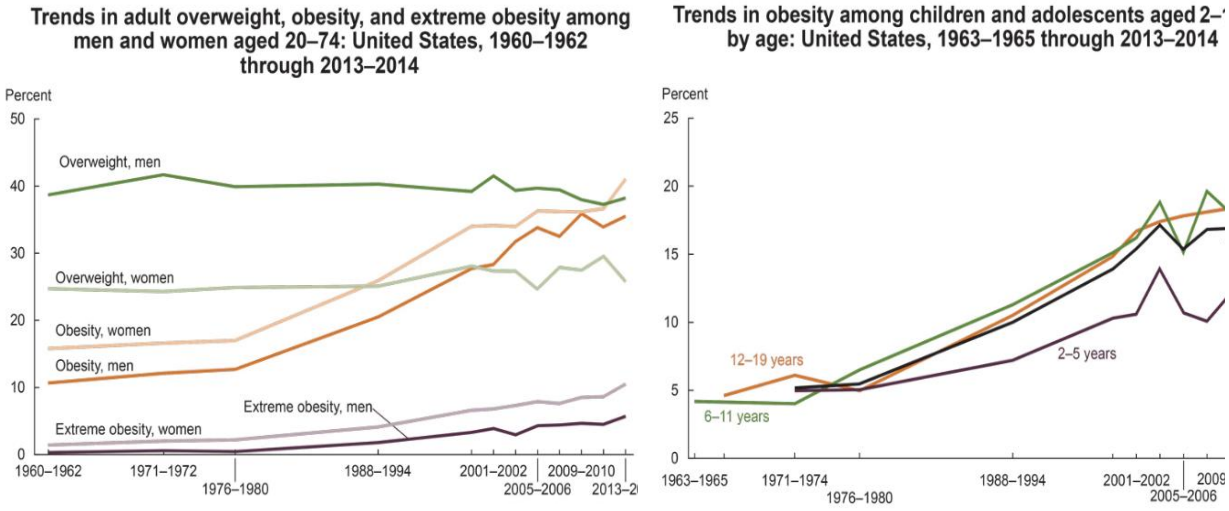
This chapter aims to harness the data extracted from smart show insoles to (i) predict the activity of an individual, including walking, sitting, running, and standing, and (ii) estimate the body weight of the individual when a standing activity is detected. Each shoe insole contains sixteen pressure sensors, three acceleration sensors, and three angular rate sensors. Because several factors affect the data obtained from the sensors, such as shoe type, posture, foot shape, and standing surface, our framework is designed to adapt to individuals rather than an entire sample population. Our framework can estimate an individual's body weight with a maximum deviation of ±3.6lbs and an average error rate of 1.4%. Also, two datasets are proposed as part of this work; The first dataset includes data collected from 30 participants in a less controlled environment in which the standing surfaces varied. The second dataset contains data collected from 10 participants in a controlled environment. Each participant is made to stand on the flat surface of a weighing scale to reduce the effect of noise introduced when changing surfaces like standing on a hardwood floor or tiles.

8.1 Overview

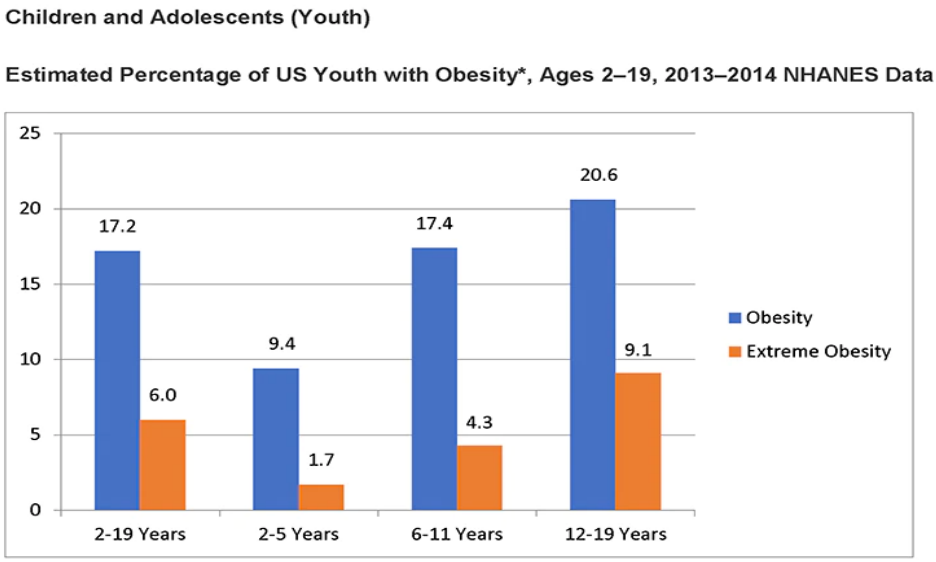
Real-time health monitoring is particularly important in maintaining good health and providing prompt response whenever an anomaly is detected. Wearable devices and sensors such as smartwatches and insertable insoles provide effective ways of retrieving meaningful data that can be used to train artificial intelligence models. While wearable device technologies have transcended to real-time monitoring for several health applications such as heart rate monitoring, sleep pattern monitoring, and oxygen level monitoring, such advancements are yet to be applied to real-time weight monitoring applications. In the context of human activity monitoring, gait analysis provides a means for achieving this and is commonly applied in medical and sporting applications. It involves a systematic study and measurement of human activities [204]. Pressure sensors, such as the pedar, provide an efficient way of estimating dynamic pressures and force distribution, making them suitable in manufacturing smart insoles [205].

The research work presented in this article explores an artificial intelligence-based solution to real-time weight monitoring using shoe insole sensors and federated learning algorithms. The contributions in this paper are as follows:

a) First two novel experimental datasets for weight monitoring and human activity identification and classification are created. The first dataset consists of data collected from 30 participants in uncontrolled settings while the second dataset consists of data collected from 10 participants in a controlled setting. Data is collected from each participant for different activities including, standing, walking, and sitting.



(a) (b)



(c)

Figure 8-1: Statistical data showing the progression of obesity rates among the US population. (a) and (b) show a trend in the obesity rate among adults and adolescents in the US, while (c) compares the severity level of obesity among adolescents [206].

b) Secondly, a sophisticated federated learning-based model is proposed for automatic body weight estimation from shoe insoles sensor data retrieved in real-time. The proposed model is shown to achieve a minimal error rate of 1.4% with a maximum deviation of about 3.6lbs. Extensive data analysis is also presented to support findings.

c) Finally, a framework solution is proposed to classify human activities (standing, sitting, and walking), and automatically track weight changes in humans. For optimal performance and noise reduction, weight measurements are only validated while the subject is in a standing position.

With about 1.4 million older adults living in nursing homes, such weight monitoring technology could prove extremely helpful in keeping an extra tab on the health status of loved ones.

The problem of body weight estimation using shoe insole sensors has recently gained popularity in the research community. A few researchers have conducted numerous experiments and proposed several techniques for body weight estimation. Tan et al. [204] conducted a research experiment in 2018 in which they tried to validate the forces measured by sensors in commercial shoe insoles. Using the Pedar-X, they collected data from a single participant for six different activities over time, including limping, slow speed walk, medium speed run, fast speed run, medium speed walk, and fast speed walk. Tan et al. [204] further proposed a mathematical model for estimating the weight by measuring the average force applied to all pressure sensors from both legs and computing the mean over a finite period. Tan et al. [204] reported a 6% to 10% error rate which could be mean a deviation of up to 20 lbs for a 200 lb candidate.

Muzzafar et al. [205] championed that existing shoe insole sensors fail to account for the effect of the ground in the sensor layout pattern. The error rate in measurements retrieved from essential pressure points is often very high. To alleviate this, Muzzafar et al. proposed a novel ergonomic

shoe insole design which involves sandwiching several sensors in a three-layered pattern for optimized and more robust force measurements.

Moon et al. [207] investigated the negative impact of postural instability and fatigue on the sensor data, increasing the error rate in downstream weight estimation modeling. Moon et al.'s experimental simulations suggest that the accuracy of data retrieved from pressure sensors embedded in shoe insoles can be greatly improved by introducing stochastic resonance to effectively propagate force across all the sensor arrays. To this effect, they proposed a sub-sensory insole vibration mechanism to reduce fatigue and imbalance posturing and restore the center area of pressure, yielding more accurate force measurements from the shoe insole. Moon et al. validated their findings through posture experimentation on 21 young adults with varied posturing.

Mey et al. [208] investigated the use of force plate data as input to mathematical and statistical models for automatically estimating and monitoring the weight in turkeys. Particularly, the authors explore the Bayesian learning approach to weight estimation and proposed intuitive techniques for arranging force plates and collecting data for optimal performance.

Visual cues have also been investigated to build models for human body weight estimation [209], [210]. Velardo et al. [209] studied the feasibility of weight estimation from anthropometric features known to be related to human appearance and correlated to weight. Multiple regression analysis is adopted for building a model on a set of anthropometric features.

Existing weight estimation methods using pressure sensors from shoe insoles are plagued with several challenges that limit their usefulness in real-life applications. This includes a) low accuracy, high error rate, and high standard deviation, b) poor experimental setup and data

collection strategies, c) a limited number of participants, d) non-robust data analysis and evaluation methodologies.

To address these challenges and limitations, this chapter introduces a more robust data collection, analysis, and evaluation methodology. First, we introduce the largest and most comprehensive dataset with over 30 participants performing numerous activities in controlled and uncontrolled settings. Next, we adopt a federated learning approach which can automatically learn to understand data patterns specific to each individual based on their postures, pressure points, and stability. The federated learning approach can automatically adjust the prediction mechanism by retraining itself on the fly in order to make more accurate predictions for every individual. Our experimental simulations achieve high accuracy and the lowest error rate and mean deviation compared to other state-of-the-art predictive models.

8.2 Methodology

8.2.1 Data Acquisition

A Moticon insole [211] is a commercially available shoe insole fitted with pressure sensors that is currently used for generating pressure maps of the foot for athletes and medical purposes. It comes with a portable battery for providing power and Bluetooth data transmission between a computer and a mobile phone. Each insole contains a total of 16 pressure sensors. The data retrieved by each pressure insole include:

i. The timestamp which depends on the sampling rate (ms).

ii. The pressure values from all 16 sensors (N/cm^2).

iii. The acceleration in the x,y,z axes (g)

iv. The angular rate in w_x, w_y, w_z (dps)

v. The computed center of pressure in the x,y coordinates (insole length/width).

v. The total force (N).

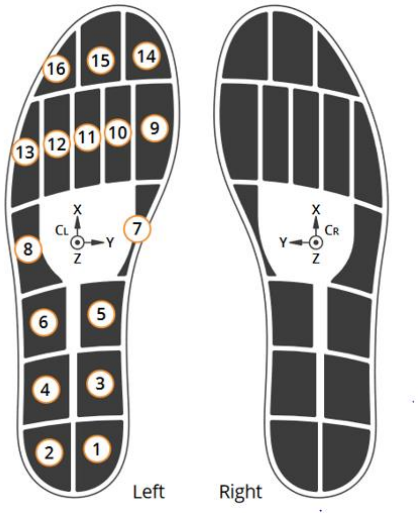


Figure 8-2: Pressure sensor distribution on the Moticon shoe insole [211]

The functionality and validity of the pressure sensors, as well as the accuracy of data transmitted has been described in [212].

