

Automated Plume Rise Measurement based on Deep Neural Networks using Video Images

MOHAMMAD KOUSHAFAR

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Doctor of Philosophy (PhD)

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

December 2023

© 2023 Mohammad Koushafar

Abstract

MOHAMMAD KOUSHAFAR

When a smokestack releases pollutants, the resulting plume cloud dissipates gradually and mixes with the surrounding air; it becomes neutrally buoyant and loses its vertical momentum. It is then carried downwind at a constant height called the Plume Rise or plume rise height. Plume rise affects how far pollutants are carried downwind, their deposition to the environment, and the amount of greenhouse gases mixed into the upper troposphere. Therefore, correctly calculating plume rise for the modelled dispersion of pollutants is of concern in air-quality transport models and local environment assessment cases. Recent studies have shown that the Briggs equations, which are a popular form of parameterization in models, significantly underpredict the plume rise. Modern computer vision methods allow the possibility of measuring plume rise under varied atmospheric conditions using video images. Most existing computer vision methods detect smoke clouds using an estimated bounding box without performing a segmentation down to the pixel level. Our proposed method can accurately detect and segment a plume cloud exiting from a smokestack and consider a hypothetical plume centerline based on an improved Deep Convolutional Neural Network (DCNN). We propose a Mask R-CNN model that can be applied for extracting the region of the plume cloud area of interest. Then, the proposed network is modified with our training dataset and used for detecting the hypothetical centerlines of the plume cloud. Finally, a comparative analysis was performed using meteorological data and smokestack measurements. This analysis involved comparing the plume rise and plume rise distance values estimated by the proposed framework with the obtained values from the Briggs parameterization equations. The outcomes confirmed a considerable underestimation of plume rise by the Briggs plume rise parameterization in the region. Using our model, the conventional plume rise model validation methods can be developed, and scientific grounds are provided for developing new physical models.

Keywords: plume rise, deep learning, plume cloud recognition

Dedicated to my beloved parents,
Rogayeh and **Hashem**,
to whom I owe everything,
and my lovely wife, **Vida**,
without whom
I could never have done this.

Acknowledgements

I would like to express my heartfelt gratitude to all those who have supported and contributed to completing my Ph.D. thesis. Their endless support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, I would like to extend my most profound appreciation to my supervisors, **Professor Mark Gordon** and **Professor Gunho Sohn** of the Earth & Space Science Department at York University, for their guidance, mentorship, and expertise. Their dedication, patience, and continuous support have been instrumental in shaping the direction and quality of my research. I am grateful for their encouragement, feedback, and valuable insights that have enriched my work.

I am also honoured to thank my committee members, **Professor Costas Armenakis** and **Professor Mojgan Jadidi** of the Earth & Space Science Department at York University, for their dedicated time and effort in examining my thesis and providing insightful comments.

My deepest gratitude goes to my first supervisor, **Professor George Vukovich** of the Earth & Space Science Department at York University, who tragically passed away at the beginning of this way. He was a kind-hearted person and an exceptional mentor, and he will always be remembered for his passion for research and dedication to advancing knowledge. Peace Be Upon Him.

I am also indebted to my friend, colleague and teacher, **Dr. Amin Alizadeh Naeini** from the AUSM laboratory at York University, who has provided me with valuable feedback, constructive criticism, and stimulating discussions. His diverse perspectives, expertise, and camaraderie have contributed immensely to my academic growth and development.

I want to thank **Dr. Sepehr Fathi**, **Dr. Stefan Miller**, **Xuanyi Zhang**, and **Rhythm Reet** from the Air Pollution Lab at York University, for their invaluable support throughout my research journey and **Mohammad Moein Sheikholeslami**, **Mohammadjavad Ghorbanalivakili**, **Mostafa Ahmadi**, and **Muhammad Kamran**, from the AUSM laboratory at York University, who willingly contributed their precious time, expertise, and insights.

Lastly, I am grateful to my family for their unwavering love, support, and encouragement throughout my Ph.D. journey. Their belief in me and their constant motivation have been a driving force in my pursuit of excellence. I am grateful for their sacrifices, understanding, and encouragement during the ups and downs of this journey.

Contents

Abstract	ii
Acknowledgements	iv
Contents	v
List of Tables	viii
List of Figures	ix
List of Abbreviations	xiv
1 Introduction	1
1.1 Plume rise original measurements	2
1.2 Airborne measurement over the Alberta oil sands operations	4
1.3 A suggested image processing-based method for Plume rise measurements	4
1.4 A suggested feature extraction method for plume cloud recognition	7
1.5 Comparison of different image processing-based methods for plume cloud recognition	9
1.6 The basic concept and motivation of the study	11
1.7 Deep learning in smoke identification	13
1.8 RCNNs' evolution	16
1.9 Goals and objectives	16
1.10 Research impact	18
1.11 Thesis overview	18
2 Background	19
2.1 Plume rise theory	19
2.1.1 Plume behaviors	19
2.1.2 Plume rise equations	20
2.2 Neural networks	26
2.2.1 Artificial neural networks	26
Neurons	27
Neural network layers	29
2.2.2 Single-layer neural network training	29
Gradient descent optimization algorithm	31

Stochastic Gradient Descent (SGD) optimization algorithm	33
Mini-batch Stochastic Gradient Descent (mini-batch SGD)	34
Root Mean Squared Error Propagation (RMSProp) method	37
Adam method	37
2.2.3 Multi-layer neural network training	39
2.2.4 Deep learning	41
Convolutional neural network and convolutional layer	45
Fully convolutional networks	46
2.2.5 Mask R-CNN	47
RPN	48
Loss function	49
3 Plume Imaging	50
3.1 Site description	50
3.2 Data collection	51
3.2.1 Recent approaches in data generation	51
3.2.2 Project data collection	53
3.2.3 Data characterization	54
4 DPRNet Pipeline	57
4.1 DPRNet	58
4.1.1 Structure	58
4.1.2 Physical module	59
4.1.3 Loss regularizer module	60
4.1.4 Training	61
4.2 NBP extraction	62
4.2.1 Centerline extraction	62
4.2.2 Asymptotic function fitting	62
4.2.3 Critical point estimation	64
5 DPRNet Comparison Results	65
5.1 Model validation metrics	65
5.2 Comparison with existing smoke recognition methods	66
5.3 Image measurement	67
6 Geometric Transformation and Validation	73
6.1 Geometric transformation	73
6.2 Uncertainty analysis of the proposed image-based measurements	79
6.2.1 Sources of uncertainties	79
Uncertainty caused by image distortion due to severe atmospheric conditions . . .	80
Uncertainty of the feature extraction performed by the proposed method	80
6.2.2 Uncertainty quantification via image inspection	80

7 Investigation of Briggs' Equations	85
7.1 Briggs plume rise parameterization in GEM-MACH	85
7.1.1 Continuous Emissions Monitoring System (CEMS)	87
7.1.2 WBEA weather stations	88
7.1.3 Plume rise variables for our study	91
7.1.4 Plume rise formulae	93
7.2 Plume rise measurements using the DCNN-based framework	95
7.3 Plume rise comparison results	95
7.4 Assessing Briggs performance in plume rise estimation	98
7.5 Plume rise horizontal distance	99
7.6 Uncertainty of the wind direction	103
7.6.1 Spatial difference	103
7.6.2 Sampling frequency and temporal difference	103
7.6.3 Wind sensor accuracy	103
7.6.4 Atmospheric conditions	103
7.7 Sensitivity analysis	104
7.8 Total uncertainty and plume rise comparison results	107
7.9 Correction factor for Briggs plume rise parameterizations	108
8 Conclusion and Future Work	109
References	111
Appendices	121
A.1 Gray Level Co-occurrence Matrix	121
A.2 Otsu threshold selection method	121
A.3 Image dilation	122

List of Tables

1.1	Summary of studies to compare plume rise parameterizations to observations.	4
1.2	Region properties in the proposed method.	8
3.1	Syncrude stacks information (Protection, 1998), including location, stack height (h_s), stack diameter (d_s), effluent velocity at the stack exit (ω_s), and effluent temperature at the stack exit (T_s). The velocities and temperatures are averages for the entire capturing period. . .	52
3.2	CCFC camera specifications (Reproduced from https://www.campbellsci.com/ccfc , last access: November 2021.	54
3.3	Sample time coding for the recorded images.	55
5.1	Comparison of different methods for plume cloud recognition using average validation metrics values.	66
5.2	Plume rise and plume rise distance values of each of the four plume images from Figure 5.4. φ is the wind direction angle from the North and θ is the wind direction angle at the smokestack location. ΔZ and D_P are respectively plume rise and plume rise distance. . .	72
6.1	Accuracy assessment of the image-based measurements of plume rise and plume rise distance vs. the visual inspection measurements, using two metrics of MAE and RSME, as well as normalized percentage values of these errors.	84
7.1	The variables used for plume rise calculations in GEM-MACH model.	85
7.2	Tha access links to find continuous data of the WBEA weather monitoring stations (Reproduced from https://wbea.org/data/network-map-station-data).	89
7.3	Comparison of the differences between modelled plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework in relation to diurnal and seasonal variations.	99
7.4	Variations of the plume rise measurements resulting from the wind direction sensitivity analysis.	106

List of Figures

1.1	Smokestack plume cloud, Syncrude facility, Alberta, Canada.	2
1.2	The flights' paths over the Alberta oil sands operations. Blue and yellow arrows, respectively, show the north and wind directions. Map image data is provided by CNES/SPOT, Digital Globe, and Google (Reproduced from Gordon et al. (2015)).	5
1.3	Flow diagram of the plume rise measurement.	7
1.4	Responses of RGB and HSV colour channels in constructing the GLCM for the direction of 45 degrees.	8
1.5	Results of unsupervised binary classification methods, (a) K-means, (b) Otsu, (c) our proposed method.	10
1.6	The scheme of the proposed framework.	12
1.7	Results of smoke recognition using learning-based fuzzy method considering colour features (Reproduced from Yuan, Liu, and Zhang (2019)).	14
1.8	Summary of evolution for Mask RCNN. RoI refers to the Region of Interest in the image.	17
2.1	Plume temperature profiles and behaviors (Reproduced from Briggs (1982)). The solid and dashed lines represent the real temperature profile and DALR with altitude, respectively. DALR is the rate at which the temperature of a parcel of dry air falls with altitude, which is 5.5°F per 1000 ft or 9.8°C per km in the Earth's atmosphere.	20
2.2	The variables used to describe the plume. Δz and x represent the plume rise and plume rise distance in the transition phase. Also, Δh and x_f show the plume rise and plume rise distance in the final phase. The green point displays the smokestack exit, and the yellow and red points show the transition and final phase of the plume, respectively. The blue curve is the plume centreline.	24
2.3	Neuron (Reproduced from Samarasinghe (2016)).	28
2.4	The mathematical model of a neuron (Reproduced from Skansi (2018)).	28
2.5	Sigmoid activation function.	29
2.6	Types of neural networks regarding the number of layers.	30
2.7	Neural network training process (Reproduced from Kim (2017)).	31
2.8	Performance of gradient descent method for a two-dimensional error function (Reproduced from Chollet (2021)).	31
2.9	Performance of descending gradient for different values of learning rate.	33
2.10	How to select training dataset in the gradient descent method (Reproduced from Kim (2017)).	33

2.11	How to select training instances in the stochastic gradient descent method (Reproduced from Kim (2017)).	35
2.12	How to select training instances in the mini-batch stochastic gradient descent method (Reproduced from Kim (2017)).	36
2.13	The role of learning rate and momentum in reducing a three-dimensional error function (Reproduced from Skansi (2018)).	36
2.14	Performance of Adam algorithm. The curves represent a quadratic error function. Yellow, Blue, green, and purple lines show the performance of the SGD, SGD+Momentum, RMSProp, and Adam algorithms, respectively, during successive iterations of network training and reaching the minimum value for the error function (red dot). A comparison of the performances indicates that the Adam optimization algorithm has taken full advantage of both using the momentum and using the technique in the RMSProp method.	39
2.15	Input information processing in different layers of the neural network (Reproduced from Chollet (2021)).	41
2.16	Graph of the sigmoid function along with its gradient.	43
2.17	Graph of the ReLU function.	43
2.18	How to slide the kernel in a two-dimensional convolutional layer (Reproduced from Skansi (2018)).	45
2.19	Sharing parameters in the convolutional network compared to the fully connected network. Assuming input units x and output units s , black arrows in each network indicate connections that used a specific parameter (Reproduced from Goodfellow, Bengio, and Courville (2016)).	46
2.20	How to apply kernels to the input 2-D image in a 2-D convolutional layer (Reproduced from Goodfellow, Bengio, and Courville (2016)).	47
2.21	An example of a fully convolutional network for pixel-wise classification applications (Reproduced from Zhang et al. (2021)).	48
2.22	Region proposal network. Red rectangles illustrate the proposal regions on the feature map of the input image.	48
3.1	Imaging situation on the satellite image of the study region. Camera mounted on Buffalo Viewpoint station, C ; wind sensors on Mildred Lake station, W ; and the tallest smokestack location, S	51
3.2	Plume imaging system. The red circle shows the camera location.	53
3.3	Sample collected images.	53
3.4	Sample images (up) and their corresponding ground truth (down) from our DPR dataset listed as (A) Clear daytime, (B)&(C) cloudy day, and (D)&(E) clear nighttime.	55
3.5	Sample outlier images.	55
3.6	Distribution of the collected images from January 2019 to December 2019.	56

4.1	Plume rise measurements system framework. x_p and z_p are the NBP coordinates in the image scale. Similarly, X_P and Z_P represent the NBP coordinates in the real-life scale. ΔZ and D_P are plume rise and plume rise distance on a real-life scale. The $x - z$ system is chosen for visualizing the plume rise measurements to be consistent with other studies in environmental science.	57
4.2	Plume rise, ΔZ ; plume rise distance, D_P ; and NBP, P ; on a sample image. θ represents the plume cloud deviation from the projection plane due to the wind, and φ denotes the wind direction. S indicates the smokestack exit, C represents the camera position, and the blue line shows the plume cloud centerline.	58
4.3	DPRNet architecture. The supplemental modules are shown in green, and the dashed blue rectangle is dismissed in the inference time.	59
4.4	Sample plume cloud segments with eight boundary points.	60
4.5	DPRNet training results.	61
4.6	NBP extraction framework. Dashed arrows show the output of each block. The red curve represents the centerline of the plume cloud. The cyan and yellow lines, respectively, display the upper and lower boundaries of the plume cloud. Green dashes demonstrate the asymptotic curve, and the magenta point is NBP.	62
4.7	Examples of alternative fitting curves for a specific plume cloud recognition result to find the trend of its centerline. The red curve represents the centerline of the plume cloud. The cyan and yellow lines, respectively, display the upper and lower boundaries of the plume cloud, and green dashes demonstrate the fitting curve accordingly.	63
5.1	Performance of different methods regarding some test images. The horizontal axis in all three figures demonstrates the image number.	68
5.2	Detailed comparison of methods over three datasets employing three validation metrics.	69
5.3	Qualitative results of recognition tasks listed as (A) Input image, (B) corresponding ground truth, (C) results of Mask R-CNN, (D) FCN, (E) results of DeepLabv3, and (F) results of DPRNet.	70
5.4	DPRNet and image measurement results. In column (c), the red curve represents the meandering of the plume cloud. The cyan and yellow lines, respectively, illustrate the upper and lower boundaries of the plume cloud. Green dashes show the asymptotic curve; the magenta asterisk is NBP. (a) Input image; (b) DPRNet result; and (c) Image measurement.	71
6.1	The coordinate systems.	74
6.2	Schematic view of our imaging situation. Camera station, C ; distance between camera and smokestack, D ; smokestack position, S ; wind direction, W ; and plume cloud, PC . The abc coordinate system is only for differentiating the side and camera views and is not used as a co-ordinate reference system.	74
6.3	The schematic 3D view of the region in the real-world coordinates.	75
6.4	The schematic top view of the region in the real-world coordinates.	76
6.5	The schematic side view of the region in the real-world coordinates.	76

6.6	Smokestack location schemes. Smokestack exit, s ; and image center, o . PC centerline is represented with a blue line, and NBP image coordinates are shown with $p(x_p, z_p)$ and $p(x_p, y_p)$, respectively, in camera view and top view schemes.	77
6.7	Sample original image and the NBP positions obtained from our measurements and from the visual inspection. The red star displays the result given from our image-based measurements, and the green star shows the results obtained from image inspection. The orange curve represents the asymptotic fitting curve of the plume cloud until it saturates on the NBP.	81
6.8	Comparison of the image-based measurement NBP locations with the NBP locations from the visual inspection.	82
6.9	Comparison of the image-based plume rise measurements with these measurements from the visual inspection.	83
7.1	Data flowchart for running Briggs parameterizations. ΔZ and D_p represent the plume rise and the plume rise distance, respectively.	88
7.2	The network map of the weather stations in the region (Reproduced from https://wbea.org/data/network-map-station-data). The under-study area is illustrated by a red rectangle. The ambient air monitoring stations are shown in blue spots; green spots are the edge plots, and purple spot shows meteorology towers.	90
7.3	The location of the following (https://wbea.org/data/network-map-station-data): Syncrude smokestack with a red circle (Lat:57.041, Long:-111.616), Mildred Lake air monitoring station with a yellow circle (Lat:57.050, Long:-111.564), Buffalo Viewpoint air monitoring station with a green circle (Lat:56.996, Long:-111.594), Lower Camp meteorological tower with a blue circle (Lat:57.026, Long:-111.500), and Mannix air monitoring station with a magenta circle (Lat:56.967, Long:-111.482). Mildred Lake air monitoring station is the closest station to the smokestack location, and our original wind direction information is obtained from this station.	90
7.4	The calculated plume rise using Briggs parameterizations (Equations 7.7 - 7.12), and the proposed DCNN-based framework with the input wind data from Mildred Lake monitoring station.	96
7.5	The distribution of plume rise calculated by Briggs parameterization (Equations 7.7 - 7.12), and the proposed DCNN-based framework with input wind data from Mildred Lake monitoring station. Distributions are illustrated for each hour and for the tallest smokestack in the area (Table 3.1). The summation of all plume rise values, which are more than 700 m, is illustrated by the right-most histogram outlined in the box. Cumulative distributions are depicted by dashed lines for each method.	96
7.6	Comparison of the predicted plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. The magenta line displays a least-square reference line, the slope of which gives the correlation coefficient of $R = 0.10$. PR stands for plume rise in this figure.	97

7.7	Comparison of the predicted plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. PR stands for plume rise in this figure.	98
7.8	The calculated plume rise distance using Briggs parameterizations (Equations 7.14 - 7.16), and the proposed DCNN-based framework with the input wind data from Mildred Lake monitoring station.	100
7.9	The distribution of plume rise distance calculated by Briggs parameterization (Equations 7.14 - 7.16), and the proposed DCNN-based framework with input wind data from Mildred Lake monitoring station. Distributions are illustrated for each hour and for the tallest smokestack in the area (Table 3.1). The right-most histogram outlined box illustrates the summation of all plume rise values, which are more than 6000 m. Cumulative distributions are depicted by dashed lines for each method.	101
7.10	Comparison of the predicted plume rise distance using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. The magenta line displays a least-square reference line, the slope of which gives the correlation coefficient of $R = -0.002$. PRD stands for plume rise in this figure.	102
7.11	Comparison of the predicted plume rise distance using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. PRD stands for plume rise in this figure.	102
7.12	Circular statistics of the wind direction angular differences between three monitoring stations in Figure 7.3.	104
7.13	Wind direction sensitivity analysis relative to the plume rise calculations.	105
7.14	Wind direction sensitivity analysis relative to the plume rise distance calculations.	106
A.1	Gray level co-occurrence matrix.	122
A.2	Dilation of set A by set B (Reproduced from Gonzalez, Woods, and Eddins (2009)).	123

List of Abbreviations

AEP	Alberta Environment Parks
AER	Alberta Energy Regulator
ANN	Artificial Neural Networks
AQHI	Air Quality Health Index
AQI	Air Quality Index
ASPP	Atrous Spatial Pyramid Pooling
CAMx	Comprehensive Air Quality Model with extensions
CEMS	Continuous Emissions Monitoring System
CFD	Computational Fluid Dynamics
CMAQ	The Community Multiscale Air Quality modeling system
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DALR	Dry Adiabatic Lapse Rate
DCNN	Deep Convolutional Neural Network
DPDR	Deep Plume Rise Dataset
DPRNet	Deep Plume Rise Network
ECCC	Environment and Climate Change Canada
EPA	Environmental Protection Agency
FCN	Fully Convolutional Network
FPN	Feature Pyramid Network
GEM-MACH	Global Environmental Multiscale model - Modelling Air quality and CHemistry
GMM	Gaussian Mixture Model
GLCM	Gray Level Co-occurrence Matrix
GLDM	Gray Level Difference Matrix
GLRLM	Gray Level Run Length Method
GPU	Graphics Processing Unit
LBPV	Local Binary Patterns based on Variance
MRF	Markov Random Field
MSE	Mean Squared Error
NBP	Neutral Buoyancy Point
NLP	Natural Language Processing
PSM	Power Spectral Method
ReLU	Rectified Linear Unit
RMSProp	Root Mean Squared Error Propagation
ROI	Region of Interest
RPN	Region Proposal Network
SCO	Synthetic Crude Oil
SGD	Stochastic Gradient Descent
SSA	Single Spectrum Analysis
SVM	Support Vector Machine
TERRA	Top-down Emission Rate Retrieval Algorithm
WBEA	Wood Buffalo Environment Association

1 Introduction

When a smokestack releases pollutants, the resulting plume will rise due to momentum and buoyancy until it dissipates and is carried downwind at a constant height. This height is called plume rise height or plume rise. Plume rise calculation is not straightforward, and it is a significant problem in predicting the dispersion of different harmful effluents into the air (Briggs, 1982). Plume rise influences how far pollutants are carried downwind, the concentration of pollutants at the surface, where the pollutants will be deposited on the green environment or inhaled by people, and the amount of greenhouse gases mixed into the upper troposphere. Therefore, accurate measurement of the plume rise is of concern for research and operational applications such as air-quality transport models, local environment assessment cases and global climate models (Ashrafi, Orkomi, and Motlagh, 2017).

The parameterizations of plume rise prediction were developed in the 1960s by Briggs (Briggs, 1969; Briggs, 1975). Dimensional analysis was used to estimate the plume rise based on smokestack parameters and meteorological measurements in different atmospheric conditions. Early observations of plume rise were used to test and modify the parameterizations developed using dimensional analysis (Bieser et al., 2011). Some of these calibration techniques were wind tunnel studies and field observations using technologies including film photography, theodolites, and cloud-height searchlights (Bringfelt, 1968). Three-dimensional air-quality models that use these equations include GEM-MACH (Makar et al., 2015), CAMx (Emery, Jung, and Yarwood, 2010), and CMAQ (Byun, 1999).

The parameterizations of plume rise prediction were tested in the 1970s and 1980s. The studies compared plume rise parameterizations to observations and showed that Briggs overestimated the plume rise (Rittmann, 1982; England, Teuscher, and Snyder, 1976; Hamilton, 1967; Moore, 1974). In 1993, an aircraft-based measurement was done to measure SO₂ emissions of a power plant which demonstrated an overestimation of about 400 m (Sharf et al., 1993). Although these earlier studies indicated some degree of overestimation, in 2002 Webster et al. performed surface measurements and concluded that the Briggs parameterizations tend to underestimate plume rise (Webster and Thomson, 2002). In 2013, as part of the Canada-Alberta Joint Oil Sands Monitoring Plan, an aerial measurement study was done in the Athabasca oil sands region of northern Alberta to study dispersion and chemical processing of emitted pollutants (Gordon et al., 2015; Gordon et al., 2018; Akingunola et al., 2018). The project consisted of 84 flight hours of an instrumented Convair aircraft over 21 flights designed to measure pollutants emissions, study the transformation of chemicals downwind of the industry, and verify satellite measurements of pollutants and greenhouse gases in the region. Using aircraft-based measurements and reported smokestack parameters and meteorological data, it was demonstrated that the Briggs equations significantly underestimated plume rise at this location, in contrast to previous results.

Given the results of this study and the gap of more than 30 years since the Briggs equations were developed and previously tested, there is a need for further testing and possible modification of the Briggs equations based on modern observation techniques. In recent decades, there have been many significant advancements in environmental monitoring activities over industrial regions for safety and pollution prevention (Isikdogan, Bovik, and Passalacqua, 2017b; Isikdogan, Bovik, and Passalacqua, 2017a; Gu, Qiao, and Lin, 2018; Gu, Qiao, and Li, 2018). Moreover, several smoke border detection and recognition models have been introduced recently using digital image analysis, such as wavelet and support vector machines (Gubbi, Marusic, and Palaniswami, 2009), LBP and LBPV pyramids (Yuan, 2011), multi-scale partitions with AdaBoost (Yuan, 2012), and high-order local ternary patterns (Yuan et al., 2016) which are well-performed and impressive. The novel results of image-based analysis and predictions of these studies lay the required foundation for innovative instrumentation that can be deployed near any smokestack to give information on pollutant dispersion and potential exposure to people downwind. This information will be based on real-time observation, i.e. video images, instead of potentially erroneous and decades-old parameterizations. Moreover, the industry might use these real-time systems to track and survey the released pollutants online and obtain information to update related criteria, such as live Air Quality Health Index (AQHI) (Logothetis et al., 2023) and Air Quality Index (AQI) (Lin and Chen, 2023) updates. In recent decades, computer vision techniques have significantly improved predictions, including weather forecasters who use radar in weather prediction. Therefore, real-time vision-based measurement techniques can be used to strengthen air-quality predictions.



FIGURE 1.1: Smokestack plume cloud, Syncrude facility, Alberta, Canada.

1.1 Plume rise original measurements

Various plume models introduced by researchers incorporate terms related to atmospheric turbulence in the momentum and buoyancy equations. An early example is Priestley's studies (Priestley, 1953; Priestley, 1956; Priestley and Ball, 1955), suggesting that buoyant parcels lose momentum and buoyancy to their surroundings, influenced by their size and ambient turbulence. He simplified the calculations by considering the loss rate as a constant, making the solution usable for parcels with regular sizes during the rise. He also divided the trajectory into buoyancy-dominated and atmosphere-dominated phases.

This definition aimed to reduce errors in ignoring atmospheric turbulence during the first phase and induced turbulence during the second.

Turner (1963) proposed a formulation of rising thermal updrafts, suggesting that ambient turbulence causes the thermal to lose mass by inducing a constant fluid removal velocity proportional to the ambient turbulence velocity. This turbulence weakens the thermal core, finally stopping its rise. Turner (1963) defined the 'active' thermal core as the portion with a systematic upward motion, surrounded by a larger passive zone. This definition aligns with the requirements for the unknown radius in the plume conservation equations.

Benjamin (1975) proposed a physical mechanism for turbulence-induced mass loss in convective turbulence, where chimney plume clouds lose mass and buoyancy by entrainment into thermal updrafts in the atmosphere-dominated phase. Csanady (1964) and Csanady (1973) used Laplacian to derive 'decay constants' for temperature and velocity fluctuations, adapting Csanady's turbulent decay concept to continuous plume clouds. They assumed that plume momentum and buoyancy are temporary 'fluctuations' charged by the surrounding ambient flow, cascading down to more minor eddy scales. Djurfors (1977) proposed a plume rise model showing levelling-off behaviour depending on decay constants. However, these terms needed more justification, and with a mechanism for plume buoyancy and momentum to cross the plume boundary, the proposed equations remained curious.

Despite qualitative agreement with observations, turbulent plume models lack widespread use due to challenges like restrictive assumptions, complicated solutions, and insufficient supporting data. These models provide valuable ideas but require further exploration beyond traditional plume rise literature.

As discussed in Section 1, a wide variety of methods have been introduced for early observation of plume rise. Visual tracking of the plumes was performed on Plexiglas screens to be compared to the height of nearby towers and measured using lidar. Other approaches to measuring the plume rise included releasing balloons from the smokestacks connected with Geiger counters, using aircraft-mounted instruments for sampling fluorescent particles and utilizing wind tunnel simulations. As summarized in Bieser et al. (2011), these observation methods tried to find appropriate values for dimensional analysis constants by confining the plume rise parameterizations. Five studies were done to compare the newly developed plume rise parameterizations to observations for an accuracy test. The results showed that the Briggs parameterizations overestimate plume rise—Table 1.1 represents these studies in more detail.

After about 20 years, Sharf et al. (1993) performed an aircraft-based experiment for measuring the power plant plume rise. They demonstrated that the Briggs parameterizations generally overestimate the plume rise by up to 400 m. In 2002, Webster and Thompson used a network of surface concentration measurements in a power plant to test a Lagrangian model and Briggs parameterizations. This time, Briggs showed a tendency to underestimate the plume heights. Most original measurements have indicated that the Briggs plume rise algorithm somewhat overestimates the actual plume height. However, the recent study demonstrates an underestimation in using these parameterizations.

TABLE 1.1: Summary of studies to compare plume rise parameterizations to observations.

Author(s)	Method	Findings
Giebel (Gielbel, 1979)	Measured pit coal power plants plumes with lidar	Observations 50% lower than the Briggs
Rittmann (Rittmann, 1982)	Measured industrial -sized sources plumes	Observations 12%-50% lower than the Briggs
England et al. (England, Teuscher, and Snyder, 1976)	Measured a gas turbine facility plumes with airborne measurements of NO _x	Observations 30% lower than the Briggs
Hamilton (Hamilton, 1967)	Measured power station plumes with lidar	Observations 50% lower than the Briggs
Moore (Moore, 1974)	Used data from seven locations measured by balloons, photography, aircraft, and lidar	Observations 10%-20% lower than the Briggs

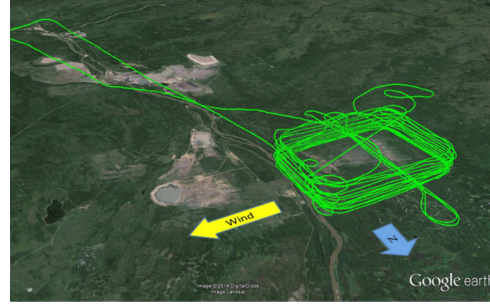
1.2 Airborne measurement over the Alberta oil sands operations

In 2013, an aerial measurement study was done in the oil sands development region north of Fort McMurray in the Athabasca region of Alberta, Canada (Gordon et al., 2015; Canadian Natural Resources Ltd, 2015). The project consisted of 84 flight hours of an instrumented Convair aircraft over 21 flights designed to measure pollutants emissions, study the transformation of chemicals downwind of the industry, and verify satellite measurements of pollutants and greenhouse gases in the region using the box method. This method is performed by repeatedly flying a square (Alfieri et al., 2010) or a polygon (Kalthoff et al., 2002) pattern around the region over different heights. These flight patterns were designed based on the maximum integrated emission rates of the facilities and the plume cloud's detailed vertical structure. Figure 1.2 shows the multiple flights' routes around the emission area.

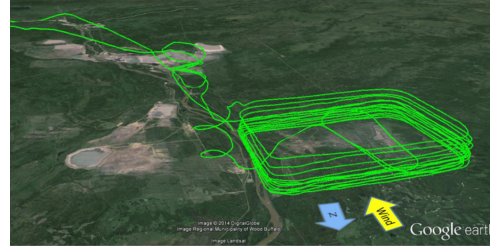
Following the flight patterns, Gordon et al. (2015) verified all possible error sources. They extrapolated the measurements to the ground to modify two analyses of Alfieri et al. (2010) and Kalthoff et al. (2002), which led to an improved data analysis algorithm called Top-down Emission Rate Retrieval Algorithm (TERRA). TERRA has improved the uncertainty of emission estimation resulting from the chosen interpolation and extrapolation algorithm. Using aircraft-based measurements, reported smokestack parameters, and meteorological data, it was demonstrated that the Briggs equations significantly underestimated plume rise at this location (Akingunola et al., 2018; Gordon et al., 2018).

1.3 A suggested image processing-based method for Plume rise measurements

The proposed approach here starts with an image analysis-based plume cloud recognition algorithm, which takes advantage of combining colour information analysis (Phillips Iii, Shah, and Vitoria Lobo,



(A) The flight path of August 20.



(B) The flight path of September 2.

FIGURE 1.2: The flights' paths over the Alberta oil sands operations. Blue and yellow arrows, respectively, show the north and wind directions. Map image data is provided by CNES/SPOT, Digital Globe, and Google (Reproduced from Gordon et al. (2015)).

2002) and a co-occurrence matrix as a texture-based analysis tool (Chen and Pavlidis, 1979; Longstaff et al., 1995; Haralick, 1979). Moreover, Thanks to the fixed camera position, there is a constant, defined ROI location. Using automated digital image analysis software, we have proposed a relatively low-cost and reproducible method to measure plume rise. The architecture of the plume cloud detection can be summarized in two steps. First, localization of the ROI and then segmentation of the suspected plume cloud area using extracted plume features and a recognition algorithm. ROI localization methods are not investigated due to the constant survey area.

One approach to plume cloud segmentation using vision-based systems is motion characteristic analysis of the plume, which is used in Yu, Mei, and Zhang (2013) and Shi et al. (2015). Also, Wu, Du, and Li (2008) and Zen et al. (2013) took advantage of the GMM to detect the motion region of the plume and Wang and Chen (2019) combined rough set and region growing methods as a plume image segmentation algorithm, which seems to be a time-consuming algorithm due to the computational burden of the region growing method. Although the use of colour information raises the false alarm rate as plume colour is primarily similar to the surrounding environment theme, the combination of motion and colour characteristics of the plume is also known as a standard method in plume segmentation (Jia et al., 2016; Jian et al., 2018; Yuan, Hou, and Pu, 2018; Zhao et al., 2019b; Zhao, 2015). Some algorithms utilize Infra-Red images and video frames in their experiments (Ajith and Martínez-Ramón, 2019), which are not easily accessible and can increase the project's costs. Moreover, using digital images makes the algorithm more flexible as it can be used with more hardware. On the other hand, some plume clouds are too close to the background temperature to be captured by the near red-channel wavelength.

Here, a plume cloud segmentation algorithm is proposed by combining colour information analysis (Phillips Iii, Shah, and Vitoria Lobo, 2002) and a co-occurrence matrix as a texture-based analysis tool (Chen and Pavlidis, 1979; Longstaff et al., 1995; Haralick, 1979). Thanks to the fixed camera position in this project, there is a constant, defined ROI location. Therefore, we have concentrated on recognizing plume clouds over their different background conditions. Among roughly 20 features to extract information from co-occurrence matrix (Longstaff et al., 1995), according to the physics of our problem, we try to extract features within certain angle attitudes. Colour information can be used to differentiate plume cloud points from their background and differentiate these two segments, which are highly similar in tone cases. The summation of wind and plume acceleration directions dictates a particular domain for varying plume cloud direction, which can help us concentrate on certain angles for feature extraction. This automated and high-speed algorithm can be a suitable approach for real-time analysis of plume rise in industrial regions, which is proposed in three main stages. Initially, a 45-degree co-occurrence feature is extracted. Then, the extracted feature is segmented based on the most well-known, non-parametric and unsupervised classification method for selecting the automatic threshold in image segmentation, namely Otsu's method (Otsu, 1979). This step is called coarse plume recognition, including small parts removal and image dilation method. Finally, a proposed unsupervised rule-based classification method is used to illustrate the trajectory of the plume and distinguish the leading plume cloud from its separated small parts, called fine plume recognition. These plume parts are not in the trajectory of the main plume stream, and using rule-based region property constraints can help us find the main plume cloud trajectory. Consequently, after processing an automated measurement, the final segmentation result and plume rise tracker with desired points are obtained.

We introduce a plume cloud segmentation method to extract the plume from its background in industrial regions and measure the plume rise for any target smokestack. A flow diagram of the plume rise measurement is shown in Figure 1.3. As highlighted, the proposed method consists of three main steps: feature extraction, coarse plume recognition and fine plume recognition to detect the plume and its properties.

Before analyzing the plume cloud borders in the image, we should look for suspected areas for plume cloud existence, called RoI. This area contains the plume regions, and the localization of this rectangular shape reduces the number of pixels for analysis, leading to an increase in the precision of our detection algorithm. In this approach, the source of plume generation is fixed and is at a constant distance from our camera station. Therefore, we can concentrate on the plume cloud region and try to extract it from its background, which makes the problem more straightforward: separating the plume objects as foreground from its sky background in different atmospheric conditions. We have concentrated on the tallest stack in the area to isolate a single plume for a desirable stack, which can help us find the plume rise for one plume generation source.

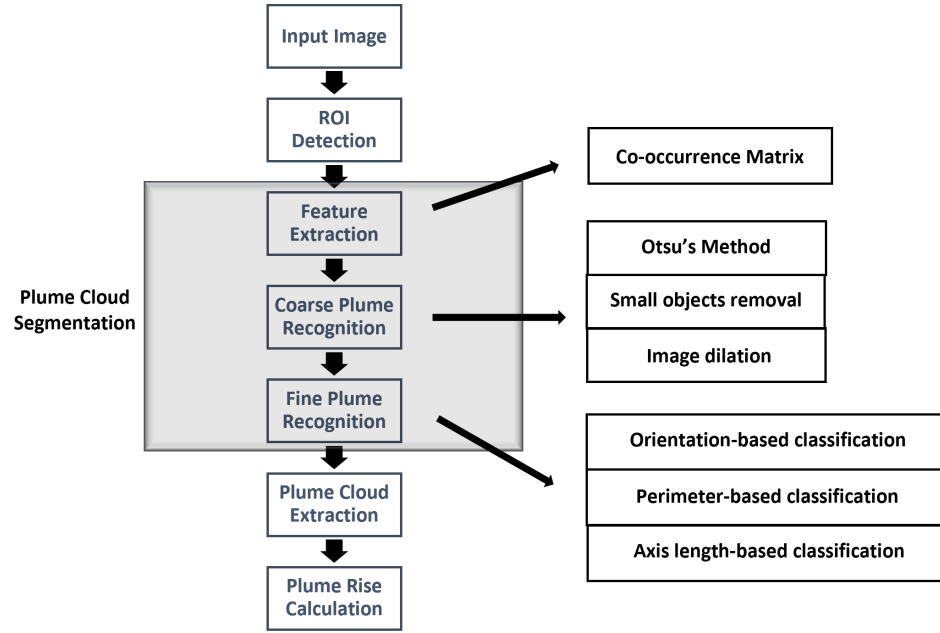


FIGURE 1.3: Flow diagram of the plume rise measurement.

1.4 A suggested feature extraction method for plume cloud recognition

Generally, object detection using a suitable and efficient feature is crucial. Due to the fuzziness of the plume cloud in colour and texture as static characteristics and frequency and shape as dynamic characteristics, it is not easy to define the plume boundaries in some images, even with human vision skills. Among different feature extraction methods, the co-occurrence matrix is a powerful tool to highlight the plume cloud from its background. As the plume is going out of the stack, due to the release acceleration, relatively high temperature of the plume, and wind speed vector above the stack, it can be expected that the plume will start its direction within certain angles near 45 degrees or 135 degrees relative to the horizontal line.

Moreover, one of the critical problems in plume detection through different atmospheric conditions is coping with the cloudy background of the plume, which is similar to the plume in colour, shape, and motion. Using the method mentioned earlier with the ability to track features in specific directions will lead to distinguishing the mostly horizontal clouds from the plume, and different direction sets can be added to the algorithm to magnify the feature extraction in that particular direction. Therefore, even if there is no wind in the region, we can add the 90-degree attitudes to the plume's possible direction sets to detect the vertical plume. The red channel of the input image outperforms other channels, even in HSV format, in representing the plume cloud region. Figure 1.4 represents the responses of RGB and HSV colour channels in constructing the GLCM for the direction of 45 degrees. Therefore, we have applied the first filter (Red channel) to extract near red wavelengths in the image, and its result is being inserted into the feature extraction algorithm to achieve the GLCM as the output of this phase.

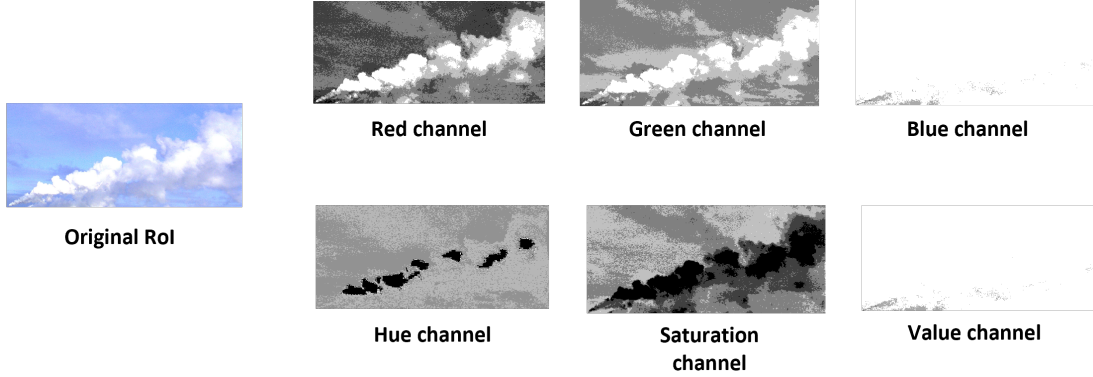


FIGURE 1.4: Responses of RGB and HSV colour channels in constructing the GLCM for the direction of 45 degrees.

In the following, we go through two parts: coarse plume recognition and fine plume recognition. Firstly, plume pixels are recognized by Otsu's method. Due to the fuzziness of the plume cloud and sometimes separated plume pixels, this classification is not accurate for scattered plume clouds. Therefore, we tried to unify the plume pixels using the dilation method to create a consistent flow of the plume morphologically and remove the smaller areas to better track its direction. In the second step, after unifying the cloud plume, we applied an unsupervised rule-based classification to find the exact pixels of the plume among non-plume pixels. This rule-based classification helps us find the plume cloud pixels using some region properties in the plume area that provide shape measurements.

The most crucial property is the direction, which defines the direction of the plume cloud mainstream and is used as the priority. Direction is the most potent property to rely on since it is clear that the plume cloud will start its trajectory according to the predicted directions with certain angles. Therefore, the algorithm crosses out the other plume objects with non-defined directions before going through the next property. Then, the perimeter of the objects is investigated, and reasonably, the more oversized plume objects are recognized to be compared by their length in the last property survey. The plume cloud should be the plume object that goes further in the image and has the maximum length of the central axis. These rules lead to finding the desirable cloud plume stream's trajectory. It is clear that adding new rules or modifying the priority in the current ones makes the algorithm more flexible for a desirable dataset. Table 1.2 represents the region properties used in our proposed method and their descriptions. We go through properties supported by GPU systems to be utilized in real-time measurement processing.

TABLE 1.2: Region properties in the proposed method.

