

**UNMANNED AERIAL VEHICLE OBJECT RECOGNITION IN BAD
WEATHER USING
DARK CHANNEL PRIOR AND CONVOLUTIONAL NEURAL NETWORKS**

By

KALOSO TOPIAS

Reg. No: 16100375

BSc (Hons) (Computing- Artificial Intelligence) (Staffordshire University)

Department of Computer Science and Information Systems,
College of Science

Botswana International University of Science and Technology
kaloso.motimedi@studentmail.biust.ac.bw, (+267) 75 690 292

A Thesis Submitted to the College of Science Partial Fulfillment of the Requirements
for the Award of the Degree of Master of Science in Computer Science of BIUST

Supervisor(s): Dimane Mpoeleng, PhD

Department of Computer Science and Information Systems,
College of Science, BIUST
mpoelengd@biust.ac.bw, (+267) 72 972 059

Signature  Date: 28-January-2021

Declaration and Copyright

DECLARATION AND COPYRIGHT

I, _____**Kaloso Topias**_____, declare that this dissertation/thesis is my own original work and that it has not been presented and will not be presented to any other university for a similar or any other degree award.

Signature 

This dissertation/thesis is copyright material protected under the Berne Convention, the Copyright Act of 1999 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of the Provost, on behalf of both the author and the BIUST.

Certification by Supervisor

CERTIFICATION

The undersigned certifies that he/she has read and hereby recommends for acceptance by the **Faculty of Science** a dissertation/thesis titled: **Unmanned Aerial Vehicle Object Recognition In Bad Weather Using Dark Channel Prior And Convolutional Neural Networks**, in fulfillment of the requirements for the degree of Master of Science in **Computer Science** of the BIUST.


.....

Dimane Mpoeleng, PhD

(Supervisor)

Date: 30-Jan-2021

Acknowledgements

I am very thankful to my supervisor, Dr Dimane Mpoeleng, for his consistent support of my MSc study; his contribution and motivation were invaluable.

I am also grateful to all the lecturers in the BIUST Department of Computer Science and Information Systems for sharing their expertise, criticisms and valuable guidance to me, I could not have academically grown this far without their continuous input.

My sincere gratitude is also expressed to my parents, for the unending encouragement, support and attention; they never had the opportunity to write masters' thesis, they made sure I did.

Last, but not least, I thank my husband for his content love and support during my study; God bless you, God bless us.

Dedication

To my darling husband, Hussein, you have shown me the true definition of love. You have always stood by and supported me; you have always encouraged me to believe in myself and spread my wings and never to hold back in pursuit doing what I love.

Thank you for being you.

To my adorable kids, Hafsa and Hamza; If there is one thing I want you to take away from me throughout this research journey this year is that when you do what you love from your heart, do it diligently and whole heartedly, so you get to enjoy the fruits of your efforts. I will forever support your dreams. You two are my reason, my why and my everything!

Table of Contents

Declaration and Copyright.....	2
Acknowledgements.....	4
Dedication.....	5
List of tables.....	8
List of Figures.....	9
List of appendices.....	12
List of abbreviations.....	13
Abstract.....	14
Chapter 1: Introduction.....	16
1.1 Problem Statement.....	16
1.2 Aims of the study.....	18
1.3 Objectives and research questions.....	19
1.3.1. General Objective.....	19
1.3.2 Specific Objectives.....	19
1.3.3 Research questions.....	19
1.4 Contribution of the study.....	20
1.5 Research approach for the study.....	22
1.6 Organization of the thesis.....	23
Chapter 2: Literature review.....	24
2.1 Introduction.....	24
2.2 Dark Channel Prior.....	24
2.2.1 Introduction.....	24
2.2.2 Background.....	26
2.2.3 Image dehazing using Dark Channel Prior.....	28
2.2.4 Limitations of the Dark Channel Prior.....	30
2.2.5 Studies on advances of the DCP image dehazing.....	31
2.3 Convolutional Neural Networks.....	33
2.3.1 Overview of convolutional neural networks.....	33
2.3.2 CNNs as state-of-the-art in object recognition.....	35
2.3.3 CNN Structure.....	39
2.4 Prior studies on object recognition in bad weather.....	46
2.5 Chapter Summary.....	52
Chapter 3: Proposed Method.....	53
3.1 Introduction.....	53
3.2 Proposed method.....	53
3.2.1 Diagrams of the proposed method.....	55
3.2.2 Discussion of the proposed method.....	58
3.3 Method development.....	59
3.3.2 DCP method development.....	59
3.3.2 CNN development.....	65