8.2.2 Data Collection Details

In our first study, participants completed a number of tasks that lasted approximately 45 minutes involving standing, walking, and jumping with and without a graded series of dumbbell weights weighing up to 15 lbs. The experiments are conducted in different settings; hence two datasets are collected. The first dataset includes data collected from 53 participants, with ages ranging from 18 to 70 years, in a less controlled environment in which the standing surfaces varied (wooden, tiled, or concrete surfaces). Data collection is done once for each individual in the first experiment. Table 1 shows the data distribution for participants in this experiment.

The second experiment includes 9 participants with age ranging from 20 to 32 years. Each participant's readings are collected for the standing position only, and the recordings are taken while standing on the Withings scale to prevent fluctuations in readings due to changes in surfaces. An average of three iterations are conducted per participant with the dumbbell weights' increasing and decreasing weights. Table 2 shows the data distribution for participants in this experiment.

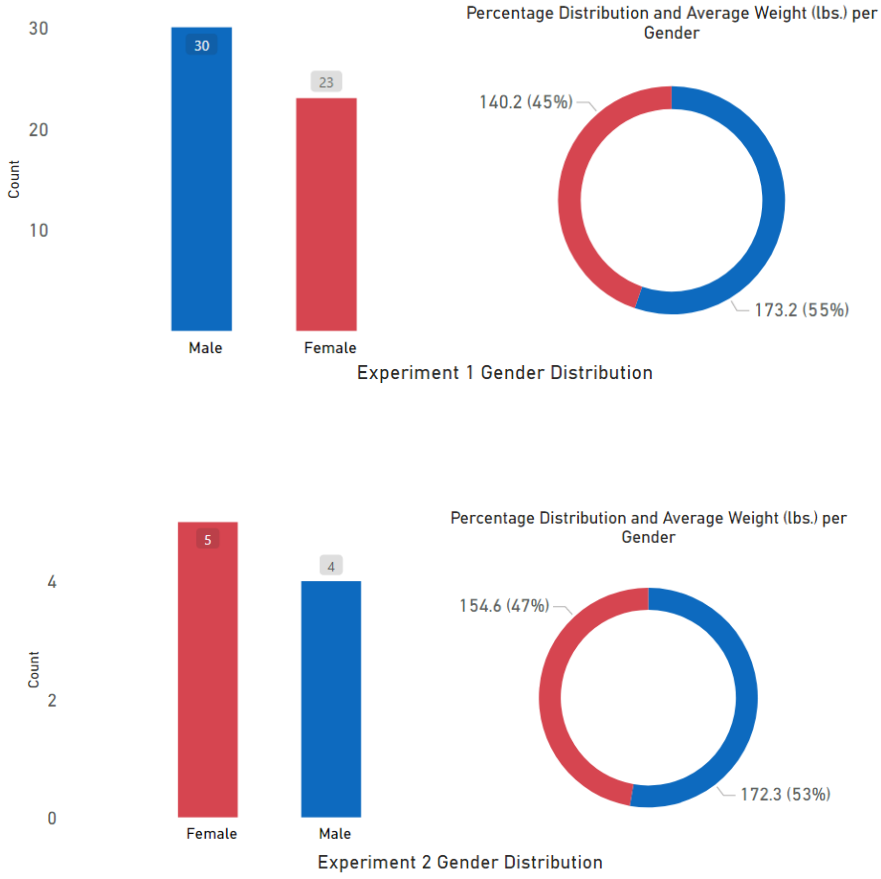


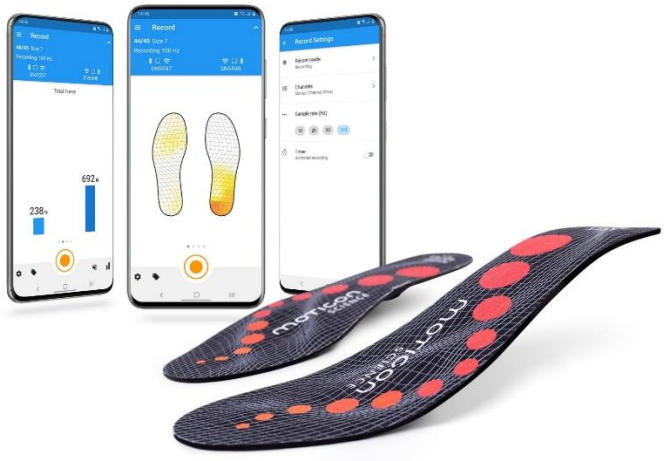
Figure 8-3: Data distribution for participants in experiment 1 (top row), and Data distribution for participants in experiment 2 (bottom row). 23 females and 30 males participated in experiment 1, while 5 females and 4 males participated in experiment 2.

The total number of features from both insoles is 50. The ground truth weight of the individual is obtained from a smart weighing scale. The sampling frequency for data collection is set at 100Hz. First, the participant's weight will be manually taken on a Withings scale to create the groundtruth

data. Next, sensor measurements are collected through the insole sensors. This process is repeated with varying weight sizes to create more diversity in the dataset.

(a)

(b)



(c)

Figure 8-4: Shows images of (a) the Moticon show insoles, (b) the Withings scale and dumbbell weights for varying differences in the load carried by the participant, and (c) Mobile software interface for retrieving and visualizing pressure sensor data via Bluetooth communication.

For each participant, data is collected in 3 or more successive iterations, with every iteration requiring calibration of the sensors. An initial calibration is performed on the sensor with respect to the participant, and the sensor data is then captured along with the ground truth weight in pounds. This procedure is carried out with the participant's base weight and repeated with the participant lifting dumbbells of weights 2, 3, 5, 7, 9, 11, and 13 lbs. These repetitions are carried out to yield a more robust dataset and eliminate noise.

8.2.3 Exploratory Data Analysis

Data exploration provides a better understanding of the data and guides the development of novel algorithms and frameworks for personalized predictions. Exploratory data analysis is carried out on a subset of the dataset collected in experiment 2. This is because multiple independent samples are collected for each participant, and these samples are not guaranteed to be good due to several factors related to posture, shoe type, improper calibration, and moisture on the shoe insole.

Of the 9 participants in experiment 2, 6 participants (3 males and 3 females) were selected for the exploratory analysis. The most strongly correlated output feature to the groundtruth weight measurements are the right and left total forces measured by the shoe insole. This correlation provides an intuitive approach for distinguishing between a noisy/unreliable data sample and a good sample. A data sample is considered to be reliable if a largely positive correlation exists between the forces measured by the sensor and the ground truth weights. Figure 8-5 and Figure 8-6 show a plot of left/right force vs. groundtruth weight for 6 participants.

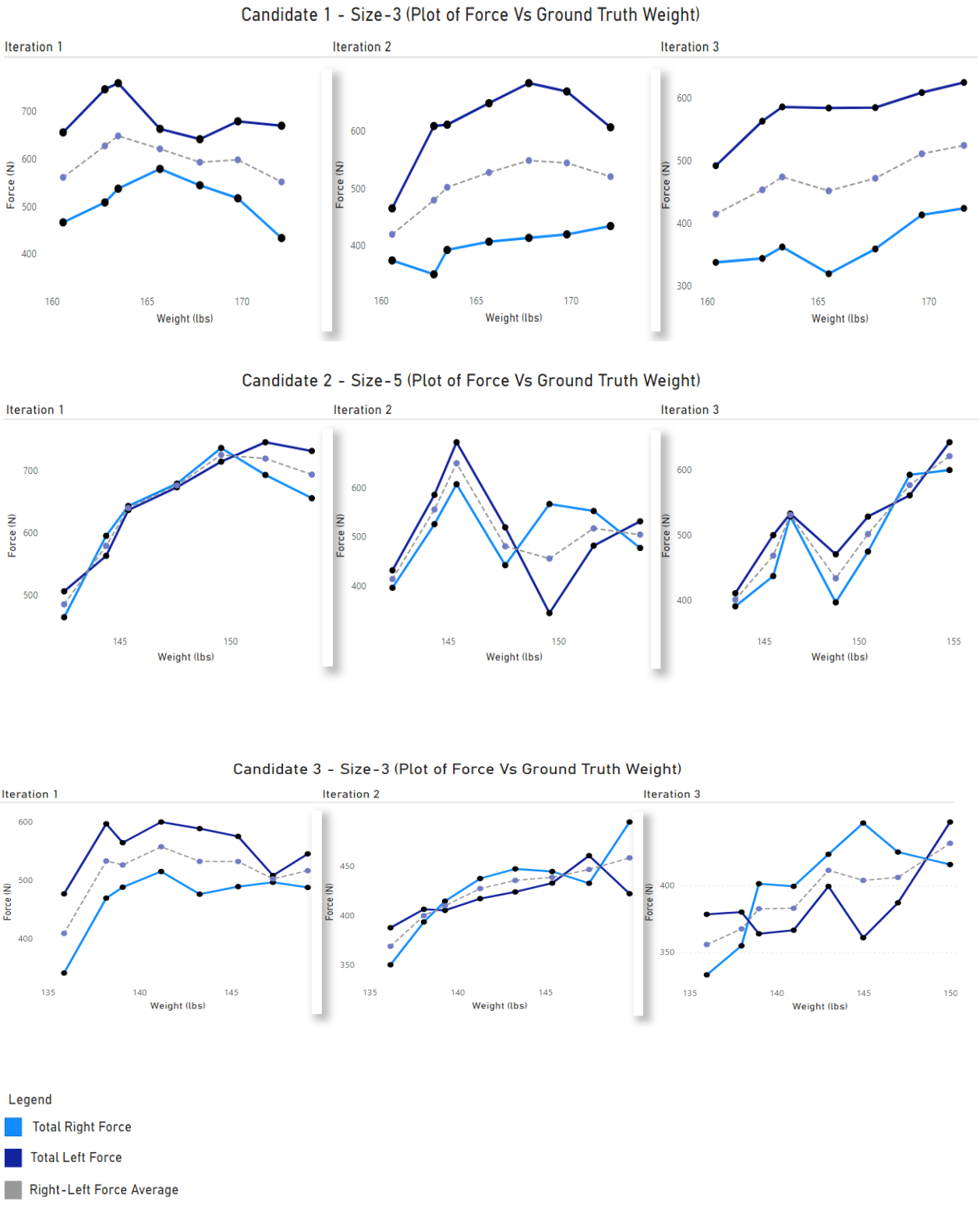


Figure 8-5: Plot of total left and right forces vs. ground truth weights measured using a Withings scale. The three rows correspond to data collected for three female candidates. The desired trend is a positively correlated graph between the x and y-axis; hence, for candidate 1, iteration 1 is discarded; for candidate 2, iteration 2 is discarded; and for candidate 3, iteration 3 is discarded. Participants holding different dumbbell weight sizes provided a means for varying the weight of individuals.