Property name	Description
Orientation	Angle between the horizontal axis and the major axis of the ellipse with same second-moments as the region
Perimeter	Distance all over the boundary of the region
Major Axis Length	Length of the major axis of the ellipse with same normalized second central moments as the region

1.5 Comparison of different image processing-based methods for plume cloud recognition

In order to evaluate the performance of the proposed method, it is compared with the K-means algorithm and Otsu's method.

- **K-means clustering** (Hartigan and Wong, 1979) an unsupervised learning algorithm which is one of the main methods used in segmentation of smoke and fire (Ajith and Martínez-Ramón, 2019).
- **Otsu's algorithm** (Otsu, 1979): our proposed method as a standard free-parametric method without using fine smoke recognition (coarse smoke recognition).
- Our proposed method is based on Otsu's method, image dilation and region property analysis.

As mentioned before, according to the physics of the problem, which restricts the direction of the released plume within certain angles at the start point, we use the 45-degree GLCM feature as an effective feature extraction method based on the motion of the pixels. Smoke pixels are much clearer in the red channel of the plume image in the domain of this feature, so we can hope to detect the plume through this domain using unsupervised methods. Figure 1.5 illustrates the results for different unsupervised binary classification methods on six different images from our dataset. The aim of the K-means algorithm is to separate a number of points and put them into some clusters in which the sum of squares is minimized. In this algorithm, a different number of iterations are required according to the dataset. Moreover, the clustering created by this algorithm is only locally optimal, and usually, it is required to have less than ten times the number of iterations to obtain an optimal solution in a local manner. We have run the algorithm ten times, and the average performances are reported here, considering the closest response to the mean average of the iterations. As we can see in the results, we cannot rely on the K-means algorithm all the time as it is not working globally for all sets of data types.

Moreover, Otsu's method is able to separate the plume cloud from its background. However, the results are not acceptable when it comes to deciding about different parts of scatter classes of the plume cloud in the image, as there are no criteria for smoke stream recognition. This crucial problem is solved by the proposed method based on rule-based region classification and can be referred to as fine smoke segmentation.

Otsu's method performs well in the unsupervised classification of the non-plume and plume parts related to the main smokestack, as seen in Figure 1.5.b for image #5, Otsu cannot connect the separated smoke parts to track the mainstream of the plume in order to follow the plume cloud until it ends. This problem has been resolved by the image dilation algorithm in our proposed method as in Figure 1.5.c for image #5, which connect the closer and smaller parts of the plume together at the end of the plume.

Furthermore, as we mentioned three important region properties as rules of the constraint classification, sample images present the vital effects of this post-processing phase on tracking the plume cloud boundaries. In Figure 1.5.b for image #1, Otsu decided to choose the primary part of the plume, which excludes the mainstream of the plume. Our proposed method has solved this problem in Figure 1.5.c for image #1 using major axis length property, which determines the mainstream according to the length

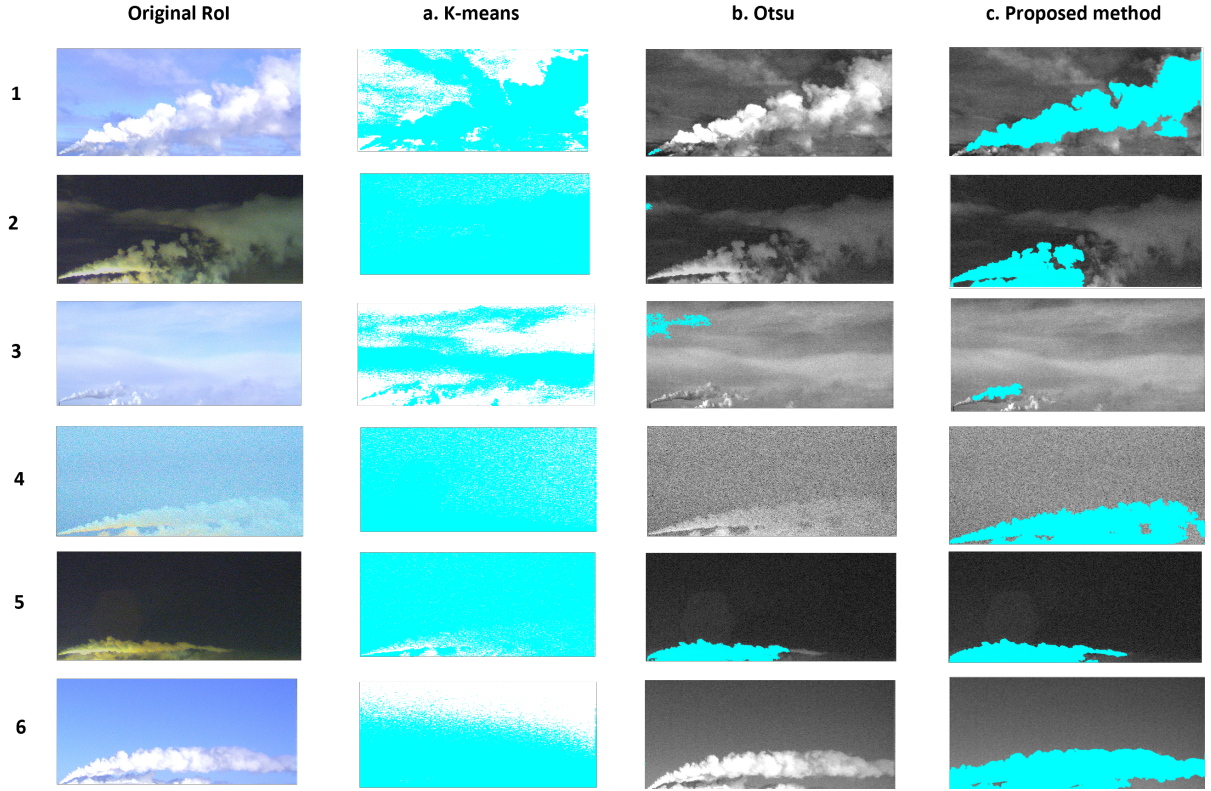


FIGURE 1.5: Results of unsupervised binary classification methods, (a) K-means, (b) Otsu, (c) our proposed method.

of the recognized region. Also, in Figure 1.5.b for image #2, Otsu decided to pick the small part of the plume, while defining perimeter property can be useful as in Figure 1.5.c for image #2, which dictates the algorithm to choose the bigger surface of plume pixels. In Figure 1.5.b for image #3, Otsu has selected an inappropriate plume part, while again direction property could solve this problem in Figure 1.5.c for image #3 dictating plume cloud angular attitude as it is released from the smokestack.

In Figure 1.5.b for image #2 as a foggy day and image #3 as a high-contrasted background, the results are not acceptable, and Otsu's classification may concentrate on small pixel lists in the image. Nevertheless, our proposed method could decide about the plume cloud in these typical images with the high-contrasted sky behind and total-grained capture as in Figure 1.5.c for image #2 and image #3. Changing the rules or their priority in the algorithm can directly affect the fine smoke segmentation section.

This approach is practical in many cases, as it can accurately identify the presence of plume clouds. However, there are many other instances where the strategy cannot recognize the plume clouds effectively. This limitation is because all the features here are manually selected. Manual feature selection involves human experts selecting specific characteristics from their visual experience, while human knowledge may only contain some of the slight patterns and variations present in the data, leading to the method's

inability to recognize plume clouds accurately in certain situations. Also, human interaction in an automated real-time recognition system is unacceptable. To address this concern, a new approach was discussed in this thesis, utilizing DCNN-based methods, which can automatically learn and extract complicated patterns and features directly from the data. Using these techniques, the process no longer depends on manually selected features but allows the algorithm to learn relevant features. This can significantly improve the method's capability to identify diverse plume clouds and become more adaptive and robust, even in challenging scenarios.

1.6 The basic concept and motivation of the study

Our preliminary objective is to design and utilize Deep Convolutional Neural Networks (DCNN) to explore images efficiently. In particular, we aim to use the power of DCNNs to discover valuable information from image data. Recent approaches for analyzing plume clouds in an industrial environment include various challenges, such as depending on manual interventions, which cause weak efficacy and low precision. Our study suggests an innovative solution to address these limitations: integrating DCNNs and imagery to automate and improve the analysis process. Image data plays the role of a window into the plume clouds' behaviours, delivering an understanding of their dispersion patterns. Taking advantage of the training process over these collected datasets of plume cloud images, we reinforced the DCNN to distinguish the dimensions of these plume clouds precisely for measurement purposes. Our research harnesses the capabilities of DCNNs to decode the complicated behaviours of industrial emissions within environmental science, such as smokestacks' plume clouds. The core of our study can be considered an intersection of image acquisition, DCNN, and environmental analysis. These concepts mutually empower us to revolutionize plume cloud analysis and contribute to a greener and cleaner environment.

We developed a novel architecture of DCNN for noise reduction, plume segmentation, plume rise model prediction and plume tracking. The designed experiments and their results will allow us to develop a DCNN for validating conventional plume rise models and provide scientific grounds for creating a new physical model.

We have proposed a novel framework using DCNN algorithms to measure plume rise. Our approach comprises three stages: (1) recognizing the plume cloud region using an improved Mask R-CNN, (2) extracting the plume cloud's Neutral Buoyancy Point (NBP) from the centerline of the recognized plume cloud, and (3) transforming the plume cloud's geometric measurement from an image-scale to real-world scale. Figure 1.6 illustrates the framework of obtaining plume rise information using DCNN.

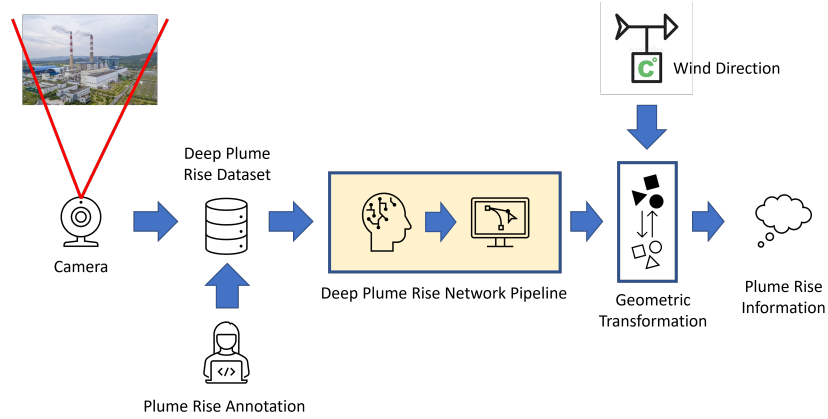


FIGURE 1.6: The scheme of the proposed framework.

The proposed visual data-driven system will be developed in the framework of DCNNs (Simonyan and Zisserman, 2014; He et al., 2016). As a definition, shallow systems refer to traditional machine learning methods, such as Support Vector Machines (SVMs). In contrast, deep learning refers to the learning process of an extended neural network from typically a considerable amount of data. The DCNN is a specific category of machine learning systems, which successfully demonstrated its success outperforming conventional shallow machine learning vision systems with significant margins (Jiang et al., 2018; Zhao et al., 2019a). So, at the ImageNet Large Scale Visual Recognition Challenge, held in 2012, the AlexNet algorithm became a computer vision breakthrough and outperformed its competitive algorithms by 10% (Krizhevsky, Sutskever, and Hinton, 2017). Nowadays, deep neural networks are able to reach more than 98% accuracy in various applications such as static and dynamic object recognition (Karimipour et al., 2019). The DCNN's success is mainly due to the combination of the availability of large-scale training datasets, advancement of computing resources and mathematical findings to estimate a few million parameters, even with small residuals resulting from the difference between training references and prediction results. The most critical benefit of the DCNN is that it can easily generalize the performance of computer vision systems working in a wide range of variations in visual datasets.

Mask RCNN is an instant recognition deep learning model and an extension of Faster RCNN, which can create pixel-level masks for the objects beside object border detection tasks. Consequently, Mask RCNN has several hyperparameters, i.e., parameters that control the learning process, and based on our application, they should be tuned carefully (Khandelwal, 2019). This is the first study to do plume cloud border detection and recognition using deep learning methods. The central theme of this thesis is to exploit the potential of visual information driven from video images captured by a typical off-the-shell camera for measuring the physical states of the smokestack plume cloud.

The strategy above recognizes the plume cloud and measures plume rise in real time. Here, we reinforce the bounding box loss function in Region Proposal Network (RPN) (Hwang et al., 2022; Girshick, 2015) through engaging a new regularization to the loss function. This regularizer restricts the search domain of RPN to the smokestack exit. In other words, it minimizes the distance between the proposed bounding boxes and the desired smokestack exit, called smokestack exit loss (L_S). The proposed method is also

computationally economical because it generates only a limited number of anchor boxes swarmed across the desired smokestack exit. Consequently, the main contributions of this research can be summarized as follows:

- We developed a plume rise vision-based measurement technique to improve plume-rise parameterizations, increasing air-quality models' accuracy. Furthermore, the industry might use this real-time visual system to track and survey the released pollutants online and obtain information to update related criteria, such as live Air-Quality Health Index (AQHI) updates.
- Introducing Deep Plume Rise Network (DPRNet), a deep learning method for plume rise measurements, by incorporating plume cloud recognition and image processing-based measurements. We have provided a reproducible algorithm to accurately recognize plume clouds from RGB images.
- To the best of our knowledge, this study automated the estimation of the plume clouds' neutral buoyancy coordinates for the first time, which is of the essence in environmental studies. This online information can help update related criteria, such as the live AQHI.
- A pixel-level recognition dataset, Deep Plume Rise Dataset (DPRD), containing (1) 2500 fine segments of plume clouds, (2) the upper and lower boundaries of plume clouds, (3) the image coordinates of smokestack exit, (4) the centerlines and NBP image coordinates of plume clouds, is presented. The DPRD dataset includes one class, namely the plume cloud. Widely used DCNN-based smoke recognition methods are employed to evaluate our dataset. Furthermore, this newly generated dataset was used for plume rise measurements.

1.7 Deep learning in smoke identification

To find the smoke within an image or a video frame, a rough location of smoke is identified using bounding boxes called smoke border detection, or pixels are identified and classified in detail using smoke recognition. Traditional smoke recognition methods mostly use manually obtained features, such as colour (Garcia-Lamont et al., 2018). Zhao (2015) proposed a colour-based way to use the Kalman filter to update the video frames' backgrounds and detect the moving pixels. Afterwards, rough set theory was used to represent the roughness of smoke-colour distribution. Finally, they initially recognized the small smoke regions using the gray wave colour feature and merged them to be recognized as candidate smoke regions. Moreover, the other manually obtained features such as shape and motion (Filonenko, Hernández, and Jo, 2017) were recently used for the purpose of smoke recognition. Yuan et al. (2015) introduced a real-time smoke border detection method using Haar-like features from original RGB images. To keep the consistency of identified regions, they used a staircase searching technique within the whole area. Last, dynamic analyses are performed to grab the candidate smoke regions completely. These manual features result in poor recognition due to significant variations of smoke-included pixels, which causes difficulties in defining or choosing the features (Yuan et al., 2019). Some other smoke recognition methods use dynamic features extracted from video frames (Yuan, Liu, and Zhang, 2019; Zhao, 2015; Filonenko, Hernández, and Jo, 2017). However, in some cases, these features are not stable, like cloudy conditions (Tian et al., 2017). Figure 1.7 shows some results from Yuan, Liu, and Zhang (2019), in

which their learning-based fuzzy smoke recognition method is compared to Otsu's recognition method (Otsu, 1979). Recently, deep learning-based methods have been utilized in border detection and object recognition, resulting in significant outcomes such as handling occlusion, i.e., when objects are merged and often wrongly tracked. Also, they performed well in clutter, when there are many objects and often challenging to focus on a specific one, as well as low-resolution problems (Jiang et al., 2018; Zhao et al., 2019a). Convolutional neural network (CNN) is the most popular network used in these deep learning studies (Simonyan and Zisserman, 2014; He et al., 2016).

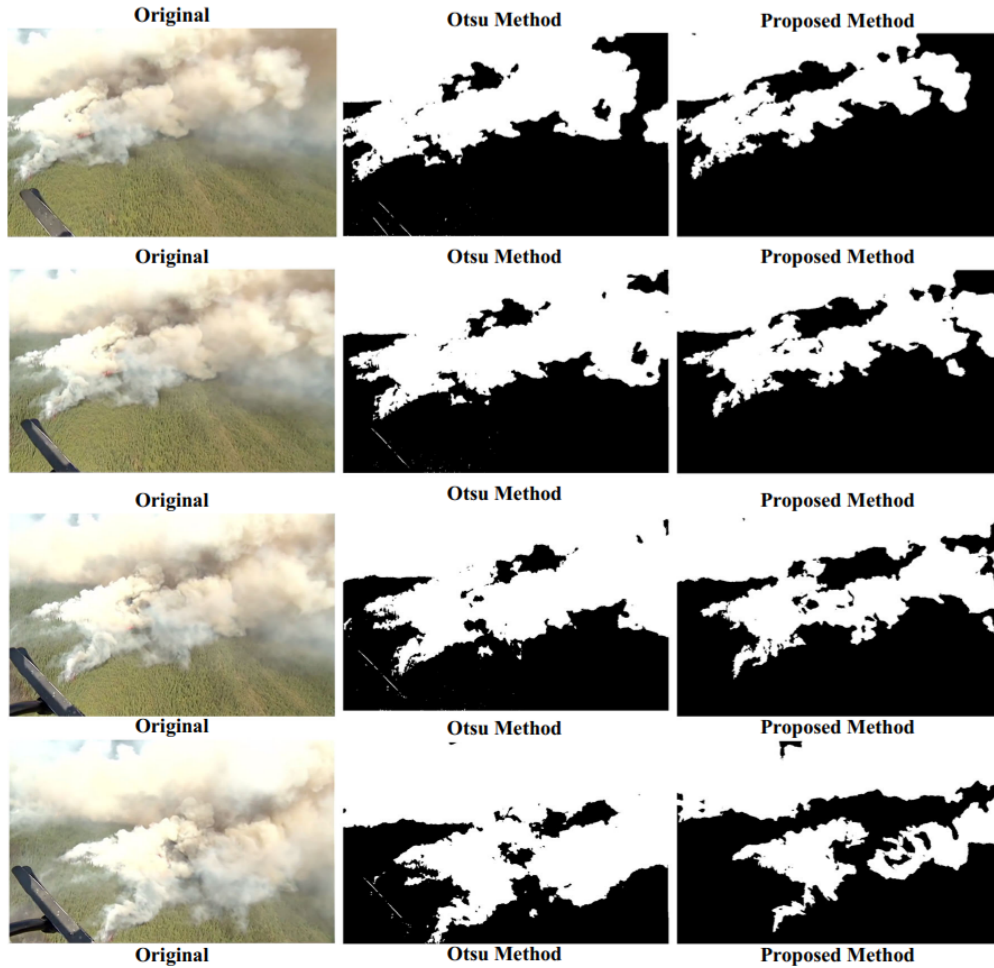


FIGURE 1.7: Results of smoke recognition using learning-based fuzzy method considering colour features (Reproduced from Yuan, Liu, and Zhang (2019)).

Fully Convolutional Network (FCN) was the first method proposed for accurate pixel-level recognition (Long, Shelhamer, and Darrell, 2015). FCN-based recognition, which is compatible with any image size, contains a set of convolutional layers and non-linear activation functions. The activation functions in neural networks decide how much of the input signal needs to be transmitted to the next layer. Convolution technique, which refers to the mathematical combination of a filter (kernel) and an input image, brings computational simplicity and effectiveness (Yuan et al., 2019). So far, using FCNs has led to the

highest average precision results among its competitors in object recognition (Ronneberger, Fischer, and Brox, 2015; Dai et al., 2016). Detailed literature reviews have recently been done on this subject, which concluded that the method had achieved great success in object recognition (Yu et al., 2018c; Yunchao and Yao, 2016).

In 2017, Badrinarayanan, Kendall, and Cipolla (2017) introduced SegNet, a deep convolutional encoder-decoder network. In this study, they added batch normalization after each convolutional layer. Furthermore, compared with FCNs in Russakovsky et al. (2015) with only upsampling operations in the decoding stage, a series of convolution and upsampling operations exist, and a consistent one-to-one match exists between the encoding and decoding stages.

U-Net, which has been mentioned in Ronneberger, Fischer, and Brox (2015), is a combination of the above two methods. However, the most critical difference between U-Net and the methods discussed above is that U-Net connects the feature maps of an encoder to its corresponding decoder and concatenates them by copying and cropping operations.

Zhang et al. (2016) proposed a Text-Block FCN using skip structures. In the Zhang et al. (2016) study, the system contains upsampling of the outputs of each convolutional stage to the same size and the final feature map is achieved by adding all the resized results. Bu et al. (2019) proposed a similar structure to skip architecture in a two-stream FCN. Luo et al. (2017) offered a 4×5 grid CNN with many skip operations to combine local and global information. Hou et al. (2017) suggested skipping structures with short connections to obtain multi-scale information.

ResNet, mentioned in He et al. (2016), constructs the basis of other methods using FCN. Jain et al. (Jain, Xiong, and Grauman, 2017) proposed a two-stream FCN, ResNet-101, in combination with dilated convolutions for the fusion of motion and appearance. Lin et al. (2017) and Peng et al. (2017) used the residual structures of ResNet to identify the foreground boundaries. Zhao et al. (2017) offered a pyramid pooling module to pool apart faint objects, and Ding et al. (2018) used novel context contrasted local features for this purpose. Also, Ding et al. (2018) utilized LSTM (Hochreiter and Schmidhuber, 1997) to convolute feature maps from two near scales.

Some related works have combined FCNs with a graphic model to obtain better performance. A series of networks, Deeplab, is a well-known method in this category. Deeplab version #1 (Chen et al., 2014) uses dilated convolutions and adds a fully connected Conditional Random Field (CRF) at the end of FCNs to improve segmentation results. Deeplab version #2 (Chen et al., 2017a) uses the Atrous Spatial Pyramid Pooling (ASPP) and ResNet101. Deeplab version #3 (Chen et al., 2017b) skips the fully connected CRF and improves ASPP and multiple grids. Deeplab version #3+ (Chen et al., 2018) uses the same encoder as version #3 and adds a decoding network with minor changes. DenseNet, proposed in Huang et al. (2017), was used by Yang et al. (2018), improving ASPP to dense ASPP.

Furthermore, the FCN is utilized for edge detection and skeleton extraction. HED (Xie and Tu, 2015) fuses the outputs of five different blocks of the VGG16 network to achieve multi-scale results. M2FCN (Shen et al., 2017b) uses several HED networks from Xie and Tu (2015) and connects each HED's original input and output to the next stage to obtain information from all scales. Shen et al. (2017a) proposed two main steps based on the HED for skeleton extraction: Loc-SSO and ScalePred-SSO.

Recently, some studies have concentrated on a two-path encoder-decoder FCN featured by skip structures for smoke recognition. DFN (Yu et al., 2018b) uses a traditional method to extract edge information to recognize the boundaries. BiSeNet (Yu et al., 2018a) proposes a two-path structure as in Yuan et al. (2019). However, the latter method can directly operate the segmentation without eight times upsampling, which is particularly important for fuzzy smoke objects (Yuan et al., 2019).

The smoke recognition studies mostly use colour characteristics of smoke (Yuan, Liu, and Zhang, 2019; Filonenko, Hernández, and Jo, 2017). Most of the time, rough sets and colour features (Zhao, 2015) come together for smoke recognition, such as Gaussian Mixture Models (GMMs). Zhao (2015) employed Kalman filters to distinguish objects with similar colours to smoke and separated smoke by the roughness distribution of smoke colours. Tian et al. (2017) offered a method based on the atmospheric scattering model. At first, there are two dictionaries for training smoke and background images. Then, an input image is classified into quasi-smoke and quasi-background components. At the end of the procedure, GMMs and image matting techniques were utilized to separate smoke from its backgrounds.

1.8 RCNNs' evolution

Mask R-CNN is a region-based CNN family member proposed in He et al. (2017). It represented the highest accuracy regarding the COCO dataset segmentation challenge, and afterwards, it is used widely in different recognition tasks (Shu et al., 2020; Lin et al., 2020; Ganesh et al., 2019). Firstly, RCNN was proposed in Albawi, Mohammed, and Al-Zawi (2017), in which computer vision techniques generated region proposals. Then, in Girshick (2015), Fast RCNN was introduced with a convolutional network before Region Proposals to increase the computational speed. In 2017, Faster RCNN continued this evolution and represented a regional proposal network to propose Regions of Interest (ROIs) (Chen and Gupta, 2017). Finally, as an extension of Faster RCNN, Mask R-CNN added a convolutional network for pixel-level segmentation of the detected objects. Mask R-CNN is a relatively simple model and can easily be generalized to similar tasks (He et al., 2017). Therefore, Mask R-CNN can create pixel-level masks for the objects besides object localization and classification tasks. Figure 1.8 shows the evolution summary for Mask RCNN (Albawi, Mohammed, and Al-Zawi, 2017; Girshick, 2015; Chen and Gupta, 2017; He et al., 2017).

1.9 Goals and objectives

The main objectives of our work can be stated in four different categories as follows:

- **Data collection:** The most significant challenge in using deep learning for plume cloud border detection and recognition is adequate labelled (annotated) images for the training stage. The labelled data can be generated by manually separating smoke pixels from their background in an input image. However, it is time-consuming and challenging as the smoke has fuzzy boundaries and translucent properties. An image dataset consists of digital images used for training, validating, and testing machine learning algorithm performance, especially in computer vision. To our knowledge, no image dataset exists for plume cloud border detection and recognition. Therefore,

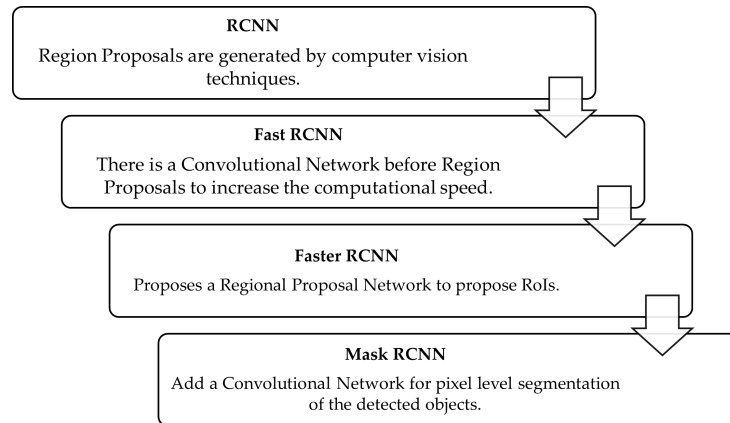


FIGURE 1.8: Summary of evolution for Mask RCNN. RoI refers to the Region of Interest in the image.

creating an image dataset for plume cloud border detection and recognition for research and industry purposes is valuable. This pixel-level recognition dataset should contain 1) fine segments of plume clouds, 2) The upper and lower boundaries of plume clouds, 3) The image coordinates of the smokestack exit, and 4) The centerlines and NBP image coordinates of plume clouds.

- **Deep model development:** Our goal is to develop a deep learning-based model to automatically identify the plume clouds in real time through captured RGB images or recorded video frames. We aim to develop a deep learning method for plume rise measurements by incorporating plume cloud recognition and image processing-based measurements. The provided reproducible algorithm should be able to recognize plume clouds from RGB images accurately. Recent studies in different fields such as computer vision, natural language processing, healthcare, autonomous vehicles, and finance have proven that developed deep models surpass traditional shallow solutions with considerable advantages.
- **Geometric transformations:** All parameters in the image scale must be transformed in real-world measurements to be comparable to the results of other methods. These transformations help us to determine the real-world plume rise as a function of distance downwind of the smokestack. The Geometric Transformation plays an important role in transferring the plume rise characteristics localized in a 2D image predicted by DPRNet to their corresponding positions in 3D world coordinates. Several studies have explored using geometric transformations to migrate from image scales to real-world scales. Some of these studies are photogrammetry, structure from motion, augmented reality, remote sensing, and robot vision and localization.
- **Briggs parameterizations validation:** The Briggs equations need to be evaluated against the image-based measurements to validate the results of Briggs estimation to our suggested visual benchmark measurements. These comparisons will be used to improve the plume rise parameterizations and help better understand a plume cloud's physics.

1.10 Research impact

After the gap of about 30 years since the plume rise parameterization equations were developed and previously tested to estimate the plume rise, as well as significant advancements in modern observation techniques, the results of this project will involve environmental monitoring techniques and improve the accuracy of air-quality models. Successful development of an automated imaging system to measure plume rise would be of direct interest to government environmental agencies such as Environment and Climate Change Canada (ECCC) and the Environmental Protection Agency (EPA), as well as industries looking to confirm or develop emissions reporting and environmental assessments.

1.11 Thesis overview

This thesis comprises eight chapters, each addressing a specific aspect of the research. The chapters are organized as follows:

1. **Introduction:** In this chapter, the research topic is introduced, and the background context and potential contributions to the field are highlighted. The literature related to the research topic is reviewed, and research objectives and impact are outlined.
2. **Background:** This chapter explains the theoretical background, information, and key concepts that form the foundation of our methodology in this research.
3. **Plume imaging** This chapter describes the study site, data collection method, data characterization, and techniques employed to create our benchmark dataset, DPRD.
4. **DPRNet Pipeline:** In this chapter, we present the DPRNet structure, a visual data-driven system developed in the framework of DCNNs, to recognize the plume cloud for plume rise measurement purposes.
5. **DPRNet Comparison Results:** This chapter assesses the performance of our proposed method, DPRNet, compared to the existing smoke recognition methods using different validation metrics.
6. **Geometric Transformation and Validation:** In this chapter, we perform calculations to migrate from image-scale measurements to real-world measurements. We also evaluate the accuracy of the imaging tool to give actual measurements by doing a validation task based on our proposed framework.
7. **Investigation of Briggs' Equations:** This chapter assesses the performance of Briggs' equations by comparing the parameterized plume rise results to the DPRNet image-based framework during a period.
8. **Conclusion and Future Work:** The research's conclusions, findings, and future studies are expressed in this chapter.

2 Background

2.1 Plume rise theory

The plume rise calculation is a non-straightforward task used to predict the dispersion of harmful effluents (Briggs, 1982). Plume rise is affected by two phenomena: buoyancy and momentum. Although some gases are lighter than air at the same temperature, mainly the plume clouds are hotter than the ambient air, meaning they are less dense and, therefore, buoyant. Thus, they rise because they are less dense than the surrounding air. Plumes have a vertical velocity and momentum when they exit the smokestack, which also causes them to rise. Plumes can also fall due to the gravitational force when cold and dense and when some surrounding obstacles cause them to move downwind (De Visscher, 2013).

2.1.1 Plume behaviors

Turbulent diffusion is the process by which particles are dispersed in a fluid due to eddies and vortices, and as a result, they are transported and distributed throughout the fluid. If we consider constant aerodynamic effects (effects of the smokestack and surrounding buildings), there are five main types of diffusion situations concerning plume rise (Briggs, 1982),

- Fanning
- Fumigation
- Coning
- Looping
- Lofting

At night, the ground radiates heat, and the air near the ground becomes cool, and consequently, a temperature inversion happens. In this situation, most plumes rise through the formed stable layer and lose their buoyancy, which is called plume fanning behaviour. The ground becomes warm during the daytime, and the convective eddies mix the plume downward. This behaviour is called fumigation. After convective eddies dissipate a temperature inversion, or when the weather is windy and cloudy, the atmospheric condition becomes neutrally stable. In this condition, the plume smoothly rises, called coning behaviour. If the ground becomes much warmer, the consequent eddies will be more intense, and it may cause a considerable twist and change its shape to a looping manner. As the day ends, the ground cools

again, and we can expect an inversion again. However, this inversion may not be strong enough to absorb the plume. Therefore, the plume rises above the inversion region, which is called lofting behaviour. Figure 2.1 represents these behaviours.

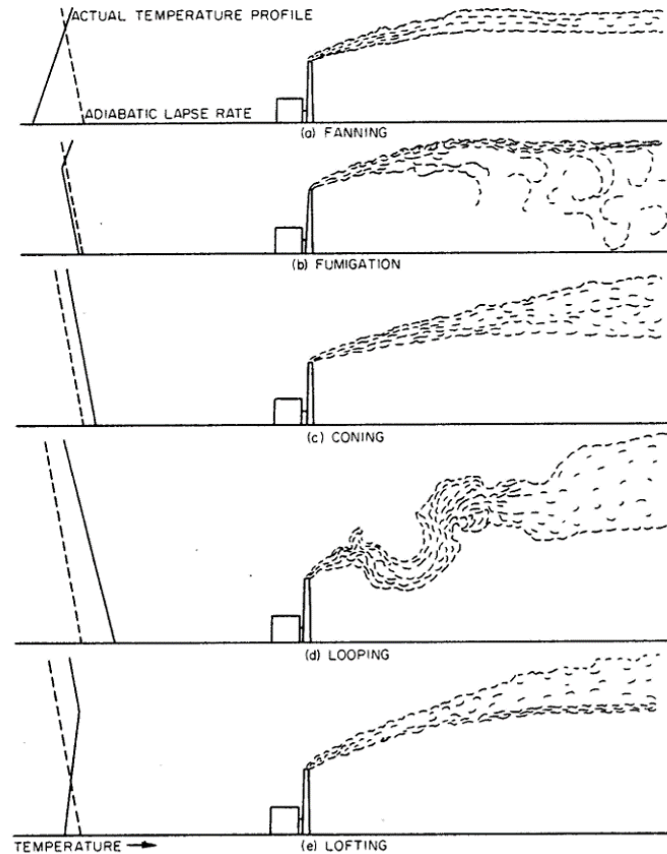


FIGURE 2.1: Plume temperature profiles and behaviors (Reproduced from Briggs (1982)). The solid and dashed lines represent the real temperature profile and DALR with altitude, respectively. DALR is the rate at which the temperature of a parcel of dry air falls with altitude, which is 5.5°F per 1000 ft or 9.8°C per km in the Earth's atmosphere.

2.1.2 Plume rise equations

There are many plume rise models with various prediction results, which can be categorized into four model types as follows (De Visscher, 2013),

- Empirical models
- Similarity-based models
- One-dimensional models
- CFD-based models

Purely empirical models are primarily inaccurate and cannot be called reliable (De Visscher, 2013). The second group is models based on similarity theory and dimensional analysis, which are the most widely used. Some other models are based on one-dimensional mass, heat, and momentum balances that require numerical integration. The last group is the models, which are based on CFD and have many computational complexities. These models are mostly used for research and academic purposes. We look at the most commonly used models here, which are based on dimensional analysis. Six variables are considered in the plume rise description (Briggs, 1965):

$$1. \text{ Momentum flux parameter : } F_m = \left[\frac{\rho_s}{\rho} \right] r_s^2 w_s^2 \text{ (m}^4 \text{ s}^{-2}\text{)}, \quad (2.1)$$

$$2. \text{ Buoyancy flux parameter : } F_b = \left[1 - \frac{\rho_s}{\rho} \right] g r_s^2 w_s^2 \text{ (m}^4 \text{ s}^{-3}\text{)}, \quad (2.2)$$

$$3. \text{ Stability Parameter : } s = \frac{g}{T} \frac{d\theta}{dz} \text{ (s}^{-2}\text{)}, \quad (2.3)$$

$$4. \text{ Mean horizontal wind speed : } \bar{u} \text{ (m s}^{-1}\text{)}, \quad (2.4)$$

$$5. \text{ Time of travel : } t \text{ (s)}, \quad (2.5)$$

$$6. \text{ Plume rise : } \Delta z(\text{transitional}); \Delta h(\text{final}) \text{ (m)}, \quad (2.6)$$

where ρ_s (kg m⁻³) is smokestack gas density, ρ (kg m⁻³) is atmospheric air density, r_s (m) is smokestack radius, w_s (m s⁻¹) is vertical velocity of the smokestack gas, g (m s⁻²) is acceleration due to gravity, T (K) is the atmosphere temperature, θ (K) is potential temperature, and z (m) is height. Plume rise and plume rise distance are denoted respectively by Δh and x_f in this section.

Since only the ratio of mass (buoyancy) is used, there are only two dimensions out of six to consider: distance and time. Based on the Buckingham π theorem, there exist four dimensionless groups as follows,

$$\pi_4 = f(\pi_1, \pi_2, \pi_3) \quad (2.7)$$

To solve the equations, some simplifications are applied as follows:

- Only one of F_m or F_b is considered in the model.
- $\frac{F_m}{\bar{u}}$ and $\frac{F_b}{\bar{u}}$ are more relevant than F_m and F_b .
- It is assumed that stability does not affect plume rise.

According to these assumptions and consequently eliminating the dependency on three variables, Buckingham π theorem can be rewritten as follows,

$$\pi_1 = \text{const} \quad (2.8)$$

For the transitional plume rise of buoyancy-dominated plumes, we eliminate F_m and s and combine F_b and $\bar{u}$ ($\frac{F_b}{\bar{u}}$). Based on Equation 2.8, we can generate a dimensionless number using the remaining variables as below,

$$\pi_1 = \frac{F_b t^2}{\bar{u}(\Delta z)^3}. \quad (2.9)$$

Hence,

$$\Delta z = \pi_1^{-1/3} \left[\frac{F_b t^2}{\bar{u}} \right]^{1/3}. \quad (2.10)$$

The transitional plume rise measurement data gives $\pi_1^{-1/3} = 1.6$ (De Visscher, 2013). If we consider $t = x/\bar{u}$, the following equation can be obtained for buoyancy-dominated plumes,

$$\Delta z = 1.6 \left[\frac{F_b x^2}{\bar{u}^3} \right]^{1/3}. \quad (2.11)$$

Similarly, for the transitional plume rise of momentum-dominated plumes, we can write the following equation,

$$\Delta z = \pi_2^{-1/3} \left[\frac{F_m t}{\bar{u}} \right]^{1/3}. \quad (2.12)$$

The transitional plume rise measurement data gives $\pi_2^{-1/3} = 2$ (De Visscher, 2013). If we consider $t = x/\bar{u}$, the following equation can be obtained for momentum-dominated plumes,

$$\Delta z = 2 \left[\frac{F_m x}{\bar{u}^2} \right]^{1/3}. \quad (2.13)$$

To calculate the final plume rise in a stable atmosphere, we substitute the stability parameter s for t . Therefore, we have,

$$\Delta h = \pi_1^{-1/3} \left[\frac{F_b}{\bar{u}s} \right]^{1/3}, \quad (2.14)$$

$$\Delta h = \pi_2^{-1/3} \left[\frac{F_m}{\bar{u}s^{1/2}} \right]^{1/3}, \quad (2.15)$$

And the final plume rise measurement data gives $\pi_1^{-1/3} = 2.6$ and $\pi_2^{-1/3} = 1.5$ (De Visscher, 2013).

Considering this approach, if the wind speed is near zero, the plume rise tends to infinity, which is unrealistic. In this situation, we use F_b and F_m variables, not $\frac{F_b}{\bar{u}}$ and $\frac{F_m}{\bar{u}}$. Also, for the final plume rise, we have the variables F_b , F_m , s and Δh . Therefore, in this case, we have,

$$\pi_1 = \frac{F_b}{s^{3/2}(\Delta h)^4}, \quad (2.16)$$

$$\pi_2 = \frac{F_m}{s(\Delta h)^4}. \quad (2.17)$$

Consequently, the final plume rise heights are as follows,

$$\Delta h = \pi_1^{-1/4} \left[\frac{F_b^2}{s^3} \right]^{1/8}, \quad (2.18)$$

$$\Delta h = \pi_2^{-1/4} \left[\frac{F_m}{s} \right]^{1/4}, \quad (2.19)$$

and the final plume rise measurement data gives $\pi_1^{-1/4} = 5.3$ and $\pi_2^{-1/4} = 2.4$ (De Visscher, 2013). Figure 2.2 demonstrates the plume measures in transitional and final situations on a sample plume cloud image.

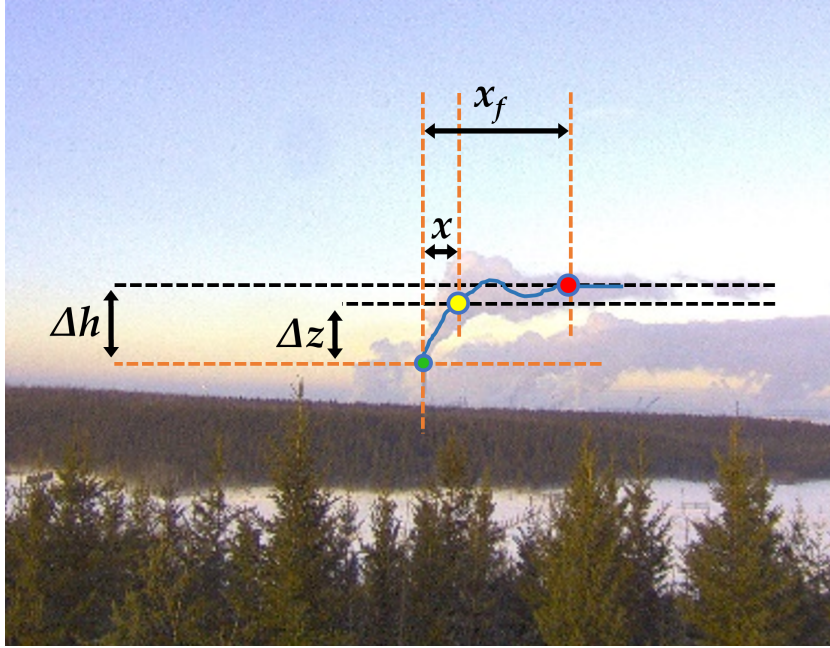


FIGURE 2.2: The variables used to describe the plume. Δz and x represent the plume rise and plume rise distance in the transition phase. Also, Δh and x_f show the plume rise and plume rise distance in the final phase. The green point displays the smokestack exit, and the yellow and red points show the transition and final phase of the plume, respectively. The blue curve is the plume centreline.

According to the Equations 2.18 & 2.19, in the neutral atmospheric condition ($s = 0$), this approach gives an infinite final plume rise. In practice, plume rise varies so slowly at a far distance from the smokestack, and it can be assumed that plume rise is constant at long distances. Therefore, the dimensionless numbers can be defined as follows,

$$\pi_1 = \frac{F_b}{\bar{u}^3 \Delta h}, \quad (2.20)$$

$$\pi_2 = \frac{F_m}{\bar{u}^2 (\Delta h)^2}, \quad (2.21)$$

and consequently, the final plume rise heights are given by,

$$\Delta h = \pi_1^{-1} \frac{F_b}{\bar{u}^3}, \quad (2.22)$$

$$\Delta h = \pi_2^{-1/2} \frac{F_m}{\bar{u}^2}, \quad (2.23)$$

and the final plume rise measurement data gives $\pi_1^{-1} = 400$ and $\pi_2^{-1/2} = 3$ (De Visscher, 2013).