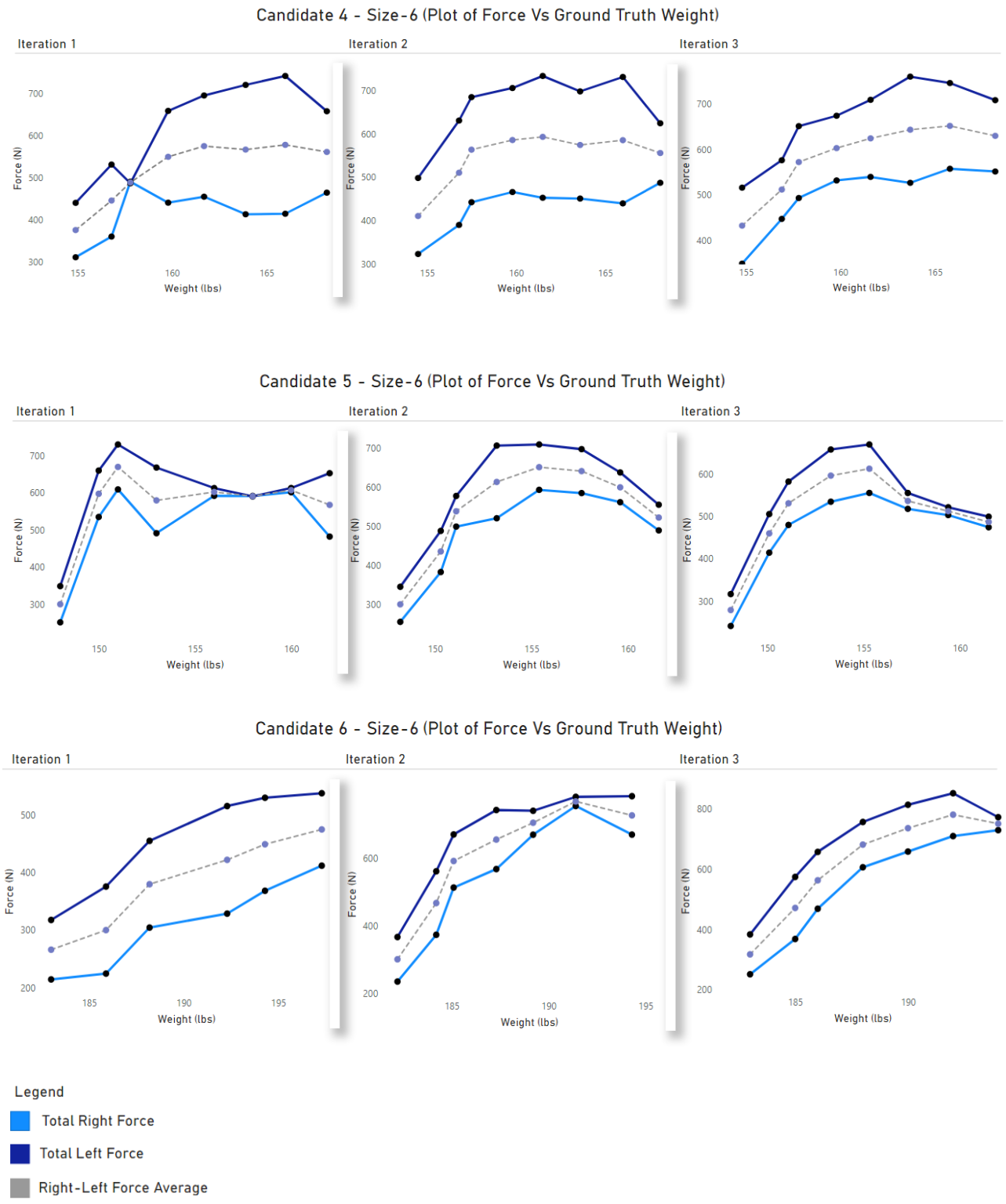


Figure 8-6: Plot of total left and right forces vs. ground truth weights measured using a Withings scale. The three rows correspond to data collected for three male candidates. The desired trend is a positively correlated graph between the x and y-axis; hence, for candidate 1, iteration 1 is discarded; for candidate 2, iteration 1 is discarded; and for candidate 3, none is discarded.

From the plots, it can be observed that the insole sensors tend to act unreliably when reaching higher weight readings for a participant, particularly when lifting dumbbell weights of 11 lbs and 13 lbs. This could be due to the sensor reaching its saturating point or the standing posture of the individual. An example of this is evident in iteration 1 of Figure 8-5 where the total force decreases as the weight carried is increased. Samples identified with this pattern are not used in the training and testing phases. The exploratory data exercise helps prevent noisy data samples from degrading the predictive ability of the machine learning model.

8.3 Predictive Modeling

8.3.1 Dynamic Windowing-based Feature Extraction

The size of data collected for each participant/dumbbell weight varied due to the length of time taken to capture each data point. This mismatch in data size across measurements can result in bias in the machine learning model developed for bodyweight prediction. Hence, we create a simple dynamic windowing approach for extracting features at multiple intervals for standardizing the training dataset.

The sliding window approach and its variants have been used for several outlier detection and prediction tasks [213]–[217]. The process of the sliding window is depicted in Figure 8-7 9-7.

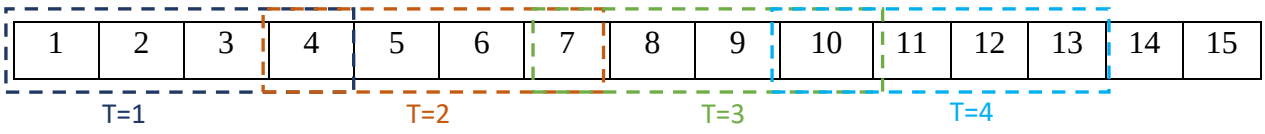


Figure 8-7: Process of sliding window at various time intervals and with some overlap.

Unlike the current adaptive sliding window, which often does not account for temporal dependencies, our dynamic windowing approach automatically estimates an overlap ratio between windows for each weight category. Equation 8.3 shows the formula for dynamic windowing

$$W_{init} = \frac{length\ of\ sequence}{number\ of\ samples} \quad 8.1$$

$$r_{new} = r_{init} * W_{init} \quad 8.2$$

$$W_{new} = \frac{length\ of\ sequence + r_{new} * W_{init}}{number\ of\ samples} \quad 8.3$$

Where,

W_{init}, r_{init} are the initial window length and initial overlap rate, respectively, which are hyperparameters defined by the user

W_{new}, r_{new} are the new window length and new overlap rate computed

The length of each sequence corresponds to the number of data samples collected for a particular weight category, while the number of samples is the approximate length of samples contained in each window.

By ensuring the window sizes are dynamically set, one doesn't need to empirically determine the best window to use since this is not practical in real-world scenarios. It is worth noting that the dynamic window doesn't guarantee the best results but provides a reasonable performance trade-off over time needed for extensive parameter search.

The data contained in a window is averaged across the feature space to form a single data point for creating a dataset corresponding to each individual for each weight category.

8.3.2 Machine Learning-Based Predictive Modeling

In this approach, the capacity of well-known machine learning regression models is trained and evaluated on our dataset. After several experiments, an open-source AutoML library called Pycaret [218] provided a quick and efficient tool for creating our data preprocessing pipelines as well as training and testing multiple machine learning models based on our hypothesis. Tree and Boosting-based methods are explored. They outperform other regression-based methods, but they also provide better insight for interpreting the features that affect the model's performance for individuals. The training pipeline is shown in Figure 8-8.

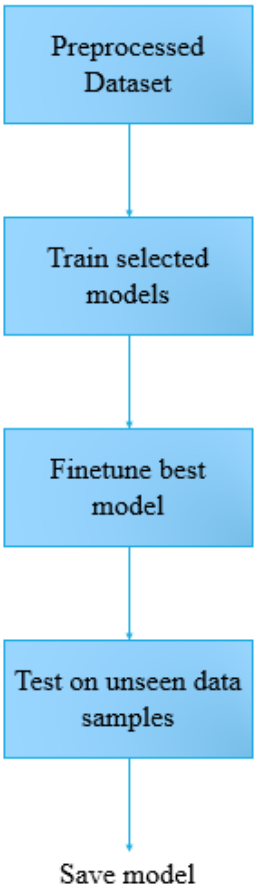


Figure 8-8: Training pipeline using Pycaret machine learning library. The preprocessed data extracted using our dynamic windowing approach serves as input to the model. Finetuning helps obtain the best hyperparameters for the individual, and finally, the model is tested on unseen data collected for a participant.

A simple deep neural network with fully connected layers is also trained on the dataset to provide a good comparison to Pycaret's model performance.

8.4 Experiments and Results

The goal of our model is to estimate a human body weight given a set of features obtained from a smart shoe pressure insole. This can be formulated as

$$weight = w\,x \sum_{i=1}^{N} Features(SI_i) + b \qquad 8.4$$

Where,

w is the learnable weight

b is the bias term

SI are the features from the left or right shoe insole

5-fold cross-validation is applied to validate the performance of each machine learning model adopted. Training and testing datasets for an individual are obtained from a set of experiments classified as 'good' data. For example, Figure 8-9: Exploratory data analysis for a participant with 8 independent experiments conducted over several days. shows the plots for a participant with 8 independent experiments carried out over several days.

Because most data collected show a positive correlation between the left and right forces and the ground truth weight, 4 samples are used for training and 4 for testing. Specifically, data samples from iterations 1, 2, 3, and 6 provide the training examples, and 4, 5, 7, and 8 provide the test examples.

For this experiment, the initial window size was set to [25, 100], and the initial overlap ratio was set to 0.2.