Although this model and the other approaches lead to relatively comprehensive analytical models, there is always a limitation in plume rising in practice (De Visscher, 2013). In other words, these models simplify the conditions, such as atmospheric states and terrain effects, to estimate the plume rise. The atmosphere is highly dynamic and affected by various factors in the real world. Furthermore, uncertainties in different input parameters, such as wind speeds, can directly affect these models' plume rise predictions.

In 1975, Briggs proposed an equation for the maximum horizontal, downwind distance of the plume rise as (Briggs, 1975),

$$x_f = d_1 F_b^{5/8} \quad \text{for } F_b \leq 55 \text{ m}^4 \text{s}^{-3}, \quad (2.24)$$

$$x_f = d_2 F_b^{2/5} \quad \text{for } F_b \geq 55 \text{ m}^4 \text{s}^{-3}, \quad (2.25)$$

where x_f represents the distance to final plume rise in meters, $d_1 = 49 \text{ s}^{15/8} \text{ m}^{-3/2}$ and $d_2 = 119 \text{ s}^{9/5} \text{ m}^{-7/5}$. Based on De Visscher (2013), these equations do not follow the dimensional analysis and the resulting relationships between properties are not fundamental. However, they provide a practical guideline for plume rise calculations.

Therefore, the final plume rise ($x = x_f$) for buoyancy-dominated plumes (Equation 2.11) can be calculated as,

$$\Delta h = \frac{c_1 F_b^{3/4}}{\bar{u}}, c_1 = 21.4 \text{ s}^{1.25} \text{ m}^{-1} \quad F_b \leq 55 \text{ m}^4 \text{s}^{-3}, \quad (2.26)$$

$$\Delta h = \frac{c_2 F_b^{3/5}}{\bar{u}}, c_2 = 38.7 \text{ s}^{0.8} \text{ m}^{-0.4} \quad F_b \geq 55 \text{ m}^4 \text{s}^{-3}. \quad (2.27)$$

We can see that except when $\bar{u} \geq 6 \text{ ms}^{-1}$, the calculated plume rise here is significantly smaller than the one from Equation 2.22.

Briggs (1975) combined both momentum and buoyancy plumes rise as,

$$\Delta z = \left(\frac{3 F_m x}{0.6^2 \bar{u}^2} + \frac{3 F_b x^2}{2 \times 0.6^2 \bar{u}^3} \right)^{1/3}, \quad (2.28)$$

where the factor 0.6 is an entrainment rate (the mean rate of increase of plume cloud in the wind direction). Equations 2.11 and 2.13 are limiting cases of Equation 2.28, derived for buoyancy-dominated and momentum-dominated plumes, respectively.

In 1988, Weil focused more on plume rise distance dependency from the source, which was ignored in most empirical works. His proposed model was based on mass, momentum, and energy equations through a plume cloud cross-section. Weil outlined that his plume rise model focuses on major fundamental aspects of the problem and cannot be used in problems such as the plume rise of multiple smokestacks. He introduced an equation for a transitional plume rise in stable and windy atmospheric conditions (an extension of Equation 2.14) as (Weil, 1988),

$$\Delta z = 2.66 \left(\frac{F_b}{\bar{u} s} \right)^{1/3} \left[\frac{0.7 s^{1/2} F_m}{F_b} \sin \left(\frac{0.7 s^{1/2} x}{\bar{u}} \right) + 1 - \cos \left(\frac{0.7 s^{1/2} x}{\bar{u}} \right) \right]^{1/3}. \quad (2.29)$$

The final plume rise distance is calculated as follows,

$$x_f = \frac{\bar{u}}{0.7 s^{1/2}} \arctan \left(\frac{F_b}{0.7 s^{1/2} F_m} \right), \quad (2.30)$$

and the final plume rise can be obtained by equation 2.14 with $\pi_1^{-1/3} = 2.66$ (De Visscher, 2013).

If we evaluate the wind speed at the local plume height and not the source height (for example, in Equation 2.28), the calculations should be solved iteratively (Cimorelli et al., 2005; Turner and Schulze, 2007). Plume rise not only depends on source conditions but is strongly affected by the local meteorological conditions, such as wind speed and direction (Briggs, 1982; Weil, 1988).

2.2 Neural networks

2.2.1 Artificial neural networks

Artificial Neural Networks (ANN) have been broadly developed to deal with machine learning problems (Aggarwal et al., 2018). These networks are inspired by the function and structure of the human nervous system (Ker et al., 2017). Indeed, from a biological point of view, the purpose of a neural network is to extract useful information and existing patterns of input data (Gurney, 2018).

The neural network is a computational structure that attempts to mimic the brain's processing task in a computer. The brain is complex and nonlinear with a parallel information processing system (Haykin, 2009). An artificial neural network was created at the same time as the first computers, but the main problem of these networks was an inability to learn. The first network was introduced called *Perceptron*, which was capable of learning (McCulloch and Pitts, 1990).

Neurons

Neurons are the main computing units of a neural network that process the data locally. We need to model the neurons mathematically to create an artificial neural network. The biological form of neurons in the brain is shown in Figure 2.3a consists of three main components,

- **Dendrites** They transmit input signals. The ratio of the input signals is adjusted by applying the connection weights to their initial value.
- **Cell body** It processes the input signals after their summation (weighted).
- **Axon** It transmits the output signal from the cell body to the next neurons.

This computational process in neurons is mathematically represented in Figure 2.3b. In this figure, the signals are received and aggregated in the body (Σ) and processed to provide output with operational functions ($f(\Sigma)$). It can also be seen in Figure 2.3c that the extraction and learning of patterns and the features of the input data depending on the appropriate values of the connection weights (Samarasinghe, 2016).

An artificial neural network is a collection of neurons corresponding to each other by some connections. As Figure 2.4 represents, if x_i are the neuron inputs from each of the connections with w_i as weights and b as bias, the output z for the neuron is calculated as follows (Skansi, 2018),

$$z = b + \sum_i w_i x_i. \quad (2.31)$$

Equation 2.31 can be written in a more general manner. Because the bias expression can be considered as one of the weighted summation expressions, we will have,

$$z = b + \sum_i w_i x_i = w_0 x_0 + w_1 x_1 + \dots = \sum_i w_i x_i, \quad (2.32)$$

where $w_0 = b$ & $x_0 = 1$, and in this way, the bias parameter is also considered as one of the weights. This process is called bias absorption. In the next step, the activation function f is applied to z and the final output of the neuron is calculated,

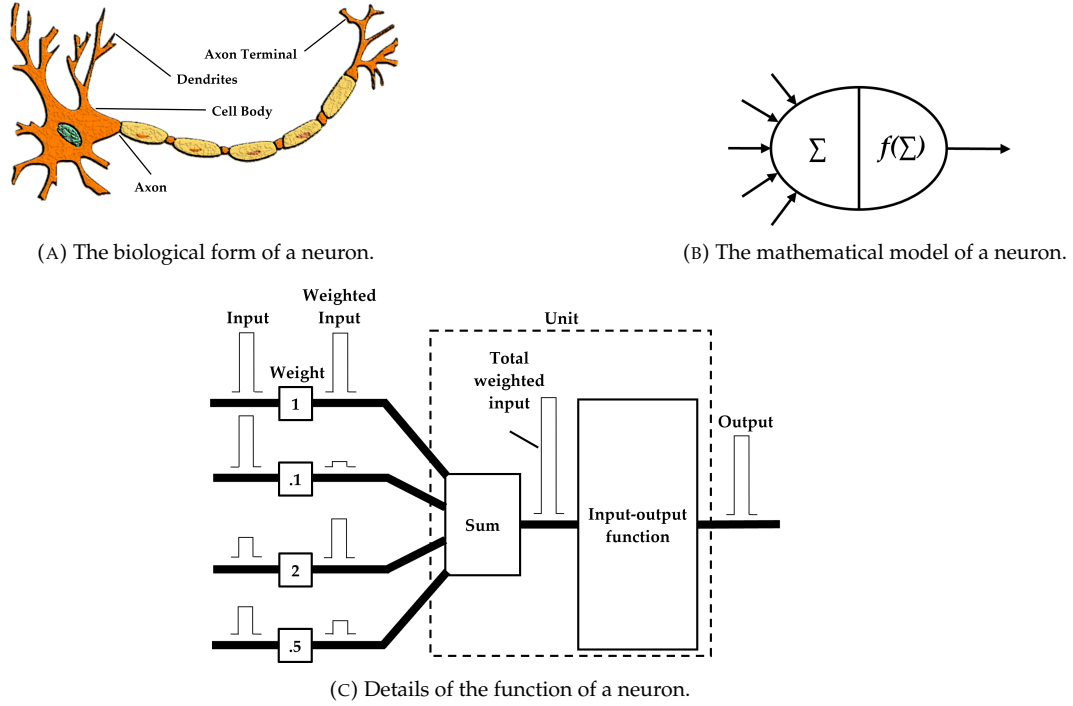


FIGURE 2.3: Neuron (Reproduced from Samarasinghe (2016)).

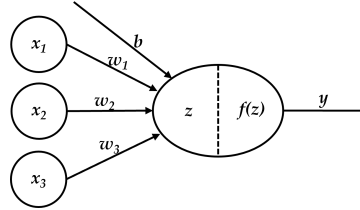


FIGURE 2.4: The mathematical model of a neuron (Reproduced from Skansi (2018)).

$$y = f(z) = f\left(\sum_i w_i x_i\right). \quad (2.33)$$

The activation function in each artificial neuron decides how much of the input signal it needs to transmit to the next level, or in other words, determines the neuron's activity level. The purpose of using these functions in networks is to add nonlinear modelling capabilities to them (Patterson and Gibson, 2017). Activation functions must also be differentiable. Because, as we will see, this feature is essential for training artificial neural networks. One of the first and most common functions used for activation was the sigmoid function shown in Figure 2.5 (Wang et al., 2020).

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2.34)$$

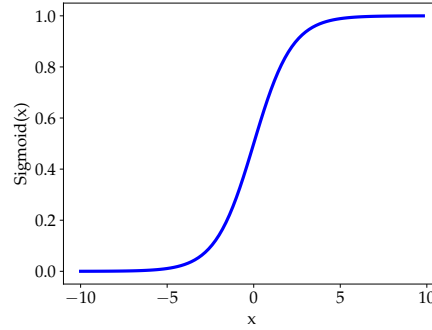


FIGURE 2.5: Sigmoid activation function.

Neural network layers

After building the first neural networks mentioned above, the researchers realized that they could improve the computing power of these networks by adding successive layers to them (Ivakhnenko and Lapa, 1965). This means a network of different layers brings a high computational ability to process information. The more neural layers there are, the deeper the processing. Accordingly, the term *deep network* is also mentioned for the first time in Ivakhnenko's research (Ivakhnenko, Lapa, and Lapa, 1967). However, in terms of the number of layers, networks are divided into three categories: single-layer, multi-layer (shallow) and deep (Figure 2.6). In other words, a network that has no hidden layer is called a single layer and a network that has one or more hidden layers is called a shallow or deep network, respectively. The main reason for separating multi-layer networks from deep networks is the difference in their development history. Deep networks were seriously and operationally introduced two decades after the emergence of multi-layer networks. Therefore, by the middle of the first decade of 2000, a multi-layer network meant a network with only one hidden layer (Kim, 2017).

2.2.2 Single-layer neural network training

Network training is required first to evaluate neural network performance. Network training means finding the optimal weights for the network so that the network output has the smallest difference from the expected answers (targets) and can be generalized to new data. Therefore, the methods used to train neural networks are optimization methods. Nowadays, there are various methods for training neural networks, and in this section, several of them have been introduced and explained.

In a single-layer neural network, input data x is given to network φ and by applying the weight matrix w , y is estimated:

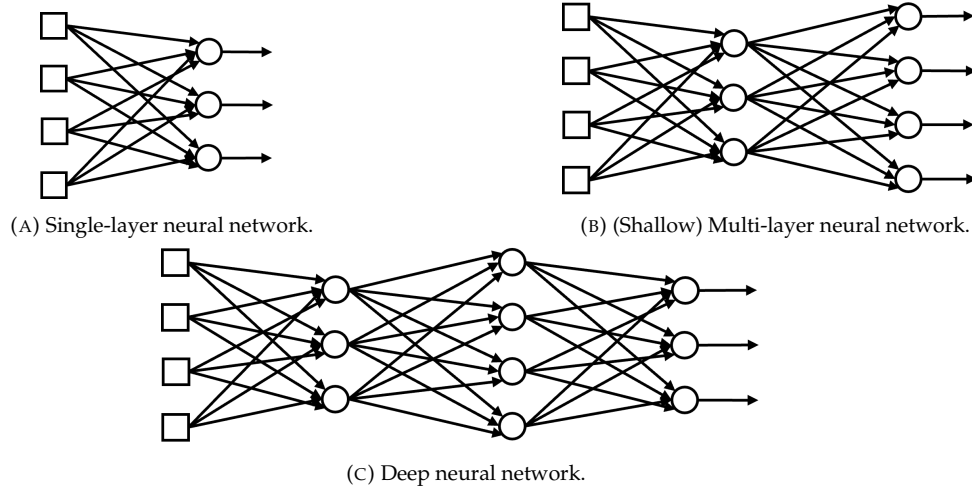


FIGURE 2.6: Types of neural networks regarding the number of layers.

$$y = \varphi(w, x). \quad (2.35)$$

Now with $\hat{y}$ which is the correct network output (target), one can calculate the network estimation error (e),

$$e = \hat{y} - y. \quad (2.36)$$

It should be noted, however, that scholars usually use an error function to estimate the error, which better represents the error and can help the network for better training. Therefore, for instance, using the Mean Squared Error (MSE) function, the above relation can be rewritten as follows,

$$E(w) = \frac{1}{2} \|\varphi(w, x) - \hat{y}\|^2. \quad (2.37)$$

Now, we have to change the network weights in a few iterations using appropriate optimization methods to minimize the value of the error function. Figure 2.7 shows the general training process of a neural network (Kim, 2017).

There are various ideas for updating neural network weights, but the idea of calculating and using gradients has become the most common method. The gradient descent method and some other methods derived from it, which are all based on calculating and using gradients, will be mentioned in this section.

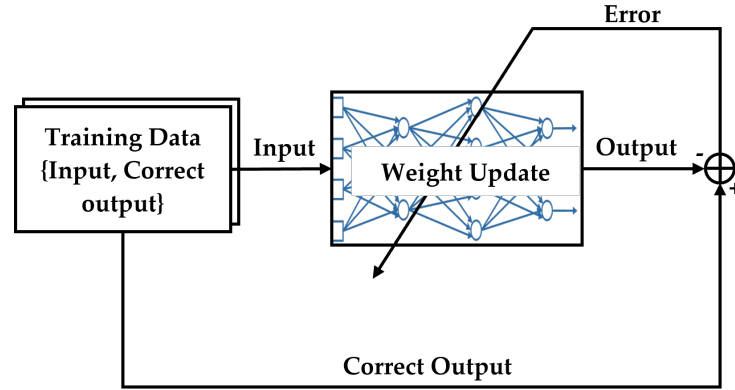


FIGURE 2.7: Neural network training process (Reproduced from Kim (2017)).

Gradient descent optimization algorithm

The idea for the gradient descent method was introduced in 1847 by Cauchy (Cauchy et al., 1847), and in 1944, Haskell Curry developed this method for nonlinear optimization problems (Curry, 1944). This method aims to find the minimum point of an arbitrary derivative function, which takes a step commensurate with the value of the gradient and in its negative direction in each iteration. Linnainmaa first used this method in the neural network in 1970 (Linnainmaa, 1970). Figure 2.8 shows how the gradient descent method works (Chollet, 2021).

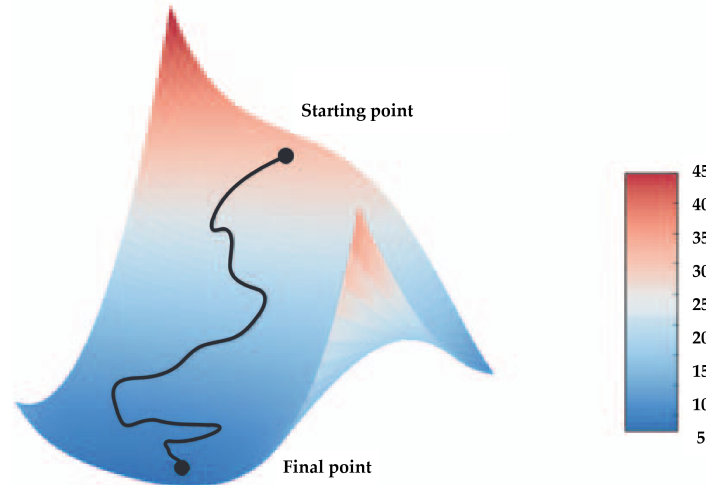


FIGURE 2.8: Performance of gradient descent method for a two-dimensional error function (Reproduced from Chollet (2021)).

Suppose $E(w)$ is a neural network cost function and $w = [w_1, w_2, \dots, w_d]^T$ is its weights vector with length of d . The gradient of $E(w)$ with respect to w is as follows (Zhang et al., 2021),

$$\nabla_w E(w) = [\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, \dots, \frac{\partial E(w)}{\partial w_d}]^T. \quad (2.38)$$

In the following, we will use $\nabla E(w)$ instead of $\nabla_w E(w)$ for brevity. Now the derivative of E in the direction of unit vector u ($\|u\| = 1$) can be defined by the following equations,

$$D_u E(w) = \lim_{h \rightarrow 0} \frac{E(w + hu) - E(w)}{h}, \quad (2.39)$$

$$D_u E(w) = \nabla E(w) \cdot u, \quad (2.40)$$

$$D_u E(w) = \|\nabla E(w)\| \cdot \|u\| \cdot \cos \theta = \|\nabla E(w)\| \cdot \cos \theta, \quad (2.41)$$

Here θ is the angle between the gradient $\nabla E(w)$ and the unit vector u . Now we have to move in the opposite direction of the gradient according to the gradient descent method, so $\theta = \pi$ and $\cos \theta = -1$. Accordingly, the general relationship of gradient descent can be defined as follows:

$$w \leftarrow w - \eta \nabla E(w). \quad (2.42)$$

$\eta > 0$ is called the learning rate in this relation. The learning rate is a hyperparameter defined and added to the relationship to expedite optimizing and updating the weights in the network (Patterson and Gibson, 2017). The idea of using a learning rate was first suggested in Jin et al. (2000). The learning rate controls how much of the update is to be applied. For example, a learning rate of 1 means applying a full update, and a learning rate of 0.1 means that only 10% of the gradient will be applied. The most commonly used values are 0.1, 0.01, and 0.001. The value of the learning rate is adjusted manually and is not updated by gradient descent, which, like the number of layers, the number of units per layer, etc., is called a hyperparameter. Figure 2.9 shows the effect of choosing a value for the learning rate (Zhang et al., 2021).

Based on the above description, the steps of the gradient descent algorithm can be defined as follows,

1. Calculate the error function for all instances in training datasets (Figure 2.10).
2. Calculate the gradient vector (∇E) according to the Equation 2.38.
3. Update network parameters based on Equation 2.42.
4. Repeat until the stop condition is met.

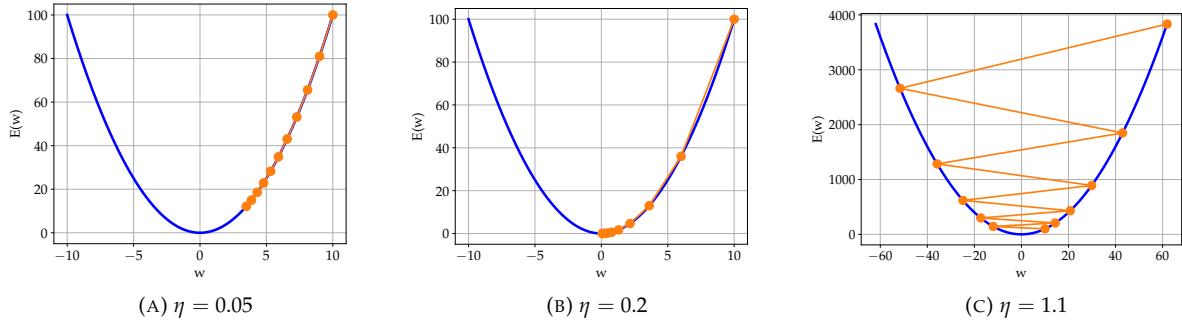


FIGURE 2.9: Performance of descending gradient for different values of learning rate.

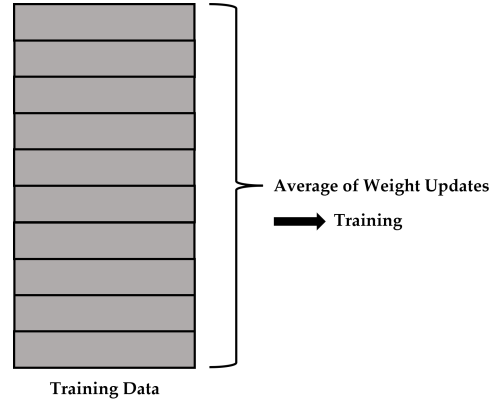


FIGURE 2.10: How to select training dataset in the gradient descent method (Reproduced from Kim (2017)).

Stochastic Gradient Descent (SGD) optimization algorithm

The stochastic gradient descent is a method derived from the gradient descent, and in this section, we will refer to their differences. In neural networks, the error function value is often the average of the value of the error on each instance in the training dataset. If $E_i(w)$ is an error function, for instance, i from a dataset with n training instances and w still represents network weights, the error function will be defined as follows (Zhang et al., 2021),

$$E(w) = \frac{1}{n} \sum_{i=1}^n E_i(w). \quad (2.43)$$

The gradient of the error function is calculated as follows,

$$\nabla E(w) = \frac{1}{n} \sum_{i=1}^n \nabla E_i(w). \quad (2.44)$$

If the gradient descent method is used, the computational cost will increase linearly by increasing n . Therefore, the computational cost will become very high by operating every iteration on datasets with a large n .

The stochastic gradient descent method was developed to reduce the computational cost for each iteration. In this method, only one random instance $i \in \{1, \dots, n\}$ is selected in every iteration, and the same instance is used to update weights. Thus, instead of Equation 2.42, stochastic gradient descent method follows the relation below,

$$w \leftarrow w - \eta \nabla E_i(w). \quad (2.45)$$

The computational cost of the stochastic gradient descent is $\frac{1}{n}$ of the computational cost of gradient descent. It should be noted that $\nabla E_i(w)$ is a good estimate of $\nabla E(w)$ on average,

$$\mathbb{E}_i \nabla E_i(w) = \frac{1}{n} \sum_{i=1}^n \nabla E_i(w) = \nabla E(w). \quad (2.46)$$

Based on the above description, the steps of the stochastic gradient descent algorithm can be defined as follows,

1. Select a random instance i (Figure 2.11).
2. Calculate the gradient vector ∇E_i .
3. Update network parameters based on Equation 2.45.
4. Repeat until the stop condition is met.

Mini-batch Stochastic Gradient Descent (mini-batch SGD)

The gradient descent method uses all the dataset instances in each iteration to calculate the gradient. For this reason, this method is also called *full-batch gradient descent*, which puts the entire dataset in one batch. On the other hand, the stochastic gradient descent method uses only one random training instance to calculate the gradient in each iteration. For this reason, it is also called *online learning*. After one or more iterations, all the data is given to the network, and a so-called training *epoch* is done.

Given the above, the mini-batch SGD suggests that a small batch of training instances (larger than one instance) is randomly selected and used to calculate the gradient in each iteration. Suppose that w_0 is

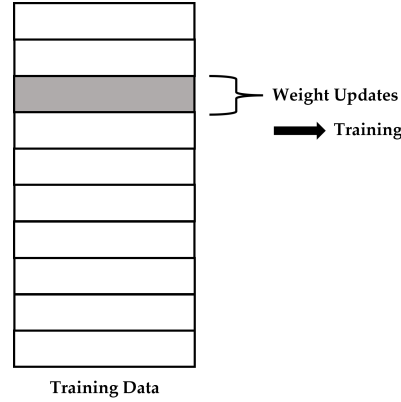


FIGURE 2.11: How to select training instances in the stochastic gradient descent method (Reproduced from Kim (2017)).

the initial value of the network weights at the time $t = 0$. The mini-batch SGD randomly selects small batches B_t from the dataset at each time step $t > 0$ and calculates the gradient g_t on it (Zhang et al., 2021),

$$g_t \leftarrow \nabla E_{B_t}(w_{t-1}) = \frac{1}{|B|} \sum_{i \in B_t} \nabla E_i(w_{t-1}). \quad (2.47)$$

In the above equation, $|B|$ is the batch size equal to the number of instances in each mini-batch. Batch size is another neural network hyperparameter and must be adjusted manually before training begins. According to Equation 2.47, the general relationship of mini-batch SGD to update weights is as follows,

$$w_t \leftarrow w_{t-1} - \eta g_t. \quad (2.48)$$

It is noteworthy that the vast majority of neural networks use the mini-batch SGD method for optimization. However, it is often called the stochastic gradient descent method or SGD due to its brevity. Based on the above explanations, the optimization steps using the mini-batch SGD algorithm can be defined as follows (Stollenga, 2016),

1. Select a random mini-batch B_t (Figure 2.12).
2. Calculate the gradient vector g_t according to Equation 2.47.
3. Update network parameters based on Equation 2.48.
4. Repeat until the stop condition is met.

It should be noted that although the mini-batch stochastic gradient descent output will be a local optimum point, optimizing a network aims to reach the optimum global point. One proposed method to

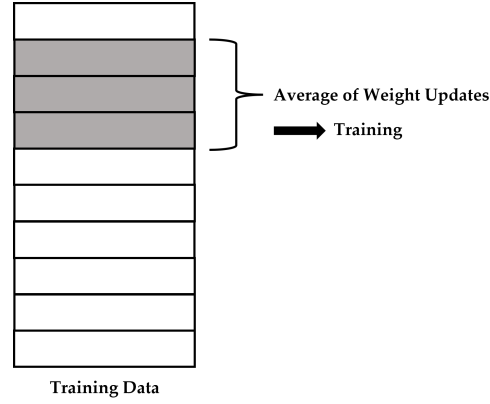


FIGURE 2.12: How to select training instances in the mini-batch stochastic gradient descent method (Reproduced from Kim (2017)).

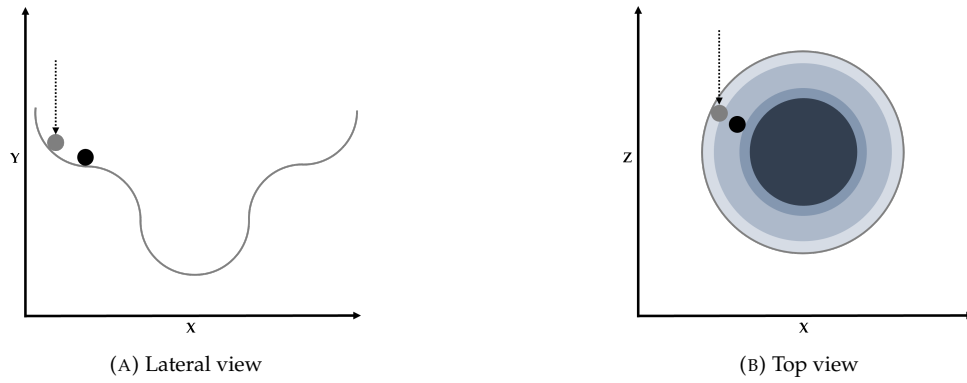


FIGURE 2.13: The role of learning rate and momentum in reducing a three-dimensional error function (Reproduced from Skansi (2018)).

prevent the algorithm from getting stuck in local optimal points is to add the *momentum* variable to the learning process (Rumelhart, Hinton, and Williams, 1985). On the other hand, momentum in neural network training, like its physical concept, prevents stopping at the local minimum points of the error function and causes it to reach the global minimum point (Skansi, 2018). Figure 2.13 shows the difference between the role of learning rate and momentum in reducing the error function.

The momentum hyperparameter sets how much of the previous weight change is maintained in the current iteration. Based on this, Equation 2.48 can be rewritten as follows (Skansi, 2018; Zhang et al., 2021),

$$w_t \leftarrow w_{t-1} - \eta g_t + \mu(|w_{t-1} - w_{t-2}|), \quad (2.49)$$

where parameter μ represents momentum and ranges between 0 and 1. Usually, the value of μ is 0.9, and it is better to choose between 0.1 and 0.99 and closer to 1 (Skansi, 2018).

Root Mean Squared Error Propagation (RMSProp) method

One of the updates of the SGD method called Root Mean Squared Error Propagation (RMSProp) (Hinton, Srivastava, and Swersky, 2012) has attracted much attention in the field of network training. The RMSProp technique has the ability to reduce the number of changes in network parameters during training. The order and amount of parameter updates are important in network optimization in deep networks. The RMSProp algorithm, with minor modifications, is an upgraded version of the AdaGrad algorithm (Duchi, Hazan, and Singer, 2011).

The optimization steps in the RMSProp algorithm can be defined as follows (Goodfellow, Bengio, and Courville, 2016),

1. Select a small random batch and its corresponding output in the dataset.
2. Calculate the gradient vector in time step t (g_t).
3. Calculate the cumulative gradient (r_t) according to the following equation (Zhang et al., 2021),

$$r_t \leftarrow \gamma r_{t-1} + (1 - \gamma) g_t \odot g_t, \quad (2.50)$$

where the hyperparameter $0 \leq \gamma < 1$ is called the reduction rate, effectively updating the cumulative gradient. The operator $\odot$ is the component product, and the initial value of r_t is considered equal to zero.

4. Update network parameters based on the following relation,

$$w_t \leftarrow w_{t-1} - \frac{\eta}{\sqrt{r_t + \epsilon}} \odot g_t, \quad (2.51)$$

where the parameter η is the learning rate, and ϵ is the numerical stability constant with a small value such as 10^{-6} .

5. Repeat until the stop condition is met.

Adam method

Adam algorithm (Kingma and Ba, 2014) is one of the adaptive learning rate methods and is, in fact, a combination of SGD methods with momentum and RMSProp method. This combination aims to increase the convergence rate by momentum and solve the problem of extreme fluctuations by RMSProp. In addition to using gradient square like the RMSProp method as the second-order moment (the phrase related to momentum), this method uses the gradient alone as the first-order moment (Goodfellow, Bengio,

and Courville, 2016; Ruder, 2016). These two moments are corrected after updating to fix bias, which is very important and effective in early iterations and is one of the advantages of this method compared to RMSProp (Aggarwal et al., 2018). It is also worth noting that Adam has relatively low sensitivity to the values of hyperparameters (Goodfellow, Bengio, and Courville, 2016).

The optimization steps in the Adam algorithm can be defined as follows (Goodfellow, Bengio, and Courville, 2016; Rumelhart, Hinton, and Williams, 1986),

1. Select a small random batch and its corresponding output in the dataset.
2. Calculate the gradient vector in time step t (g_t).
3. Update the first moment v_t (with bias),

$$v_t \leftarrow \beta_1 v_{t-1} + (1 - \beta_1) g_t, \quad (2.52)$$

where $0 \leq \beta_1 < 1$ is the exponential reduction rate and, by default, is set to 0.9 in the main reference.

4. Update the second moment r_t (with bias),

$$r_t \leftarrow \beta_2 r_{t-1} + (1 - \beta_2) g_t \odot g_t, \quad (2.53)$$

where $0 \leq \beta_2 < 1$ is the exponential reduction rate and, by default, is set to 0.999 in the main reference.

5. Correct the first-moment bias,

$$\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_1^t}. \quad (2.54)$$

6. Correct the second-moment bias,

$$\hat{r}_t \leftarrow \frac{r_t}{1 - \beta_2^t}. \quad (2.55)$$

7. Update the network parameters,

$$w_t \leftarrow w_{t-1} - \eta \frac{\hat{v}_t}{\sqrt{\hat{r}_t + \epsilon}}. \quad (2.56)$$

Learning rate η and numerical stability constants ϵ are generally considered to be 10^{-3} and 10^{-8} , respectively.

8. Repeat until the stop condition is met.

As mentioned earlier, Figure 2.14 compares the performance of the Adam algorithm with other methods in the form of a hypothetical example.

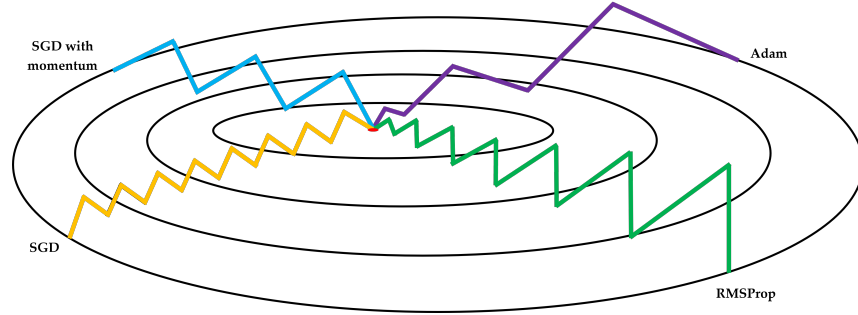


FIGURE 2.14: Performance of Adam algorithm. The curves represent a quadratic error function. Yellow, Blue, green, and purple lines show the performance of the SGD, SGD+Momentum, RMSProp, and Adam algorithms, respectively, during successive iterations of network training and reaching the minimum value for the error function (red dot). A comparison of the performances indicates that the Adam optimization algorithm has taken full advantage of both using the momentum and using the technique in the RMSProp method.

2.2.3 Multi-layer neural network training

Given the limitations of the single-layer neural network in learning patterns and solving complex problems, the researchers tried to increase the network learning ability by increasing the number of its layers and migrating to multi-layer architecture.

The error back-propagation algorithm (LeCun et al., 1989) can be implemented along with one of the conventional neural network training methods, including gradient descent and its derivatives, and can be used to train multi-layer networks.

Suppose $(x, \hat{y})$ is a hypothetical dataset that x is the input and $\hat{y}$ is the target output. Network output is also displayed as y_o . To calculate the error, the following relationship is established between the target output and the network output,

$$E = \frac{1}{2}(\hat{y} - y_o)^2, \quad (2.57)$$

where E is the error function.

In the first step, we need to calculate the gradient of the error function concerning the network output,

$$\frac{\partial E}{\partial y_o} = -(\hat{y} - y_o). \quad (2.58)$$

Now that we have the above relation, we must compute the gradient for z_o using the chain rule and, according to the following relation,

$$\frac{\partial E}{\partial z_o} = \frac{\partial E}{\partial y_o} \frac{\partial y_o}{\partial z_o} = \frac{\partial E}{\partial y_o} y_o (1 - y_o). \quad (2.59)$$

Since all operators and neural network activation functions are derivative, we can reach the hidden layer from the output gradient using the chain rule,

$$\frac{\partial E}{\partial y_h} = \sum_o \frac{\partial E}{\partial z_o} \frac{\partial z_o}{\partial y_h} = \sum_o \frac{\partial E}{\partial z_o} w_{ho}. \quad (2.60)$$

The process that went from $\frac{\partial E}{\partial y_o}$ to $\frac{\partial E}{\partial y_h}$ in the above relationships is, in fact, the main idea of the error back-propagation algorithm. In the above relation, w_{ho} is the weight of the connection between the hidden layer (h) and the last or output layer (o). Therefore, the error gradient to the weights (in this example: w_{ho}) can be calculated simply from the following relation,

$$\frac{\partial E}{\partial w_{ho}} = \frac{\partial E}{\partial z_o} \frac{\partial z_o}{\partial w_{ho}} = \frac{\partial E}{\partial z_o} y_h. \quad (2.61)$$

The resulting gradient can now be used in gradient-based training methods such as gradient descent for updating the value of w_{ho} ,

$$w_{ho} \leftarrow w_{ho} - \eta \frac{\partial E}{\partial w_{ho}}. \quad (2.62)$$

Although the gradient descent method is used in the above example, other methods, such as mini-batch SGD and Adam, can be used.

The last point is that according to the above equations, the computational volume for networks with many layers will be very high. Hence, the error back-propagation method reduces its computational load using techniques such as gradients finite approximation.

2.2.4 Deep learning

Deep learning is a subfield of machine learning with a hierarchical structure of many layers that models high-level abstract concepts using linear and nonlinear transformations. The neural structure inspires the motivation for using this learning structure in the human brain. In the brain's neural network, the process of perception and recognition is carried out by neural cells through processing in several successive stages. The essence of this process is to understand hierarchically but continuously highly complex data. Different models and structures have been proposed and studied in this field depending on the various hypotheses about how these neurons connect. However, these models do not naturally exist in the human brain, and the human brain is more complex in how it connects and conducts information. These models have made good progress in various areas such as Natural Language Processing (NLP) (Collobert and Weston, 2008) and image processing (Krizhevsky, Sutskever, and Hinton, 2012).

Unlike many machine learning methods, the deep learning model eliminates the need to extract and select features to a great extent. There is usually no need to do such preprocessing. The network performs the extraction and feature selection steps within the training process by feeding the data to a deep network. In these networks, in the initial layers, low-level features are extracted, and the further we move to the last layer, the higher the feature levels are until the final expected output is produced (Figure 2.15). A fundamental problem in deep learning is how to present data. The data submitted to the network during the training job should be sufficiently diverse so that the network can reach a correct and comprehensive understanding of the problem. For example, when processing speech, diversity factors such as the speaker's accent, age or gender can be influential. Also, when processing an image, for example, a satellite image, the amount of sunshine, atmospheric conditions, the location of objects and the conditions of obtaining the satellite image can be essential factors of diversity and, in some problems, should be considered.

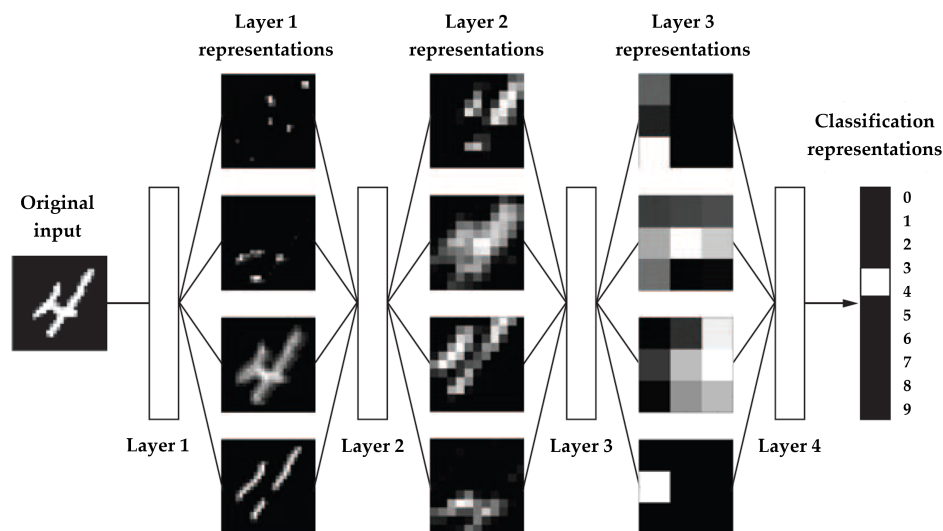


FIGURE 2.15: Input information processing in different layers of the neural network (Reproduced from Chollet (2021)).

Indeed, deep learning is the training process of deep neural networks for various applications. As seen in Section 2.2.1, single-layer networks have zero, multi-layer networks have one, and deep networks have two or more hidden layers. Although the error back-propagation algorithm made single-layer networks transition to multi-layer networks after 30 years, the use of deep networks was postponed to 20 years later due to their poor performance.

The main reason for the poor performance of deep networks was the inability to train them properly, even with the help of the back-propagation algorithm. The three basic challenges and problems faced by the back-propagation algorithm were (Kim, 2017),

- Vanishing/Exploding gradient
- Overfitting
- Computational burden

Since in the error back-propagation algorithm, the gradient of the input and hidden layers is calculated from the product of the gradient of their subsequent layers, if the value of the gradients of the layers close to the output is large or small, the gradient of the hidden layers close to the input will be very large or very small. These two modes are called exploding and vanishing gradient, respectively. In mathematical terms, if a network has d layers, for input x , output o and any arbitrary layer i with function f_i (Zhang et al., 2021),

$$h^{i+1} = f_i(h^i) \rightarrow o = f_d \circ \dots \circ f_1(x). \quad (2.63)$$

Therefore, for each layer f_i with parameter group w_i ,

$$\frac{\partial o}{\partial w_i} = \frac{\partial h^d}{\partial h^{d-1}} \dots \frac{\partial h^{i+1}}{\partial h^i} \frac{\partial h^i}{\partial w_i}. \quad (2.64)$$

It is obvious from the above relationship that if the values of the gradients are small, it will lead to a minimal value for the gradient of the network's hidden layers. This causes the number of small corrections applied to the hidden layers close to the input to be such that almost no update happens in these layers. Obviously, in this case, training and learning will not work.

On the other hand, if the values of the gradients are large, it will lead to a substantial value for the gradient of the network's hidden layers. In this case, the amount of corrections applied to the hidden layers is considerable, which prevents the network's convergence and makes the network's training unstable. Also, the enormous values of the gradients and the affected weights may be large enough to exceed computers' storage and processing power.

Fortunately, there are simple solutions for each of these problems. One reason for the vanishing gradient phenomenon is the extensive use of the sigmoid activation function. According to Figure 2.16, the gradient of the sigmoid function for large and small values is tiny and will disappear in the back-propagation.

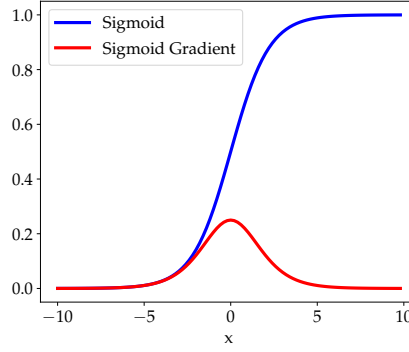


FIGURE 2.16: Graph of the sigmoid function along with its gradient.

The proposed simple but effective solution is to use the Rectified Linear Unit (ReLU) activation function (Goodfellow, Bengio, and Courville, 2016),

$$\varphi(x) = \begin{cases} 0 & x \leq 0, \\ x & x > 0. \end{cases} \quad (2.65)$$

Figure 2.17 shows the ReLU diagram.

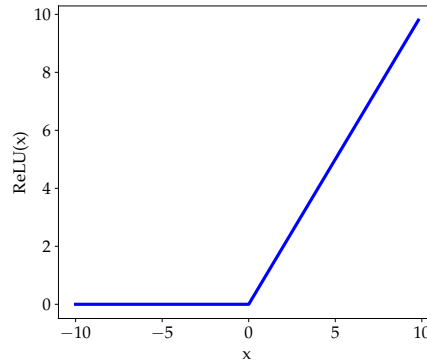


FIGURE 2.17: Graph of the ReLU function.

Now, if the ReLU function is derived, the following relation is obtained,

$$\varphi'(x) = \begin{cases} 0 & x \leq 0, \\ 1 & x > 0. \end{cases} \quad (2.66)$$

As it is clear from this relationship, the gradient of the mentioned function will have a value equal to 0 or 1, preventing the gradient of the hidden layers from vanishing. Other side solutions are suggested to help deal with the gradient explosion. One of these solutions is to limit the gradient in a specific interval during training (Goodfellow, Bengio, and Courville, 2016; Zhang et al., 2019). Another solution is appropriate initialization of the network parameters (Glorot and Bengio, 2010).

Overfitting will not be unexpected with the increase in the number of layers and, consequently, in the number of network parameters. Overfitting means that the model is fitted to the training data so accurately that the results will be terrible for any new data, such as test data. Several ways were proposed to solve this challenge, including dropout (Srivastava et al., 2014) or accidental removal of several parameters in each training iteration, which resolves overfitting despite being easy to implement. Another suggestion is to add a regularization term to the error function, which adds a condition to the network weights and partially prevents overfitting. Another common solution can be using an extensive dataset (Kim, 2017).

Another thing that deep learning needs to improve is its high computational burden. Because the high number of layers and functions that must be applied to the input and the error back-propagation require calculations. To solve this problem, two main approaches were adopted,

- Offering deep and effective structures with less computational load
- Increasing the computing power of computers

Structure-wise, CNN, which was invented by Fukushima (Fukushima and Miyake, 1982; Fukushima, 1988), provided the possibility of local feature extraction from images while reducing parameters and computational load. After that, LeCun et al. (1989) and LeCun, Denker, and Solla (1989) trained this network more optimally using the back-propagation algorithm, which became the cornerstone of current convolutional networks. On the other hand, with the advancements in hardware, in particular, the prevalence of using Graphics Processing Units (GPUs) and the development of software frameworks to perform calculations in parallel on the GPU, the computing power of computers has dramatically increased. For neural network training, GPU was used in various research such as Krizhevsky, Sutskever, and Hinton (2012) and Ciresan et al. (2011). For example, in Krizhevsky, Sutskever, and Hinton (2012), it was shown that GPU is very suitable for processing convolutional networks and with its help, a very deep network was trained. The results of the fast training of this network with the use of GPU were surprising, and in many cases, it performed better than human power in the visual recognition of objects. Ever since deep learning has moved towards the use of GPU-based convolutional networks, the capabilities of these processors have also improved significantly. This high processing power has made it possible to train complex networks with deep structures. Continuing this discussion, the convolutional neural network is discussed as a powerful and widely used neural network in deep models.

Convolutional neural network and convolutional layer

Convolutional neural networks are special neural networks suitable for processing grid data and data with significant spatial dependencies. Image data also has a two-dimensional or three-dimensional mesh structure of pixels. The name given to a convolutional neural network is derived from its convolutional layers. The main spark for the creation of convolutional neural networks was struck in the 1960s during research on the visual cortex (Hubel and Wiesel, 1959). According to this research, neurons show sensitivity to edges in the early brain layers and to more complex features in later layers. Also, one of their important findings is that the local structures of images are preserved in the brain. Based on these biological inspirations, the first neural model was designed by Fukushima (Fukushima and Miyake, 1982), although it could not be trained well due to some problems. After the failure of Fukushima, LeCun (LeCun et al., 1998) designed a network based on convolutional layers, which achieved outstanding results. Since then, convolutional networks have been widely used in many fields, and outstanding results have been obtained.

Each layer receives input from the previous adjacent layer in convolutional neural networks. This feature allows learning more complex patterns in a hierarchical method, combining elementary low-level and high-level features. For example, edges can be extracted from raw image pixels. On the other hand, edges can be used to extract simple shapes, and finally, shapes can be used to identify objects. In convolutional networks, meaningful and useful features can be extracted by sliding kernels on the input image for the network's primary purpose. These kernels, which have weights and small dimensions, are applied to all the pixels of the image by sliding on them (Figure 2.18). Therefore, through the sharing of parameters, convolutional networks have fewer parameters compared to fully connected networks, which will reduce the computational load and the possibility of overfitting (Figure 2.19) (Goodfellow, Bengio, and Courville, 2016). In addition, with the local application of kernels, these networks, like the function of the human visual system, can recognize a feature regardless of its displacement in the image by considering the relative positions, which is an available feature in many fields.

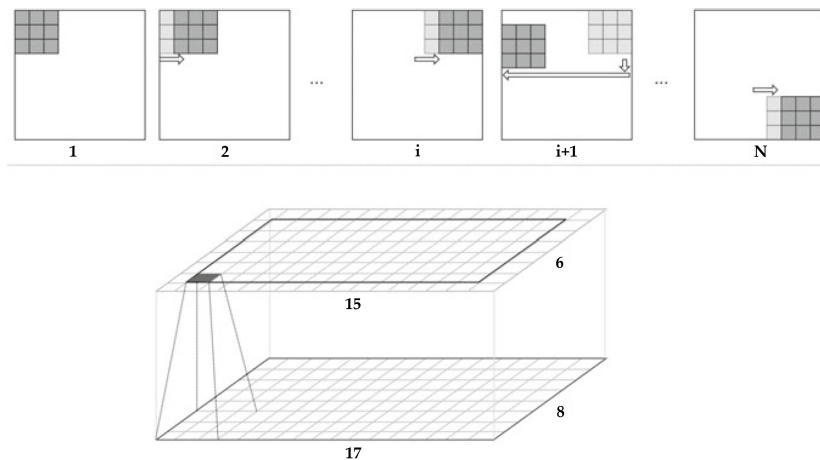
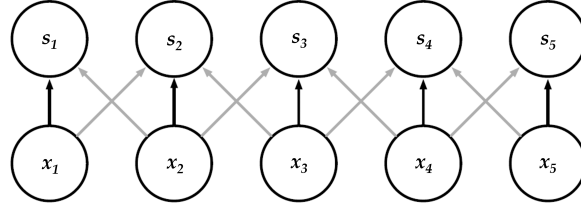
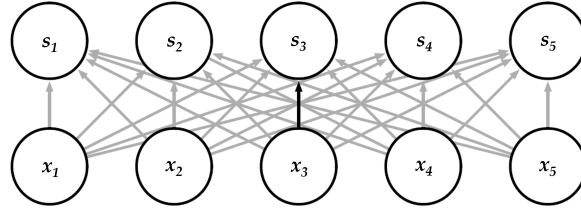


FIGURE 2.18: How to slide the kernel in a two-dimensional convolutional layer (Reproduced from Skansi (2018)).



(A) The black arrows indicate the use cases of the central component of a convolution network kernel, which has three components. It can be seen that x has been applied to all input units due to the sharing of parameters.



(B) The black arrow indicates the only used central component of the weight matrix in a fully connected network. As can be seen, the parameters are not shared in this network; therefore, each parameter is used only once.

FIGURE 2.19: Sharing parameters in the convolutional network compared to the fully connected network. Assuming input units x and output units s , black arrows in each network indicate connections that used a specific parameter (Reproduced from Goodfellow, Bengio, and Courville (2016)).

Each convolutional layer contains several kernels and biases that are locally applied to the input and produce a feature map or an activation map according to the number of filters. The cross-correlation operator performs calculations in these kernels, contrary to their names, which reminds us of the convolution operator, whose kernel is the negative of its definition in the convolution operator. If this convolutional layer is applied to the input image in a two-dimensional manner, its j^{th} feature map O_j , which is obtained by applying the j^{th} kernel, is calculated at the position (x, y) from the following relation (Figure 2.20) (Ji et al., 2012),

$$O_j^{xy} = B_j + \sum_k \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} w_j^{mn} z_k^{(x+m)(y+n)}, \quad (2.67)$$

where k moves along the depth dimension of input $Z \in \mathbb{R}^{p \times q \times r}$ and w_j^{mn} is the two-dimensional kernel weight $W \in \mathbb{R}^{M \times N}$ at position (m, n) . Also, B_j is the bias matrix.

Fully convolutional networks

Conventional convolutional networks often use fully connected layers in their last layers because their goal is to estimate the output for the entire input image. Fully convolutional networks use convolutional layers and their coordinated layers from the beginning to the end. As a result, the grid state of the input data is maintained until the end (output estimation). Therefore, instead of estimating the output for each

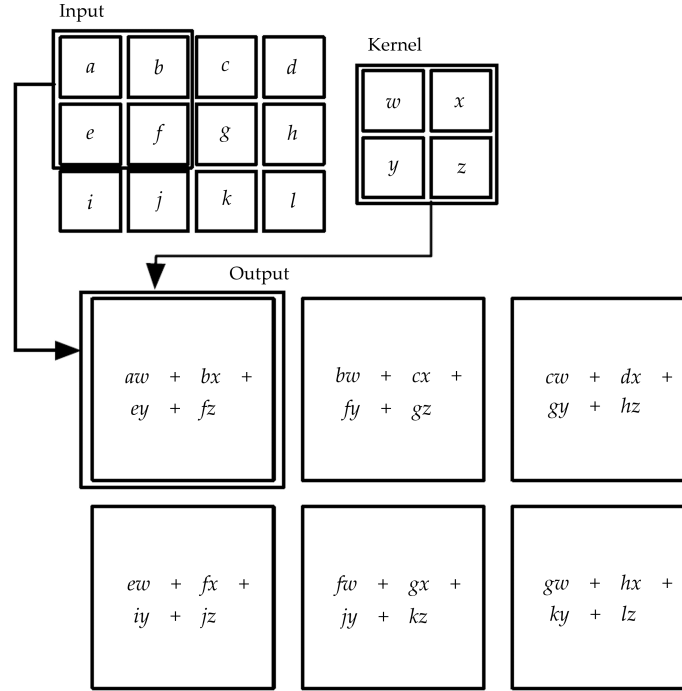


FIGURE 2.20: How to apply kernels to the input 2-D image in a 2-D convolutional layer (Reproduced from Goodfellow, Bengio, and Courville (2016)).

input data, as in conventional convolutional networks, the estimation is done per pixel (Figure 2.21). This feature makes fully convolutional networks an ideal choice in pixel-wise classification and segmentation (Zhang et al., 2021).

2.2.5 Mask R-CNN

Mask R-CNN, a member of the region-based CNN family, was proposed in He et al. (2017) and has been widely employed for various identification tasks, including recognition within the COCO dataset. Initially, RCNN was presented in Albawi, Mohammed, and Al-Zawi (2017), where computer vision techniques were utilized to generate region proposals. Subsequently, Fast RCNN was introduced in Girshick (2015), incorporating a CNN before region proposals to decrease execution time. In 2017, Faster RCNN continued this progression by introducing the RPN to propose ROIs (Chen and Gupta, 2017). Ultimately, Mask R-CNN, an extension of Faster RCNN, integrated an additional CNN for pixel-level recognition of detected object boundaries. Mask R-CNN is a relatively straightforward model and is easily adaptable to similar tasks (He et al., 2017). As a result, Mask R-CNN can generate pixel-level masks for objects in addition to performing object localization and classification tasks.

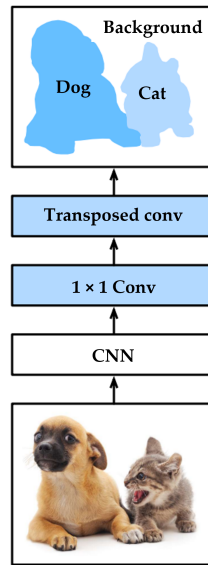


FIGURE 2.21: An example of a fully convolutional network for pixel-wise classification applications (Reproduced from Zhang et al. (2021)).

RPN

An RPN is a deep FCN that suggests regions and plays a critical role in Mask R-CNN. RPN assists in selectively concentrating on valuable aspects within input images. This network inputs an image and outputs a set of region proposals alongside their objectness scores. It slides a window over the convolutional feature map (backbone output) and maps it to a lower-dimensional feature. This generated feature is then fed into two fully connected layers to determine the proposed region's class (object vs. non-object) and the four corresponding coordinates (Chen and Gupta, 2017). A maximum of k potential proposals are parameterized for each sliding window location relative to k reference boxes or anchors. RPN is trained end-to-end using back-propagation and stochastic gradient descent. Figure 2.22 illustrates an RPN scheme that produces the proposed regions as module outputs.

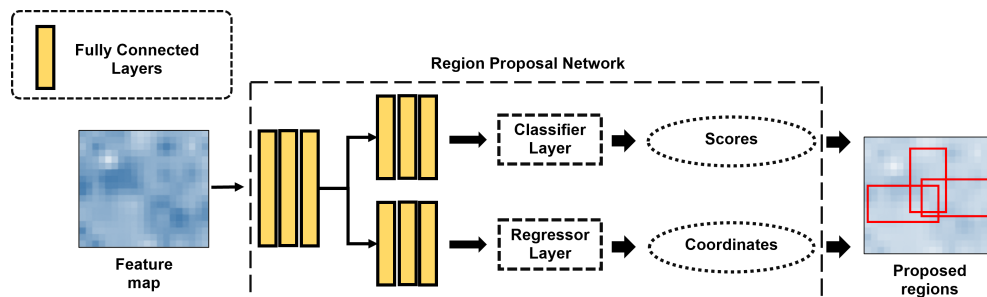


FIGURE 2.22: Region proposal network. Red rectangles illustrate the proposal regions on the feature map of the input image.

Loss function

Mask R-CNN loss function is a weighted summation of other losses related to different sections of this comprehensive model. As a definition based on He et al. (2017), a multi-task loss function is proposed on each sampled ROI as,

$$L = L_{cls} + L_{reg} + L_{msk}, \quad (2.68)$$

where L_{cls} recognizes the class type of each object while L_{reg} attempts to find the optimum anchor box for each object. Note that, in this study, we have one class, the plume cloud. L_{msk} tries to recognize the optimum object's segment in each bounding box.

3 Plume Imaging

In this chapter, we describe the dataset of plume cloud measurements in an industrial environment, obtained utilizing a field camera, along with wind direction data obtained from surrounding monitoring stations. We refer to this dataset in the following as the Deep Plume Rise Dataset (DPRD). DPRD comprises a collection of high-resolution RGB images and corresponding ground truths, date and time of capturing, weather conditions of each image, and geometric features of plume clouds. To create DPRD, image capturing was carried out over several months and in different lighting and atmospheric conditions, allowing us to record a wide range of plume cloud characteristics, including size, shape, and dispersion patterns. DPRD can be used to develop and validate algorithms for the automatic identification and measurements of plume clouds in industrial environments. Moreover, it can serve as a resource for scholars and policymakers on air pollution mitigation strategies. We believe that DPRD will facilitate significant advances in the field of plume cloud measurements and contribute to the improvement of accuracies in air-quality models.

3.1 Site description

The imaging system was deployed on a meteorological tower with a clear sightline to the desired stack. It is situated at the south end of Syncrude’s South Mine (Latitude: 56.99° , Longitude: -111.59°), approximately 5.2 km south of the Syncrude oil sands processing facility north of Fort McMurray, Alberta, Canada. Wood Buffalo Environment Association (WBEA) operates this 10-meter-tall meteorological tower with a clear sightline to the smokestacks at the south of the Syncrude facility (<https://wbea.org/stations/buffalo-viewpoint>, accessed on 1 January 2022). The camera system is mounted on this tower above the tree canopy, as the station is on a hill sloping downward from the tower location, and the largest smokestack and its plume cloud are always visible.

Syncrude Canada Ltd. is one of the world’s largest producers of synthetic crude oil from oil sands and the largest single-source producer in Canada. Emissions ranging from 62 to 164 kg CO₂ equivalent per barrel of Synthetic Crude Oil (SCO) were reported in Charpentier, Bergerson, and MacLean (2009), which results from surface mining and upgrading processes. Also, Charpentier, Bergerson, and MacLean (2009) reported emission rates of 99 to 176 kg CO₂ equivalent per barrel of SCO, produced through *in situ* and upgrading processes. Figure 3.1 shows the region’s satellite images, the camera’s location, the imaged smokestack, and the wind sensor’s location.

The Syncrude processing facility has six main smokestacks (Protection, 1998). The tallest one is about 183 m (600 ft), and the height of the other five is between 31 m and 76 m. In order to isolate a single

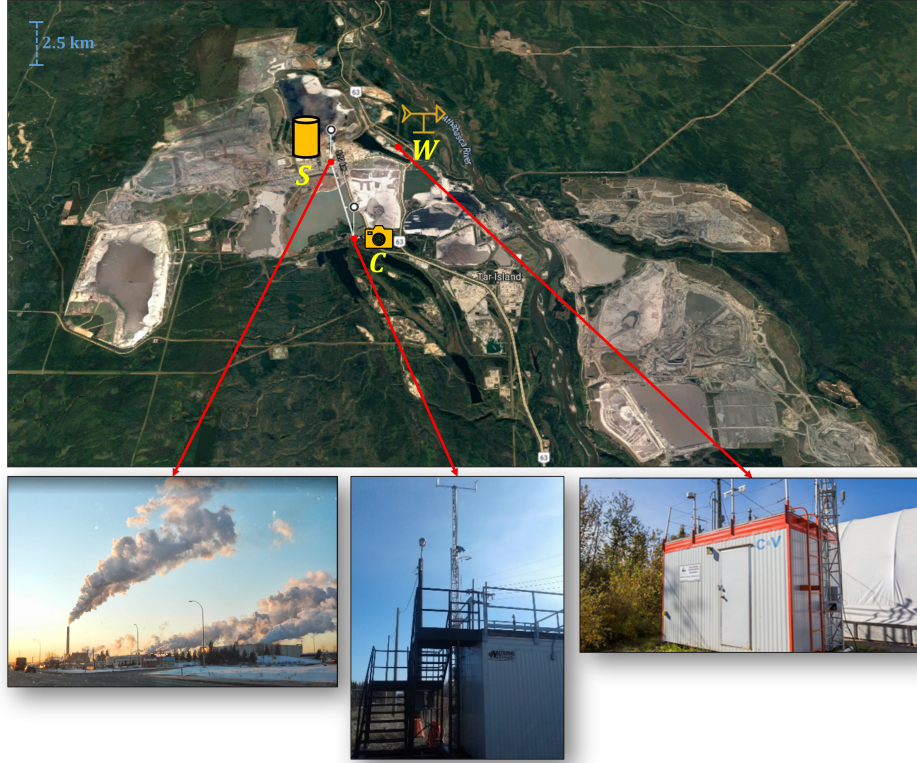


FIGURE 3.1: Imaging situation on the satellite image of the study region. Camera mounted on Buffalo Viewpoint station, C; wind sensors on Mildred Lake station, W; and the tallest smokestack location, S.

smoke plume rise, we have concentrated on the tallest one in the area. All six stacks are listed in Table 3.1.

Wind directions were determined from the Mildred Lake Air Monitoring Station (<https://wbea.org/stations/mildred-lake>, accessed on 1 January 2022), which is located at the Mildred Lake airstrip (AMS02: Latitude: 57.05° , Longitude: -111.56°), approximately 5km from the Syncrude facility. WBEA reports the wind direction in 1-hour averages.

3.2 Data collection

The main challenge when applying deep learning to identify plume clouds is the need for more availability of appropriately labelled images for the training phase. Thus, developing a dataset of images specifically for plume cloud recognition to support research and industry applications is beneficial.

3.2.1 Recent approaches in data generation

Recently, scholars have proposed different methods with dataset generation approaches based on their purposes. Zeng et al. (2018) used 1000 fire smoke pictures as training data and 400 pictures as verification

TABLE 3.1: Syncrude stacks information (Protection, 1998), including location, stack height (h_s), stack diameter (d_s), effluent velocity at the stack exit (ω_s), and effluent temperature at the stack exit (T_s). The velocities and temperatures are averages for the entire capturing period.

Reported ID	Latitude	Longitude	$h_s(\text{m})$	$d_s(\text{m})$	$\omega_s(\text{m s}^{-1})$	$T_s(\text{K})$
Syn. 12908	57.041	-111.616	183.0	7.9	12.0	427.9
Syn. 12909	57.048	-111.613	76.2	6.6	10.1	350.7
Syn. 13219	57.296	-111.506	30.5	5.2	8.8	355.0
Syn. 16914	57.046	-111.602	45.7	1.9	12.0	643.4
Syn. 16915	57.046	-111.604	31.0	5.0	9.0	454.5
Syn. 16916	57.297	-111.505	31.0	5.2	9.2	355.0

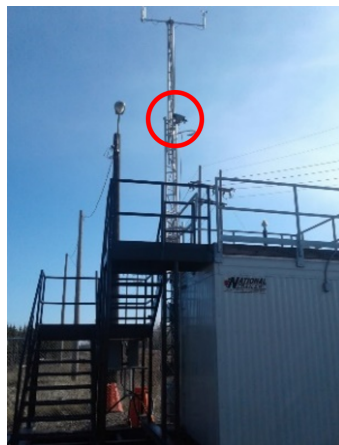
data for verified original faster R-CNN + SSD + R-FCN implementation. Lin et al. (2019) used synthetic smoke images generated by graphic software for faster R-CNN (smoke target location) and 3D CNN (smoke recognition). Yin et al. (2019) proposed two individual networks (motion network and space network) to be fused in an RNN for every two consecutive frames using three kinds of datasets. Small and large smoke images are labelled manually, and non-smoke datasets include many images similar to smoke but are non-smoke (gathered from the internet).

In 2019, Pundir and Raman (2019) used their own data for training the network. The Pundir and Raman (2019) study mainly focused on cloudy and foggy imaging conditions working based on SVM and CNN (separately for smoke and non-smoke images). LIU et al. (2019) used their own data for the training phase, splitting the smoke images, which are adjusted manually. Liu et al. (2019) used a public dataset on the site (smoke and non-smoke images) to train two separate CNNs (AlexNet); the first shallow network for dark channel extraction and the second deeper one to extract general features of the images. Jakubowski, Solarczyk, and Wiśnios (2019) created a set of training data using smoke and non-smoke images to modify the last part of the pre-defined network, which was AlexNet. Matlani and Shrivastava (2019) utilized data of smoke with different backgrounds from their own database in the VGG-net CNN classifier. Cheng, Ma, and Zhang (2019) used a set of manually labelled smoke and non-smoke video frames in a set of encoders and decoders with smoke edge optimization. Yuan et al. (2019) generated synthetic data using different channels of RGB in a linear equation to be applied to an FCN with two paths: coarse path and fine path. Aslan et al. (2019) used pre-processed data (motion-based geometrical transformation) on captured camera video frames in DCGAN with two stages of training. Xu et al. (2019) utilized a pre-determined training dataset from the web to train a combination of pixel-level, object-level and region-level salient CNN.

To the best of our knowledge, there is no image dataset for smokestack plume cloud recognition. Therefore, it is valuable to create this dataset for both research and industry purposes.

3.2.2 Project data collection

Image dataset collection for this project started on 5 November 2018 and finished on 20 March 2020. The plume cloud imaging camera system consists of a CCFC field camera (<https://www.campbellsci.com/ccfc>, last access: November 2021) with shutter control and a camera housing for weather protection with interior heating for window defrost and de-icing. The housing's camera activation and temperature control are controlled by a data logger, which is powered by a small solar cell. It was programmed to take images multiple times throughout the day, with a pre-determined frequency dependent on the length of the project and available memory and save them to a local computer. Figure 3.2 shows the imaging camera and the monitoring station with the meteorological tower. Table 3.2 gives the CCFC field camera specifications. The CCFC is designed to meet the stringent operational requirements necessary for remote battery-powered installations while producing HD video and photos of up to 5 megapixels. It can operate over a wide temperature range and has several advanced power-saving modes to suit a variety of needs.



(A) Monitoring station
(<https://wbea.org/data/network-map-station-data/>,
last access: August 2020)

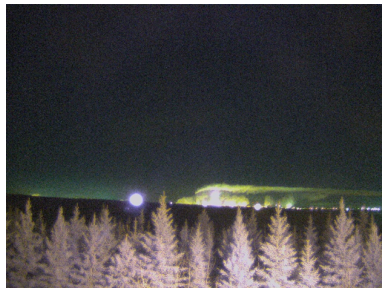


(B) CCFC camera
(<https://www.campbellsci.com/ccfc>,
last access: November 2021)

FIGURE 3.2: Plume imaging system. The red circle shows the camera location.



(A) Daytime



(B) Nighttime



(C) Cloudy/foggy

FIGURE 3.3: Sample collected images.

Forty-four thousand seven hundred forty images were collected from the area, and the size of each image is about 840 to 880 KB. Figure 3.3 shows three sample images from the region.

TABLE 3.2: CCFC camera specifications
(Reproduced from <https://www.campbellsci.com/ccfc>, last access: November 2021.

Specification	Description
Focal length	4.7 mm
Pixel dimension	2.4 μm
Field of view	55 degrees
Power supply	9 – 30 VDC input voltage
Operating temperature	-40°C to +60°C
Weight	2.4 kg (5.25 lb)
Clock Accuracy	± 2 Minutes/Year (-40°C to +60°C)
Dimensions	Length: 28.4 cm (11.2"); Height: 13.0 cm (5.1"); Width: 13.2 cm (5.2")
Lens	4.7 to 84.5 mm, 3° to 55° horizontal field of view
Photo Capture Times	Fully On: < 1 s; Partially On: 10 sec; Deep Sleep: 10 sec; Off State: 90 sec
Photo Resolutions (JPEG)	2592 x 1944; 1200 x 960; 1280 x 720; 640 x 480
Modem Power Control	Maximum Output Current: 750 mA; 12 VDC
Communication Protocols	Web interface via web browser; Email; PakBus®

3.2.3 Data characterization

Totally 96 images were captured every 24 hours as a part of our image dataset, which demonstrated many types of plume shapes in different atmospheric conditions. To classify the dataset, we considered the atmospheric conditions and categorized the images into three classes: daytime, night times, and cloudy/foggy. The collected datasets showed that among 96 images every 24 hours, we have 48 day and 48 night images. Therefore, 1450 day images and 1450 night images exist for every month of the year. Image classification has been initiated with time coding for every captured image. Table 3.3 shows the time coding for a sample image to represent the date and time of image recording.

For our study, images were collected and annotated based on two classes: "Plume cloud" and "background," and 60% of the datasets are considered as training data, and 15% remains for validation and 25% for testing purposes. To achieve the results for Sections 4 and 7.3, we have chosen 6000 valid plume images, which consist of 3600 training images, 850 validation images and 1550 test images. Rows in Figure 3.4 shows sample images from the region and their corresponding ground truth, which are generated by the "Labelme" graphical image annotation tool at <https://github.com/wkentaro/labelme>. We tried to select images of different atmospheric conditions, such as clear daytime, nighttime, cloudy, and foggy, to represent the results of different situations.

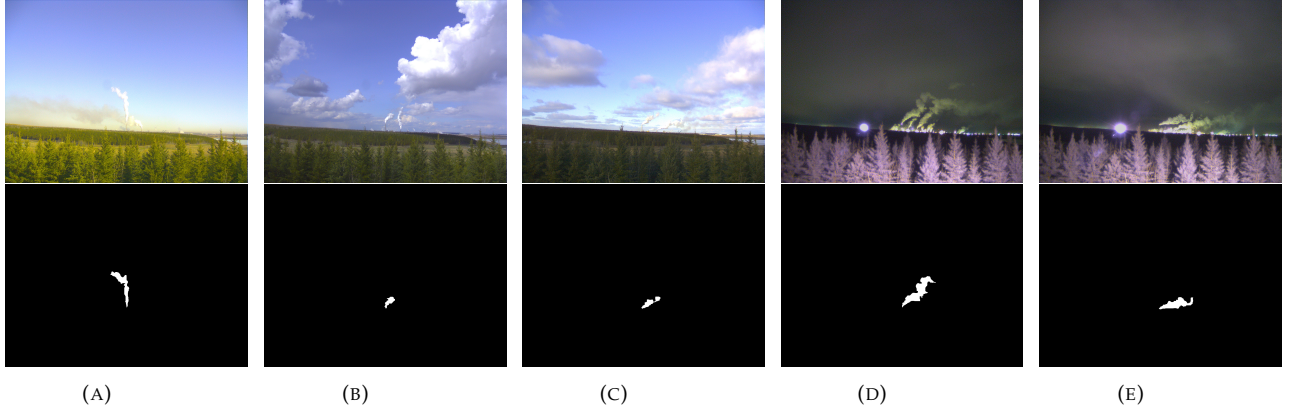


FIGURE 3.4: Sample images (up) and their corresponding ground truth (down) from our DPR dataset listed as (A) Clear daytime, (B)&(C) cloudy day, and (D)&(E) clear nighttime.

TABLE 3.3: Sample time coding for the recorded images.

Photo1_2019_02_20_13_00_12						
Photo1:	2019:	02:	20:	13:	00:	12:
Camera #1	Year	Month	Day	Hour	Minute	Second

Some images are not captured well for different reasons, such as camera handle shaking, auto-focus problems, disturbing snow (frost) and severe hail falling (severe precipitation). Furthermore, even upon visual inspection, some plume clouds could not be distinguished from their background. Moreover, some images include plume clouds from several smoke-generation sources in the area, and some of them were taken during periods when the facility was not operational. Figure 3.5 displays some samples of these invalid images from the region and a sample of the images with no plume, so-called valid non-plume images.

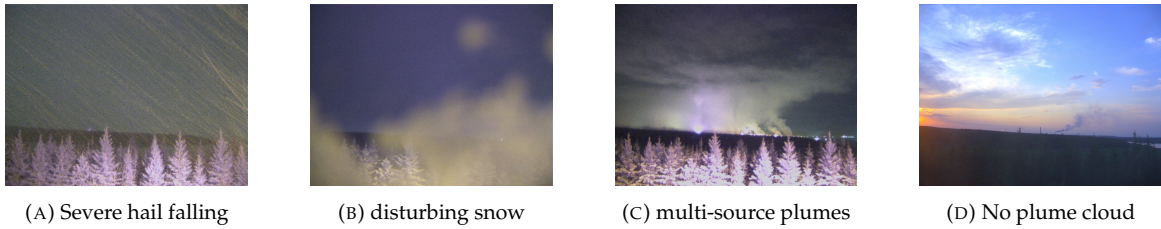


FIGURE 3.5: Sample outlier images.

Also, Figure 3.6 shows the number of valid and invalid images in the chosen capturing period. As seen in Figure 3.6, a large amount of captured images are invalid, and the main portion of the valid images

have no plume in them. For this chosen capturing period, 7.35% of total images are valid plume images and, consequently, usable.

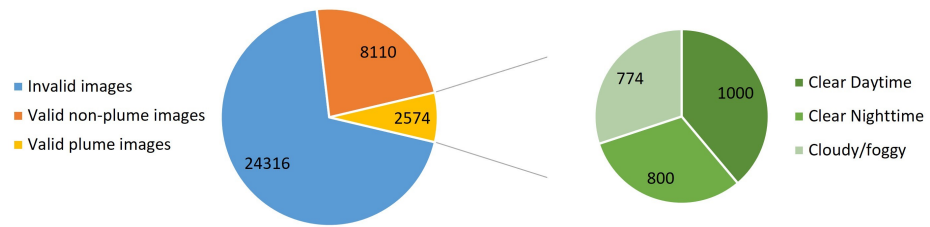


FIGURE 3.6: Distribution of the collected images from January 2019 to December 2019.

4 DPRNet Pipeline

The proposed framework for plume rise measurement is depicted in Figure 4.1. Images containing plume cloud(s) are input to the DPRNet for plume cloud boundary detection and recognition. Plume rise and plume rise distance are subsequently measured based on the extraction of the NBP from DPRNet's output. The image coordinates of the NBP are combined with wind direction data for processing through geometric transformation calculations. The primary output of the system provides the plume rise as a physical height and the plume rise distance downwind at which the plume rise occurs.

Additionally, a schematic diagram in Figure 4.2 illustrates the definition of the plume cloud centerline, plume rise, and plume rise distance. We have concentrated on recognizing plume clouds over their different background conditions. Then, we propose a deep network that can be applied to extracting the region of the plume cloud area of interest. An improved Mask R-CNN model is proposed for extracting the region of the plume cloud area of interest (He et al., 2017). In the next step, geometric transformations are performed to represent the plume cloud measurements in real-world scales. Therefore, the proposed model produces two outputs. First, the plume cloud foreground segments and second, multiple lines for representing the central lines and upper and lower boundaries from the segmented regions.

The plume rise measurement system based on the DPRNet model designed in this research is used via sending data to the DPRNet for plume cloud border detection and recognition, and the results are combined with the wind direction information to be processed by geometric transformations. The system's main output will reveal the plume rise in the input data.

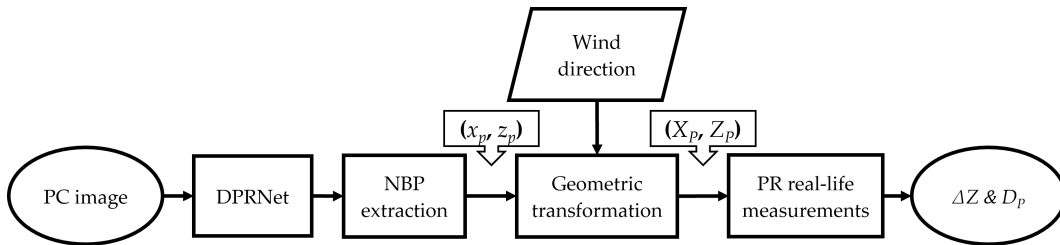
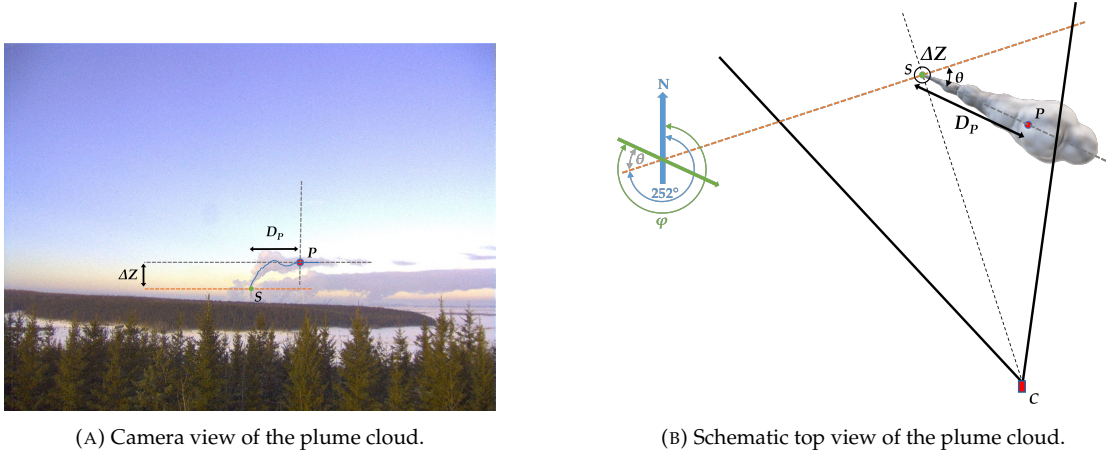


FIGURE 4.1: Plume rise measurements system framework. x_p and z_p are the NBP coordinates in the image scale. Similarly, X_p and Z_p represent the NBP coordinates in the real-life scale. ΔZ and D_p are plume rise and plume rise distance on a real-life scale. The $x - z$ system is chosen for visualizing the plume rise measurements to be consistent with other studies in environmental science.



(A) Camera view of the plume cloud.

(B) Schematic top view of the plume cloud.

FIGURE 4.2: Plume rise, ΔZ ; plume rise distance, D_P ; and NBP, P ; on a sample image. θ represents the plume cloud deviation from the projection plane due to the wind, and φ denotes the wind direction. S indicates the smokestack exit, C represents the camera position, and the blue line shows the plume cloud centerline.

4.1 DPRNet

This research aims to accurately identify the plume cloud of the target smokestack from a broad range of image datasets captured within the study area. DPRNet is a customized version of Mask R-CNN, featuring two innovative smokestack plume rise measurement modules: (1) the physical module and (2) the loss regularizer module. These modules enhance the RPN performance in pinpointing the most probable proposal plume clouds. Mask R-CNN, the foundation of the proposed method, is a widely used boundary detection and recognition technique. This robust framework accounts for the irregular shapes of plume clouds, their translucent edges, and similar pixel values compared to their background (He et al., 2017). As illustrated in Figure 4.3, DPRNet is an application-oriented adaptation of Mask R-CNN, augmented with two novel modules.

4.1.1 Structure

In this architecture, ResNet50 (He et al., 2016) serves as the backbone network for extracting feature maps from input images. The Feature Pyramid Network (FPN) utilizes these feature maps to produce multi-scale feature maps containing more valuable information than conventional feature pyramids. Subsequently, the RPN identifies the plume cloud by sliding a window over these feature maps, predicting the presence of a plume cloud and locating it by creating bounding boxes. Consequently, a set of plume cloud proposals from RPN and the feature map generated by the backbone network are obtained. The ROI Align module scales the proposals to the feature map level and prevents misalignment by standardizing the proposals' aspect ratios. Finally, these refined feature maps are directed to three separate outputs. The first output is a classification block determining whether the ROI corresponds to the foreground (plume cloud). The second output is a regression block that predicts bounding boxes based on

the provided ground truth. The final output is a recognition mask for the detected plume cloud, generated using an FCN (Long, Shelhamer, and Darrell, 2015).

Two modules are incorporated into Mask R-CNN to enhance effectiveness and decrease computational complexity. The first module, a basic image processing module, estimates the smokestack. The second module aims to improve the loss associated with L_{reg} by introducing a regularizer loss, detailed in Section 4.1.3. These modules are thoroughly discussed in the subsequent subsections.

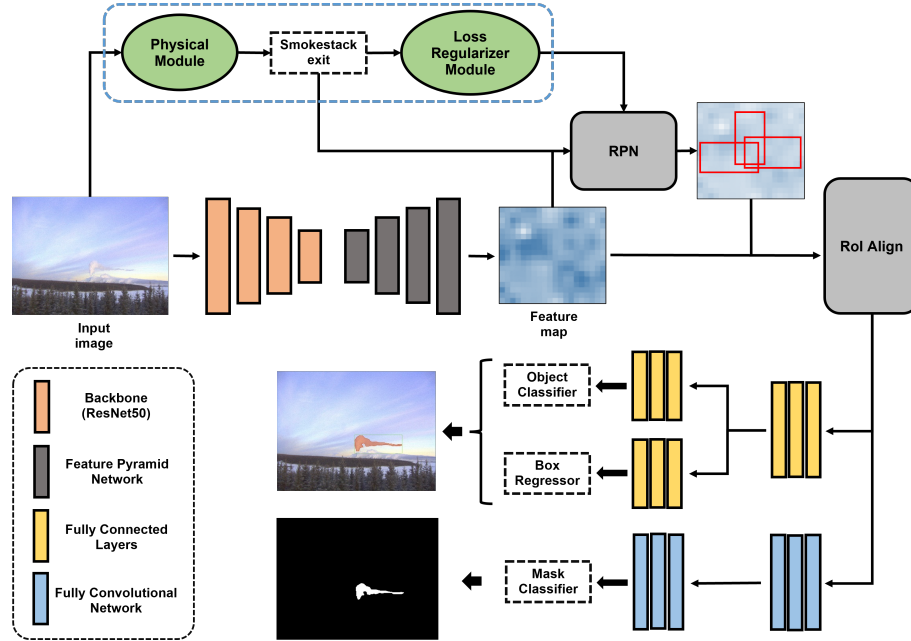


FIGURE 4.3: DPRNet architecture. The supplemental modules are shown in green, and the dashed blue rectangle is dismissed in the inference time.

4.1.2 Physical module

The physical module is designed to identify an exit point from which the plume cloud of interest emerges from a chimney stack in the image's center. It is important to note that this module is utilized only during the training phase. At the same time, the exit point is automatically predicted during the network's inference process without the physical module. The smokestack exit is detected in the physical module from the ground truth binary image using an image processing technique that identifies extrema points of foreground boundaries (Gonzalez, Woods, and Eddins, 2009). Logically, the smokestack exit is the plausible region for the plume rise. Consequently, proposed regions can be considered around this point (Figure 4.3). This module ensures that the method does not detect small plume cloud fragments occasionally visible in different parts of images apart from the smokestack exit.

To identify the smokestack exit, eight extrema points on each plume cloud's boundary are extracted based on image processing analysis (Gonzalez, Woods, and Eddins, 2009), as depicted in Figure 4.4. This algorithm first smooths noise in the foreground regions (i.e., the plume cloud region labelled in the ground truth imagery). It then tracks boundary pixels of the foreground region and extracts eight

extrema points exhibiting the locally highest geometric curvature. The bottom-left corner of each plume cloud segment is regarded as the smokestack exit point of S in the world coordinate depicted in Figure 4.2. Furthermore, the corresponding point in the image coordinate is s .

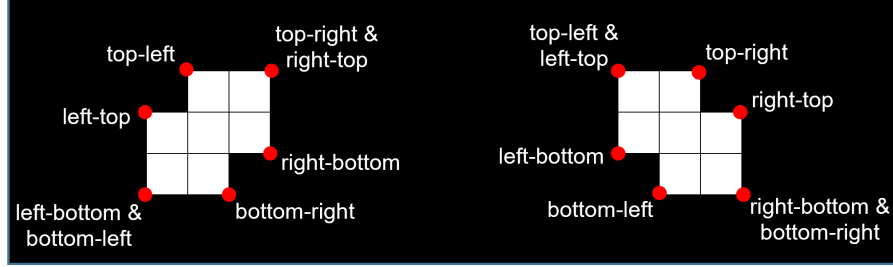


FIGURE 4.4: Sample plume cloud segments with eight boundary points.

4.1.3 Loss regularizer module

As discussed in Section 2.2.5, defining an effective loss function is crucial for stabilizing the model. In this context, a new regularizer is added to the loss function, which regresses the coordinates of the most plausible plume cloud regions. The aim is to minimize the distance between the bounding boxes proposed by the RPN and the smokestack exit. Consider a box with coordinates (x, z, w, h) defined here. In that case, the regression loss related to the smokestack exit can be defined as,

$$L_S = R(u - u^*), \quad (4.1)$$

in which,

$$\begin{aligned} u_x &= \frac{x - x_a}{w_a}, \quad u_z = \frac{z - z_a}{h_a}, \quad u_w = \log\left(\frac{w}{w_a}\right), \quad u_h = \log\left(\frac{h}{h_a}\right), \\ u_x^* &= \frac{x^* - x_a}{w_a}, \quad u_z^* = \frac{z^* - z_a}{h_a}, \quad u_w^* = \log\left(\frac{w^*}{w_a}\right), \quad u_h^* = \log\left(\frac{h^*}{h_a}\right) \end{aligned} \quad (4.2)$$

where u and u^* represent the coordinates of our predicted and ground truth smokestack exit, and R is the robust loss function. Note that the subscript a and superscript $*$ variables denote the anchor coordinate and ground truth coordinates, respectively, while the remaining variables represent the predicted coordinates. The point (x, z) specifies the location of the bounding box's top-left corner, and the parameters w and h correspond to the width and height of the bounding box, respectively.