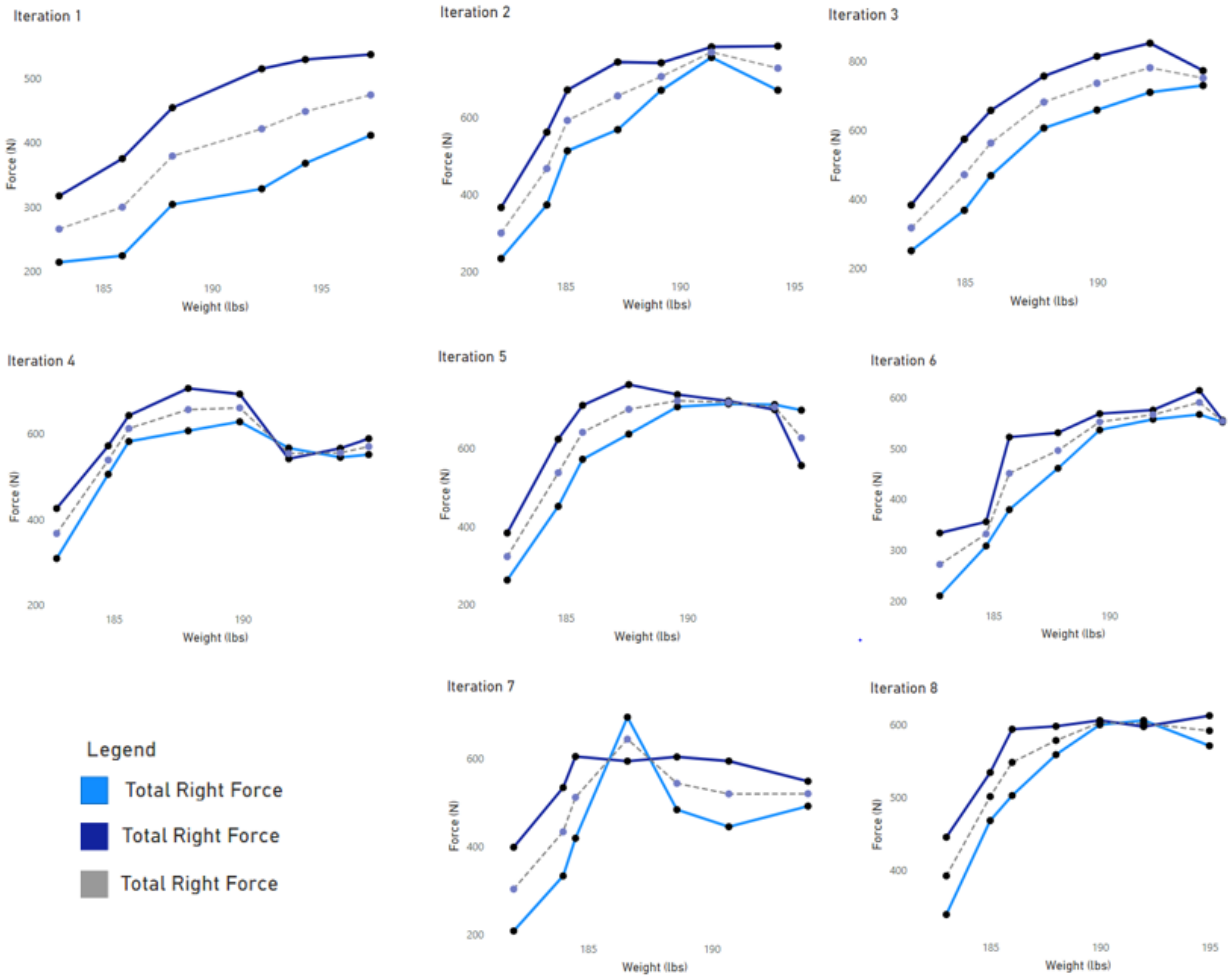


Figure 8-9: Exploratory data analysis for a participant with 8 independent experiments conducted over several days. The plots show a graph of Force (N) vs. groundtruth weight (lbs). A positive correlation between the forces and measured weight is desired and gives a simple indication of the usability of a data sample for training the model.

The results obtained from this experiment are presented in **Figure 9.**

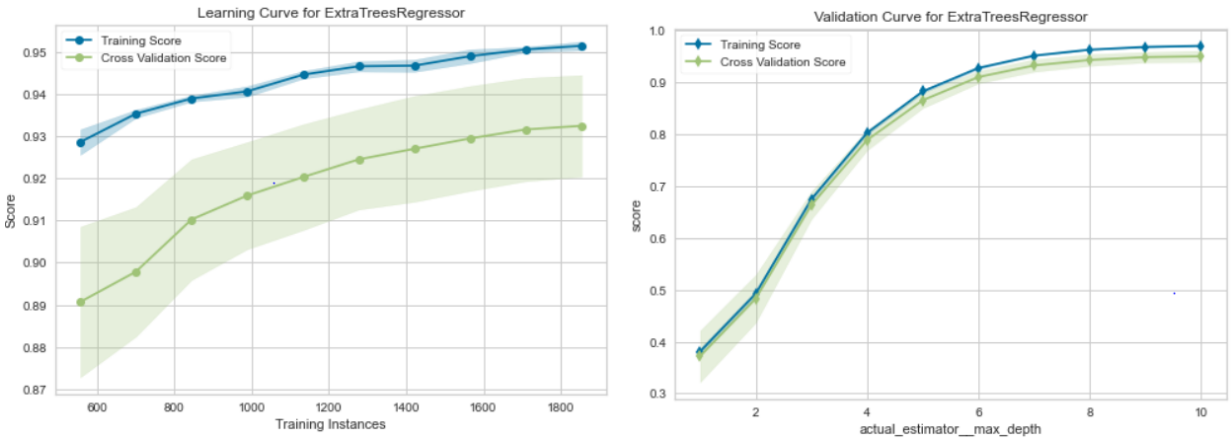


Figure 8-10: Plot of the mean absolution error (MAE) vs. number of training instances (left) and the MAE vs. tree depth (right). This shows that increasing the number of training instances improves the model's predictive ability. A maximum tree depth of 8 is sufficient for this candidate to obtain the best results.

Figure 8-10 shows the plots of the mean absolute error vs. the number of training instances on the left, and the right plot represents the effect of increasing the tree depth on the MAE score. The left plot shows that increasing the number of training instances significantly improves the error rate of ExtraTreesRegressor model. The right plot indicates that a tree depth of 8 is optimal for obtaining the best error rate, reducing the training time and computational cost incurred when performing real-time inference.

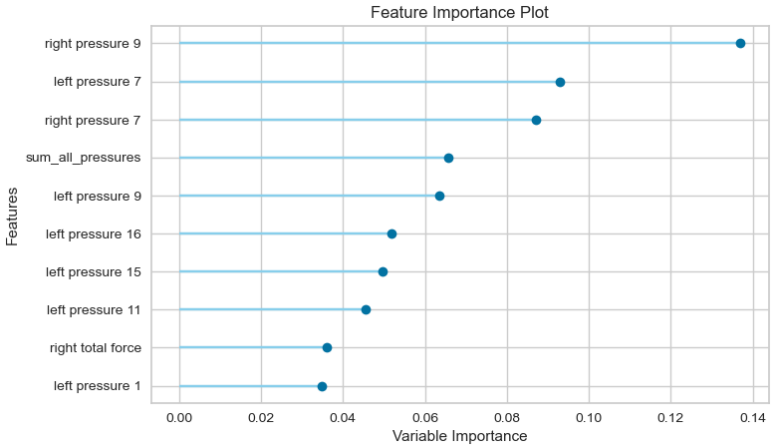


Figure 8-11: Plot of feature importance for participant 6. This ranks the features that contribute most to the performance of the model. The top 10 ranked features are represented in this figure and vary for each candidate.

Figure 8-11 shows a visual example of the feature ranking for candidate 6. The number label corresponds to the position of the sensor in the insole described in Figure 8-2. The plot shows that the right pressure insole at positions 7, 9 and the left pressure insole at position 7 contribute the most to the model's predictive capacity.

Table 8-1 shows results obtained using a window size of 25 and 100. With a window size of 25, more samples are extracted from each iteration, hence, increasing the likelihood of extracting noisy, uncorrelated samples.

Table 8-1: Evaluation of three machine learning models on all four test data samples. The fourth model is an ensemble of ExtraTreeRegressor, LightGBM, and CatBoost models. The Ensemble-based approach significantly outperforms other models in three of the four experiments.

Window size	**Model**	**Mean Absolute Error (lbs)**			
		Iteration 4	Iteration 5	Iteration 7	Iteration 8
25	**ExtraTreeRegressor**	1.081	6.254	2.776	2.473
	LightGBM	1.321	6.081	3.102	2.612
	CatBoost	1.000	6.269	2.753	2.218
	Ensembled	0.926	5.900	2.664	2.323
100					
	ExtraTreeRegressor	0.956	6.027	2.644	2.582
	LightGBM	1.014	6.176	2.803	2.541
	CatBoost	1.000	6.100	2.803	2.388
	Ensembled	**0.810**	**5.424**	**2.559**	2.420

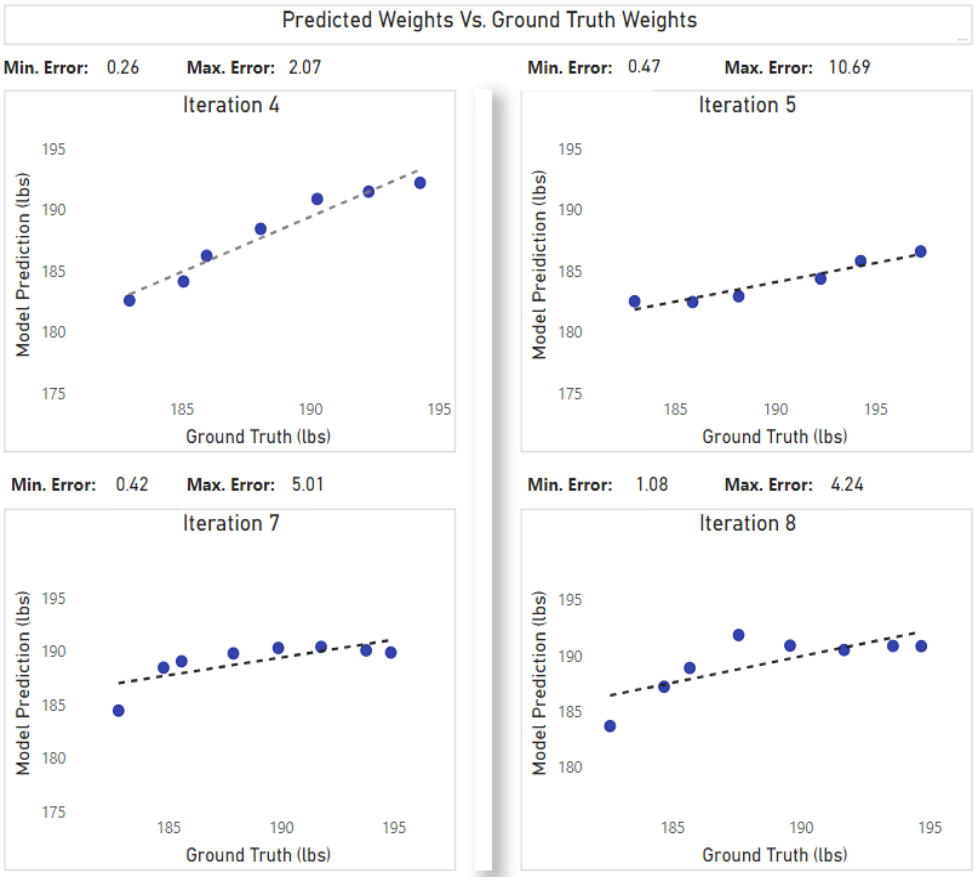


Figure 8-12: Scatter plots of the predicted weights vs. ground truth weights. The minimum and maximum difference between the estimated and groundtruth weight is included because the regression line is not always accurate, as depicted in the plot for Experiment 5.

Table 8-1 and Figure 8-12 presents the results for candidate 6. The best error rate is obtained using an ensembled regressor model (highlighted in red). This model is obtained by building a meta-model that generates the final prediction using the predictions of multiple base estimators [219]. In our case, these base estimators are ExtraTreeRegressor, Light Gradient Boosting Machine (LightGBM), and Categorical Boosting (CatBoost).

Table 8-2: Average Mean Absolute Error for data samples collected in the first and second experiment. The MAE score is lower for females because of the lower weight distribution while it is larger in males because of the wider weight distribution. Results are obtained using the Ensemble model.

Dataset	**Sex**	**Average MAE (lbs)**
Experiment 1	Female	2.202
	Male	3.681
Experiment 2	Female	2.140
	Male	3.447

Table 8-2 shows the mean absolute error for data samples collected in experiments 1 and 2. The MAE score in females is much lower than that of males. This makes sense since female participants' minimum and maximum weight is much lower than in males (see Figure 8-3).

8.5 Summary

This chapter introduces the first large-scale human body weight estimation dataset using smart shoe insole pressure sensors. The proposed dataset comprises 28 female and 34 male participants with corresponding groundtruth weights. An extensive exploratory data analysis is also introduced in this chapter to provide better understanding on the data retrieved and the potential of real-time body weight monitoring. The average error rate obtained using the proposed machine learning predictive modeling pipeline is ~1.4%, with a maximum deviation of ±3.6lbs.

Chapter 9. Concluding Remarks and Future Work

9.1 Conclusion

This dissertation focuses on identifying limitations of current Artificial Intelligence systems and developing novel solutions for addressing these issues, focusing on multimodal data enhancement and augmentation. To accomplish this, this dissertation investigated three major research topics:

(a) Human visual system-inspired multimodal enhancement algorithms utilized deep neural networks to dynamically adapt to several operating environments, coupled with reduced inference and retraining time. Also, these algorithms help overcome challenges faced with different distortions due to underwater, low-light, and hazy environments.

(b) Adaptive data augmentation methodologies to compensate for inadequate training data encountered in various tasks. This is achieved by developing novel pseudo-labeled data generation techniques for large unannotated datasets with discriminative features for augmenting the original datasets. These strategies provided additional data for training, validating, and testing our algorithms and served as a regularization strategy to reduce overfitting.

(c) An adaptive and automated approach that considers individual uniqueness in building AI systems. AI has a tendency for algorithmic bias, especially in the development of systems for tools intended to aid individuals hence the importance of minimizing variability in data captured and errors in predictive modeling for different individuals.

Figure 9-1 shows a detailed overview of the contributions of this dissertation.

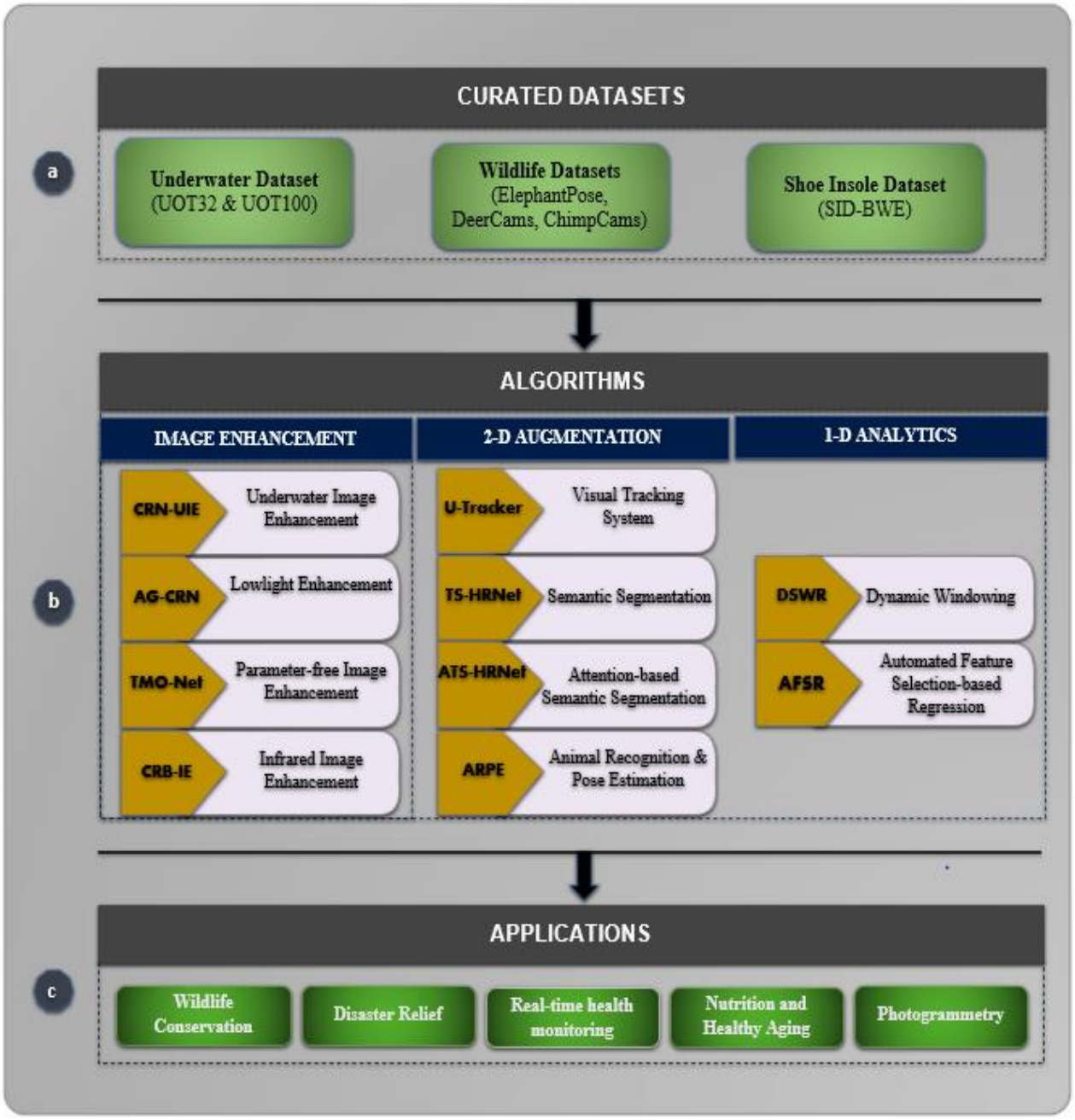


Figure 9-1: Overview of the dissertation contributions. (a) depicts the datasets created as part of this dissertation, (b) outlines the various algorithms and techniques implemented during the course of this dissertation, and (c) highlights the various areas of applications explored in this dissertation.

For multimodal data enhancement, this dissertation proposed:

- A novel attention-guided cascaded residual network (AG-CRN) for improving the visual quality of low-light images (Chapter 2). Quantitative and qualitative evaluation shows that AG-CRN outperforms existing low-light enhancement algorithms while avoiding artifacts and preserving the natural contents of the input image (2.3). We also showed that including AG-CRN as a preprocessing step for other computer vision tasks such as face detection and facial landmark localization significantly improves the performance of baseline models (2.3).
- Cascaded residual networks are also harnessed for underwater image enhancement (Chapter 3). CRN-UIE was proposed for translating distorted underwater images to open-air-like/enhanced images (3.4). The contents of the enhanced underwater image are more visible, making it better suited for exploration. The effectiveness of CRN-UIE is tested on a novel underwater single object tracking dataset (UOT100) (3.6). Results show a significant performance improvement of state-of-the-art trackers on the enhanced UOT100 dataset over the original un-enhanced data.

Pseudo-labeling as a form of data augmentation is explored for animal detection and pose estimation. This dissertation proposed:

- A feature-based sliding window approach for elephant detection (4.2). This explored conventional feature extraction techniques for binary classification and a sliding window approach for patch-wise classification (4.2.4). Non-maximum suppression is used to filter out repetitive bounding boxes while merging overlapping predictions. A shadow-based detection algorithm is also explored for detecting elephants from drone footage (4.3). To address the limitations of feature- and shadow-based elephant detection algorithms, deep

learning-based approaches for feature extraction and elephant detection is explored under diverse scenarios and environmental conditions to evaluate the robustness of each methodology (4.4).

- A unified approach for pose estimation in mammals (Chapter 5). A novel approach for self-paced training using pseudo-labels generated from many unannotated datasets is also proposed to ensure that these labels do not degrade the model's performance (5.3). Results shows a significant improvement in the predictive ability of the trained model on unseen animal species and categories (5.5) and outperform other state-of-the-art models by as high as 5% on some benchmark validation datasets.

For exploring the effectiveness of two-stream networks for multi-task learning and the importance of data augmentation and pseudo-labeling on AI systems, this dissertation proposed:

- A two-stream high-resolution network for joint building localization and damage classification (Chapter 6). This approach is different from other two-stage approaches where individual models are run sequentially i.e. the output of a building localization model serves as input to a building damage classification model. By training the proposed TS-HRNet to learn both tasks via weight sharing, each stream can learn the best parameters for each task and speed up inference time. A simple pseudo-label generation technique is also introduced to maximize the information that can be learned from a large number of unlabeled datasets. Results show that training the proposed TS-HRNet with groundtruth and pseudo-labeled data significantly outperforms the baseline model (6.3).
- Spatial-Aware CutMix (SA-CutMix) data augmentation strategy (7.2.1) is also introduced to address class imbalance in the training dataset. An attention module is also introduced in the TS-HRNet (ATS-HRNet) to harness long-range feature interactions in the training

process (7.3). Quantitative and qualitative results show a significant improvement over other state-of-the-art approaches (7.5).

Finally, an extensive exploratory analysis of streaming sensor data is carried out. This dissertation proposed:

- A personalized approach for extracting features obtained from wearable sensor data. This helps in significantly reducing bias in our predictive model.
- A machine learning framework for estimating human body weights from pressure and force data provided by shoe insoles.

9.2 Future Work

Although most of the algorithms proposed in this dissertation outperform the current state-of-the-art algorithms demonstrated by the quantitative and qualitative comparisons, certain limitations were observed, providing a space for improvement in subsequent research work for the future. The limitations identified and the suggestions for addressing each are:

- The enhancement algorithms proposed are trained on images only and do not account for temporal dependencies, hence making them unsuitable for real-time applications in videos. Hence, part of the future work related to this would involve incorporating temporal information in the model design and training process.
- The low-light and underwater enhancement algorithms proposed are currently used as a preprocessing step, increasing the latency at inference time since enhancement has to be done first before the next task (detection, tracking, etc.). It also creates two points of failure in which the effectiveness of a model depends on another model's performance. This design

is inefficient; hence, part of the future is exploring multi-task learning for combining related tasks into a single network, thus decreasing latency.