In contrast to the Mask R-CNN model, DPRNet's loss regularizer module (L_S) minimizes recognition task errors for a specific plume cloud, addressing the primary issues of this model, such as missing the desired smokestack exit and proposing multiple boxes for a single plume cloud. L_S aids in preventing the need to span the entire image pixels, which leads to high training times and computational complexities.

4.1.4 Training

The training of DPRNet was conducted on a dedicated computing infrastructure equipped with a computer equipped with Core i9, 3.70 GHz/4.90 GHz, 20 MB cache CPU, 64GB RAM and NVIDIA GeForce RTX 3080, 10 GB graphic card. The total training time of the network was about nine hours using Python 3.8 with PyTorch Deep Learning framework. The running time for a single 2592×1944 test image was about 0.27 seconds. Finally, for the geometric transformation and image processing analysis, we used MATLAB R2022b software. The dataset used for training comprised a diverse set of image datasets captured within the study area, containing various environmental conditions and scenarios. The training process aimed to enable DPRNet to accurately identify the plume cloud of the target smokestack from this broad range of images. We utilized SGD as the optimization algorithm with a learning rate of 0.001. The weights were initialized using the random initialization method, and the training process spanned 60 epochs.

To evaluate the performance of DPRNet during training, we monitored two key metrics: training loss and accuracy. The training loss measures the error between the predicted plume clouds and the ground truth, while accuracy reflects the network's ability to identify plume clouds in the images correctly. The training results are depicted in Figure 4.5, displaying the evolution of training loss and accuracy over the 60 epochs. Figure 4.5a illustrates the training loss's convergence and the error reduction as the network learns from the dataset. Figure 4.5b provides insights into the training accuracy, showing the improvement in the model's ability for accurate plume identification.

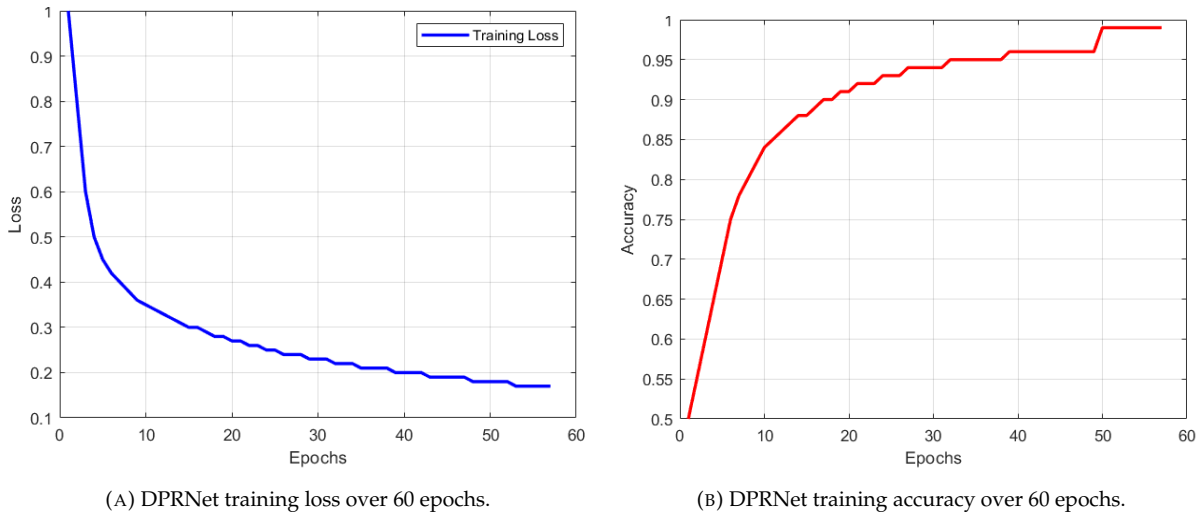


FIGURE 4.5: DPRNet training results.

The observed trends in training loss and accuracy indicate the effectiveness of the training process. The convergence of these metrics certifies the robustness of DPRNet, highlighting its adaptability to the complexities of plume cloud identification in various environmental conditions.

4.2 NBP extraction

As illustrated in Figure 4.1, plume rise measurements are carried out based on extracting the NBP for each plume cloud image. This point is depicted as p in the image coordinate and P in the real-world coordinates in the figures. The NBP extraction phase can be summarized in the following three steps, with an overall processing chain shown in Figure 4.6

4.2.1 Centerline extraction

In the first step, the plume cloud's centerline is extracted as a curve representing the meandering of the plume cloud. This is achieved by dividing the boundary pixels of the plume cloud into two categories: upper and lower boundary pixels, displayed as cyan and yellow-coloured lines, respectively, in Figure 4.6. The upper and lower boundaries are determined by identifying the highest and lowest white pixels in the binary plume cloud image. This method enables the extraction of the plume cloud's skeletonized curve, the centerline, by calculating the mean of the upper and lower points. This binary classification of boundary lines is performed by tracking the upper pixel at the highest row index. In contrast, the lowest one for the lower boundary pixel is determined from each fixed column index with the coordinate origin at the lower-left corner of the image space. The centerline pixel per column is found by simply averaging the two-row indices.

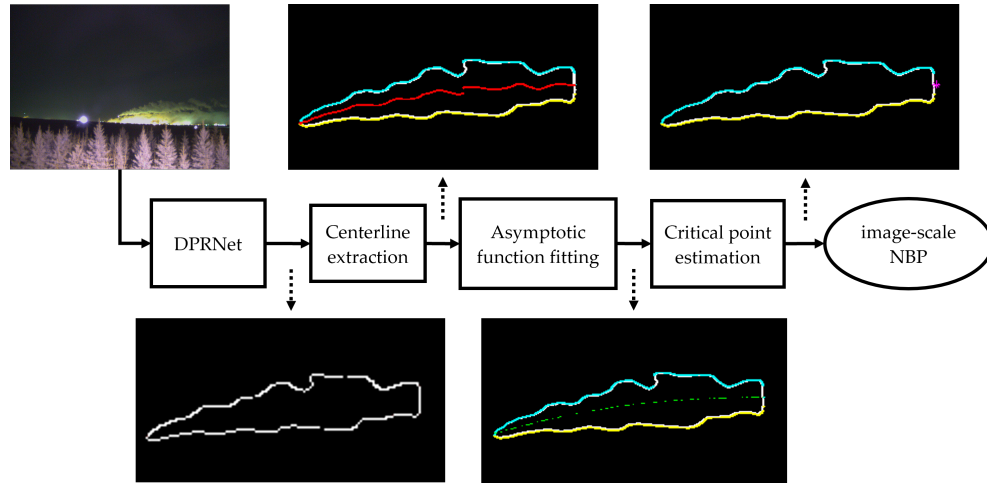


FIGURE 4.6: NBP extraction framework. Dashed arrows show the output of each block. The red curve represents the centerline of the plume cloud. The cyan and yellow lines, respectively, display the upper and lower boundaries of the plume cloud. Green dashes demonstrate the asymptotic curve, and the magenta point is NBP.

4.2.2 Asymptotic function fitting

Measuring the NBP for plume rise in the context of atmospheric science using imagery requires identifying the point at which the buoyant force from the plume is equal to the gravitational force acting on it. At this point, the plume stops rising further and begins to spread laterally. The central lines detected from the imagery display prominent visual cues caused by various factors involved in atmospheric dynamics.

However, they cannot explicitly estimate the NBP and necessitate a physical interpretation to determine the NBP from the centerlines. Consequently, an asymptotic function (Berg, 2013) is fitted to the centerline pixels using a conventional least-squared method to derive this function's horizontal asymptote, which helps determine the NBP from the imagery. Equation 4.3 presents the function employed in this study as

$$z = z_{stack} + (z_{NBP} - z_{stack})(1 - \exp(-\frac{x - x_{stack}}{x_{NBP} - x_{stack}}b)) \quad (4.3)$$

where (x_{stack}, z_{stack}) and (x_{NBP}, z_{NBP}) are representatives of the smokestack exit and NBP pixel location on the image, and b is a constant that defines what we consider "asymptotic", which is discussed in Section 4.2.3. This equation is defined in the image coordinates of x as the horizontal axis and z as the vertical axis.

An exponential equation was utilized since it is recognized as one of the field's commonly used functions, which approaches a particular value (in this case, the maximum height of the plume cloud) as the input variable (in this case, downwind distance, x) increases without bound. This behaviour is similar to what we might expect from the plume cloud rising from a smokestack, which will continue rising until it reaches a certain height and disperses into the surrounding air.

It's essential to recognize that choosing an exponential function in modelling the plume cloud centerline is just one of many possible choices, and there may be better fits for a given dataset or problem with specific underlying physical processes. Exponential functions are often chosen as default or initial models because they are relatively simple and have desirable properties, such as representing growth or decay rates. However, this simplicity comes with certain limitations. For example, an exponential function may not work efficiently if the plume cloud complex behaviour with multiple stages or transitions needs to be studied. Figure 4.7 provides some examples of alternative functions to be fitted for a specific plume cloud from our dataset.

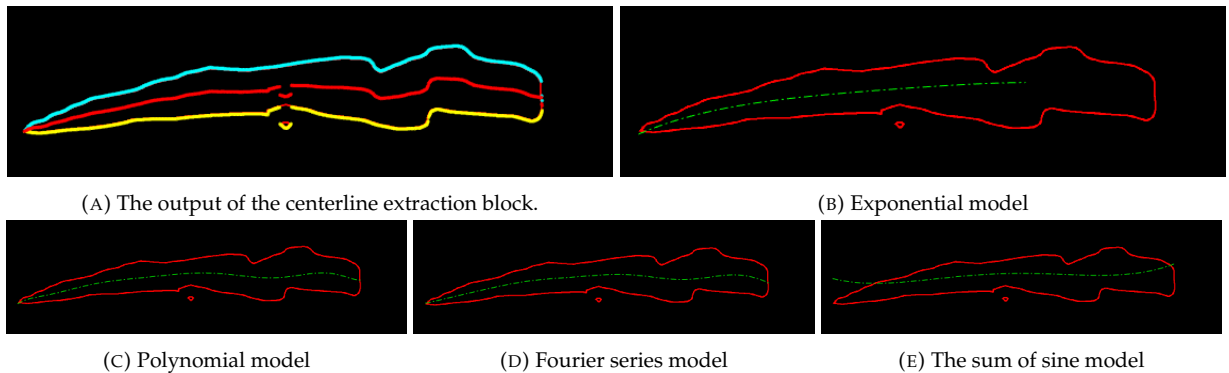


FIGURE 4.7: Examples of alternative fitting curves for a specific plume cloud recognition result to find the trend of its centerline. The red curve represents the centerline of the plume cloud. The cyan and yellow lines, respectively, display the upper and lower boundaries of the plume cloud, and green dashes demonstrate the fitting curve accordingly.

To sum up, making a more informed choice of modelling the smokestack plume cloud requires a more rigorous analysis of potential limitations, including comparing the fit quality of various functions using the R^2 value.

4.2.3 Critical point estimation

In the final stage of our NBP extraction process, we aim to specify the critical point on the fitted exponential curve. The critical point on the exponential curve symbolizes when the buoyant force acting on the plume equals the gravitational force, causing it to stop its vertical ascent and initiate lateral dispersion. This critical point corresponds to the NBP. To estimate this critical point, we employ a mathematical technique that involves analyzing the behaviour of the fitted exponential curve and identifying the point at which the exponential curve approaches its horizontal asymptote. On the other hand, we know that the suggestion of a zero slope is impossible, as the exponent function will never reach a zero slope. Therefore, if we substitute the NBP coordinates in Equation 4.3, the obtained value for b becomes ∞ , which is unacceptable. To address this concern, we considered the NBP to be where $z = 0.99(z_{NBP} - z_{stack})$, then the equation can be solved for b as

$$1 - \exp(-b) = 0.99 \quad \text{which gives} \quad b = 4.605. \quad (4.4)$$

The fitting function determines the decay rate to the generic equation, $\hat{y} = A(1 - \exp(-B\hat{x}))$, where $\hat{x}$ and $\hat{y}$ are the distances from the stack, and the best fit determines A and B . The choice of b determines how we define where the NBP is. In other words, if we chose values of b to give the NBP location at either $z/(z_{NBP} - z_{stack}) = 0.90, 0.99, \text{ or } 0.9999$, it changes where you place the NBP along the curve. As x increases, the plume concentration gradually approaches z_{NBP} . This behaviour resembles an asymptote in practice because the concentration change becomes very slow but never reaches a specific constant value. While the choice of b doesn't affect the plume rises in the vertical distance, it does impact the plume behaviour in the horizontal distance. Therefore, due to the exponential decay term, the provided equation does not have an actual asymptotic point. Choosing a value close to 1, 0.99 here can make the plume behaviour appear asymptotic over long horizontal distances. However, it technically never reaches a constant value. This choice affects the horizontal distance of the plume while maintaining its vertical rise characteristics, which causes uncertainty in plume rise distance measurements from an image. In this context, the horizontal asymptote represents the theoretical limit beyond which the plume cannot rise further due to the balance of buoyant and gravitational forces.

It should also be noted that the obtained critical point, considered the NBP point of the plume cloud, can be displayed both inside and outside of the observed plume, depending on its specific dynamics and conditions. In our dataset, where the observation is centred around an active smokestack continuously emitting a plume, it is observed that the NBP mainly occurs within the plume. However, it's crucial to admit that the NBP can also display outside the visible plume cloud, in the surrounding air rather than within the dense plume cloud structure. Our approach, which employs the fitted exponential curve and mathematical analysis, allows us to account for these variations, which comes back to the definition of the plume cloud neutral buoyancy moment.

5 DPRNet Comparison Results

This chapter first discusses the validation metrics used to compare our proposed DPRNet to other competitive methods in smoke border detection and recognition. Our discussion then proceeds to the two final sections, titled “Comparison with existing smoke recognition methods” and “Plume rise measurement”, where the performance of the proposed method is evaluated, and the PR and PR distance are calculated in the image scale based on our “DPRNet”, respectively. To validate the performance of our proposed method, we used a computer equipped with a Core i9, 3.70 GHz/4.90 GHz, 20 MB cache CPU, 64 GB RAM, and an NVIDIA GeForce RTX 3080, 10 GB graphics card. The total training time for the network was approximately one hour, utilizing Python 3.8 with the PyTorch Deep Learning framework. Finally, for image processing analysis and comparison tasks, we employed MATLAB R2022b software.

5.1 Model validation metrics

The performance of the considered methods is assessed using accuracy, recall, precision, and F1 score metrics. These metrics are defined based on four values, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which are derived from the confusion matrix of each introduced method (Dougherty, 2012). The accuracy validation metric represents the proportion of correctly predicted observations to the total number of observations. In our application, the model’s accuracy signifies its ability to recognize plume cloud pixels accurately. This criterion is valid as long as FP and FN values are roughly equal (Dougherty, 2012). If not, alternative validation metrics should be taken into account. The foreground pixel coverage of the sample images indicates that accuracy is not appropriate for this study. Recall, or sensitivity, is the ratio of correctly predicted positive observations to all actual positive observations. Recall demonstrates the number of plume cloud pixels labelled among all the actual plume cloud pixels. The recall is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \quad (5.1)$$

Precision is the ratio of positive observations that are predicted correctly to all observations that are predicted as positive. This metric represents how many plume cloud pixels exist among all the pixels labelled as plume cloud. Therefore, a low rate of FP can achieve high precision. This validation metric is obtained as follows,

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

As it is implied from Equations 5.1 and 5.2, precision and recall take either FP or FN into account. The last validation measure in this paper, the F1 score, considers both FP and FN as a weighted average of recall and precision metrics. Unlike accuracy, this metric is more useful when FP and FN are not the same as in our study. Our FP is less than FN, or the amount of non-actual plume cloud pixels predicted as plume cloud pixels is less than that of actual plume cloud pixels predicted as non-plume pixels. Therefore, the F1 score helps us look at both recall and precision validation metrics as follows,

$$F1score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (5.3)$$

5.2 Comparison with existing smoke recognition methods

In this section, we evaluate the performance of DPRNet and compare it with several competitors. To select suitable smoke recognition methods for comparison, we considered both the identification accuracy and computational complexity of the reviewed approaches, leading us to choose DeepLabv3+ (Chen et al., 2018), FCN (Long, Shelhamer, and Darrell, 2015), and regular Mask R-CNN. These methods have demonstrated good performance in object border detection and recognition tasks in previous studies and employ deep convolutional neural network architecture, effectively capturing and processing spatial features in the input image, resulting in improved smoke border detection and recognition accuracy. Our proposed DPRNet is evaluated using three metrics introduced in Section 5.1.

As evident from Table 5.1, DPRNet outperforms competitive methods for 90 test images selected from various day, night, foggy, and cloudy conditions. Specifically, the recall and precision metrics highlight a significant difference between the models, demonstrating the effectiveness of the proposed model in recognizing actual plume cloud pixels. The higher F1 score value for DPRNet compared to the competitors guarantees that DPRNet performs better than the other three methods, showcasing its efficacy. Among our competitive methods, DeepLabv3+ performed better in terms of all validation metrics, while Mask R-CNN had the worst performance.

TABLE 5.1: Comparison of different methods for plume cloud recognition using average validation metrics values.

Mode	Recall	Precision	F1 Score
Mask R-CNN	0.556	0.727	0.607
FCN	0.591	0.859	0.599
DeepLabv3	0.654	0.892	0.721
DPRNet	0.846	0.925	0.881

In addition to these average values, the detailed statistics for each model are provided in Figure 5.1 concerning each validation metric used. This figure shows that our proposed method exhibits the most robust performance in all situations. Among the competitors, Mask R-CNN and FCN have the worst

performance, while DeepLabv3 has slightly better efficiency. It is also evident from Figure 5.1 that some competitors have validation metrics of zero. This indicates that these methods were unable to detect any plume clouds in the corresponding images. In other words, the model might have been trained on a specific data type, and consequently, it might misclassify unfamiliar types of data, resulting in lower accuracy, recall, and other performance metrics.

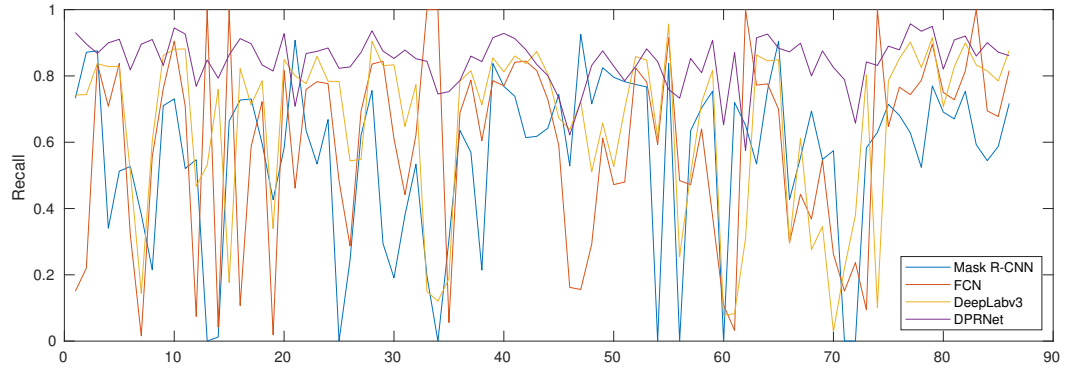
To further validate our DPRNet performance, we compared the models over the day, night, and foggy and cloudy datasets using a variety of validation metrics, as shown in Figure 5.2. The number of test images in both figures is 90, as shown on the horizontal axis in Figure 5.1. While this figure shows the results for each test image, Figure 5.2 presents the average outcomes for each validation metric across all test datasets. Furthermore, as the vertical axis in this figure represents the average value, it does not contain any zeros. All methods, with the exception of Mask R-CNN, exhibit acceptable performance on day and night datasets. Even with night precision, FCN is better than our proposed method. However, as discussed in Section 5.1, this metric can only partially convey the merit of a model individually, and it needs to be analyzed with the F1 score. Our proposed DPRNet outperforms the other rival methods by recognizing roughly all of the plume cloud pixels correctly. Most datasets are related to cloudy and foggy conditions and are frequently seen within image batches. The strength of our DPRNet is its robust performance in this case, which is of paramount importance in our application. The DPRNet could improve the recall metric by 66%, 58%, and 87% on average in cloudy and foggy conditions relative to FCN, DeepLabv3, and Mask R-CNN frameworks, respectively, which means that the proposed method is able to find the plume cloud regions appropriately, using L_5 . This capability produces high-quality image recognition with a more complicated mixture of plume clouds and the sky behind. These high recall values help us meet our research application requirement to identify the entire plume cloud stream for PR distance measurement.

To demonstrate the qualitative results of the proposed method, we show some visual results to compare competitive methods. Figure 5.3 depicts these recognition results. The first two rows represent the input images and their corresponding ground truths, respectively, and the other rows give the output of different models. We tried to visualize samples from all classes such that the first two images are related to cloudy/foggy conditions, the second two are from the nighttime dataset, and the last two are obtained from our daytime dataset. It is observed that DPRNet outperformed the other methods by attaining high accuracy of plume cloud localization and, consequently, correctly recognizing the desired smokestack plume cloud.

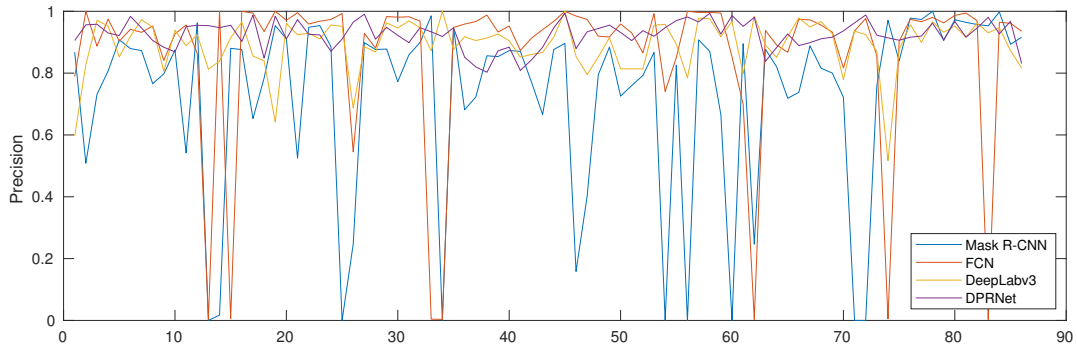
5.3 Image measurement

DPRNet gives plume cloud border detection and recognition. Then, we take advantage of the NBP image coordinates and the wind direction information from the meteorological tower to obtain plume rise real-life measurements through geometric transformations. Figure 5.4 illustrates the asymptotic curve for four plume cloud images and the automatically chosen point NBP, where the plume cloud reaches neutral buoyancy. Apart from the plume rise and plume rise distance values of each sample plume cloud,

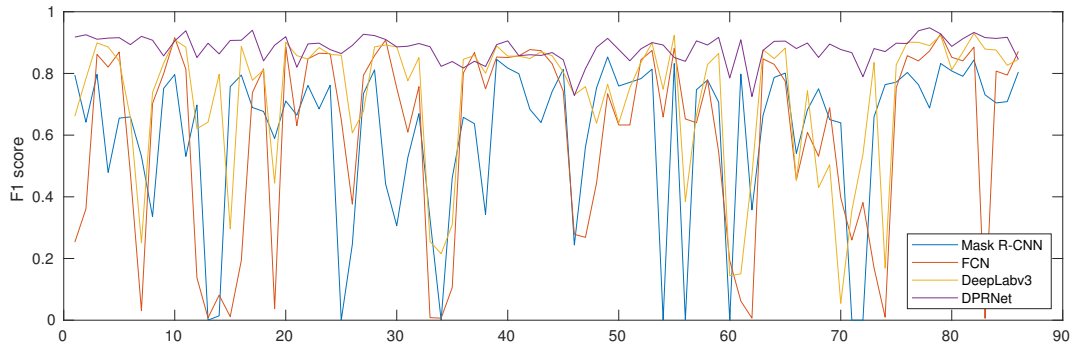
estimated by the proposed framework (tabulated in Table 5.2), the averaged hourly measured wind directions at the image sampling times are given as prior information of this study. The primary aim of this study is to assess the accuracy of Briggs equations by utilizing the images. These realistic plume rise and plume rise distance values within an extended period are required for future work, where our scope is a comparative study, and we will use image-based measurements to test and verify the Briggs equations.



(A) Recall

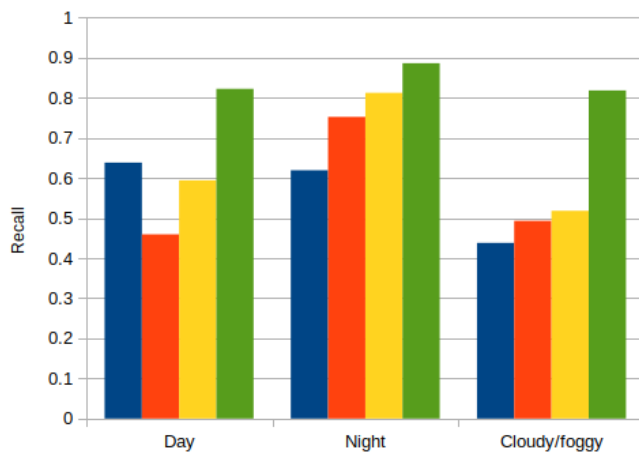


(B) Precision

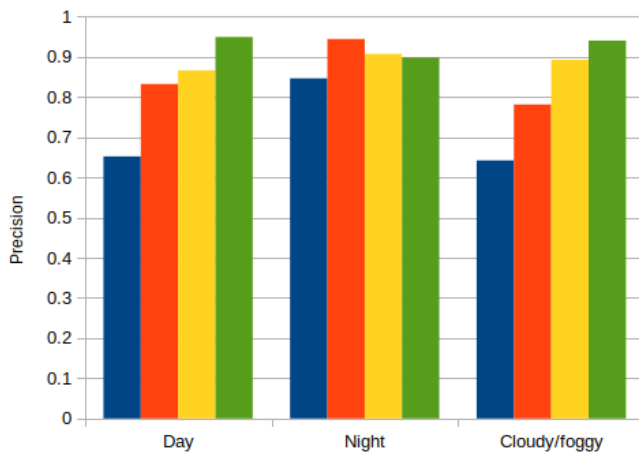


(C) F1 score metrics

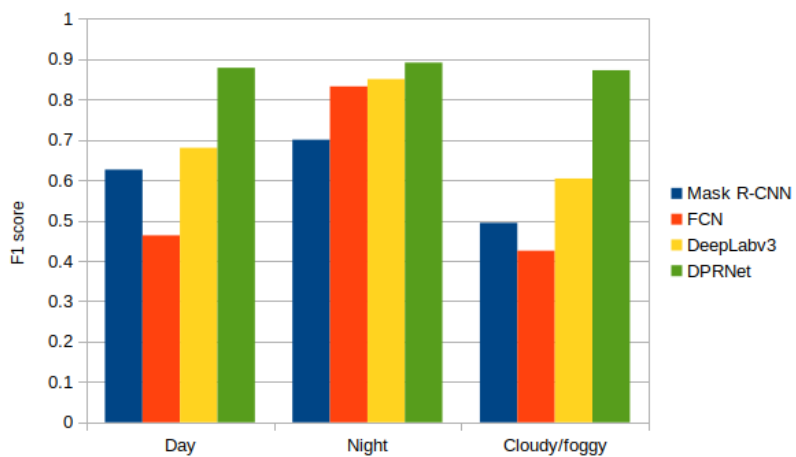
FIGURE 5.1: Performance of different methods regarding some test images. The horizontal axis in all three figures demonstrates the image number.



(A) Recall



(B) Precision



(C) F1 score

FIGURE 5.2: Detailed comparison of methods over three datasets employing three validation metrics.

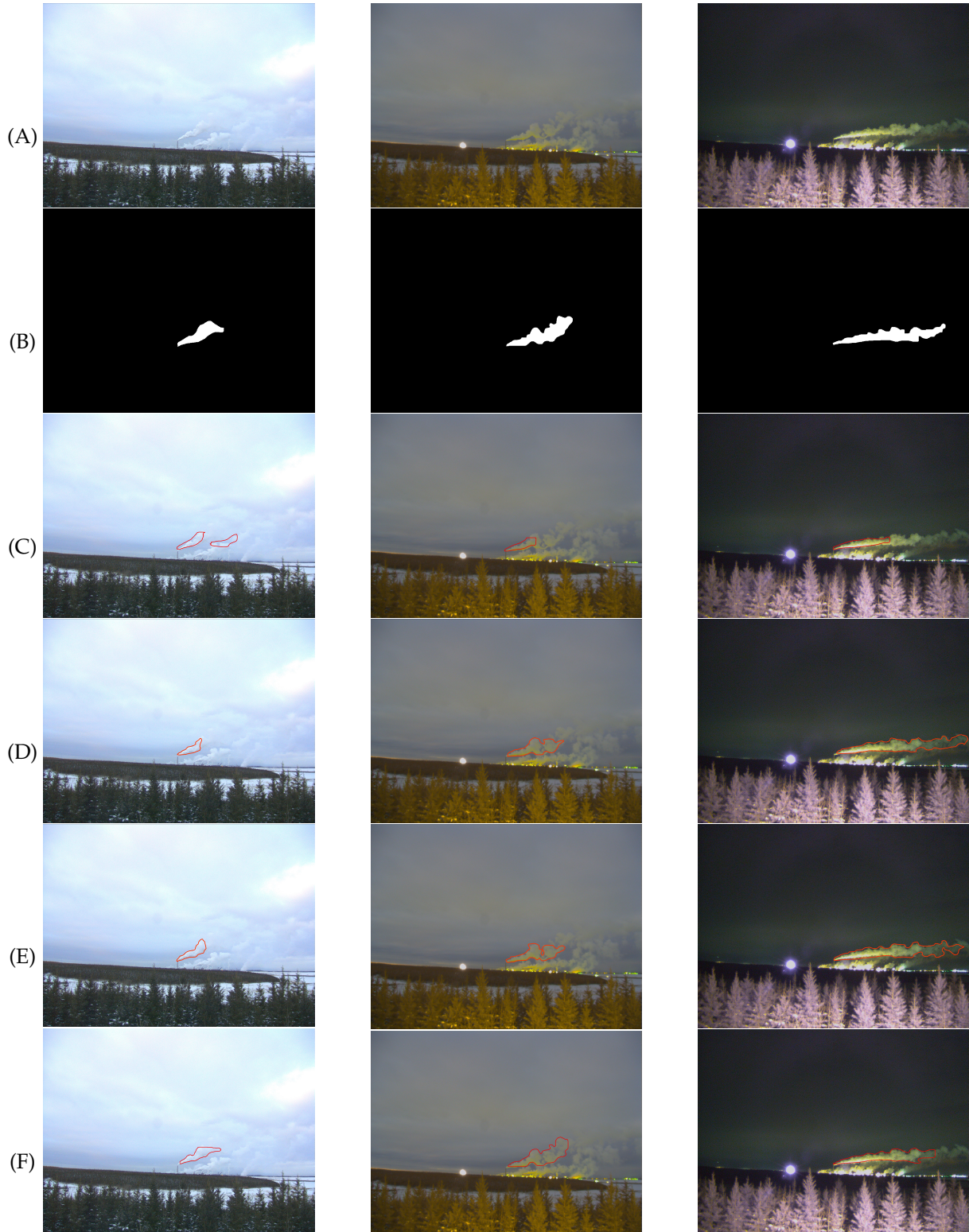


FIGURE 5.3: Qualitative results of recognition tasks listed as (A) Input image, (B) corresponding ground truth, (C) results of Mask R-CNN, (D) FCN, (E) results of DeepLabv3, and (F) results of DPRNet.

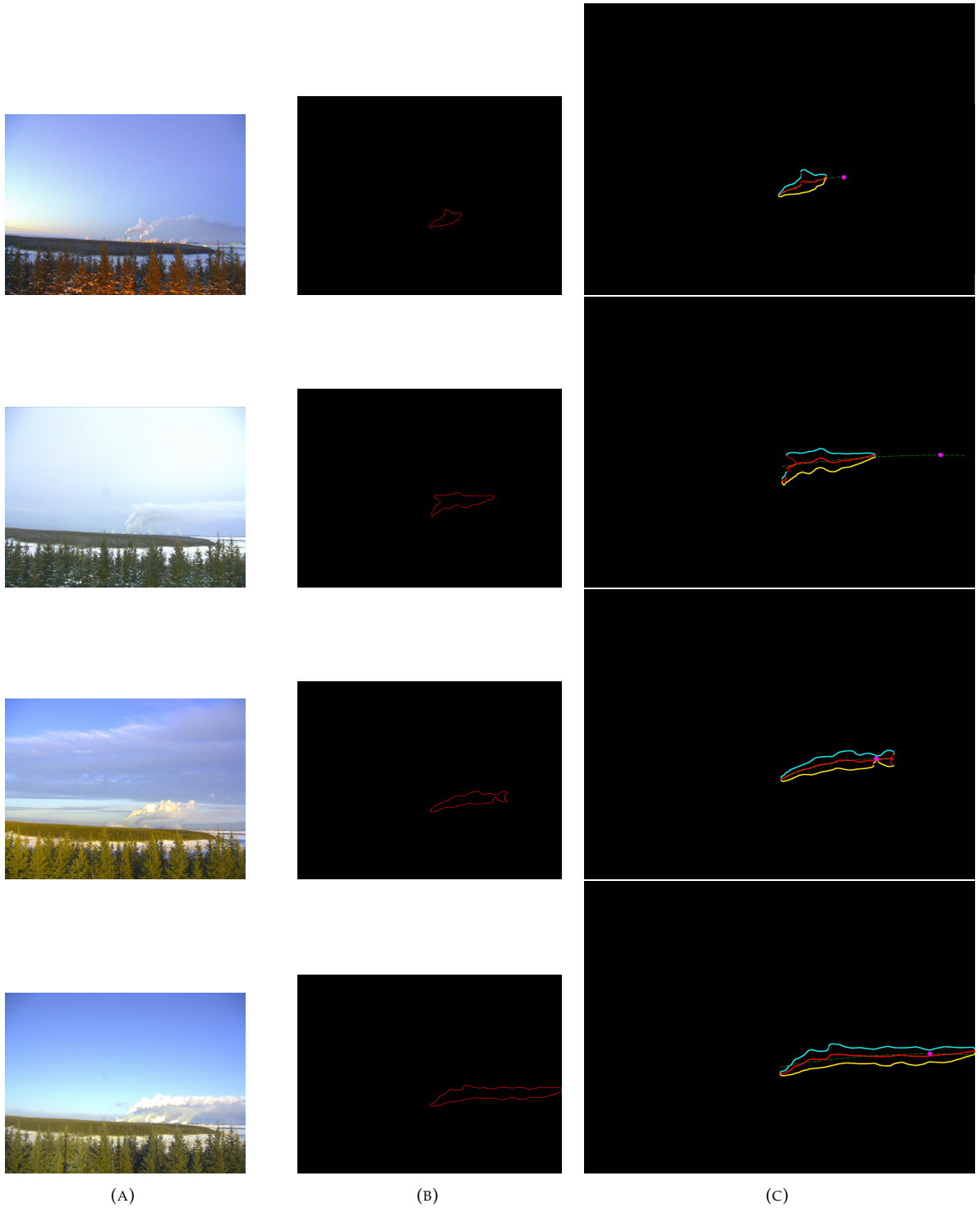


FIGURE 5.4: DPRNet and image measurement results. In column (c), the red curve represents the meandering of the plume cloud. The cyan and yellow lines, respectively, illustrate the upper and lower boundaries of the plume cloud. Green dashes show the asymptotic curve; the magenta asterisk is NBP. (a) Input image; (b) DPRNet result; and (c) Image measurement.

TABLE 5.2: Plume rise and plume rise distance values of each of the four plume images from Figure 5.4. φ is the wind direction angle from the North and θ is the wind direction angle at the smokestack location. ΔZ and D_P are respectively plume rise and plume rise distance.

Image	Date	Time	φ (deg.)	θ (deg.)	ΔZ (m)	D_P (m)
Photo1_2019_11_08_18_00_13	November 08, 2019	18-00-13	12.1	59.8	460	842
Photo1_2019_11_09_15_00_13	November 09, 2019	15-00-13	3.4	68.5	126	1707
Photo1_2019_11_14_10_00_16	November 14, 2019	10-00-16	10.4	61.5	338	1960
Photo1_2019_11_16_11_00_12	November 16, 2019	11-00-12	10.8	61.1	427	3143

6 Geometric Transformation and Validation

6.1 Geometric transformation

To compare the visual results with the real-world results, we need to perform calculations to obtain plume rise from the photos using the measured wind direction. Usually, two cameras are used to find an object in 3D space for a stereoscopic image. However, for an evolving and relatively homogeneous plume, it would be impossible to identify features to triangulate the 3D points. Hence, we use wind direction to constrain the location of the plume using only one camera. The main target here is the conversion from the location in the photograph to a real-life distance. In order to determine the plume rise height from the images, we need to know what direction the plume is blowing, which requires knowing the wind direction. Software such as Adobe Photoshop, AutoCAD, ArcGIS, MATLAB, and Adobe Premiere Pro work under a pixel coordinate system, while based on the physics of our problem, parameters are defined and measured in an image coordinate system. The differences between these two systems are the location of the origin, the direction and the scale of the axes. The origin of the pixel coordinate system is the top left corner of the top left pixel in the image, while the origin of the image coordinate system is the center of the image. Both coordinate systems represent the distance of the pixels within the image space and the Equation 6.1 shows the relation between these systems (Luhmann et al., 2006). Figure 6.1 demonstrates these two coordinate systems.

$$\begin{aligned} x &= [c - \frac{W}{2}] \\ z &= [\frac{H}{2} - r] \end{aligned} \tag{6.1}$$

where (x, z) represents a typical point position in the image coordinate system, (r, c) represents its position in the pixel coordinate system, and W and H show the Width and Height of the image, respectively.

Two important parameters in the calculations are Pixel Size (P) and Ground Sample Distance (G), which help us convert the image scale measurements to real scale measurements. P can be calculated via camera Field of View (α) and camera Focal Length (F) based on the camera specifications tabulated in Table 3.2. Figure 6.2 shows a schematic view of our imaging situation.

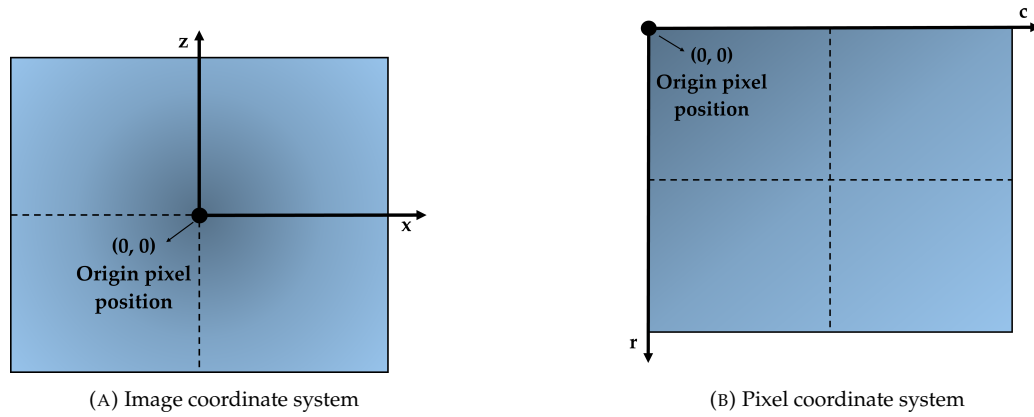
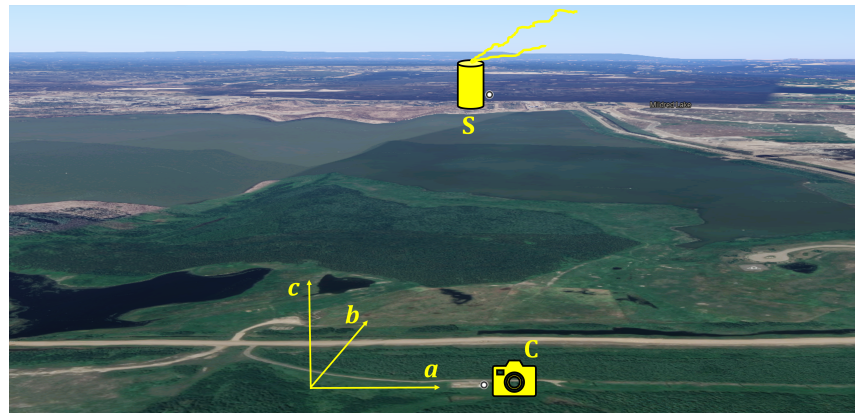
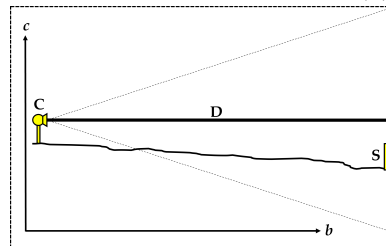


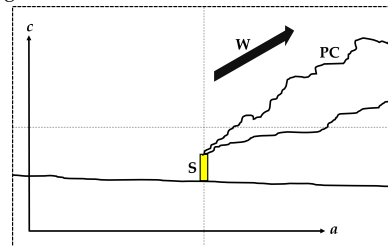
FIGURE 6.1: The coordinate systems.



(A) 3D view of the region



(B) Side view scheme



(C) Camera view scheme

FIGURE 6.2: Schematic view of our imaging situation. Camera station, C; distance between camera and smokestack, D; smokestack position, S; wind direction, W; and plume cloud, PC. The abc coordinate system is only for differentiating the side and camera views and is not used as a co-ordinate reference system.

If NBP is introduced as a point on the plume centerline in which the plume rise height happens (the point of maximum plume rise closest to the stack), the plume rise distance can be defined as the horizontal distance between the smokestack exit and the NBP. Figure 4.2 shows these definitions on a sample plume cloud image captured from the study region.

The Geometric Transformation module plays an important role in transferring the plume rise characteristics localized in a 2D image predicted by DPRNet to their corresponding positions in 3D world coordinates. The plume cloud characteristics that need to be transformed from 2D-to-3D are the plume rise and the plume rise distance. Figure 4.2 shows a schematic diagram explaining the geometry of the plume cloud characteristics shown in a camera view and its orthographic view. In the following, the geometry of the plume cloud characteristics is illustrated with respect to a camera in 3D (Figure 6.3), 2D top view (Figure 6.4), and 2D side view (Figure 6.5). As shown in Figure 6.3, the plume rise distance, D_P , is a distance measured between S and P on a horizontal plane made by two vectors, an optical axis, CY , and its orthogonal vector, CX , where C represents the camera centre's position in the world coordinate and (X, Y, Z) are the world coordinate vectors. Figure 6.5 defines the plume rise, ΔZ , which is an effective height measured between the smokestack's exit point, S , and the NBP, P .