- Current open-air trackers can be improved to work on underwater scenarios without the need for enhancement. This can be done by reducing the discrepancy between the number of open-air and underwater datasets used to train the object tracker. The tracking performance would significantly increase by incorporating UOT100 with the existing datasets and training on both. Additionally, new model architectures can be developed to adapt to multimodal data using a two-stream approach proposed in this dissertation for satellite imagery.
- Similarly, proposed models for animal detection, recognition, pose estimation, and tracking are currently implemented separately. This also is inefficient as the tasks can be combined into a single network and framework designed as a multi-task learning problem since detection, recognition, pose estimation, and tracking can benefit from weight sharing to improve its performance.
- Animal pose estimation is still a difficult problem becomes of the variety of poses an animal can take as well as joint occlusion. Novel model architectures that account for 'context' in the localization of each keypoint need to be developed to help improve the limitations of existing systems.
- The model's performance still needs to be improved for satellite imagery because of the issues observed when localizing and classifying damages to small buildings. More datasets also need to be curated to improve the robustness of current systems to diversity in building design patterns that vary across ethnic groups, regions, climates, and continents.

- From the experiments carried out as part of this dissertation, certain limitations were observed in the current shoe insoles used for data collection. The sensitivity of the sensors isn’t suitable for detecting very low changes in body weight up to 100g. A new shoe insole with customized pressure sensors needs to be designed to address this challenge. This would also help improve the current models to adapt better to the calibrations of the new system.
- The algorithms developed for multimedia data enhancement can be integrated to the data augmentation methodologies developed for animal pose estimation and satellite image processing and analysis to improve the input images' visual quality (as a preprocessing step) for obtaining the best results under non-ideal conditions. Part of the future work is to integrate these tasks as a single-stage process and use a single network to improve efficiency for real-time applications.
- Finally, a software solution for wildlife conservation experts is to be developed to integrate the separate components proposed in this dissertation into a simple, easy-to-use, and efficient tool for facilitating wildlife research.

References

[1] K. P. Acharya, P. K. Paudel, P. R. Neupane, and M. Köhl, "Human-wildlife conflicts in Nepal: Patterns of human fatalities and injuries caused by large mammals," *PLoS One*, vol. 11, no. 9, Sep. 2016.

[2] L. J. Shaffer, K. K. Khadka, J. Van Den Hoek, and K. J. Naithani, "Human-elephant conflict: A review of current management strategies and future directions," *Front. Ecol. Evol.*, vol. 6, no. JAN, p. 235, Jan. 2019.

[3] "Environmental health in emergencies." [Online]. Available: https://www.who.int/teams/environment-climate-change-and-health/emergencies. [Accessed: 08-Nov-2021].

[4] L. Kompaniyets *et al.*, "Body Mass Index and Risk for COVID-19–Related Hospitalization, Intensive Care Unit Admission, Invasive Mechanical Ventilation, and Death — United States, March–December 2020," *MMWR. Morb. Mortal. Wkly. Rep.*, vol. 70, no. 10, pp. 355–361, 2021.

[5] U. Epa and C. Change Division, "Climate Change and the Health of Older Adults."

[6] M. Wadman, "Why COVID-19 is more deadly in people with obesity—even if they're young," *Science (80-.).*, Sep. 2020.

[7] D. J. Granrath, "The Role of Human Visual Models in Image Processing," *Proc. IEEE*, vol. 69, no. 5, pp. 552–561, 1981.

[8] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[9] V. Oludare, L. Kezebou, K. Panetta, and S. S. Agaian, "Attention-guided cascaded networks for improved face detection and landmark localization under low-light conditions," *https://doi.org/10.1117/12.2558397*, vol. 11399, pp. 108–118, Apr. 2020.

[10] S. Ai and J. Kwon, "Extreme Low-Light Image Enhancement for Surveillance Cameras Using Attention U-Net," *Sensors (Basel).*, vol. 20, no. 2, Jan. 2020.

[11] K. A. Panetta, E. J. Wharton, and S. S. Agaian, "Human visual system-based image enhancement and logarithmic contrast measure," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 38, no. 1, pp. 174–188, Feb. 2008.

[12] O. Patel, Y. P. S. Maravi, and S. Sharma, "Signal & Image Processing," *An Int. J.*, vol. 4, no. 5, 2013.

[13] W. Zhihong and X. Xiaohong, "Study on histogram equalization," *Proc. - 2011 Int. Symp. Intell. Inf. Process. Trust. Comput. IPTC 2011*, pp. 177–179, 2011.

[14] P. P. Banik, R. Saha, and K. D. Kim, "Contrast enhancement of low-light image using histogram equalization and illumination adjustment," *Int. Conf. Electron. Inf. Commun. ICEIC 2018*, vol. 2018-January, pp. 1–4, Apr. 2018.

[15] A. M. Reza, “Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement,” *J. VLSI signal Process. Syst. signal, image video Technol. 2004 381*, vol. 38, no. 1, pp. 35–44, Nov. 2004.

[16] H. Ibrahim and N. S. P. Kong, “Brightness preserving dynamic histogram equalization for image contrast enhancement,” *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1752–1758, Nov. 2007.

[17] D. J. Jobson, Z. U. Rahman, and G. A. Woodell, “Properties and performance of a center/surround retinex,” *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 451–462, 1997.

[18] D. J. Jobson, Z. U. Rahman, and G. A. Woodell, “A multiscale retinex for bridging the gap between color images and the human observation of scenes,” *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, 1997.

[19] B. Jiang, G. A. Woodell, and D. J. Jobson, “Novel multi-scale retinex with color restoration on graphics processing unit,” *J. Real-Time Image Process.*, vol. 10, no. 2, pp. 239–253, Jun. 2015.

[20] X. Fu, D. Zeng, Y. Huang, X. P. Zhang, and X. Ding, “A Weighted Variational Model for Simultaneous Reflectance and Illumination Estimation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2782–2790, Dec. 2016.

[21] X. Guo, “LIME: A Method for Low-light IMage Enhancement,” *MM 2016 - Proc. 2016 ACM Multimed. Conf.*, pp. 87–91, May 2016.

[22] C. Wei, W. Wang, W. Yang, and J. Liu, “Deep Retinex Decomposition for Low-Light Enhancement,” *Br. Mach. Vis. Conf. 2018, BMVC 2018*, Aug. 2018.

[23] L. Shen, Z. Yue, F. Feng, Q. Chen, S. Liu, and J. Ma, “MSR-net:Low-light Image Enhancement Using Deep Convolutional Network,” Nov. 2017.

[24] C. Li, J. Guo, F. Porikli, and Y. Pang, “LightenNet: A Convolutional Neural Network for weakly illuminated image enhancement,” *Pattern Recognit. Lett.*, vol. 104, pp. 15–22, 2018.

[25] K. G. Lore, A. Akintayo, and S. Sarkar, “LLNet: A Deep Autoencoder Approach to Natural Low-light Image Enhancement,” *Pattern Recognit.*, vol. 61, pp. 650–662, Nov. 2015.

[26] J. Cai, S. Gu, and L. Zhang, “Learning a deep single image contrast enhancer from multi-exposure images,” *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2049–2062, Apr. 2018.

[27] F. Lv, F. Lu, J. Wu, and C. Lim, “MBLLEN: Low-light Image/Video Enhancement Using CNNs.”

[28] C. Chen, Q. Chen, J. Xu, and V. Koltun, “Learning to See in the Dark,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3291–3300, May 2018.

[29] C. Guo *et al.*, “Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1777–1786, Jan. 2020.

[30] Y. Jiang *et al.*, "EnlightenGAN: Deep Light Enhancement without Paired Supervision," *IEEE Trans. Image Process.*, vol. 30, pp. 2340–2349, Jun. 2019.

[31] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 5967–5976, Nov. 2016.

[32] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9906 LNCS, pp. 694–711, Mar. 2016.

[33] M. Saquib Sarfraz, C. Seibold, H. Khalid, and R. Stiefelhagen, "Content and Colour Distillation for Learning Image Translations with the Spatial Profile Loss," *30th Br. Mach. Vis. Conf. 2019, BMVC 2019*, Aug. 2019.

[34] W. Yang *et al.*, "UG^{2+} Track 2: A Collective Benchmark Effort for Evaluating and Advancing Image Understanding in Poor Visibility Environments," Apr. 2019.

[35] K. Ma, K. Zeng, and Z. Wang, "Perceptual quality assessment for multi-exposure image fusion," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3345–3356, Nov. 2015.

[36] X. Guo, Y. Li, and H. Ling, "LIME: Low-light image enhancement via illumination map estimation," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 982–993, Feb. 2017.

[37] S. Wang, J. Zheng, H. M. Hu, and B. Li, "Naturalness preserved enhancement algorithm for non-uniform illumination images," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3538–3548, Sep. 2013.

[38] "Vasileios Vonikakis - Datasets." [Online]. Available: https://sites.google.com/site/vonikakis/datasets. [Accessed: 02-Jan-2022].

[39] C. Lee, C. S. Kim, and C. Lee, "Contrast enhancement based on layered difference representation of 2D histograms," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5372–5384, 2013.

[40] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, 2013.

[41] Y. Zhenqiang, G. Li, and W. Gao, "A Bio-Inspired Multi-Exposure Fusion Framework for Low-light Image Enhancement."

[42] Z. Ying, G. Li, Y. Ren, R. Wang, and W. Wang, "A New Low-Light Image Enhancement Algorithm Using Camera Response Model," *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017*, vol. 2018-January, pp. 3015–3022, Jul. 2017.

[43] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage Dense Face Localisation in the Wild," May 2019.

[44] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A Face Detection Benchmark," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 5525–5533, Nov. 2015.

[45] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile

Vision Applications," Apr. 2017.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2015.

[47] J. R. Ahamed, J. Raihan, P. E. Abas, and L. C. De Silva, "PCF design and simulation View project Maritime Container terminals View project Review of underwater image restoration algorithms," 2019.

[48] L. Kezebou, V. Oludare, K. Panetta, and S. S. Agaian, "Underwater Object Tracking Benchmark and Dataset," *2019 IEEE Int. Symp. Technol. Homel. Secur. HST 2019*, Nov. 2019.

[49] K. Panetta, L. Kezebou, V. Oludare, and S. Agaian, "Comprehensive Underwater Object Tracking Benchmark Dataset and Underwater Image Enhancement With GAN," *IEEE J. Ocean. Eng.*, 2021.

[50] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[51] "OTB Toolkit | otb-toolkit." .

[52] "VOT Challenge| Challenges." .

[53] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking," Apr. 2015.

[54] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," Mar. 2016.

[55] "MOT Challenge - Results." .

[56] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[57] M. Danelljan, G. Bhat, S. Khan, and M. Felsberg, "ECO: Efficient Convolution Operators for Tracking."

[58] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9914 LNCS, pp. 850–865.

[59] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 8926, pp. 254–265.

[60] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning Background-Aware Correlation Filters for Visual Tracking."

[61] M. Danelljan, G. Häger, S. Khan, and M. Felsberg, "Learning Spatially Regularized

Correlation Filters for Visual Tracking."

[62] H. Nam and B. Han, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking."

[63] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking."

[64] L. Kezebou, V. Oludare, K. Panetta, and S. S. Agaian, "Underwater Object Tracking Benchmark and Dataset," in *2019 IEEE International Symposium on Technologies for Homeland Security, HST 2019*, 2019.

[65] S. Anwar and C. Li, "Diving deeper into underwater image enhancement: A survey," *Signal Process. Image Commun.*, vol. 89, p. 115978, Nov. 2020.

[66] M. Yang, J. Hu, C. Li, G. Rohde, Y. Du, and K. Hu, "An in-depth survey of underwater image enhancement and restoration," *IEEE Access*, vol. 7, pp. 123638–123657, 2019.

[67] E. J. McCartney and F. F. H. Jr, "Optics of the Atmosphere: Scattering by Molecules and Particles," *Phys. Today*, vol. 30, no. 5, p. 76, Dec. 2008.

[68] J. Y. Chiang and Y. C. Chen, "Underwater image enhancement by wavelength compensation and dehazing," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1756–1769, Apr. 2012.

[69] D. X. Jia and Y. R. Ge, "Underwater image de-noising algorithm based on Nonsubsampled Contourlet Transform and Total Variation," *undefined*, pp. 76–80, 2012.

[70] S. Bazeille, I. Quidu, L. Jaulin, J.-P. Malkasse, and J.-P. Malkasse, "Automatic underwater image pre-processing," p. xx, 2006.

[71] A. Arnold-Bos, J. Malkasse, and G. Kervern, "A preprocessing framework for automatic underwater images denoising," *undefined*, 2005.

[72] L. A. Torres-Méndez and G. Dudek, "Color Correction of Underwater Images for Aquatic Robot Inspection."

[73] A. Majed Chambah, D. Semani, A. Renouf, P. Courtellemont, and A. Rizzi, "Underwater color constancy: enhancement of automatic live fish recognition," *https://doi.org/10.1117/12.524540*, vol. 5293, no. 18, pp. 157–168, Dec. 2003.

[74] A. S. Abdul Ghani and N. A. Mat Isa, "Underwater image quality enhancement through composition of dual-intensity images and Rayleigh-stretching," *Springerplus*, vol. 3, no. 1, pp. 1–14, Dec. 2014.

[75] A. S. Abdul Ghani and N. A. Mat Isa, "Underwater image quality enhancement through composition of dual-intensity images and Rayleigh-stretching," *Springerplus*, vol. 3, no. 1, pp. 1–14, Dec. 2014.

[76] S. Anwar, C. Li, and F. Porikli, "Deep Underwater Image Enhancement," Jul. 2018.

[77] C. Li *et al.*, "An Underwater Image Enhancement Benchmark Dataset and beyond," *IEEE Trans. Image Process.*, vol. 29, pp. 4376–4389, 2020.

[78] M. Hou, R. Liu, X. Fan, and Z. Luo, “Joint Residual Learning for Underwater Image Enhancement,” *Proc. - Int. Conf. Image Process. ICIP*, pp. 4043–4047, Aug. 2018.

[79] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8798–8807, Nov. 2017.

[80] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, May 2015.

[81] I. Goodfellow *et al.*, “Generative Adversarial Networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Jun. 2014.

[82] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Dec. 2014.

[83] M. J. Islam, Y. Xia, and J. Sattar, “Fast Underwater Image Enhancement for Improved Visual Perception,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3227–3234, Mar. 2019.

[84] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 2242–2251, Mar. 2017.

[85] C. Fabbri, M. J. Islam, and J. Sattar, “Enhancing Underwater Imagery using Generative Adversarial Networks,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 7159–7165, Jan. 2018.

[86] K. Panetta, C. Gao, and S. Agaian, “Human-visual-system-inspired underwater image quality measures,” *IEEE J. Ocean. Eng.*, vol. 41, no. 3, pp. 541–551, Jul. 2016.

[87] Y. Wang *et al.*, “An imaging-inspired no-reference underwater color image quality assessment metric,” *Comput. Electr. Eng.*, vol. 70, pp. 904–913, Aug. 2018.

[88] S. Bei, Z. Zhen, L. Wusheng, D. Liebo, and L. Qin, “Visual object tracking challenges revisited: VOT vs. OTB,” *PLoS One*, vol. 13, no. 9, Sep. 2018.

[89] M. Kristan *et al.*, “A Novel Performance Evaluation Methodology for Single-Target Trackers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

[90] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, “Staple: Complementary Learners for Real-Time Tracking.”

[91] J. Xiao, R. Stolkin, and A. Leonardis, “Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models.”

[92] X. Jia, H. Lu, and M. H. Yang, “Visual tracking via adaptive structural local sparse appearance model,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1822–1829.

[93] V. Ramalakshmi and M. G. Alex, “Visual object tracking using discriminative correlation filter,” in *Proceedings of the International Conference on Communication and Electronics Systems, ICCES 2016*, 2016.

[94] Y. Song *et al.*, “VITAL: VIsual Tracking via Adversarial Learning,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8990–8999, Apr. 2018.

[95] C. Ma SJTU, X. Yang SJTU, and M.-H. Yang, “Hierarchical Convolutional Features for Visual Tracking.”

[96] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9909 LNCS, pp. 472–488.

[97] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, “End-to-end representation learning for Correlation Filter based tracking.”

[98] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, “Discriminative Scale Space Tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[99] S. Hare *et al.*, “Struck: Structured Output Tracking with Kernels,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[100] “Homepage - Subnautica.” [Online]. Available: https://unknownworlds.com/subnautica/. [Accessed: 12-Jan-2022].

[101] H. Grabner, M. Grabner, and H. Bischof, “Real-Time Tracking via On-line Boosting.”

[102] E. Richardson *et al.*, “Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation.” pp. 2287–2296, 2021.

[103] A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, and M. Kristan, “Discriminative Correlation Filter with Channel and Spatial Reliability,” *Int. J. Comput. Vis.*, vol. 126, no. 7, pp. 671–688, Nov. 2016.

[104] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-backward error: Automatic detection of tracking failures,” in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 2756–2759.

[105] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast Online Object Tracking and Segmentation: A Unifying Approach,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 1328–1338, Dec. 2018.

[106] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks.” pp. 4282–4291, 2019.

[107] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.

[108] M. J. Chase *et al.*, “Continent-wide survey reveals massive decline in African savannah elephants,” *PeerJ*, vol. 2016, no. 8, p. e2354, Aug. 2016.

[109] P. J. Clemins and M. T. Johnson, “Application of speech recognition to African elephant (Loxodonta Africana) vocalizations,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 1, pp. 484–487, 2003.

[110] K. B. Payne, M. Thompson, and L. Kramer, “Elephant calling patterns as indicators of group size and composition: The basis for an acoustic monitoring system,” *Afr. J. Ecol.*, vol. 41, no. 1, pp. 99–107, Mar. 2003.

[111] J. D. Wood, C. E. O’Connell-Rodwell, and S. L. Klemperer, “METHODOLOGICAL INSIGHTS: Using seismic sensors to detect elephants and other large mammals: a potential census technique,” *J. Appl. Ecol.*, vol. 42, no. 3, pp. 587–594, Jun. 2005.

[112] “Feature Detectors - Sobel Edge Detector.” [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm?iframe=true&width=80%25&height=80%25. [Accessed: 01-Dec-2021].

[113] B. Schiele and J. L. Crowley, “Object recognition using multidimensional receptive field histograms,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1064, pp. 610–619, 1996.

[114] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[115] S. Brahnam, L. Jain, L. Nanni, and A. Lumini, “Local Binary Patterns: New Variants and Applications,” *undefined*, vol. 506, 2014.

[116] C. H. Chen, L. F. Pau, P. S. P Wang, M. Tuceryan, and A. K. Jain, “TEXTURE ANALYSIS,” *Handb. Pattern Recognit. Comput. Vis.*, pp. 235–276, Aug. 1993.

[117] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[118] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, “Computer Vision Using Local Binary Patterns,” *Comput. Imaging Vis.*, vol. 40, pp. E1–E2, 2011.

[119] D. G. Lowe, “Accepted for publication in the,” *Int. J. Comput. Vis.*, 2004.

[120] H. R. Kher and V. K. Thakar, “Scale invariant feature transform based image matching and registration,” *Proc. - 2014 5th Int. Conf. Signal Image Process. ICSIP 2014*, pp. 50–55, 2014.

[121] T. Lindeberg, “Scale Invariant Feature Transform,” *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.

[122] B. Schölkopf, “SVMs - A practical consequence of learning theory,” *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–21, Jul. 1998.

[123] J. Wang and Y. Yagi, “Shadow extraction and application in pedestrian detection,” *Eurasip J. Image Video Process.*, vol. 2014, no. 1, pp. 1–10, Feb. 2014.

[124] M. Russell, J. J. Zou, and G. Fang, “Human shadow detection for real-time applications,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 1610–1613, Feb. 2017.

[125] J. Xiang, H. Fan, H. Liao, J. Xu, W. Sun, and S. Yu, “Moving object detection and shadow removing under changing illumination condition,” *Math. Probl. Eng.*, vol. 2014,

2014.

[126] “Chromaticity - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Chromaticity. [Accessed: 06-Dec-2021].

[127] A. Sanin, C. Sanderson, and B. C. Lovell, “Shadow Detection: A Survey and Comparative Evaluation of Recent Methods,” *Pattern Recognit.*, vol. 45, no. 4, pp. 1684–1695, Apr. 2013.

[128] C. T. Chen, C. Y. Su, and W. C. Kao, “An enhanced segmentation on vision-based shadow removal for vehicle detection,” *1st Int. Conf. Green Circuits Syst. ICGCS 2010*, pp. 679–682, 2010.

[129] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2015.

[130] S. Sahil *et al.*, “A Survey of Modern Deep Learning based Object Detection Models.”

[131] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020.

[132] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.”

[133] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression,” *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 12993–13000, Nov. 2019.

[134] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, Aug. 2016.

[135] D. Misra, “Mish: A Self Regularized Non-Monotonic Activation Function,” Aug. 2019.

[136] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 1571–1580, Nov. 2019.

[137] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8691 LNCS, no. PART 3, pp. 346–361, Jun. 2014.

[138] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8759–8768, Mar. 2018.

[139] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[140] G. Welch and G. Bishop, “An Introduction to the Kalman Filter.”

[141] X. Li, K. Wang, W. Wang, and Y. Li, “A multiple object tracking method using Kalman filter,” *2010 IEEE Int. Conf. Inf. Autom. ICIA 2010*, pp. 1862–1866, 2010.

[142] “Kalman filter - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter. [Accessed: 07-Dec-2021].

[143] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Dec. 2018.

[144] T. Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, May 2014.

[145] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2D human pose estimation: New benchmark and state of the art analysis,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3686–3693, Sep. 2014.

[146] S. Li, J. Li, H. Tang, R. Qian, and W. Lin, “ATRW: A Benchmark for Amur Tiger Re-identification in the Wild,” *MM 2020 - Proc. 28th ACM Int. Conf. Multimed.*, vol. 20, pp. 2590–2598, Jun. 2019.

[147] A. Mathis *et al.*, “Pretraining boosts out-of-domain robustness for pose estimation,” *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, pp. 1858–1867, Sep. 2019.

[148] R. Labuguen *et al.*, “MacaquePose: A Novel ‘In the Wild’ Macaque Monkey Pose Dataset for Markerless Motion Capture,” *Front. Behav. Neurosci.*, vol. 14, p. 268, Jan. 2021.