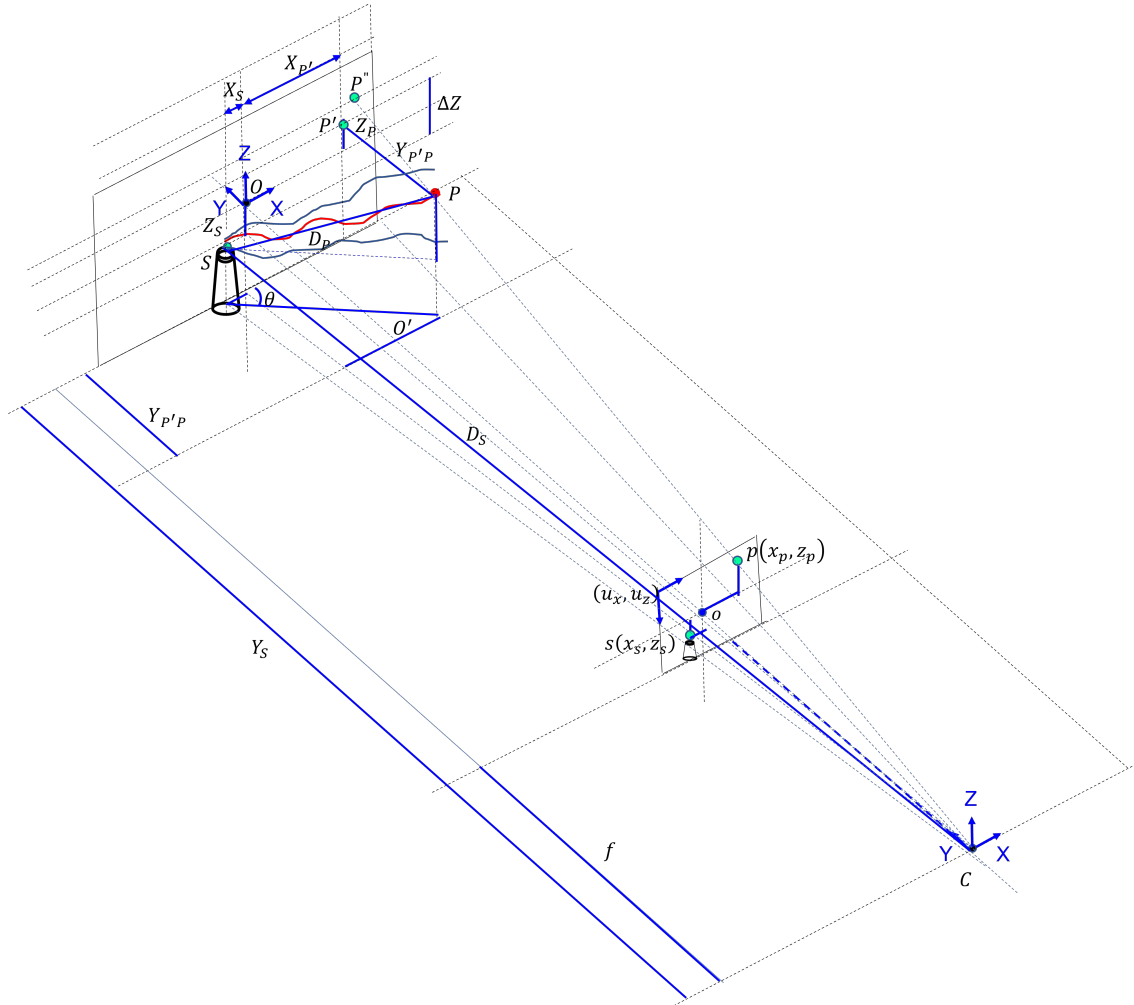


FIGURE 6.3: The schematic 3D view of the region in the real-world coordinates.

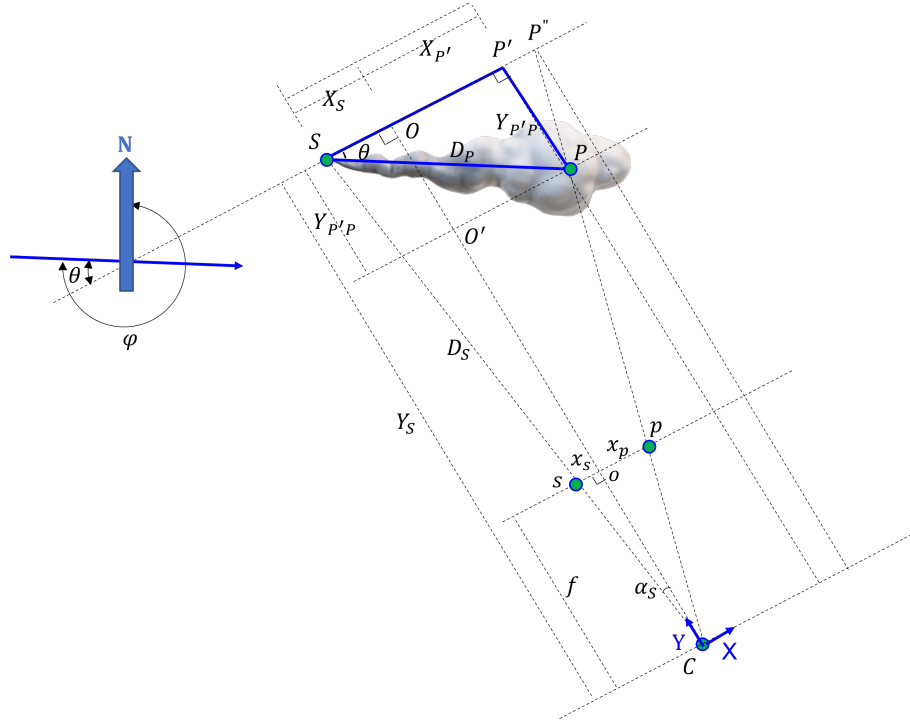


FIGURE 6.4: The schematic top view of the region in the real-world coordinates.

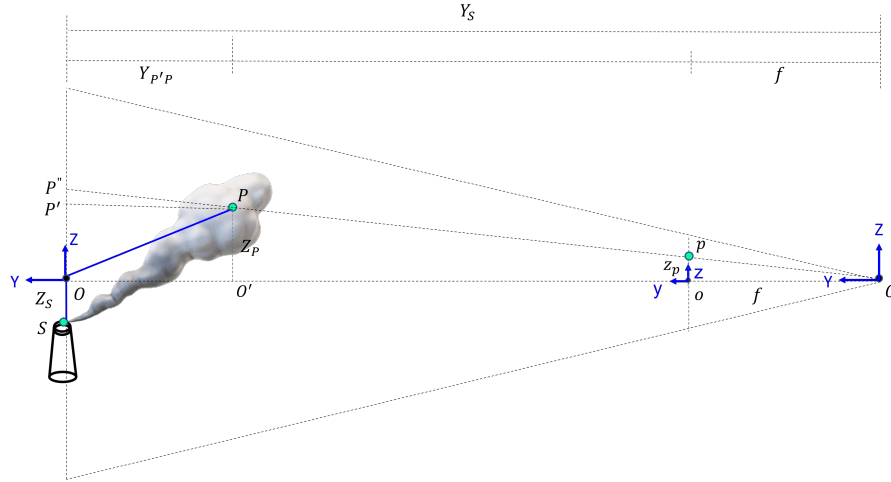


FIGURE 6.5: The schematic side view of the region in the real-world coordinates.

In this study, we assume that several parameters are known in advance. The distance between the camera and the position of the smokestack, D_S in Figure 6.4, is measured by Google Maps (<https://maps.google.com>), a web-based service providing detailed information about geographical regions and sites worldwide. Given the geographical location information about the camera and the smokestack, we measured D_S using Google Maps. In addition, we assume that the information about the wind direction is pre-known. For instance, $\varphi = 90^\circ$ is wind from the east, and $\varphi = 180^\circ$ is wind from the south. Finally, the camera's

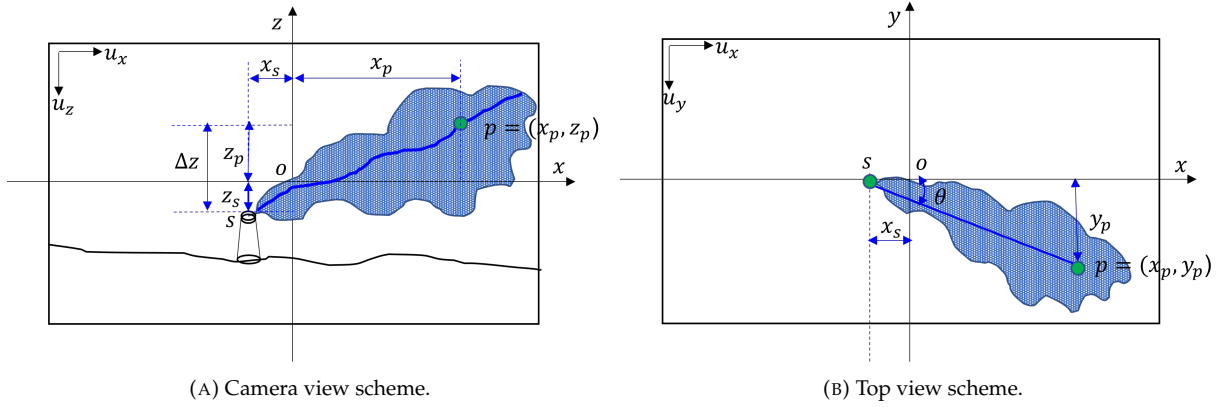


FIGURE 6.6: Smokestack location schemes. Smokestack exit, s ; and image center, o . PC centerline is represented with a blue line, and NBP image coordinates are shown with $p(x_p, z_p)$ and $p(x_p, y_p)$, respectively, in camera view and top view schemes.

intrinsic parameters, such as focal length, f , principal point's position, o and extrinsic camera parameters, including camera position with respect to the origin of the world coordinates, are shown in Figure 6.6.

With the known f and D_S , we can compute the position of the smokestack exit point S in the world coordinates. First, we can derive a geometric relationship between (x_s, y_s) and (X_S, Y_S) or (x_s, z_s) and (X_S, Z_S) using similarity triangle theorem with ΔCOS and ΔCos in Figures 6.4 and 6.5, respectively (Luhmann et al., 2006).

$$X_S = Y_S \frac{x_s}{f}, \quad (6.2)$$

$$Z_S = Y_S \frac{z_s}{f}. \quad (6.3)$$

Using the trigonometry theorem with ΔCos and ΔCOS in Figure 6.4, we can also compute the horizontal angle of the smokestack, α_s , and Y distance of the stack exit point, S , as below:

$$\alpha_s = \tan^{-1} \frac{x_s}{f} \quad (6.4)$$

$$Y_S = D_S \cos \alpha_s \quad (6.5)$$

Thus, we can calculate the position of the smokestack exit point, S , from ΔCOS in Figure 6.4, and Equations 6.2 and 6.3 as follows:

$$X_S = D_S \frac{x_s}{f} \cos \alpha_s, \quad (6.6)$$

$$Z_S = D_S \frac{z_s}{f} \cos \alpha_s. \quad (6.7)$$

Similarly, we can compute the position of the PR point, P . From $\Delta SPP'$, we can calculate the following:

$$\tan \theta = \frac{Y_{P'P}}{X_S + X_{P'}}. \quad (6.8)$$

Since two triangles, ΔCop and $\Delta CO'P$, are similar, we can derive:

$$\frac{x_p}{X_{P'}} = \frac{f}{Y_S - Y_{P'P}}, \quad (6.9)$$

And since θ is given from the weather station, we can calculate $X_{P'}$ by substituting Equation 6.8 in Equation 6.9 as follows:

$$X_{P'} = \frac{Y_S - X_S \tan \theta}{\frac{f}{x_p} + \tan \theta}, \quad (6.10)$$

$$Y_{P'P} = \frac{X_S f - Y_S x_p}{f \cot \theta + x_p}. \quad (6.11)$$

Also, we can compute the following PR ratio from Figure 6.5:

$$\frac{z_p}{Z_P} = \frac{f}{Y_S - Y_{P'P}}. \quad (6.12)$$

Thus, the PR height, Z_P , can be calculated as below:

$$Z_P = \frac{Y_S - Y_{P'P}}{f} z_p. \quad (6.13)$$

By substituting Equations 6.5 and 6.11 in Equation 6.13:

$$Z_P = \frac{z_p}{f} \frac{D_S \cos \alpha_s \sin \theta - X_S \cos \theta}{\frac{x_p}{f} \cos \theta + \sin \theta}. \quad (6.14)$$

Therefore, PR and PR distance can be calculated as follows:

$$\Delta Z = |Z_S - Z_P|, \quad (6.15)$$

$$D_P = \sqrt{(X_S + X_{P'})^2 + Y_{P'P}^2}. \quad (6.16)$$

6.2 Uncertainty analysis of the proposed image-based measurements

Identifying and quantifying the sources of uncertainties in image-based measurement procedures are critical for accurately interpreting results and making informed decisions. Therefore, validating the proposed method, including the plume cloud recognition and the geometric transformation, is essential to ensure that this approach is adequately accurate and reliable.

6.2.1 Sources of uncertainties

To carry out a comprehensive validation process here, we have identified various uncertainty sources, including instrumentation uncertainty, uncertainty caused by image distortion due to severe atmospheric conditions, and uncertainty of the feature extraction performed by the proposed method. This process allows us to identify inaccuracies during image analysis, which leads to improvement in our proposed algorithms.

To test the Equations 6.15 and 6.16 with known distances, we physically measured the location of the checkerboard origin in the lab (the distance from the camera to the checkerboard origin, simulating the distance from the camera to the smokestack exit) and the distance along the checkerboard to an object points on the last column of the checkerboard with a known pixel location (simulating the NBP affected by the known wind direction based on the checkerboard angle from the image plane). Then we use equations 6.15 and 6.16 to calculate the plume rise and plume rise distance. Finally, we compared the calculated values to the physically measured height and distance and repeated these steps for other images to get a representative average of the errors.

To extend this analysis for plume rise to the actual area with a larger scale in distances, the obtained lab errors should be propagated based on camera distance to the checkerboard origin in the lab, D_{lab} , and distance between the camera and the smokestack location in our real-world site, D_{site} . The error in these distances can arise from factors such as instrumental precision, calibration uncertainties, and inherent limitations in the measurement devices. The error propagation law provides a systematic way to estimate how errors in the input quantities affect the error in the final result. Based on the Thales theorem, the plume rise in the site (ΔZ_{site}) can be related to the plume rise in the lab (ΔZ_{lab}) as below,

$$\Delta Z_{site} = f(D_{site}, D_{lab}, \Delta Z_{lab}). \quad (6.17)$$

Therefore, the propagated error in the calculated image location can be expressed using the following formulas:

$$\delta_{\Delta Z_{site}}^2 = \left(\frac{\partial f}{\partial D_{site}}\right)^2 \delta_{D_{site}}^2 + \left(\frac{\partial f}{\partial D_{lab}}\right)^2 \delta_{D_{lab}}^2 + \left(\frac{\partial f}{\partial \Delta Z_{lab}}\right)^2 \delta_{\Delta Z_{lab}}^2, \quad (6.18)$$

where $\frac{\partial f}{\partial x}$ represents the partial derivative of the variable x , and δy denotes the variation of the variable, y .

Based on the lab simulation using different images, the average error of $\delta_{\Delta Z_{lab}}$ became 0.3 cm. Also, the average distance for D_{lab} is about 402 cm. Therefore, based on Equation 6.18 and the known distance of 5200 m for D_{site} , the plume rise propagated error in the actual space becomes about 3.9 m.

This approach provides a more accurate representation of the overall uncertainty in our measurements. It allows us to understand better how errors in the lab setting might propagate into the real-world scenario over the extended distance of approximately 5.2 km.

Uncertainty caused by image distortion due to severe atmospheric conditions

The atmosphere is a dynamic system, and its different effects, such as turbulence, heat haze, and severe precipitation, can cause uncertainties by covering the emitted plume cloud and distorting the captured images. Therefore, plume cloud identification and the corresponding measurements are potentially affected.

Uncertainty of the feature extraction performed by the proposed method

The automatic extraction of relevant information from images, performed by the proposed DCNN-based plume cloud detection and recognition method, involves various algorithms such as edge detection and image segmentation. The inaccuracies of these algorithms can raise uncertainties in locating plume cloud translucent boundaries and, consequently, other measures such as centerline and the NBP, which potentially affect the plume rise measurements using the proposed method. The accuracy of this approach was validated in Section 5.2 using different metrics and plots. The following section will quantify this uncertainty and the previous case caused by severe atmospheric conditions.

6.2.2 Uncertainty quantification via image inspection

Image inspection or validation with a ground truth dataset is a visual approach that involves manually inspecting the image results of the proposed method. For this purpose, we compare our results with ground truth data obtained through annotation. This comparison allows us to assess the reliability of the proposed approach and helps us understand the restrictions and possible biases in our image-based measurements. As discussed in Section 4.2, the last step of the image measurements is finding the NBP on the image, which leads to calculating the plume rise and plume rise distance after geometric transformations. Therefore, comparing this point acquired from our measurements to the coordinates of this point obtained from our ground truth data can validate our method of image-based measurements. For this purpose, we have selected 300 images, samples of which are displayed in Figure 6.7. In this figure, two different positions of the NBP obtained from our measurements and visual inspection are depicted, whose differences are considered as a criterion to validate our image-based measurements. Although none of these are ground truth, the visual inspection has strong biases (depending on who is doing the inspection and how they define a plume cloud). As we compare the coordinates of the NBP points, Euclidean distance is calculated between the corresponding points to quantify the difference.

Now, we calculate the difference in the NBP positions between our image-based measurements and the ground truth data. Euclidean distance will help us quantify the biases between these two approaches. Also, we take advantage of common statistics tools, Mean Square Error (MAE) and Root Mean Square Error (RMSE), to validate our comparison. Figure 6.8 visualizes this comparison leveraging two plots, including a scatter plot and an error histogram.

As seen in Figure 6.8a, an MAE of 4.9, suggests an average deviation of approximately 5.0 pixels between our image-based measurements and visual inspection. An RMSE of 5.6 indicates the spread of errors, while the larger errors significantly affect its value. Moreover, Figure 6.8b indicates that these errors mostly occur below 5 pixels, which is expected from the statistic metrics above. Although these errors seem reasonable for plume cloud recognition at about 5 km, considering the atmospheric conditions and the complexities of translucent plume cloud borders, we need to calculate the actual percentage error (the uncertainty) for the plume rise and plume rise distance in pixels.

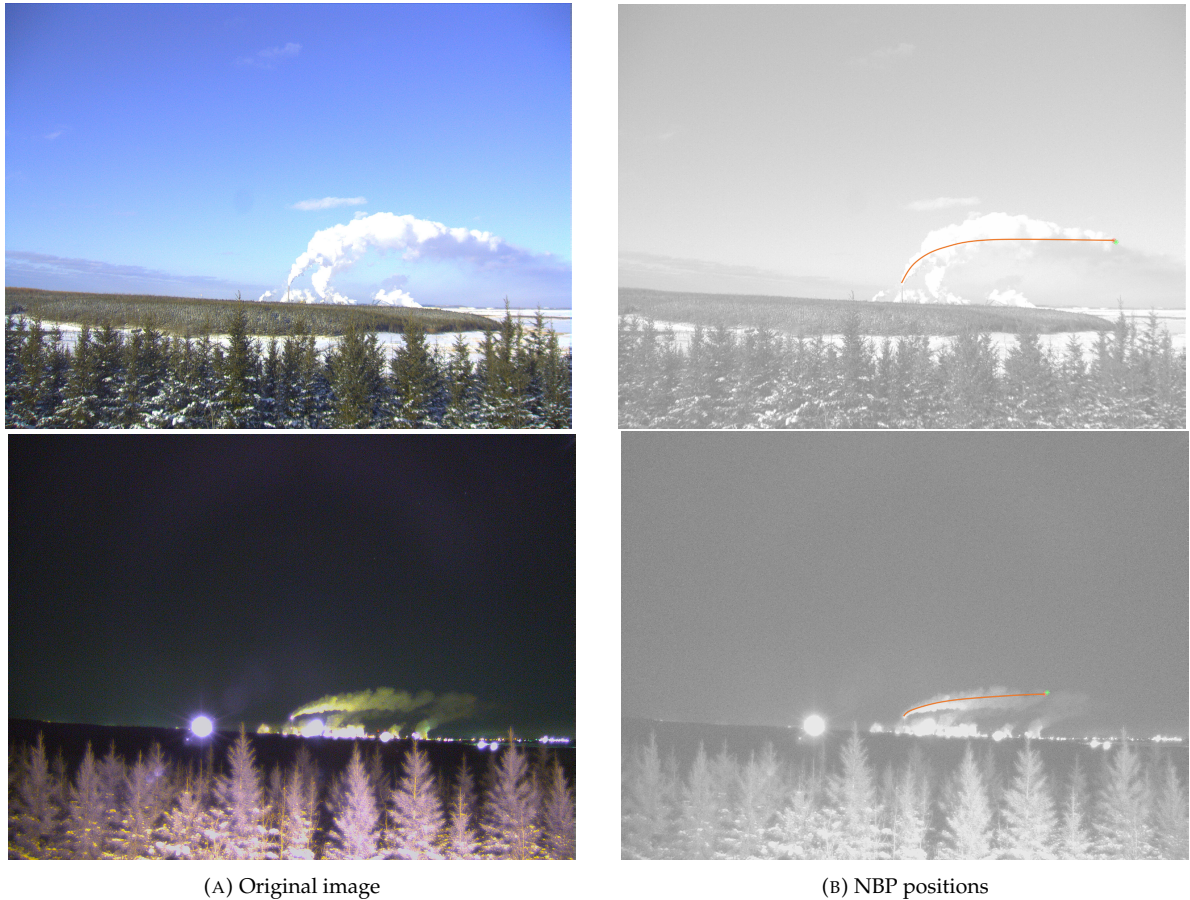
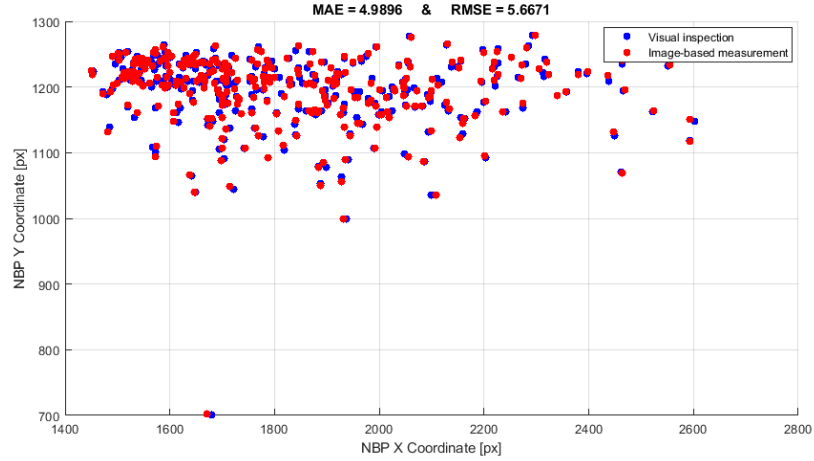
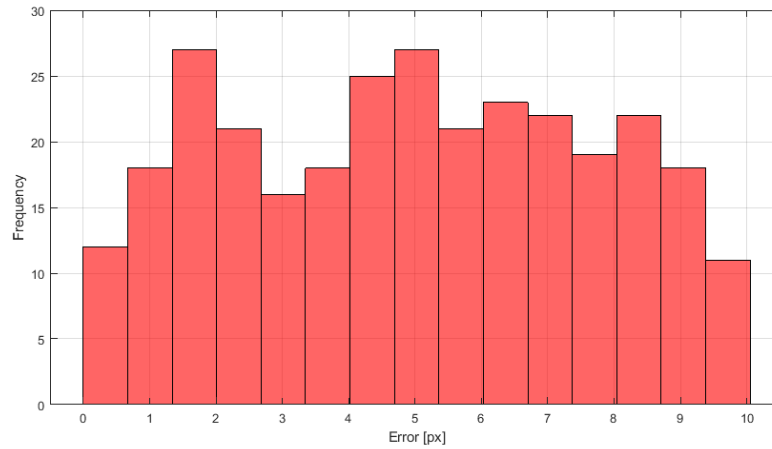


FIGURE 6.7: Sample original image and the NBP positions obtained from our measurements and from the visual inspection. The red star displays the result given from our image-based measurements, and the green star shows the results obtained from image inspection. The orange curve represents the asymptotic fitting curve of the plume cloud until it saturates on the NBP.



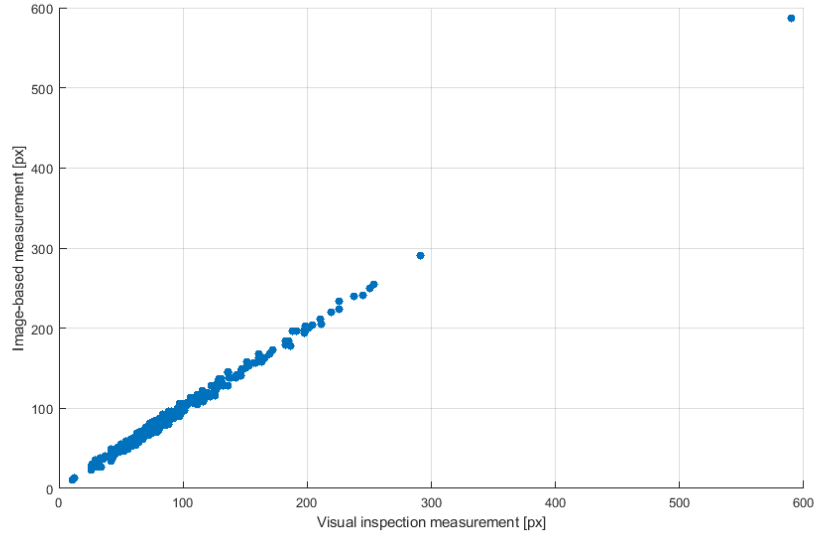
(A) The NBP coordinates obtained from our image-based measurement vs. visual inspection



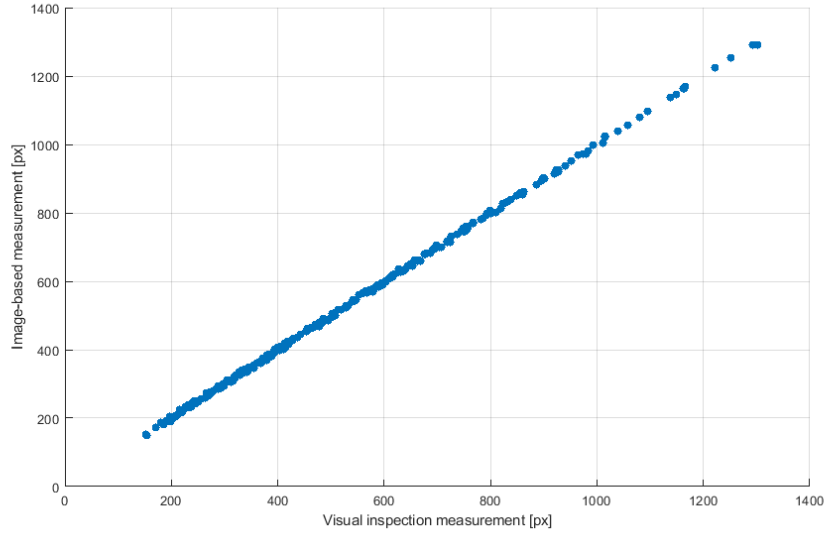
(B) Error histogram of the NBP locations

FIGURE 6.8: Comparison of the image-based measurement NBP locations with the NBP locations from the visual inspection.

To assess this uncertainty in plume rise measurements and understand the deviation between two measurements, we utilized the visual inspection values as the reference, while the obtained values from our image-based measurement are considered as the measured values. Figure 6.9 shows the relation between these measurements (for both plume rise and plume rise distance) using a single scatter plot. Table 6.1 displays the assessment of accuracy using MAE and RMSE. MAE quantifies the average absolute difference between the plume rise and plume rise distance values between the visual inspection and image-based measurements. RMSE gives the root mean squared difference between two measurements, considering the magnitude and direction of deviations. We further normalized these errors to percentage values to help us understand the differences relative to the average value of the visual inspection measurements.



(A) Plume rise values obtained from our image-based measurement vs. visual inspection



(B) Plume rise distance values obtained from our image-based measurement vs. visual inspection

FIGURE 6.9: Comparison of the image-based plume rise measurements with these measurements from the visual inspection.

The MAE percentage of 3.25% and the RMSE percentage of 4.12% indicate that the deviations between the two measurements are relatively low. This suggests that our image-based method can produce measures very close to the visually precise values. The even lower MAE percentage of 0.61% and RMSE percentage of 0.77% for the plume rise distance values demonstrate an even higher accuracy and consistency level than the plume rise values. The low MAE and RMSE percentage errors in both cases reveal that our measurements agree with the reference measurements, which emphasizes the reliability of our measurement method.

TABLE 6.1: Accuracy assessment of the image-based measurements of plume rise and plume rise distance vs. the visual inspection measurements, using two metrics of MAE and RSME, as well as normalized percentage values of these errors.

Plume rise			
Average MAE [px]	Average RSME [px]	MAE percentage [%]	RSME percentage [%]
3.07	3.98	3.25	4.12
Plume rise distance			
Average MAE [px]	Average RSME [px]	MAE percentage [%]	RSME percentage [%]
3.24	4.11	0.61	0.77

For the purpose of this research, the level of uncertainty seems to be acceptable, considering the impact of the uncertainties above on the final results of our application. For a higher level of accuracy, further improvement of the measurement process, such as camera calibration refinement, computer vision algorithm enhancement, and validation with more reliable data sources (e.g. satellite images), can be considered. Finally, it is crucial to consider a trade-off between obtaining the required accuracy and feasibility, especially in such applications dealing with plume cloud identification processes with very complicated borders from a distance of about 5 km.

7 Investigation of Briggs' Equations

Previously in Chapter 4, we talked about the proposed DPRNet (Koushafar, Sohn, and Gordon, 2023), a visual data-driven system developed in the framework of DCNNs (Simonyan and Zisserman, 2014; He et al., 2016), to recognize the plume cloud for measuring plume rise. The DPRNet is used for noise reduction, plume recognition, plume rise model prediction and plume tracking. The designed experiments and their results led to the development of DPRNet for validating conventional plume rise models and provided scientific grounds for developing a new physical model.

In this chapter, the accuracy of the Briggs algorithms is evaluated using the proposed DCNN-based framework for measuring plume rise. The Briggs plume rise algorithms are driven by stack parameters and meteorological observations. In our companion study (Gordon et al., 2018), the Briggs parameterizations were compared with aircraft observations of emitted SO_2 , as well as the Briggs another model for irregular stability profiles (Briggs, 1984).

7.1 Briggs plume rise parameterization in GEM-MACH

This section's plume rise (Δh) calculations are performed based on Briggs parameterizations formulated by the GEM-MACH model in ECCC. Using this model, nine variables drive these calculations, which are tabulated in Table 7.1.

TABLE 7.1: The variables used for plume rise calculations in GEM-MACH model.

No.	Variable	Symbol	Unit
1	Smokestack height	h_s	m
2	Smokestack exit temperature	T_s	K
3	Smokestack emission volumetric flow rate	V	$\text{m}^3 \text{s}^{-1}$
4	Ambient temperature at the smokestack height	T_a	K
5	Wind speed at the smokestack height	U	m s^{-1}
6	Surface temperature	T_{surface}	K
7	Obukhov length	L	m
8	Boundary-layer height	H	m
9	Friction velocity	u_*	m s^{-1}

These parameters and their calculation/measurement processes are described more in the following:

1) Smokestack height

It defines the physical height of the smokestack exit from the ground level. This variable can affect the plume rise in a way that the higher the smokestack, the more different atmospheric interactions. Smokestack height can be measured using different techniques, such as GPS, Laser range finder, LiDAR, manual measurements and aerial photography.

2) Smokestack exit temperature

This is the temperature of the plume cloud as it leaves the smokestack exit. This variable can influence the plume rise by affecting the buoyancy of the plume cloud. This temperature is measured using various thermometers, thermocouples, and resistance temperature detectors.

3) Smokestack emission volumetric flow rate

It represents the initial amount of the plume cloud volume released from the smokestack exit per unit of time. This concept is measured using different tools such as pitot tubes, ultrasonic flow meters, hot wire anemometers, and volumetric flow meters.

4) Ambient temperature at the smokestack height

This temperature refers to the temperature of the surrounding air at the height of the smokestack exit location. This variable can influence the plume rise by affecting the buoyancy of the emitted plume cloud. This temperature is measured considering various factors, such as temperature lapse rate, atmospheric stability, and thermal radiation, which affect atmospheric conditions.

5) Wind speed at the smokestack height

It refers to the speed of the air currents at the height of the smokestack exit location. Wind speed is one of the critical meteorological factors affecting plume rise by impacting the plume cloud's spread, dilution, and direction. Moreover, it influences how far the plume cloud will travel. Some of the tools for measuring wind speed are anemometers, meteorological towers, LiDAR, Sodar, and wind profiling radars.

6) Surface temperature

The Earth's surface temperature is measured at or near ground level. Atmospheric stability, local weather conditions, time of day, and terrain are the factors that affect this temperature, which consequently influences the plume rise. This temperature is measured using different instruments, such as ground-based and infrared thermometers, remote sensing satellites, soil temperature probes, and microwave radiometers.

7) Obukhov length

It characterizes the atmospheric boundary layer stability. This variable measures the balance between turbulence driven by buoyancy due to temperature variations and turbulence caused by the changes in wind speed near the Earth's surface. The Obukhov length calculations require detailed consideration of atmospheric stability, surface conditions, and accurate temperature and wind speed measurements.

8) Boundary-layer height

It defines the height above the Earth's surface at which the atmospheric characteristics, such as temperature and wind, start to differ significantly from their surface values. This height can change day and night due to solar radiation, surface heating, and atmospheric stability. This variable is measured using radiosondes, LiDAR, Sodar, ceilometers, and radio wind profiles.

9) Friction velocity

It refers to the effect of surface roughness on the wind velocity near the Earth's surface. Friction velocity impacts the vertical dispersion and behaviour of the plume cloud emitted from the smokestack exit.

These stack parameters affect the emitted plume cloud and its upper and lower boundaries, increasing it above the smokestack height. In this model, it is assumed that the plume cloud has reached neutral buoyancy when it is distributed through the area.

7.1.1 Continuous Emissions Monitoring System (CEMS)

CEMS (Protection, 1998; ECCC, 2017) is a regulatory requirement for industries that emit environmental pollutants, including our research area, Northern Alberta, Canada. CEMS measures, records, and reports air emissions from industrial processes in power plants, refineries, and other extensive facilities. The data collected by CEMS are crucial to evaluating compliance with environmental regulations, tracking emissions, and coping with the impact of industrial activities on air quality and public health criteria. The CEMS operators in Alberta are required to collect and report the CEMS data to the Government of Alberta Ministry of Environment and Parks (AEP) and Alberta Energy Regulator (AER) (ECCC and AEP, 2016). This process contains four steps as follows:

1. **Installation:** The Syncrude industry has CEMS equipment installed at appropriate locations of emission points. Different sensors, analyzers, and data-recording devices continuously measure and record emissions in these systems.
2. **Data collection:** CEMS devices continuously monitor the pollutants emitted from industrial processes, record the collected data at frequent intervals, and store them electronically.
3. **Data validation:** Quality assurance processes are applied to the collected data to ensure its accuracy and reliability.
4. **Data report:** CEMS operators regularly report monitoring information to the Alberta Government. These reports deliver detailed information on the emissions of various pollutants electronically.

These collected data include pollutant concentrations, emission rates, time series, and operational data such as flow rate, temperature, and pressure. These data are retained in an emission inventory system, which collects, organizes, and manages the database. An emission inventory system aims to provide a detailed emissions report, which is essential for air quality management. Then, a set of procedures and methodologies are used to convert the raw data, kept in the emission inventory system, into meaningful and usable emission estimates. This system is called the emission processing system. To have access to the industrial reports and the government database for our study, we followed a procedure of five steps below:

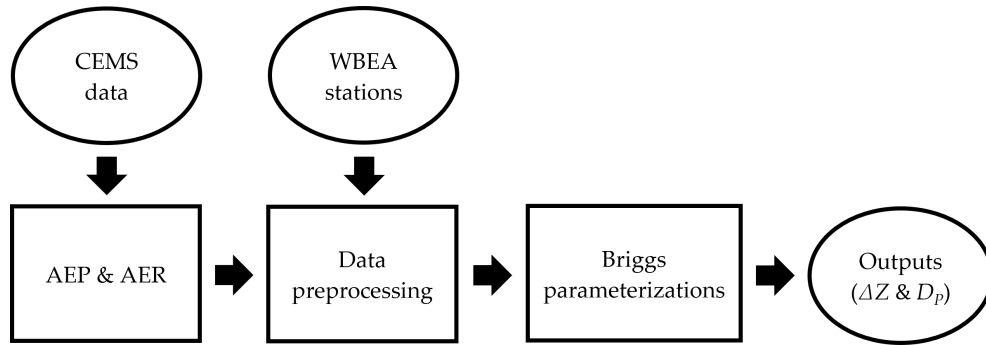


FIGURE 7.1: Data flowchart for running Briggs parameterizations. ΔZ and D_p represent the plume rise and the plume rise distance, respectively.

1. **Access request:** We needed to submit a formal request to AEP and AER to let them know our interest in obtaining CEMS data.
2. **Sharing agreements:** We were required to enter into data sharing agreements or meet specific data access requirements.
3. **Usage purpose:** We outlined the purpose of our research and how we aim to utilize the CEMS data.
4. **Data privacy:** Some agreements were to be signed regarding some privacy guidelines.
5. **Data format:** Among different data types, we could access the requested specific format of the CEMS data upon approval. The data is often in electronic files, which can be handled using various software.

Figure 7.1 illustrates the process of obtaining CEMS data to be applied to the GEM-MACH model. As seen in this figure, we also take advantage of the meteorological data coming from the WBEA monitoring stations, which is discussed in the following section.

7.1.2 WBEA weather stations

The WBEA employs a network of weather stations in the Wood Buffalo region of Alberta, Canada, which are designed to monitor various meteorological parameters to assess the region's environmental conditions and air quality. These weather stations provide helpful data for multiple purposes, such as evaluating the effect of industrial activities on air quality and understanding atmospheric processes. This data is available from over 25 continuous air monitoring stations, six meteorological towers and 50+ integrated monitoring sites. The data type helps the researchers find the right place to obtain, download, or view the data. The network map of these stations is displayed in Figure 7.2 and Figure 7.3 depicts the locations of the adjacent stations to our under-study area.

The most critical category is the continuous data acquired from ambient air monitoring stations and meteorological towers. In this category, both the data interval and level of review are of concern. The access links for these data are tabulated in Table 7.2. Mostly, the data here are available in 5-minute and 1-hour periods and are divided into two primary levels (<https://wbea.org/data/network-map-station-data>):

- **Level 1 (L1):** This data is raw air quality monitoring data depicted directly from monitoring instrument output signals. Although L1 data generally indicates the ambient conditions, qualified technicians must review and validate it. Therefore, the final processed data may vary from the L1 values shown on the site.
- **Level 2 (L2):** This data is quality-controlled and processed monthly, including a summary and validation of the data. WBEA further reviews data as an acceptance test before generating monthly compliance reports.

TABLE 7.2: The access links to find continuous data of the WBEA weather monitoring stations (Reproduced from <https://wbea.org/data/network-map-station-data>).

Continuous Level 1 Data	
Link	Description
https://cdv.wbea.org/wbea.html	Used for multiple parameters and stations
http://wbea.org/data/station-datanetwork-map	Used for the most recent hour at a station
Continuous Level 2 Data (available 30 days following month end)	
Link	Description
https://wbea.org/data/continuous-monitoring-data	Used for validated data (historical data)

The weather stations commonly measure meteorological variables, such as below,

- **Temperature** is a primary meteorological parameter, and it is measured in degrees Celsius.
- **Wind speed and direction** deliver information about air movement in the atmosphere, and they are measured, respectively, in meters per second and degrees.
- **Relative humidity** represents the amount of moisture in the air relative to its maximum capacity at a provided temperature, and it is expressed as a percentage.
- **Atmospheric pressure** indicates the weight of the air above a particular point, and it is measured in hectopascals or millibars.
- **Precipitation data** includes rainfall and snowfall measurements necessary for local hydrological and meteorological processes.
- **Solar radiation data** explains how much solar energy reaches the Earth's surface, which is used in energy balance and weather patterns.
- **Visibility measurements** help estimate the clearness of the atmosphere and can be leveraged in air quality safety applications.

The data collected from WBEA weather stations is used as input for atmospheric dispersion models like the GEM-MACH model, in which accurate meteorological information is essential for accurate modelling, reliability of the simulations, validity of the air quality assessments and environmental impacts.



FIGURE 7.2: The network map of the weather stations in the region (Reproduced from <https://wbea.org/data/network-map-station-data>). The under-study area is illustrated by a red rectangle. The ambient air monitoring stations are shown in blue spots; green spots are the edge plots, and purple spot shows meteorology towers.



FIGURE 7.3: The location of the following (<https://wbea.org/data/network-map-station-data>): Syncrude smokestack with a red circle (Lat:57.041, Long:-111.616), Mildred Lake air monitoring station with a yellow circle (Lat:57.050, Long:-111.564), Buffalo Viewpoint air monitoring station with a green circle (Lat:56.996, Long:-111.594), Lower Camp meteorological tower with a blue circle (Lat:57.026, Long:-111.500), and Mannix air monitoring station with a magenta circle (Lat:56.967, Long:-111.482). Mildred Lake air monitoring station is the closest station to the smokestack location, and our original wind direction information is obtained from this station.

7.1.3 Plume rise variables for our study

This section details the methods employed in our study for calculating and/or obtaining (from CEMS or WBEA stations) the variables tabulated in Table 7.1, which are used for plume rise calculations by Briggs parameterizations.

Smokestack height (h_s), exit temperature (T_s), and emission volumetric flow rate (V):

These variables were obtained through an established formal request for comprehensive stack information from AEP and AER collected by CEMS, following regulatory compliance standards and protocols. This hourly data was concurrent with the time of plume imaging, which is used in the plume rise comparison procedure in Section 7.3. The government is frequently approached for detailed documentation of the industries' emissions sources, including stack specifications, in order to keep the emission information accurate and up-to-date.

Temperature (T_a) and wind speed (U) at the smokestack height and surface temperature ($T_{surface}$):

We leveraged the WBEA Lower Camp meteorological tower and Mannix meteorological tower (Figure 7.3) data to access this information. Lower Camp Tower provides temperature, wind speed, and wind direction measurements at 20, 45, 100, and 167 m heights. Mannix Tower provides temperature measurements at heights of 2 (near-surface), 20, 45, and 75 m and wind speed and direction measurements at heights of 20, 45, 75, and 90 m. These data are provided as the average value of 1-hour measurements.

Obukhov length (L), boundary-layer height (H), and friction velocity (u_*):

Obukhov length, boundary-layer height, and friction velocity were calculated based on the temperature and wind speed profiles at different heights over the ground. These variables all are calculated to drive the Briggs plume rise parameterizations (Equations 7.7 - 7.12) regarding observations from two WBEA meteorological towers (Lower Camp and Mannix), on which anemometers and temperature sensors are installed at various heights between 2 and 167 m.

For the calculation process, we need to classify the atmospheric stability conditions by determining the stability parameter S . This parameter is obtained using the bulk Richardson Number (Garratt, 1994), R_i ,

$$R_i = \frac{gz_h}{\theta} \frac{\Delta\theta}{\Delta U^2}. \quad (7.1)$$

In this relation, $\Delta\theta$ is the potential temperature, z_h is the height range between the lower and higher measurement locations, and ΔU is wind speed differences within z_h . The z_h value for the Lower Camp and Mannix towers are 147 and 55 m, respectively. Then, the stability parameter was developed as (Kaimal and Finnigan, 1994)

$$z/L = \begin{cases} R_i, & R_i < 0, \\ \frac{R_i}{1 - R_i/R_{ic}}, & 0 < R_i < R_{ic}, \\ +\infty, & R_i > R_{ic}. \end{cases} \quad (7.2)$$

Here, R_{ic} is the critical Richardson number and is considered 0.25, which is the average of the reported values in Mahrt (1981). Now, Obukhov length could be obtained from the stability parameter relation as

$$L = \frac{z_{max}}{z/L}, \quad (7.3)$$

where z_{max} is the highest measurement height, which is 167 m for the Lower Camp Tower and 90 m for the Mannix Tower.

Based on Mahrt (1981), boundary-layer height in stable or unstable conditions is formulated as

$$H = \frac{R_i T_{surface}}{g} \frac{U_H^2}{\theta_H - \theta_{surface}}, \quad (7.4)$$

where g is the gravitational acceleration, U_H is wind speed at the boundary-layer height, θ_H is potential temperature at this height, and $\theta_{surface}$ is the potential temperature near surface.

Using information from the Lower Camp Tower, we could determine the H value based on Equation 7.4. However, we substituted U_H and θ_H , respectively, with $U_{z_{max}}$ and $\theta_{z_{max}}$, since we might not be available to measure wind and temperature at the boundary-layer height.

Finally, based on a semi-empirical log wind speed profile in Garratt (1994), the friction velocity is determined as

$$u_z = \frac{u_*}{\kappa} [\ln(\frac{z}{z_0}) - \Phi], \quad (7.5)$$

where κ is the Von Kármán constant (≈ 0.41), u_z is the wind speed at the measurement height of z , and z_0 is the surface roughness length. Typically obtained from local observations, this length defines how much the surface impacts the wind flow and depends on the surface type, including urban, rural, and water. For each tower, z_0 was considered as the median value of the hourly profiles, which were determined individually by the least-square method. This value was 1.5 m for the Lower Camp Tower and 0.75 m for the Mannix Tower. z_0 then can be used in Equation 7.5 to calculate u_* , while wind speed at the highest location is employed for u_z . Φ denotes the stability parameter as

$$\Phi = \begin{cases} 2 \ln \left(\frac{1}{2}(1 + x_0) \right) + \ln \left(\frac{1}{2}(1 + x_0^2) \right) - 2 \tan^{-1}(x_0) + \frac{\pi}{2}, & z/L < 0, \\ -5 z/L, & z/L > 0, \end{cases} \quad (7.6)$$

where $x_0 = (1 - 16 z/L)^{1/4}$.

Some approximations here, such as ignoring heat flux measurement in stability formulation and boundary-layer height estimation based on near-surface measurements, can impact the u_* estimation process and may cause high uncertainty in these calculations. To address this concern, an uncertainty analysis has been performed in Gordon et al. (2018) to test the sensitivity of the plume rise to the three variables of Obukhov length, boundary-layer height, and friction velocity.

7.1.4 Plume rise formulae

As we discussed in Section 2.1.2, plume rise calculations are different because the atmospheric condition is neutral, stable, or unstable. Briggs (1984) calculated the buoyancy flux, F_b as

$$F_b = \begin{cases} \frac{g}{\pi} V \frac{(T_s - T_a)}{T_s}, & T_s > T_a, \\ 0, & T_s \leq T_a, \end{cases} \quad (7.7)$$

where g is the gravitational acceleration, V is the smokestack emission volumetric flow rate, and T_s and T_a , respectively, are the smokestack exit temperature and the ambient temperature at the smokestack height. Based on Equation 2.3, the stability parameter, S , is obtained as,

$$S = \frac{g}{T_a} \left(\frac{dT}{dz} + \frac{g}{c_p} \right), \quad (7.8)$$

where z is the coordinate of the height and $c_p = 1005 \text{ J K}^{-1} \text{ kg}^{-1}$. Also $\frac{dT}{dz} = (T_a - T_{surface})/h_s$, in which a minimum value is set at -5 K km^{-1} (i.e. $S \geq 0.047/T_a$).

And finally, the convective velocity, H_* , is defined as (Briggs, 1985)

$$H_* = -2.5 u_*^3 / L, \quad (7.9)$$

where u_* is the friction velocity and L is the Obukhov length. Now, based on different atmospheric conditions discussed in Briggs (1984), plume rise formulations are described as follows:

- **Neutral atmospheric conditions** ($L > 2h_s$ or $L < -0.25h_s$)

The plume rise is obtained from the minimum value of two Briggs formulations mentioned in Sharf et al. (1993) and Byun (1999):

$$\Delta h = \min \left[39 \frac{F_b^{3/5}}{U}, 1.2 \left(\frac{F_b}{u_*^2 U} \right)^{3/5} (h_s + 1.3 \frac{F_b}{u_*^2 U})^{2/5} \right]. \quad (7.10)$$

- **Stable atmospheric conditions** ($0 < L < 2h_s$ or $h_s \geq H$)

The plume rise is calculated as (Briggs, 1984)

$$\Delta h = 2.6 \left(\frac{F_b}{SU} \right)^{1/3} \quad (7.11)$$

- **Unstable atmospheric conditions** ($-0.25h_s < L < 0$)

The plume rise is taken as the minimum of two Briggs formulations mentioned in Byun (1999) as,

$$\Delta h = \min \left[3 \left(\frac{F_b}{U} \right)^{3/5} H_*^{-2/5}, 30 \left(\frac{F_b}{U} \right)^{3/5} \right]. \quad (7.12)$$

In this model, the calculations are also modified for *bumping* situation (Briggs, 1984), where the boundary-layer height is larger than the smokestack height ($h_s < H$), but the plume cloud rises higher enough to reach the boundary-layer height. Here, the plume cloud's vertical depth is considered equal to the plume rise value. However, if any portion of the plume cloud, p , is above the boundary-layer height, the plume rise is obtained as

$$\Delta h = (0.62 + 0.38p)(H - h_s), \quad (7.13)$$

where p is the portion of the plume cloud above the boundary-layer height, H .

The aforementioned formulae are used in the GEM-MACH model and the other models for plume rise measurements. In Gordon et al. (2018), a layer-based approach is examined, which is also suggested in Briggs (1984) for plume rise calculations. Also, Akingunola et al. (2018) studies the impact of this layer-based approach within the GEM-MACH model.

7.2 Plume rise measurements using the DCNN-based framework

As discussed in Chapters 4, 5, and 6, we introduced a novel framework that utilizes DCNNs to monitor smokestack plume clouds and make real-time, long-term measurements of plume rise. The framework comprises three stages. In the first stage, the plume cloud is identified using an enhanced Mask R-CNN, known as the DPRNet. Next, image processing analysis and least squares theory are applied to determine the plume cloud's boundaries and fit an asymptotic model to its centerlines. The z-coordinate of the critical point of this model represents the plume rise. Finally, a geometric transformation phase converts image measurements into real-world values. This study's findings indicate that the DPRNet outperforms conventional smoke border detection and recognition networks. In quantitative terms, the proposed approach yielded a 22% enhancement in the F1 score compared to its closest competitor, DeepLabv3. In the following section, we will compare the results of this proposed framework to Briggs parameterization results for plume rise measurements.

7.3 Plume rise comparison results

The landscape of the Athabasca oil sands region can be characterized as a river valley running from north to south, spanning about 1 to 5 kilometres in width. This river valley is situated within a gently sloping north-south valley measuring 10 and 50 kilometres in width. The area also experiences significant differences in elevation, with up to 500 meters of vertical variation. Local surface wind patterns within the valley can be heterogenous, and between two stations of Mildred Lake station and Lower Camp Tower, which are the nearest stations to the smokestack area (Table 3.1), the Mildred Lake station is in the vicinity of the smokestack. As previously discussed in Chapter 3, this monitoring station is used to obtain the wind direction information for plume rise calculations (Figure 7.3).

We compare the plume rise values obtained from the Briggs parameterizations (Equations 7.7 - 7.12) with the proposed DCNN-based framework results to get an idea of the uncertainty related to the local weather conditions. Figure 7.4 shows the plume rise values for both methods. The Briggs results are missing for some date and time; therefore, the corresponding images of these missing values are crossed out. The comparison is done for the image-capturing period starting from November 11, 2018, to July 31, 2019. Plume clouds' shape and background conditions are similar in 4-5 successive images with capturing intervals of 15 minutes. Therefore, we can increase our step size to one hour in order to see more plume variety within a limited number of images. With hourly results of the Briggs Equations, there are 1550 images, which are all in concurrent times within the capturing period. The distributions of the obtained plume rise values for the aforementioned capturing time are represented in Figure 7.5.

Based on Figure 7.4, plume rise values using Briggs parameterizations are in a similar pattern to those calculated with the proposed framework. Also, the distribution of the calculated plume rise values in Figure 7.5 demonstrates that values for the Briggs parameterization are mostly below 150 m. Approximately 93% of these values are under approximately 180 m, with more than half below 90 m. With the proposed framework, approximately 40% of the values are below 180 m, and half or more are above 200 m; However, approximately 10% of the Briggs values are above 200 m.

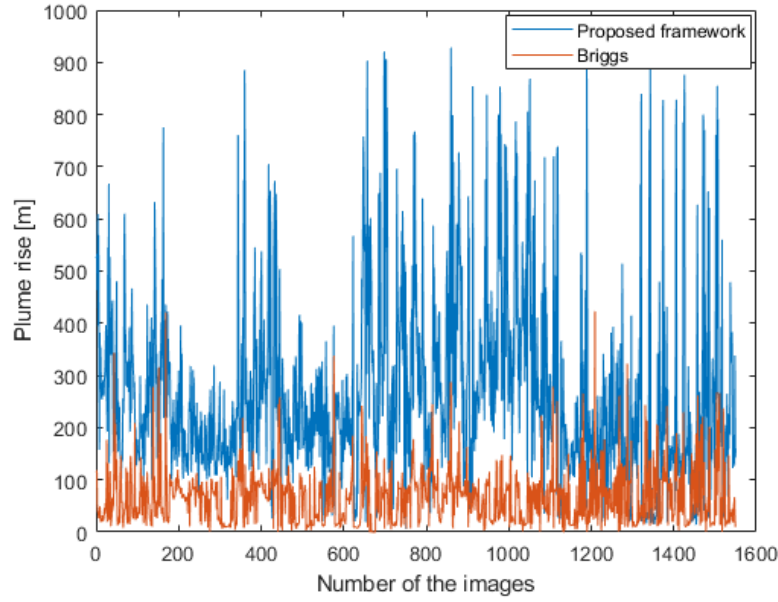


FIGURE 7.4: The calculated plume rise using Briggs parameterizations (Equations 7.7 - 7.12), and the proposed DCNN-based framework with the input wind data from Mildred Lake monitoring station.

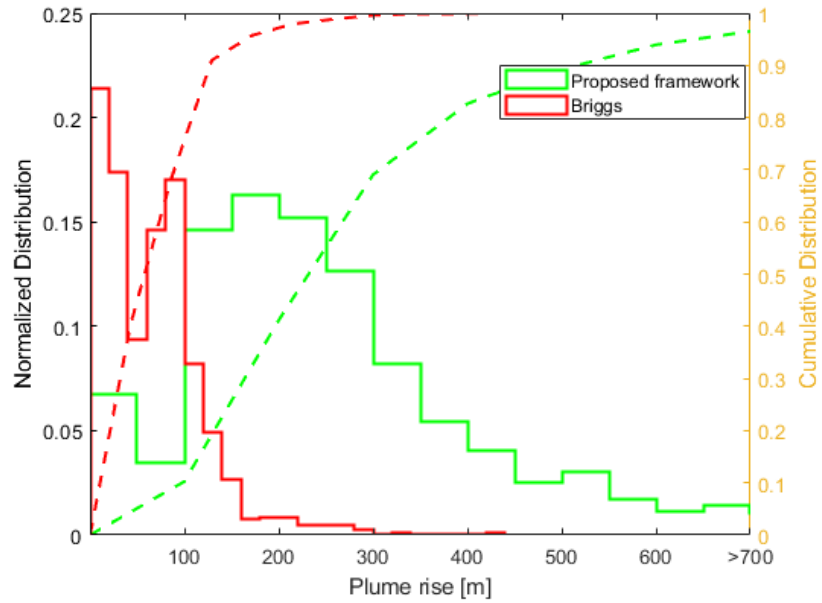


FIGURE 7.5: The distribution of plume rise calculated by Briggs parameterization (Equations 7.7 - 7.12), and the proposed DCNN-based framework with input wind data from Mildred Lake monitoring station. Distributions are illustrated for each hour and for the tallest smokestack in the area (Table 3.1). The summation of all plume rise values, which are more than 700 m, is illustrated by the right-most histogram outlined in the box. Cumulative distributions are depicted by dashed lines for each method.

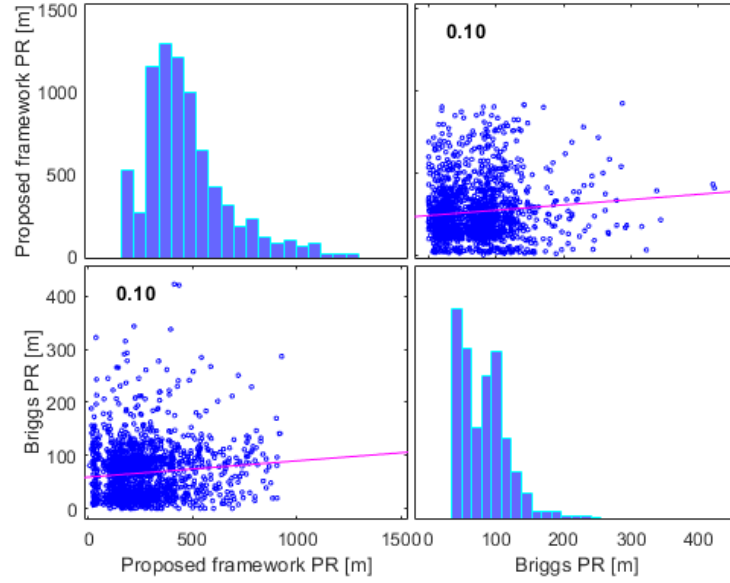


FIGURE 7.6: Comparison of the predicted plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. The magenta line displays a least-square reference line, the slope of which gives the correlation coefficient of $R = 0.10$. PR stands for plume rise in this figure.

The plume rise results obtained from the Briggs parameterization as well as from the proposed DCNN-based framework are also compared in a paired Pearson's correlation coefficients plot as shown in Figure 7.6. The R-value and P-value for this comparison are 0.1 and 0.00004, respectively. This correlation coefficient suggests a weak positive correlation between the measured plume and the predicted values. This means that an increase in plume rise value will lead to a slight tendency for Briggs values to increase, and the relationship is not strong. Moreover, a P-value of less than 0.001 indicates that the measured correlation is statically significant, and this relationship is unlikely to happen by chance. Also, the bivariate histogram plot of the Briggs parameterization and the proposed DCNN-based framework is depicted in Figure 7.7, in which the bins are represented as 3-D rectangular bars whose heights show the number of images in each bar. Statistics comparing the plume rise values using both methods give a low slope of 0.03 (smaller than 0.5), an effective intercept of 59.11 (between 44 m and 107 m), and a low correlation coefficient of 0.1 (smaller than 0.2) indicated that the Briggs plume rise parameterization has a poor performance in predicting plume rise.

Using wind information from the Mildred Lake monitoring station, the results suggest that the Briggs parameterization underestimates the plume rise by an average of 74%. The proposed DCNN-based framework gives a plume rise value, on average, approximately 200 m larger than that of the Briggs equations. Among all cases, 85% of the plume rise values are underestimated, and 18% of plume rise predictions by Briggs are within a factor of 2 of the measured values using the proposed method.

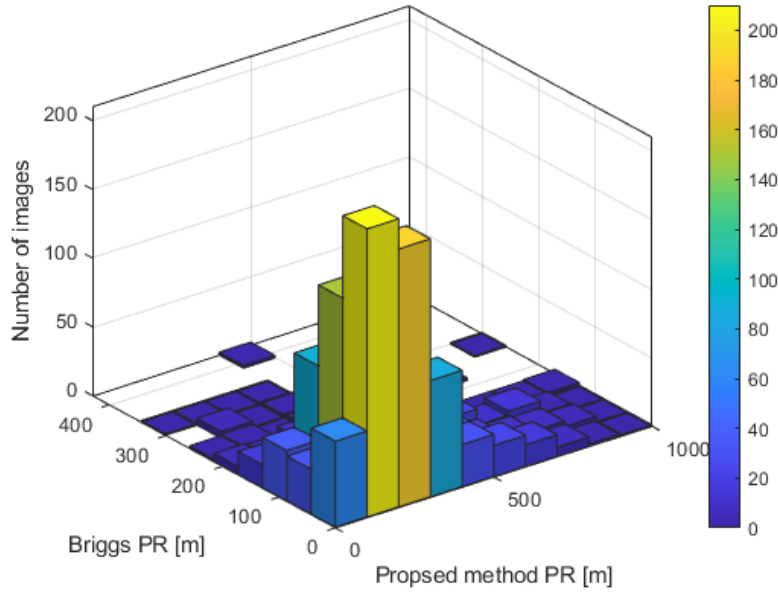


FIGURE 7.7: Comparison of the predicted plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. PR stands for plume rise in this figure.

7.4 Assessing Briggs performance in plume rise estimation

This section aims to evaluate the performance of the Briggs plume rise prediction algorithm under various environmental conditions. Specifically, we investigate whether the parameterizations show differing degrees of accuracy during different times of the day and different seasonal conditions. To assess Briggs performance, we collected the measured values of plume rise and compared them to Briggs estimations using the abovementioned factors. As mentioned in Section 7.3, the comparison was made for the image-capturing period of about nine months, starting from November 11, 2018, to July 31, 2019. We divided the obtained values into daytime and nighttime periods to explore whether Briggs accuracy varies throughout the day. Table 7.3 shows this comparison, in which we can see that the Briggs algorithm's performance does not vary significantly between nighttime and daytime conditions, and diurnal variations do not substantially affect the Briggs accuracy in predicting plume rise.

To better understand Briggs behaviour under varying conditions, we aimed to determine if Briggs performance differs with changing meteorological conditions. Therefore, we categorized the plume rise values into winter and summer-time values for the subsequent comparison. Based on our capturing period, the winter time for this analysis is considered from November 11, 2018, to February 25, 2019, and the summer time is considered from February 26, 2019, to July 31, 2019. Table 7.3 indicates that Briggs performs roughly the same during winter as in summer. It should be noted that these factors can change interchangeably based on the sources and specific conditions of the study area.

TABLE 7.3: Comparison of the differences between modelled plume rise using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework in relation to diurnal and seasonal variations.

	Measured [m]	Modelled [m]	Difference [m]
Average daytime plume rise	252.2	67.8	184.4
Average nighttime plume rise	246.1	66.8	179.3
Average summertime plume rise	242.8	64.4	178.3
Average wintertime plume rise	255.7	70.2	185.5

The Briggs algorithm depends on certain input parameters such as stack height, exhaust velocity, and atmospheric stability class. These parameters remain relatively constant over short time scales, like day-to-day variations, and even between seasons. Furthermore, the impact of diurnal variations on local atmospheric conditions and, consequently, plume rise may be minimal for specific industrial processes or sources, especially if they operate continuously. These factors could contribute to the consistent performance of the Briggs algorithm. Finally, it is expected that the Briggs plume rise algorithm is designed to be robust and applicable in a wide range of conditions with the complex interplay of various factors, which helps it maintain its accuracy even in different environmental conditions.

7.5 Plume rise horizontal distance

As there are many smokestacks around the research's prominent smokestack, there should be cases in which the plume cloud cannot travel a distance long enough to reach the maximum plume rise point mentioned in the Briggs equations. Another approach parameterized in Briggs (1984) suggests a downwind distance to the maximum point of the plume rise, which is appropriate to be leveraged here. The final rise of a plume cloud in stable conditions is calculated as (Briggs, 1984)

$$x_e = 4.7 \left(\frac{U}{\sqrt{S}} \right). \quad (7.14)$$

The final rise of a plume cloud in neutral conditions is calculated as (Briggs, 1975)

$$x_e = \begin{cases} 49F_b^{5/8}, & F_b < 55\text{m}^4\text{s}^{-3}, \\ 119F_b^{2/5}, & F_b > 55\text{m}^4\text{s}^{-3}. \end{cases} \quad (7.15)$$

When the atmospheric conditions are unstable, fumigation happens where the plume cloud reaches the surface. In this condition, if the plume cloud's width is assumed to be Δh , the plume cloud spreads

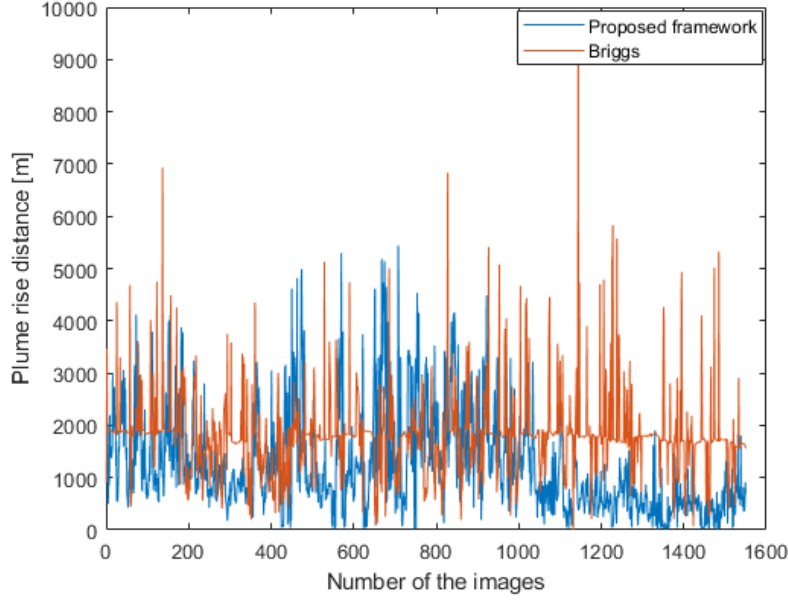


FIGURE 7.8: The calculated plume rise distance using Briggs parameterizations (Equations 7.14 - 7.16), and the proposed DCNN-based framework with the input wind data from Mildred Lake monitoring station.

out evenly between a height of $1.5\Delta h$ and the surface. No specific parameterization for these conditions determines the distance to maximum plume rise. However, a parameterization has been developed by Briggs (1984) to calculate the average horizontal distance to fumigation as

$$x_f = \frac{U}{w}(h_s + 0.5\Delta h), \quad (7.16)$$

where $w = 0.8u_*$ is introduced as the average downdraft speed in Briggs (1984).

The plume rise distance values obtained from the Briggs parameterizations (Equations 7.14 - 7.16) with the proposed DCNN-based framework results are compared in Figure 7.8.

Based on Figure 7.8, plume rise distance values using Briggs parameterizations are not in a similar pattern to those calculated with the proposed framework. Considering that the measured plume rise is usually more than the calculated plume rise by Briggs, the distance to the maximum plume rise is mostly expected to be underestimated. Also, it is discussed in Gordon et al. (2018) that some plume clouds reach their maximum height beyond the measurement location, and their distance to maximum plume rise surpasses the limited box of measurements. Therefore, there should be some cases in which the plume clouds could not reach their maximum rise, and the Briggs parameterization may underestimate the plume rise at these moments.

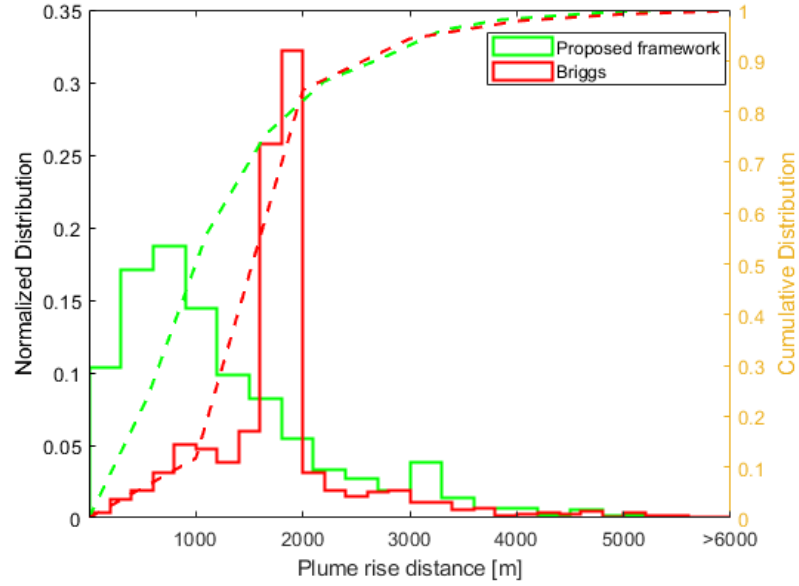


FIGURE 7.9: The distribution of plume rise distance calculated by Briggs parameterization (Equations 7.14 - 7.16), and the proposed DCNN-based framework with input wind data from Mildred Lake monitoring station. Distributions are illustrated for each hour and for the tallest smokestack in the area (Table 3.1). The right-most histogram outlined box illustrates the summation of all plume rise values, which are more than 6000 m. Cumulative distributions are depicted by dashed lines for each method.

In measuring the plume rise distance using an image-capturing system, the limitation lies in the image frame, which only allows us to observe the plume cloud within the frame boundaries. Consequently, we can only measure the plume rise distance according to the information available within that specific frame, and any portions of the plume cloud beyond it cannot be processed. This concern has been addressed in Section 4.2, in which the proposed method tries to find the NBP either within the captured image or beyond the frame boundaries on a real-world scale. The distributions of the obtained plume rise distance values for the capturing time are represented in Figure 7.9. This figure demonstrates that the distance values for the Briggs parameterization are primarily between 1500 m and 2000 m. The plume rise distance results obtained from the Briggs parameterization as well as from the proposed DCNN-based framework are also compared in a paired Pearson's correlation coefficients plot as shown in Figure 7.10. The R-value and P-value for this comparison are -0.002 and 0.93, respectively. Also, the bivariate histogram plot of the Briggs parameterization and the proposed DCNN-based framework results for plume rise distance are depicted in Figure 7.11.

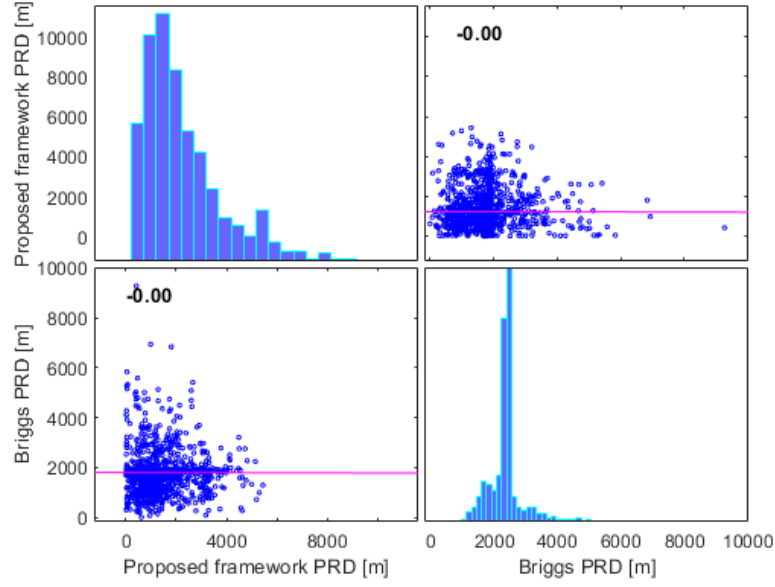


FIGURE 7.10: Comparison of the predicted plume rise distance using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. The magenta line displays a least-square reference line, the slope of which gives the correlation coefficient of $R = -0.002$. PRD stands for plume rise in this figure.

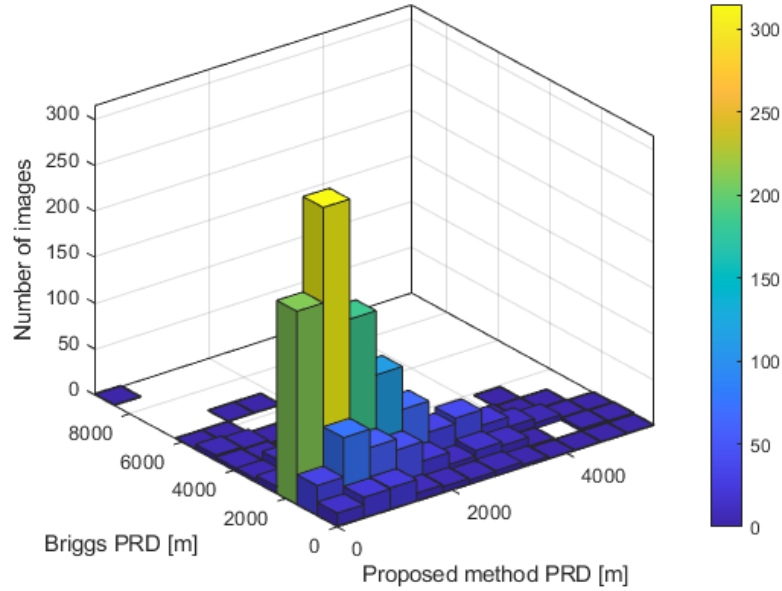


FIGURE 7.11: Comparison of the predicted plume rise distance using the Briggs parameterization in GEM-MACH with the measured plume rise obtained from the proposed DCNN-based framework. PRD stands for plume rise in this figure.

7.6 Uncertainty of the wind direction

As discussed in Chapter 6, wind direction is crucial to our calculations, and it is essential to address the uncertainties associated with wind direction estimation. We need to estimate the uncertainty caused by the assumption that our wind direction represents the wind direction at the smokestack location.

The accuracy of wind direction data is important, as it determines the location of the plume outside of the image plane. Here, we discuss the various sources of uncertainty that may affect the wind direction obtained from a monitoring station located approximately 4 km away from the smokestack location, including spatial difference, sampling frequency, wind sensor accuracy, and atmospheric conditions.

7.6.1 Spatial difference

The foremost concern in using wind direction data from a remote monitoring station is the separate location of the monitoring station and the smokestack. A distance of about 4 km raises the potential for changes in wind patterns due to atmospheric conditions and obstacles such as buildings and vegetation, which lead to deviations in wind direction between the monitoring station and the smokestack. This issue can impact the accuracy of the wind direction information.

7.6.2 Sampling frequency and temporal difference

Temporal synchronicity between the wind direction data and what is happening at the location of the plume cloud emissions is of concern. Monitoring stations deliver data at discrete time intervals, which might not adapt to the timing of the plume cloud generation from the smokestack. This temporal difference raises uncertainty, as wind patterns can change significantly over short intervals.

7.6.3 Wind sensor accuracy

The accuracy of the instruments employed to measure wind direction at the monitoring station affects the obtained data quality. Inaccuracies from instrumentation errors and limitations could cause uncertainties in the derived real-world measurements of plume rise.

7.6.4 Atmospheric conditions

Atmospheric conditions, such as turbulence, can generate fluctuations in wind direction over short distances and time scales. This problem is more observable in industrial areas, such as our research area in Northern Alberta, where complex air patterns occur due to various structures. These variations can affect the accuracy of the wind direction data, especially when the monitoring station is far enough from the smokestack location.

In summary, using wind direction data from a monitoring station remote from the smokestack contains uncertainties that can impact the accuracy of the following calculations to migrate from image-scale measurements to real-world values. Recognizing and quantifying these uncertainties is essential for assessing the validity of the measures via the proposed framework. In the following section, we will

quantify the actual measures of these uncertainties by comparing the wind directions at three different weather stations, including our base tower, followed by a test for wind direction estimation.

7.7 Sensitivity analysis

Here, we conduct an analysis to determine the uncertainty in plume rise caused by the wind direction input variables by comparing wind directions measured at different weather stations, including Mildred Lake station (our base tower), Lower Camp station, and Mannix station (Figure 7.3). We have used about four months of hourly samples of wind direction for all three stations to choose a representative actual number that encapsulates the differences between the wind directions from the three stations above. We leveraged RMSE criteria to calculate the average angular deviation between each station pair to achieve a representative value for the wind direction variability. This should demonstrate the uncertainty in the measured wind direction over the same spatial scale as the distance between your measurement location and the smokestack. Figure 7.12 visualizes the average angular differences between the station pairs during the capturing period. It should be noted that wind direction information should be handled as circular data, and we should use circular statistics and calculations considering the wind direction space's circular nature (Mardia and Jupp, 2009).

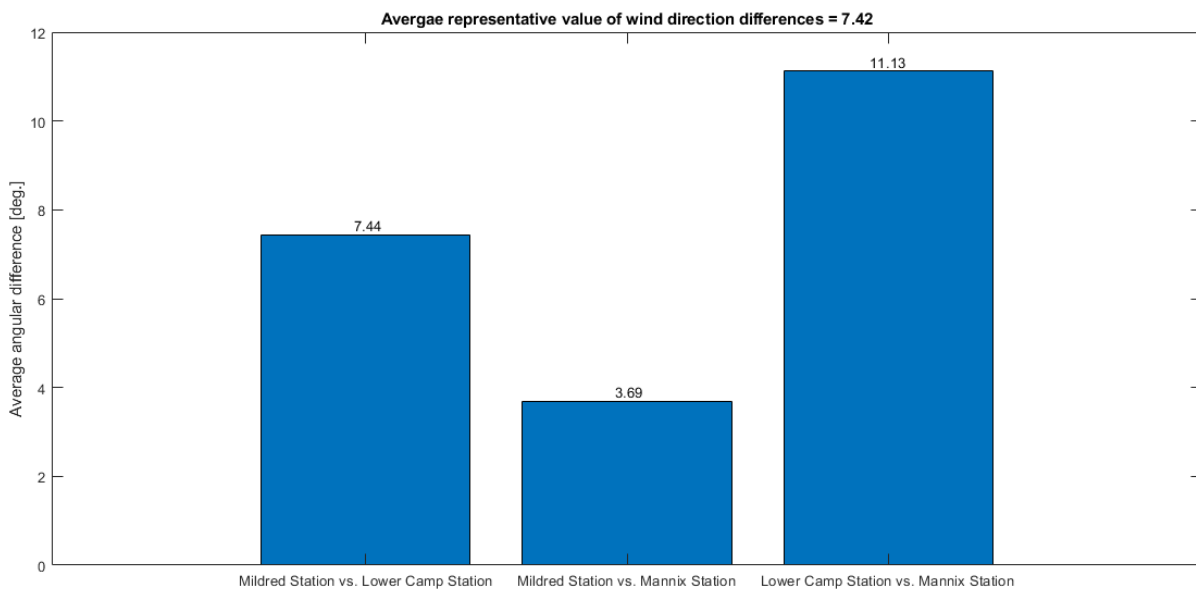


FIGURE 7.12: Circular statistics of the wind direction angular differences between three monitoring stations in Figure 7.3.

Based on the comparison results derived from Figure 7.12, all the original measured wind directions (measured by the base station, Mildred Lake Tower) are increased by 7.42 degrees for our sensitivity analysis, and all the plume rise and plume rise distance values are recalculated. Then, we do the same process for a decrease of 7.42 degrees in wind directions. The plume rise calculations are again performed with the new wind direction values to determine the plume rise and plume rise distance variations in

regard to the previous results with the original wind direction information. The changes in the plume rise and plume rise distance for each condition above are displayed in Figures 7.13 and 7.14.

As seen in Figure 7.13, the increase of the wind direction leads to a higher rise of the plume cloud than the original wind direction. Also, the calculated plume rise is less when the wind direction decreases by 7.42 degrees. Therefore, the wind direction shifts influence a plume cloud's higher and lower rise. However, Figure 7.14 indicates that the wind direction variations have a minor impact on the distance the plume cloud travels.

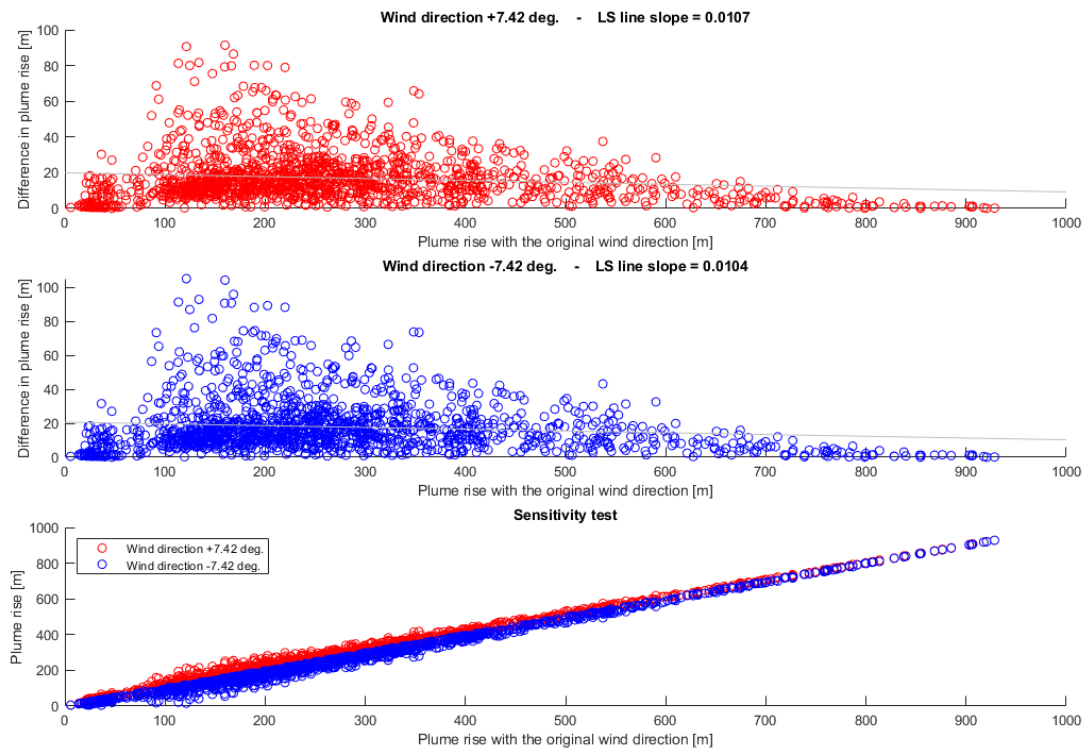


FIGURE 7.13: Wind direction sensitivity analysis relative to the plume rise calculations.

Moreover, the plume rise measurement values affected by the wind direction changes are tabulated in Table 7.4, resulting from sensitivity analysis. These results also show that the wind direction has a more noticeable effect on the plume rise than the plume rise distance. The sizeable average change of about 17 m for the plume rise indicates that the measurements could be unreliable without accurate wind direction estimates. The increase in wind direction by 7.42 degrees leads to an average increase in the plume rise of 9.24 %, while this increase of 7.42 degrees leads to an average increase in plume rise distance of 0.04 %. Although the difference in wind direction by about 15 degrees has a relatively slight influence on the plume rise distance (0.04 % to 1.78 %), this has a powerful consequence on the plume rise, ranging from a 9.24 % increase to a 9.58 % decrease in average. The table specifies the wind direction significantly impacts the plume rise results; therefore, it requires more accuracy when acquired from a meteorological tower or a meteorological model forecast. The uncertainty in estimating this variation will indirectly lead to uncertainty in plume rise assessments.

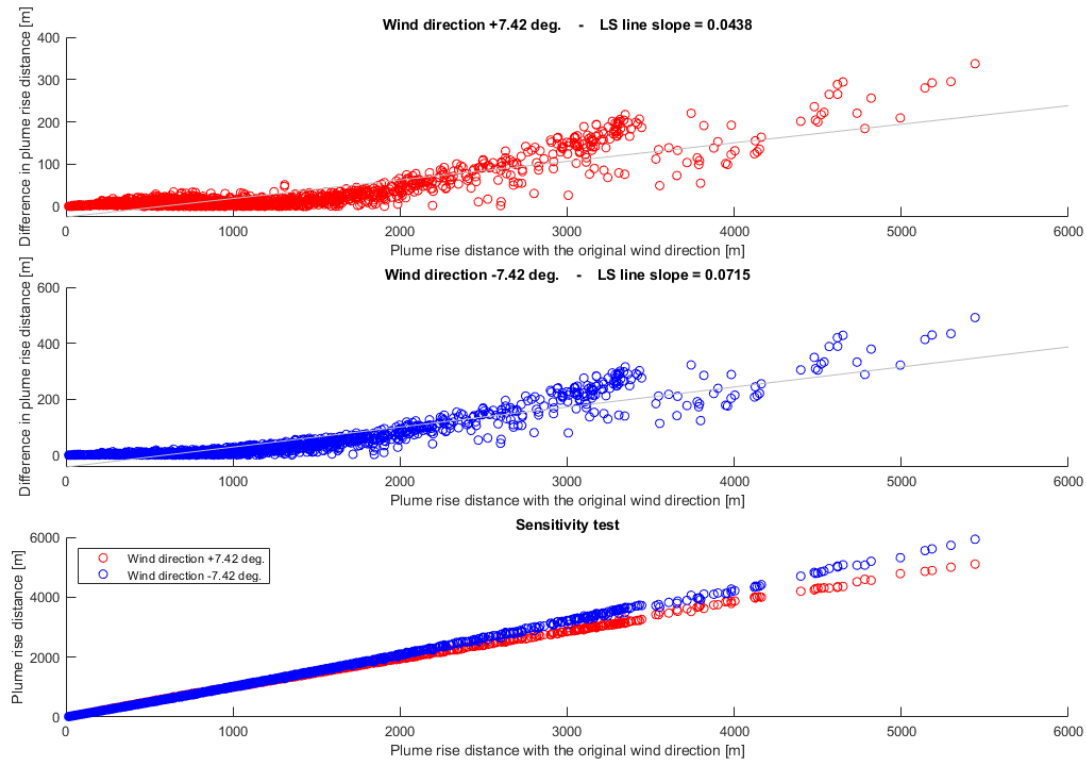


FIGURE 7.14: Wind direction sensitivity analysis relative to the plume rise distance calculations.