[149] J. M. Graving *et al.*, “Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning,” *Elife*, vol. 8, Oct. 2019.

[150] J. Cao, H. Tang, H. S. Fang, X. Shen, C. Lu, and Y. W. Tai, “Cross-Domain Adaptation for Animal Pose Estimation,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 9497–9506, Aug. 2019.

[151] H. Yu, Y. Xu, J. Zhang, W. Zhao, Z. Guan, and D. Tao, “AP-10K: A Benchmark for Animal Pose Estimation in the Wild,” Aug. 2021.

[152] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, “Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7577 LNCS, no. PART 6, pp. 852–863, 2012.

[153] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, “Vision-based hand pose estimation: A review,” *Comput. Vis. Image Underst.*, vol. 108, no. 1–2, pp. 52–73, Oct. 2007.

[154] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation in computer vision: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, 2009.

[155] G. Fanelli, J. Gall, and L. Van Gool, “Real time head pose estimation with random regression forests,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp.

617–624, 2011.

[156] S. H. Zhang *et al.*, “Pose2Seg: Detection Free Human Instance Segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 889–898, Mar. 2018.

[157] G. Ning, P. Liu, X. Fan, and C. Zhang, “A Top-down Approach to Articulated Human Pose Estimation and Tracking,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11130 LNCS, pp. 227–234, Jan. 2019.

[158] Z. Geng, K. Sun, Z. Zhang, and J. Wang, “Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression.”

[159] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1302–1310, Nov. 2016.

[160] M. Li, Z. Zhou, J. Li, and X. Liu, “Bottom-up Pose Estimation of Multiple Person with Bounding Box Constraint.”

[161] H.-S. Fang, S. Xie, Y.-W. Tai, C. Lu, S. Jiao Tong University, and T. YouTu, “RMPE: Regional Multi-Person Pose Estimation,” 2018.

[162] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep High-Resolution Representation Learning for Human Pose Estimation.”

[163] L. Pishchulin *et al.*, “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 4929–4937, Nov. 2015.

[164] A. Newell, Z. Huang, and J. Deng, “Associative Embedding: End-to-End Learning for Joint Detection and Grouping,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 2278–2288, Nov. 2016.

[165] S. Yang, Z. Quan, M. Nie, and W. Yang, “TransPose: Keypoint Localization via Transformer,” Dec. 2020.

[166] K. Sun *et al.*, “High-Resolution Representations for Labeling Pixels and Regions,” Apr. 2019.

[167] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” *ACM Int. Conf. Proceeding Ser.*, vol. 382, 2009.

[168] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, “Distribution-Aware Coordinate Representation for Human Pose Estimation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7091–7100, Oct. 2019.

[169] J. Wang *et al.*, “Deep High-Resolution Representation Learning for Visual Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Aug. 2019.

[170] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” 2019.

[171] “A Comprehensive Guide to Human Pose Estimation.” [Online]. Available: https://www.v7labs.com/blog/human-pose-estimation-guide#pose-estimation-metrics. [Accessed: 20-Dec-2021].

[172] “COCO - Common Objects in Context.” [Online]. Available: https://cocodataset.org/#keypoints-eval. [Accessed: 20-Dec-2021].

[173] H. Liu, F. Liu, X. Fan, and D. Huang, “Polarized Self-Attention: Towards High-quality Pixel-wise Regression,” Jul. 2021.

[174] V. Oludare, L. Kezebou, K. Panetta, and S. S. Agaian, “Semi-supervised learning for improved post-disaster damage assessment from satellite imagery,” *https://doi.org/10.1117/12.2586232*, vol. 11734, pp. 172–182, Apr. 2021.

[175] J. Sublime and E. Kalinicheva, “Automatic Post-Disaster Damage Mapping Using Deep-Learning Techniques for Change Detection: Case Study of the Tohoku Tsunami,” *Remote Sens. 2019, Vol. 11, Page 1123*, vol. 11, no. 9, p. 1123, May 2019.

[176] R. Gupta *et al.*, “xBD: A Dataset for Assessing Building Damage from Satellite Imagery,” Nov. 2019.

[177] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark,” *Int. Geosci. Remote Sens. Symp.*, vol. 2017-July, pp. 3226–3229, Dec. 2017.

[178] A. J. Cooner, Y. Shao, J. B. Campbell, R. Tomas, Z. Li, and P. S. Thenkabail, “Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms: Revisiting the 2010 Haiti Earthquake,” *Remote Sens. 2016, Vol. 8, Page 868*, vol. 8, no. 10, p. 868, Oct. 2016.

[179] M. Ji, L. Liu, and M. Buchroithner, “Identifying Collapsed Buildings Using Post-Earthquake Satellite Imagery and Convolutional Neural Networks: A Case Study of the 2010 Haiti Earthquake,” *Remote Sens. 2018, Vol. 10, Page 1689*, vol. 10, no. 11, p. 1689, Oct. 2018.

[180] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, “Satellite Image Classification of Building Damages Using Airborne and Satellite Image Samples in a Deep Learning Approach,” *ISPAn*, vol. IV2, no. 2, pp. 89–96, May 2018.

[181] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, pp. 234–241, May 2015.

[182] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11045 LNCS, pp. 3–11, Jul. 2018.

[183] E. Weber and H. Kané, “Building Disaster Damage Assessment in Satellite Imagery with Multi-Temporal Fusion,” Apr. 2020.

[184] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object

Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Aug. 2017.

[185] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10553 LNCS, pp. 240–248, Jul. 2017.

[186] S. Jadon, "A survey of loss functions for semantic segmentation," *2020 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol. CIBCB 2020*, Jun. 2020.

[187] "DIU's xVIEW2 - Assessing Building Damage | Challenge.gov." [Online]. Available: https://www.challenge.gov/challenge/diu-xview2-assessing-building-damage/. [Accessed: 12-Nov-2021].

[188] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning."

[189] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[190] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient Object Localization Using Convolutional Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 648–656, Nov. 2014.

[191] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, Feb. 2015.

[192] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[193] "ImageNet." [Online]. Available: https://image-net.org/. [Accessed: 02-Dec-2021].

[194] D. Erhan, Y. Bengio, A. Courville, P.-A. M. Ca, P. V. Ca, and B. Com, "Why Does Unsupervised Pre-training Help Deep Learning? Pierre-Antoine Manzagol Pascal Vincent Samy Bengio," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010.

[195] S. Yun, D. Han, S. Chun, S. J. Oh, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 6022–6031, May 2019.

[196] Y. Shen *et al.*, "BDANet: Multiscale Convolutional Neural Network with Cross-directional Attention for Building Damage Assessment from Satellite Images," *IEEE Trans. Geosci. Remote Sens.*, May 2021.

[197] L. Chen *et al.*, "SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning."

[198] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module."

[199] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-Cross Attention for Semantic Segmentation," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 603–612, Nov. 2018.

[200] "DIUx-xView/xView2_fifth_place: Dual-HRNet." [Online]. Available: https://github.com/DIUx-xView/xView2_fifth_place. [Accessed: 06-Jan-2022].

[201] "Understanding Dice Loss for Crisp Boundary Detection | by Shuchen Du | AI Salon | Medium." [Online]. Available: https://medium.com/ai-salon/understanding-dice-loss-for-crisp-boundary-detection-bb30c2e5f62b. [Accessed: 06-Jan-2022].

[202] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving Object-Centric Image Segmentation Evaluation," Mar. 2021.

[203] W. Sirko *et al.*, "CONTINENTAL-SCALE BUILDING DETECTION FROM HIGH RESOLUTION SATELLITE IMAGERY," 2021.

[204] A. Ming Tan, Y. Weizman, and F. Konstantin Fuss, "Measurement Accuracy of the Body Weight with Smart Insoles †," *Proceedings*, vol. 2, p. 274, 2018.

[205] S. Muzaffar and I. M. Elfadel, "Shoe-Integrated, Force Sensor Design for Continuous Body Weight Monitoring," *Sensors (Basel).*, vol. 20, no. 12, pp. 1–22, Jun. 2020.

[206] M. D. Jensen *et al.*, "2013 AHA/ACC/TOS guideline for the management of overweight and obesity in adults: A report of the American College of cardiology/American Heart Association task force on practice guidelines and the obesity society," *Circulation*, vol. 129, no. 25 SUPPL. 1, Jun. 2014.

[207] J. Moon *et al.*, "Shoes with active insoles mitigate declines in balance after fatigue," *Sci. Reports 2020 101*, vol. 10, no. 1, pp. 1–11, Feb. 2020.

[208] J. Hageman, "Body Weight Prediction of Turkeys-From Walk to Mass," 2019.

[209] C. Velardo and J. L. Dugelay, "Weight estimation from visual body appearance," *IEEE 4th Int. Conf. Biometrics Theory, Appl. Syst. BTAS 2010*, 2010.

[210] C. Pfitzner, S. May, and A. Nüchter, "Evaluation of Features from RGB-D Data for Human Body Weight Estimation," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10148–10153, Jul. 2017.

[211] "Sensor insoles for clinical grade mobile gait & motion analysis." [Online]. Available: https://moticon.com/. [Accessed: 08-Dec-2021].

[212] T. Stöggl and A. Martiner, "Validation of Moticon's OpenGo sensor insoles during gait, jumps, balance and cross-country skiing specific imitation movements," *J. Sports Sci.*, vol. 35, no. 2, pp. 196–206, Jan. 2017.

[213] Y. Benyahmed, A. Abu Bakar, A. Razakhamdan, A. Ahmed, S. Mastura, and S. Abdullah, "ADAPTIVE SLIDING WINDOW ALGORITHM FOR WEATHER DATA SEGMENTATION," *J. Theor. Appl. Inf. Technol.*, vol. 80, no. 2, 2015.

[214] H. S. Hota, R. Handa, and A. K. Shrivas, "Time Series Data Prediction Using Sliding Window Based RBF Neural Network," *Int. J. Comput. Intell. Res.*, vol. 13, no. 5, pp.

1145–1156, 2017.

[215] Y. Yu, Y. Zhu, S. Li, and D. Wan, “Time series outlier detection based on sliding window prediction,” *Math. Probl. Eng.*, vol. 2014, 2014.

[216] M. Vafaeipour, O. Rahbari, M. A. Rosen, F. Fazelpour, and P. Ansarirad, “Application of sliding window technique for prediction of wind velocity time series,” *Int. J. Energy Environ. Eng.*, vol. 5, no. 2–3, pp. 1–7, Jul. 2014.

[217] L. Mozaffari, A. Mozaffari, and N. L. Azad, “Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads,” *Eng. Sci. Technol. an Int. J.*, vol. 18, no. 2, pp. 150–162, Jun. 2015.

[218] “Home - PyCaret.” [Online]. Available: https://pycaret.org/. [Accessed: 10-Jan-2022].

[219] “Stack Models - PyCaret.” [Online]. Available: https://pycaret.org/stack-models/. [Accessed: 11-Jan-2022].