TABLE 7.4: Variations of the plume rise measurements resulting from the wind direction sensitivity analysis.

Plume rise variations				
	Average change [m]	Low value [m]	High value [m]	Average percentage [%]
Increased wind direction	+17.11	-4.78	+91.34	+9.24
Decreased wind direction	-17.94	-105.31	+4.80	-9.58
Plume rise distance variations				
	Average change [m]	Low value [m]	High value [m]	Average percentage [%]
Increased wind direction	-19.58	-337.44	+51.28	+0.04
Decreased wind direction	+43.78	-26.86	+491.63	+1.78

The sensitivity analysis outcomes emphasize the sensitivity of the research outcomes, especially plume rise, to changes in wind direction. This analysis shows the importance of accurate wind direction estimation for the purpose of reliable plume rise measurements. A slight change in wind direction can lead to significant changes in plume cloud behaviour and deviations in our geometric transformations, consequently affecting the plume rise calculations.

Finally, we outline the methodology employed to calculate the angle between the image projection plane and the geographical north, enabling precise determination of wind direction in the smokestack location. As our pre-defined parameters, the latitude and longitude coordinates of the smokestack location and the camera position are known. This information serves as the foundation for subsequent calculations. We utilize the geographical coordinates to determine a line representing the direction of the sight line from the camera to the smokestack. Using these two defined points, the angle of this line can be calculated from the geographical north. This leads to calculating our image projection plane angle, perpendicular to the line connecting both the camera and the smokestack. Assume the latitude and longitude of the camera and the smokestack locations are $(lat1, lon1)$ and $(lat2, lon2)$. The initial bearing formula between two points on the earth's surface can be written as follows, which is based on spherical trigonometry,

$$\theta_N = \arctan 2[\sin(\Delta\lambda) \cdot \cos(lat2), \cos(lat1) \cdot \sin(lat2) - \sin(lat1) \cdot \cos(lat2) \cdot \cos(\Delta\lambda)], \quad (7.17)$$

where θ_N is the angle from the north, $\Delta\lambda$ is the longitude difference between the two points, $\arctan 2$ is the function of two-variable arctangent. It should be noted that the latitude and longitude of the points should be considered in radians.

Several factors contribute to the uncertainty of the calculated angle, including the accuracy of GPS measurements for the smokestack and camera locations and elevation differences between these two locations. Considering these factors would enhance the credibility of the wind direction calculations and support a thorough understanding of the limitations of the proposed methodology.

7.8 Total uncertainty and plume rise comparison results

Here, we discuss the impact of our total calculated uncertainty on interpreting our comparison results in Section 7.3. The summation of all estimated uncertainties of the proposed framework in this thesis, which were discussed in Sections 6.2 and 7.7 should be considered to evaluate our conclusions upon comparing plume rise calculations using Briggs parameterizations to our DCNN-based framework.

In Section 7.3, we demonstrated that the Briggs equations underestimate the measured plume rise by 200 meters, while Our analysis revealed a total estimated uncertainty of approximately 8% in our calculation. Although our results indicate a significant difference between the plume rise measurements and the predictions made by the Briggs equations, considering the uncertainty quantification changes the interpretation of the results. The 8% uncertainty suggests that our measurements may show some level of

variability due to factors such as measurement accuracy, instrument calibration, and experimental and environmental conditions. Consequently, we can say that the Briggs equations underestimate the plume rise by roughly 200 ± 16 meters. While the Briggs parameterizations occur to underestimate plume rise, the presence of uncertainty highlights the need for further investigation and consideration of potential sources of error in future studies.

7.9 Correction factor for Briggs plume rise parameterizations

This section investigates a multiplication factor to adjust the results obtained from the Briggs plume rise prediction algorithm with the values obtained from the DCNN-based proposed framework as discussed in Section 7.3. By analyzing the differences between the two methods, we aim to derive this factor to systematically enhance the Briggs accuracy and its alignment with the image-based measurements. The development of the correction factor fills the gap between algorithmic predictions and measured plume rise values, leading to closer measurements to real-world observations. For this purpose, the plume rise values are structured for analysis, including corresponding pairs of algorithmic and measured values. Then, the ratio between values for each data point is calculated. Finally, the geometric mean of these ratios is computed to determine the overall correction factor. This factor, which became **3.6432**, represents the average ratio between values obtained from the two methods. This adjustment approach contributes to more reliable pollutant dispersion modelling. Further advancements may be necessary to fine-tune this multiplication factor for specific procedures and data sets.

The accurate prediction of plume rise is crucial for understanding the dispersion of pollutants in the atmosphere. Briggs parameterizations have long been employed for this purpose, relying on a combination of theoretical foundations and empirical constants derived from historical measurements. However, concerns have been raised about these equations' tendency to under or overestimate the plume rise. Our investigation in this chapter included an analysis of the variables within Briggs equations. Inaccurately modelled stability, a critical factor affecting plume rise, can lead to deviations in predicted plume rise. Similarly, friction velocity and Obukhov length effectively impact the model's accuracy. A significant aspect of Briggs equations lies in the empirical constants derived from historical plume rise measurements. Our study delves into the possibility that the constants, calibrated on data from previous decades, may not be universally applicable. We explore the hypothesis that specific features of the location under consideration might necessitate adjustments to these constants. Examining the constants in intersection with local characteristics may lead to potential differences contributing to the observed deviations. In other words, we conclude that it is not the physics but the constant in front of each equation that might be wrong. Those constants are empirical fits based on previous plume rise measurements decades ago. Something specific about this location might make this constant different or something different about previous measurements.

8 Conclusion and Future Work

To measure the plume rise through remote sensing images, plume cloud recognition is of the essence as the first step. In this regard, a novel deep learning-based method, inspired by the nature of the problem, is proposed in this research to detect and recognize the PC accurately. In the next stage, image processing analysis is leveraged to extract the PC centerline. Afterward, the critical point of this curve is estimated, the y-component coordinate of which is equivalent to plume rise. Lastly, this image measurement is transformed into a real-life world under the geometric transformation stage. Experimental results indicate that the proposed method, DPRNet, significantly outperformed its rivals. This work also demonstrated that plume rise could be determined with a single-camera system and wind direction measurements, allowing further investigation and understanding of the physics of how buoyant plume clouds interact with their environment under different meteorological conditions. Researchers can gain new insights into the fundamental physical processes that drive PC behaviour by systematically varying these conditions and observing how the plume cloud responds. In addition, the method may allow them to study the effects of various external factors on buoyant plume cloud behaviour. The proposed DPRNet plays the main role in the proposed DCNN-based framework in this study, which includes DPRNet, image measurements, and geometric transformations to measure plume rise. DPRNet can be extended to a more comprehensive method, and its limitations can be addressed through several advancements in future research:

- Generalizing DPRNet to simultaneously predict the plume cloud and plume cloud centerline. DPRNet predicts two characteristics of the chimney stack's exit point and plume cloud segments. The DPRNet pipeline estimates detailed information on the plume cloud's central line and plume rise point through post-processing steps. In our future study, we will extend DPRNet's capacity to predict these properties.
- Reinforcing DPRNet to recognize multi-source plume clouds occurring in industrial environments. DPRNet's novelty is to predict a focusing anchor, with which the network's mask prediction can be made only on the single exit point. However, many chimney stacks produce plume clouds in most industrial sites. Thus, our future study will extend DPRNet's capacity to detect multiple exit points anchored for predicting multiple plume clouds, which is required.
- Validating the proposed methods over different sites and cameras with various focal lengths, perspective views and resolutions for generalizing DPRNet's performance.

As the final step of this study, a comparative study using meteorological and smokestack measurements was conducted between the estimated plume rise and plume rise distance from the proposed framework

and the Briggs parameterization equations. The results indicated that the Briggs plume rise parameterization significantly underestimates the plume rise. These results contradict the previously discussed studies in Table 1.1, which suggest a definite overestimation in Briggs plume rise equations. The VDI report (Kommission Reinhaltung der Luft (KRdL) im VDI und DIN – Normenausschuss, 1985), which has summarized these studies, suggests a reduction by a factor of 30% for the Briggs equations in neutral atmospheric conditions. However, this research demonstrates that an increase of more than 30% for the Briggs equations would improve the results. Also, if we consider an overestimation in surface concentration measurements in Webster and Thomson (2002), this study also showed a tendency in Briggs equations to underestimate plume rise.

The low height of a smokestack can affect the plume rise calculations, especially in aircraft-based measurements, when the smokestack is lower than the lowest aircraft measurement height. However, this limitation does not influence our proposed method of plume rise measurement as far as the smokestack and its plume cloud are visible in the captured images. Moreover, for this comparison, we chose the largest smokestack in the area, which has the best results of the Briggs parameterization used in the GEM-MACH model, in comparison with the smaller smokestacks (Gordon et al., 2018). Therefore, any assessment using the Briggs parameterization for tall smokestacks in the region is acceptably precise, and it is recommended to concentrate on the moderately sized smokestacks for future endeavours.

In comparing the plume rise values using both the Briggs parameterization and the proposed framework, the correlation coefficient is low (smaller than 0.2), and the slope of the least-square fit is smaller than 0.5. This indicates that the statement from Carson and Moses (1969) is still valid after about 50 years. Carson and Moses (1969) says that “no plume rise equation can be expected to predict short-term plume rise accurately,” and the corresponding parameterizations were “to be used for general design considerations.” Also, leveraging the same equations thus far demonstrates limited advancements in the subject.

This image-based method for plume rise measurements provides results based on the captured images over the course of several months, in which the plume cloud and its related smokestack are visible. Although it seems reasonable to lay on the obtained results, the invariant underestimation of the plume rise implies more investigation into this subject. With the recent improvement in atmospheric measurements, technologies such as automated lidar and RASS can be utilized to conduct long-term measurements of plume rise with higher frequencies, which would enhance predictability. Although the Briggs algorithms are broadly cited in different air quality and dispersion textbooks (Arya et al., 1999; Beychok, 2005) and have been applied in various air-quality models, such as GEM-MACH, SMOKE, SMOKE-EU, SCREEN3, AEROPOL, RADM, and CALGRID, there should be new measurement techniques with new approaches to verify these algorithms. Accurate in situ measurements are required to target the uncertainties, which leads to enhancement in prediction algorithms. One of these uncertainties is the wind direction around the smokestack, which is strongly affected by the significant horizontal heterogeneity across the region. Accordingly, the wind direction information obtained from the monitoring station (Mildred Lake), approximately 4 km from the smokestack region, needs to be investigated more accurately. High-resolution models can be utilized for meteorological observations to develop measurement methods further.

References

- Aggarwal, Charu C et al. (2018). "Neural networks and deep learning". In: *Springer* 10, pp. 978–3.
- Ajith, Meenu and Manel Martínez-Ramón (2019). "Unsupervised segmentation of fire and smoke from infra-red videos". In: *IEEE Access* 7, pp. 182381–182394.
- Akingunola, Ayodeji et al. (2018). "A chemical transport model study of plume-rise and particle size distribution for the Athabasca oil sands". In: *Atmospheric Chemistry and Physics* 18.12, pp. 8667–8688.
- Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi (2017). "Understanding of a convolutional neural network". In: *2017 international conference on engineering and technology (ICET)*. Ieee, pp. 1–6.
- Alfieri, S et al. (2010). "Quantifying trace gas emissions from composite landscapes: a mass-budget approach with aircraft measurements". In: *Atmospheric Environment* 44.15, pp. 1866–1876.
- Arya, S Pal et al. (1999). *Air pollution meteorology and dispersion*. Vol. 310. Oxford University Press New York.
- Ashrafi, Khosro, Ali Ahmadi Orkomi, and Majid Shafipour Motlagh (2017). "Direct effect of atmospheric turbulence on plume rise in a neutral atmosphere". In: *Atmospheric Pollution Research* 8.4, pp. 640–651.
- Aslan, Süleyman et al. (2019). "Early wildfire smoke detection based on motion-based geometric image transformation and deep convolutional generative adversarial networks". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 8315–8319.
- Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12, pp. 2481–2495.
- Benjamin, Stephen F (1975). "The behaviour of hot plumes from tall chimneys under conditions of strong thermal convection". In.
- Berg, Lothar (2013). *Introduction to the operational calculus*. Elsevier.
- Beychok, Milton R (2005). *Fundamentals of stack gas dispersion*. MR Beychok.
- Bieser, J et al. (2011). "Vertical emission profiles for Europe based on plume rise calculations". In: *Environmental Pollution* 159.10, pp. 2935–2946.
- Briggs, GA (1984). "Plume rise and buoyancy effects, atmospheric sciences and power production". In: *US Department of Energy DOE/TIC-27601 (DE 84005177), Washington, DC, USA*.
- Briggs, Gary A (1965). "A plume rise model compared with observations". In: *Journal of the Air Pollution Control Association* 15.9, pp. 433–438.
- (1969). *PLUME RISE: A CRITICAL SURVEY*. Tech. rep. Air Resources Atmospheric Turbulence and Diffusion Lab., Oak Ridge, Tenn.
- (1975). *Plume rise from multiple sources*. Tech. rep.

- Briggs, Gary A (1982). "Plume rise predictions". In: *Lectures on air pollution and environmental impact analyses*. Springer, pp. 59–111.
- (1985). "Analytical parameterizations of diffusion: The convective boundary layer". In: *Journal of Applied Meteorology and Climatology* 24.11, pp. 1167–1186.
- Bringfelt, Björn (1968). "Plume rise measurements at industrial chimneys". In: *Atmospheric Environment* (1967) 2.6, pp. 575–598.
- Bu, XiaoQing et al. (2019). "One-Shot Video Object Segmentation Initialized with Referring Expression". In: *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, pp. 416–428.
- Byun, D (1999). "Science algorithms of the EPA Models-3 community multiscale air quality (CMAQ) modeling system". In: *EPA/600/R-99/030*.
- Canadian Natural Resources Ltd (2015). *Operations: Horizon Oil Sands*. <http://www.cnrl.com/operations/north-america/horizon-oil-sands.html>, last access: 4 May 2015.
- Carson, James E and Harry Moses (1969). "The validity of several plume rise formulas". In: *Journal of the air pollution control association* 19.11, pp. 862–866.
- Cauchy, Augustin et al. (1847). "Méthode générale pour la résolution des systemes d'équations simultanées". In: *Comp. Rend. Sci. Paris* 25.1847, pp. 536–538.
- Charpentier, Alex D, Joule A Bergerson, and Heather L MacLean (2009). "Understanding the Canadian oil sands industry's greenhouse gas emissions". In: *Environmental research letters* 4.1, p. 014005.
- Chen, Liang-Chieh et al. (2014). "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062*.
- (2017a). "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4, pp. 834–848.
- Chen, Liang-Chieh et al. (2017b). "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587*.
- Chen, Liang-Chieh et al. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.
- Chen, Patrick C and Theodosios Pavlidis (1979). "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm". In: *Computer graphics and image processing* 10.2, pp. 172–182.
- Chen, Xinlei and Abhinav Gupta (2017). "An implementation of faster rcnn with study for region sampling". In: *arXiv preprint arXiv:1702.02138*.
- Cheng, Shuhong, Jiyong Ma, and Shijun Zhang (2019). "Smoke detection and trend prediction method based on Deeplabv3+ and generative adversarial network". In: *Journal of Electronic Imaging* 28.3, p. 033006.
- Chollet, Francois (2021). *Deep learning with Python*. Simon and Schuster.
- Cimorelli, Alan J et al. (2005). "AERMOD: A dispersion model for industrial source applications. Part I: General model formulation and boundary layer characterization". In: *Journal of applied meteorology* 44.5, pp. 682–693.
- Ciresan, Dan Claudiu et al. (2011). "Flexible, high performance convolutional neural networks for image classification". In: *Twenty-second international joint conference on artificial intelligence*.

- Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*, pp. 160–167.
- Connors, Richard W and Charles A Harlow (1980). "A theoretical comparison of texture algorithms". In: *IEEE transactions on pattern analysis and machine intelligence* 3, pp. 204–222.
- Csanady, Gabriel T (1973). *Turbulent diffusion in the environment*. 3. Springer Science & Business Media.
- Csanady, GT (1964). "Turbulent diffusion in a stratified fluid". In: *Journal of Atmospheric Sciences* 21.4, pp. 439–447.
- Curry, Haskell B (1944). "The method of steepest descent for non-linear minimization problems". In: *Quarterly of Applied Mathematics* 2.3, pp. 258–261.
- Dai, Jifeng et al. (2016). "Instance-sensitive fully convolutional networks". In: *European Conference on Computer Vision*. Springer, pp. 534–549.
- De Visscher, Alex (2013). *Air dispersion modeling: foundations and applications*. John Wiley & Sons.
- Ding, Henghui et al. (2018). "Context contrasted feature and gated multi-scale aggregation for scene segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2393–2402.
- Djurfors, S (1977). "On the rise of buoyant plumes in turbulent environments". In.
- Dougherty, Geoff (2012). *Pattern recognition and classification: an introduction*. Springer Science & Business Media.
- Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7.
- ECCC (2017). *Environment and Climate Change Canada: CEMS data*. http://collaboration.cmc.ec.gc.ca/cmc/arqi/ACP-2017-1215/CAC_inventory.tz, last access: October 2018b.
- ECCC and AEP (2016). *Environment and Climate Change Canada and Alberta Environment and Parks: Joint Oil Sands Monitoring Program Emissions Inventory Compilation Report*, 146 pp). <http://aep.alberta.ca/air/reports-data/documents/JOSM-EmissionsInventoryReport-Jun2016.pdf>, last access: November 2017.
- Emery, Christopher, Jaegun Jung, and Greg Yarwood (2010). "Implementation of an alternative plume rise methodology in CAMx". In: *Novato, CA*.
- England, Walter G, Lynn H Teuscher, and R Bruce Snyder (1976). "A measurement program to determine plume configurations at the Beaver Gas Turbine Facility, Port Westward, Oregon". In: *Journal of the Air Pollution Control Association* 26.10, pp. 986–989.
- Filonenko, Alexander, Danilo Cáceres Hernández, and Kang-Hyun Jo (2017). "Fast smoke detection for video surveillance using CUDA". In: *IEEE Transactions on Industrial Informatics* 14.2, pp. 725–733.
- Fukushima, Kuniyoshi (1988). "Neocognitron: A hierarchical neural network capable of visual pattern recognition". In: *Neural networks* 1.2, pp. 119–130.
- Fukushima, Kuniyoshi and Sei Miyake (1982). "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". In: *Competition and cooperation in neural nets*. Springer, pp. 267–285.
- Ganesh, P et al. (2019). "Deep orange: Mask R-CNN based orange detection and segmentation". In: *IFAC-PapersOnLine* 52.30, pp. 70–75.

- Garcia-Lamont, Farid et al. (2018). "Segmentation of images by color features: A survey". In: *Neurocomputing* 292, pp. 1–27.
- Garratt, John Roy (1994). "The atmospheric boundary layer". In: *Earth-Science Reviews* 37.1-2, pp. 89–134.
- Gielbel, J (1979). "Messungen der Abgasfahnenüberhöhung eines Steinkohlekraftwerkes mit Hilfe von LIDAR (Plume Rise measurements of a pit coal power plant by means of LIDAR)". In: *Schriftenreihe der Landesanstalt für Immissionsschutz des Landes NRW* 47 S 42, p. 59.
- Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.
- Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.
- Gonzalez, RC (2006). *Woods. RE, (2002) "Digital Image Processing"*.
- Gonzalez, RC, RE Woods, and SL Eddins (2009). *Digital image processing using Matlab* "Gatesmark Publishing".
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.
- Gordon, M et al. (2015). "Determining air pollutant emission rates based on mass balance using airborne measurement data over the Alberta oil sands operations". In: *Atmospheric Measurement Techniques* 8.9, pp. 3745–3765.
- Gordon, Mark et al. (2018). "A comparison of plume rise algorithms to stack plume measurements in the Athabasca oil sands". In: *Atmospheric Chemistry and Physics* 18.19, pp. 14695–14714.
- Gu, Ke, Junfei Qiao, and Xiaoli Li (2018). "Highly efficient picture-based prediction of PM2. 5 concentration". In: *IEEE Transactions on Industrial Electronics* 66.4, pp. 3176–3184.
- Gu, Ke, Junfei Qiao, and Weisi Lin (2018). "Recurrent air quality predictor based on meteorology-and pollution-related factors". In: *IEEE Transactions on Industrial Informatics* 14.9, pp. 3946–3955.
- Gubbi, Jayavardhana, Slaven Marusic, and Marimuthu Palaniswami (2009). "Smoke detection in video using wavelets and support vector machines". In: *Fire Safety Journal* 44.8, pp. 1110–1115.
- Gurney, Kevin (2018). *An introduction to neural networks*. CRC press.
- Hamilton, PM (1967). "Paper III: plume height measurements at Northfleet and Tilbury power stations". In: *Atmospheric Environment* (1967) 1.4, pp. 379–387.
- Haralick, Robert M (1979). "Statistical and structural approaches to texture". In: *Proceedings of the IEEE* 67.5, pp. 786–804.
- Hartigan, John A and Manchek A Wong (1979). "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the royal statistical society. series c (applied statistics)* 28.1, pp. 100–108.
- Haykin, Simon (2009). *Neural networks and learning machines*, 3/E. Pearson Education India.
- He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- He, Kaiming et al. (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.
- Hinton, Geoffrey, Nitsh Srivastava, and Kevin Swersky (2012). "Neural networks for machine learning". In: *Coursera, video lectures* 264.1, pp. 2146–2153.

- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.
- Hou, Qibin et al. (2017). "Deeply supervised salient object detection with short connections". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3203–3212.
- Huang, Gao et al. (2017). "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.
- Hubel, David H and Torsten N Wiesel (1959). "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of physiology* 148.3, p. 574.
- Hwang, Byunghoon et al. (2022). "Automatic detection and segmentation of thrombi in abdominal aortic aneurysms using a mask region-based convolutional neural network with optimized loss functions". In: *Sensors* 22.10, p. 3643.
- Isikdogan, Furkan, Alan Bovik, and Paola Passalacqua (2017a). "RivaMap: An automated river analysis and mapping engine". In: *Remote Sensing of Environment* 202, pp. 88–97.
- Isikdogan, Furkan, Alan C Bovik, and Paola Passalacqua (2017b). "Surface water mapping by deep learning". In: *IEEE journal of selected topics in applied earth observations and remote sensing* 10.11, pp. 4909–4918.
- Ivakhnenko, AGe and VG Lapa (1965). "Cybernetic Predicting Devices. CCM Information Corporation". In: *First working Deep Learners with many layers, learning internal representations*.
- Ivakhnenko, Alekse Grigorevich, Valentin Grigorevich Lapa, and Valentin Grigorevich Lapa (1967). *Cybernetics and forecasting techniques*. Vol. 8. American Elsevier Publishing Company.
- Jain, Suyog Dutt, Bo Xiong, and Kristen Grauman (2017). "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos". In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, pp. 2117–2126.
- Jakubowski, Jacek, Maciej Solarczyk, and Michał Wiśnios (2019). "Smoke detection in a digital image with the use of convolutional network". In: *XII Conference on Reconnaissance and Electronic Warfare Systems*. Vol. 11055. SPIE, pp. 123–133.
- Ji, Shuiwang et al. (2012). "3D convolutional neural networks for human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1, pp. 221–231.
- Jia, Yang et al. (2016). "A saliency-based method for early smoke detection in video sequences". In: *Fire technology* 52.5, pp. 1271–1292.
- Jian, Wenlin et al. (2018). "Smoke regions extraction based on two steps segmentation and motion detection in early fire". In: *MIPPR 2017: Pattern Recognition and Computer Vision*. Vol. 10609. SPIE, pp. 281–288.
- Jiang, Xiaoheng et al. (2018). "Deep neural networks with elastic rectified linear units for object recognition". In: *Neurocomputing* 275, pp. 1132–1139.
- Jin, Wen et al. (2000). "The improvements of BP neural network learning algorithm". In: *WCC 2000-ICSP 2000. 2000 5th international conference on signal processing proceedings. 16th world computer congress 2000*. Vol. 3. IEEE, pp. 1647–1649.
- Kaimal, Jagadish Chandran and John J Finnigan (1994). *Atmospheric boundary layer flows: their structure and measurement*. Oxford university press.
- Kalthoff, N et al. (2002). "Emissions of the city of Augsburg determined using the mass balance method". In: *Atmospheric Environment* 36, pp. 19–31.

- Karimipour, Hadis et al. (2019). "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids". In: *IEEE Access* 7, pp. 80778–80788.
- Ker, Justin et al. (2017). "Deep learning applications in medical image analysis". In: *Ieee Access* 6, pp. 9375–9389.
- Khandelwal, R (2019). "Computer Vision-A journey from CNN to Mask R-CNN and YOLO-Part 2". In: *Dostupné z: <https://towardsdatascience.com/computervision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-1d141eba6e04>*.
- Kim, Phil (2017). "Matlab deep learning". In: *With machine learning, neural networks and artificial intelligence* 130.21.
- Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.
- Kommission Reinhaltung der Luft (KRdL) im VDI und DIN – Normenausschuss (1985). *VDI: Ausbreitung von Luftverunreinigungen in der Atmosphäre; Berechnung der Abgasfahnen-überhöhung (Dispersion of air pollutants in the atmosphere; determination of plume rise)*. <http://www.vdi.de>, last access: February 2018.
- Koushafar, Mohammad, Gunho Sohn, and Mark Gordon (2023). "Deep Convolutional Neural Network for Plume Rise Measurements in Industrial Environments". In: *Remote Sensing* 15.12, p. 3083.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.
- (2017). "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6, pp. 84–90.
- LeCun, Yann, John Denker, and Sara Solla (1989). "Optimal brain damage". In: *Advances in neural information processing systems* 2.
- LeCun, Yann et al. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.
- LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- Lin, Fengyi and Xingwei Chen (2023). "Effects of Landscape Patterns on Atmospheric Particulate Matter Concentrations in Fujian Province, China". In: *Atmosphere* 14.5, p. 787.
- Lin, Gaohua et al. (2019). "Smoke detection on video sequences using 3D convolutional neural networks". In: *Fire Technology* 55.5, pp. 1827–1847.
- Lin, Guosheng et al. (2017). "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1925–1934.
- Lin, Kaihan et al. (2020). "Face detection and segmentation based on improved mask R-CNN". In: *Discrete dynamics in nature and society* 2020.
- Linnainmaa, S (1970). "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors.(in Finnish) Master's Thesis, Department of Computer Science, University of Helsinki, 1970." In.
- LIU, Mao-shen et al. (2019). "Simple and Efficient Smoke Segmentation Based on Fully Convolutional Network". In: *DEStech Trans. Comput. Sci. Eng.(ica)*.

- Liu, Yanbei et al. (2019). "A dual convolution network using dark channel prior for image smoke classification". In: *IEEE Access* 7, pp. 60697–60706.
- Logothetis, Ioannis et al. (2023). "A Case Study of Air Quality and a Health Index over a Port, an Urban and a High-Traffic Location in Rhodes City". In: *Air* 1.2, pp. 139–158.
- Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.
- Longstaff, Dennis et al. (1995). "Improving co-occurrence matrix feature discrimination". In: *Proc. of DICTA'95, 3rd International Conference on Digital Image Computing: Techniques and Applications*. Citeseer.
- Luhmann, Thomas et al. (2006). *Close range photogrammetry: principles, techniques and applications*. Vol. 3. Whittles publishing Dunbeath.
- Luo, Zhiming et al. (2017). "Non-local deep features for salient object detection". In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 6609–6617.
- Mahrt, L (1981). "Modelling the depth of the stable boundary-layer". In: *Boundary-Layer Meteorology* 21.1, pp. 3–19.
- Makar, PA et al. (2015). "Feedbacks between air pollution and weather, Part 1: Effects on weather". In: *Atmospheric Environment* 115, pp. 442–469.
- Mardia, Kanti V and Peter E Jupp (2009). *Directional statistics*. John Wiley & Sons.
- Matlani, Princy and Manish Shrivastava (2019). "Hybrid deep VGG-NET convolutional classifier for video smoke detection". In: *Computer Modeling in Engineering & Sciences* 119.3, pp. 427–458.
- McCulloch, Warren S and Walter Pitts (1990). "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of mathematical biology* 52.1, pp. 99–115.
- Moore, DJ (1974). "A comparison of the trajectories of rising buoyant plumes with theoretical/empirical models". In: *Atmospheric Environment (1967)* 8.5, pp. 441–457.
- Ohanian, Philippe P and Richard C Dubes (1992). "Performance evaluation for four classes of textural features". In: *Pattern recognition* 25.8, pp. 819–833.
- Otsu, Nobuyuki (1979). "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1, pp. 62–66.
- Patterson, Josh and Adam Gibson (2017). *Deep learning: A practitioner's approach*. "O'Reilly Media, Inc."
- Peng, Chao et al. (2017). "Large kernel matters—improve semantic segmentation by global convolutional network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361.
- Phillips Iii, Walter, Mubarak Shah, and Niels da Vitoria Lobo (2002). "Flame recognition in video". In: *Pattern recognition letters* 23.1-3, pp. 319–327.
- Priestley, CHB (1953). "Buoyant motion in a turbulent environment". In: *Australian Journal of Physics* 6.3, pp. 279–290.
- (1956). "A working theory of the bent-over plume of hot gas". In: *Quarterly Journal of the Royal Meteorological Society* 82.352, pp. 165–176.
- Priestley, CHB and FK Ball (1955). "Continuous convection from an isolated source of heat". In: *Quarterly Journal of the Royal Meteorological Society* 81.348, pp. 144–157.
- Protection, Alberta Environmental (1998). "Continuous Emission Monitoring System (CEMS) Code". In: *Environmental Service*.

- Pundir, Arun Singh and Balasubramanian Raman (2019). "Dual deep learning model for image based smoke detection". In: *Fire technology* 55.6, pp. 2419–2442.
- Rittmann, Bruch E (1982). "Application of two-thirds law to plume rise from industrial-sized sources". In: *Atmospheric Environment* (1967) 16.11, pp. 2575–2579.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.
- Ruder, Sebastian (2016). "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747*.
- Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science.
- (1986). "Learning representations by back-propagating errors". In: *nature* 323.6088, pp. 533–536.
- Russakovsky, Olga et al. (2015). "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3, pp. 211–252.
- Samarasinghe, Sandhya (2016). *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. Auerbach publications.
- Sharf, Geulah et al. (1993). "Plume rise measurements from large point sources in Israel". In: *Atmospheric Environment. Part A. General Topics* 27.11, pp. 1657–1663.
- Shen, Wei et al. (2017a). "Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images". In: *IEEE Transactions on Image Processing* 26.11, pp. 5298–5311.
- Shen, Wei et al. (2017b). "Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2391–2400.
- Shi, Yu-kun et al. (2015). "A Study on Smoke Detection Based on Multi-feature [J]". In: *Journal of Signal Processing* 31.10, pp. 1336–1341.
- Shu, Jian-Hua et al. (2020). "An improved mask R-CNN model for multiorgan segmentation". In: *Mathematical Problems in Engineering* 2020.
- Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.
- Skansi, Sandro (2018). *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer.
- Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.
- Stollenga, Marijn Frederik (2016). "Advances in humanoid control and perception". PhD thesis. Università della Svizzera italiana.
- Tian, Hongda et al. (2017). "Detection and separation of smoke from single image frames". In: *IEEE Transactions on Image Processing* 27.3, pp. 1164–1177.
- Turner, D Bruce and Richard Schulze (2007). "Atmospheric Dispersion Modeling: Trinity Consultants." In.
- Turner, JS (1963). "The motion of buoyant elements in turbulent surroundings". In: *Journal of Fluid Mechanics* 16.1, pp. 1–16.

- Wang, Haitao and Yanli Chen (2019). "A smoke image segmentation algorithm based on rough set and region growing". In: *Journal of Forest Science* 65.8, pp. 321–329.
- Wang, Yingying et al. (2020). "The influence of the activation function in a convolution neural network model of facial expression recognition". In: *Applied Sciences* 10.5, p. 1897.
- Webster, HN and DJ Thomson (2002). "Validation of a Lagrangian model plume rise scheme using the Kincaid data set". In: *Atmospheric Environment* 36.32, pp. 5031–5042.
- Weil, Jeffrey C (1988). "Plume rise". In: *Lectures on air pollution modeling*. Springer, pp. 119–166.
- Wu, A, C Du, and M Li (2008). "Smoke detection method based on mixed Gaussian model and wavelet transformation". In: *Chinese Journal of Scientific Instrument* 29.8, pp. 1622–1626.
- Xie, Saining and Zhuowen Tu (2015). "Holistically-nested edge detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403.
- Xu, Gao et al. (2019). "Video smoke detection based on deep saliency network". In: *Fire Safety Journal* 105, pp. 277–285.
- Yang, Maoke et al. (2018). "Denseaspp for semantic segmentation in street scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3684–3692.
- Yin, Mengxia et al. (2019). "Recurrent convolutional network for video-based smoke detection". In: *Multimedia Tools and Applications* 78.1, pp. 237–256.
- Yu, Changqian et al. (2018a). "Bisenet: Bilateral segmentation network for real-time semantic segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341.
- (2018b). "Learning a discriminative feature network for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1857–1866.
- Yu, Chunyu, Zhibin Mei, and Xi Zhang (2013). "A real-time video fire flame and smoke detection algorithm". In: *Procedia Engineering* 62, pp. 891–898.
- Yu, Hongshan et al. (2018c). "Methods and datasets on semantic segmentation: A review". In: *Neurocomputing* 304, pp. 82–103.
- Yuan, Chi, Zhixiang Liu, and Youmin Zhang (2019). "Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance". In: *Journal of Intelligent & Robotic Systems* 93.1, pp. 337–349.
- Yuan, Feiniu (2011). "Video-based smoke detection with histogram sequence of LBP and LBPV pyramids". In: *Fire safety journal* 46.3, pp. 132–139.
- (2012). "A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with AdaBoost for video smoke detection". In: *Pattern Recognition* 45.12, pp. 4326–4336.
- Yuan, Feiniu et al. (2015). "Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis". In: *IET Image Processing* 9.10, pp. 849–856.
- Yuan, Feiniu et al. (2016). "High-order local ternary patterns with locality preserving projection for smoke detection and image classification". In: *Information Sciences* 372, pp. 225–240.
- Yuan, Feiniu et al. (2019). "Deep smoke segmentation". In: *Neurocomputing* 357, pp. 248–260.
- Yuan, Peng, Xinguo Hou, and Leping Pu (2018). "A Smoke Recognition Method Combined Dynamic Characteristics and Color Characteristics of Large Displacement Area". In: *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, pp. 501–507.

- Yunchao, WEI and ZHAO Yao (2016). "A review on image semantic segmentation based on DCNN". In: *journal of beijing jiaotong university* 40.4, pp. 82–91.
- Zen, Revaldo IM et al. (2013). "Dangerous smoke classification using mathematical model of meaning". In: *Procedia Engineering* 62, pp. 963–971.
- Zeng, Junying et al. (2018). "An improved object detection method based on deep convolution neural network for smoke detection". In: *2018 international conference on machine learning and cybernetics (ICMLC)*. Vol. 1. IEEE, pp. 184–189.
- Zhang, Aston et al. (2021). "Dive into deep learning". In: *arXiv preprint arXiv:2106.11342*.
- Zhang, Jingzhao et al. (2019). "Why gradient clipping accelerates training: A theoretical justification for adaptivity". In: *arXiv preprint arXiv:1905.11881*.
- Zhang, Zheng et al. (2016). "Multi-oriented text detection with fully convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4159–4167.
- Zhao, Hengshuang et al. (2017). "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890.
- Zhao, Juanping et al. (2019a). "A coupled convolutional neural network for small and densely clustered ship detection in SAR images". In: *Science China Information Sciences* 62.4, pp. 1–16.
- Zhao, Wangda et al. (2019b). "A smoke segmentation algorithm based on improved intelligent seeded region growing". In: *Fire and Materials* 43.6, pp. 725–733.
- Zhao, Yaqin (2015). "Candidate smoke region segmentation of fire video based on rough set theory". In: *Journal of Electrical and Computer Engineering* 2015.

Appendices

A.1 Gray Level Co-occurrence Matrix

The Gray Level Co-occurrence Matrix (GLCM) (Haralick, 1979) is a powerful tool which is used to extract second-order texture information from an image (Longstaff et al., 1995). These features provide helpful information about the relative positions of the different gray scale levels in the image. For example, the first-order features in a face detection process are eye colour and a facial scar. At the same time, the second-order features are mouth shape, the distance between the mouth and nose, face shape, and such relative features. It has been proven that this technique outperforms the Gray Level Difference Matrix (GLDM), Gray Level Run Length Method (GLRLM), and Power Spectral Method (PSM) as its peer methods (Conners and Harlow, 1980). Also, this extractor has more useful features than the Markov Random Field (MRF) and Gabor filter in texture classification (Ohanian and Dubes, 1992).

Generally, the co-occurrence matrix is symmetric and normalized; thus, the values of the elements are between 0 and 1, and the sum of all these values is equal to 1. Consider a rectangular region of an image with N_r number of row pixels and N_c number of column pixels for texture analysis. The GLCM consists of relative frequencies P_{ij} with which two pixels are separated on the image region by the distance of d in a certain angle θ (direction), one with the value of i and the other with a value of j . Therefore, a matrix is formed as a function of angular (θ) relationship and distance (d) between the neighbouring pixels. According to the dataset and physical behaviour of a typical plume cloud released from a stack, the angular attitudes of the plumes are mostly near 45 or 135 degrees relative to the horizontal line, depending on wind direction. Furthermore, we ignore the unstable atmospheric conditions around the smoke stack in this experiment. Thus, considering this angle for feature extraction can improve the feature extraction process in this direction.

For instance, for a $N_r \times N_c$: 5×5 region of an image, GLCM for $d=1$ and $\theta=45^\circ$ and $\theta=135^\circ$ can be calculated as shown in Figure A.1.

A.2 Otsu threshold selection method

Determining a suitable grayscale threshold to separate objects from their background is crucial. Otsu's method is a standard non-parametric method in image processing. Furthermore, it is an unsupervised classification technique that automatically selects an optimal threshold, making it a suitable criterion for evaluating the efficiency of other methods (Otsu, 1979).

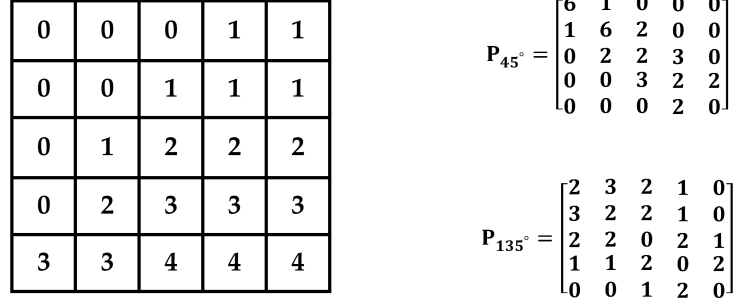


FIGURE A.1: Gray level co-occurrence matrix.

Considering a given image represented by L gray levels, let n_i represent the number of pixels with level i . Therefore, the total number of pixels can be stated as,

$$N = n_1 + n_2 + \dots + n_L \quad (1)$$

Hence, the normalized probability distribution can be written as follows,

$$p_i = \frac{n_i}{N}; \quad p_i \geq 0, \sum_{i=1}^L p_i = 1. \quad (2)$$

Based on Otsu (1979), the optimal threshold k^* is,

$$\sigma_B^2(k^*) = \max_{\{1 \leq k < L\}} \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}; \quad \sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \quad (3)$$

where $\mu_k = \sum_{i=1}^k i p_i$ and $\omega(k) = \sum_{i=1}^k p_i$ are the first-order and the zeroth-order cumulative moments of the histogram up to the k^{th} level, respectively and $\mu_T = \mu(L)$ is the total mean level of the image. An advantage of this method is that its procedure is simple and straightforward, which can be helpful in real-time processing and high-speed measurements. Moreover, the stable optimal threshold is globally selected based on a global property and the exhaustive searching process related to this algorithm (Otsu, 1979).

A.3 Image dilation

Image dilation is one of the fundamental operations in morphological image analysis (Gonzalez, 2006; Gonzalez, Woods, and Eddins, 2009). As a definition, for A and B as sets in Z^2 domain, the dilation

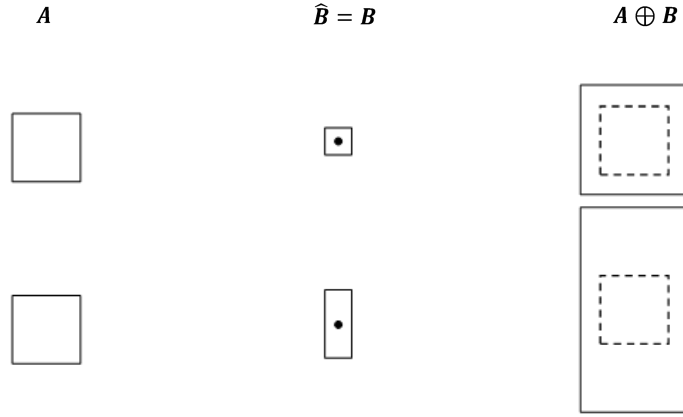


FIGURE A.2: Dilation of set A by set B (Reproduced from Gonzalez, Woods, and Eddins (2009)).

denoted $A \oplus B$ is,

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (4)$$

where we assume that B as a structuring element dilating the set A (image object). Image dilation is a nonlinear operation as it is based on set operations. It grows or thickens objects in a binary image controlled by the structuring element's shape. Figure A.2 shows the results of two different dilations by two different structuring elements (B) of the same image object (A) (Gonzalez, 2006). The dot in B denotes the origin, and the dashed outlines are the boundaries of A for reference.

Bridging the gaps in an image is one of the applications of image dilation. Also, one of the advantages of dilation as a morphological approach compared to the low-pass filtering method is that the dilation gives us a binary image. In contrast, low pass filtering results in a grayscale image, which a threshold function should operate on to return to binary format.