3.3 Chapter summary	73
Chapter 4: Method experiments.....	74
4.1 Introduction	74
4.2 Overview of the experiment method	74
4.3 Justification for using experimental method	75
4.4 Discussion of experiments	76
4.4.1 Experiments on UAV haze/fog removal using DCP method	76
4.4.2 Experiments on UAV object recognition using CNN	85
4.4.3 Experiments on UAV object recognition in haze/fog using the proposed method.....	90
4.4.4 Computation time of the proposed method	100
4.5 Chapter summary	101
Chapter 5: Presentation and Discussion of Results.....	102
5.1 Introduction	102
5.2 Discussions on UAV fog/haze removal using the DCP method.....	103
5.2.1 Results and discussion of UAV haze/fog removal using the DCP method	103
5.2.2.1 Object visibility level at varied fog densities.....	105
5.2.2.2 Object visibility level at varied object distance in fog/haze	109
5.3 Discussion on object recognition using CNN.....	119
5.4 Discussion on UAV object recognition in fog/haze using the proposed method	126
5.4.1 Discussion of results for UAV object recognition in varied fog densities	127
5.4.1 Discussion of results for UAV object recognition at varied distances in fog.....	132
5.5 Discussion on computation time of proposed method	137
5.6 Summary of findings.....	141
Chapter 6: Conclusions and recommendations	143
Bibliography.....	147
List of appendices.....	163
Appendix 1: CNN programming code	163
Appendix 2: DCP programming code	167

List of tables

Table 1: Research questions of the study	54
Table 2: CNN performance accuracy of related studies	125
Table 3: Computation time (ms) of the DCP sub-operations.....	139

List of Figures

Figure 2.1: Formation of a hazy image	26
Figure 2.2: Dark channel construction of a hazy image	27
Figure 2.3: Atmospheric light estimation	39
Figure 2.4: DCP image dehazing using guided image filter	31
Figure 2.5: DCP dehazing using weighted guided image filter	32
Figure 2.6: Modified DCP versus DCP	32
Figure 2.8: CNN structure/topology	33
Figure 2.9: Sparse connectivity in CNN versus ANN	35
Figure 2.10: Parameter sharing in CNN versus in ANN	36
Figure 2.11: The filter operation in CNN	37
Figure 2.12: Structure of a CNN	39
Figure 2.13: The convolution operation	40
Figure 2.14: The pooling operation in CNN	43
Figure 2.15: Average pooling in CNN	43
Figure 2.16: Image classification task using CNN	44
Figure 2.17: Image dehazing using CLAHE and CURVELET transform	45
Figure 2.18: Image dehazing using atmospheric light estimation	47
Figure 2.19: De-hazing using deep-learning image restoration	48
Figure 2.20: Dehazing using cascaded CNN	49
Figure 3.1: Block diagram of the proposed method	54
Figure 3.2: Visual diagram of the proposed method	55
Figure 3.3: Block diagram of the DCP method	57
Figure 3.4: Formation of a hazy image	58
Figure 3.5: Implementation of dark channel construction	60
Figure 3.6: Python implementation of atmospheric light estimation	61
Figure 3.7: Python implementation of transmission map estimation	61
Figure 3.8: Python implementation of guided filter	62
Figure 3.9: Python implementation of radiance recovery	63
Figure 3.10: Dataset used for training the CNN	64
Figure 3.11: Python implementation of grayscale conversion	64
Figure 3.12: Python implementation of image resizing	65
Figure 3.13: Python implementation of array image conversion	65
Figure 3.14: Python implementation of dataset scaling	65
Figure 3.15: Python implementation of dataset reshaping	66
Figure 3.16: Assigning labels to datasets	66
Figure 3.17: Converting class labels to one-hot encoding	67
Figure 3.18: Shuffling and splitting the dataset	67
Figure 3.19: Python implementation of the CNN model	68
Figure 3.20: Summary of the Proposed CNN model	68
Figure 3.22: Python implementation of model training	69
Figure 3.23: CNN model evaluation	70
Figure 3.24: Predicting recognition on test images	70
Figure 3.25: Predicting recognition on new images	71
Figure 4.1: Apparatus used for fog removal experiments	75
Figure 4.2: Configurations of apparatus for DCP experiments	76

Figure 4.3: Setup of scene for fog removal experiments	77
Figure 4.4: Experiment for light fog removal	78
Figure 4.5: Experiment for medium fog removal	78
Figure 4.6: Experiment for thick fog removal	79
Figure 4.7: Scene setup for UAV fog removal at varied distances	80
Figure 4.8: Light fog removal at $\approx 6\text{m}$	81
Figure 4.9: Medium fog removal at $\approx 9\text{m}$	81
Figure 4.10: Light fog removal at $\approx 9\text{m}$	82
Figure 4.11: Configuration of apparatus for CNN experiments	84
Figure 4.12: Setup of UAV object recognition experiments	85
Figure 4.13: UAV recognition ≈ 4 meters from object	85
Figure 4.14: UAV recognition ≈ 8 meters from object	86
Figure 4.15: UAV recognition ≈ 10 meters from object	86
Figure 4.16: UAV recognition ≈ 12 meters from object	87
Figure 4.17: Configuration of apparatus for UAV object recognition in fog	89
Figure 4.18: Apparatus for UAV object recognition in fog	90
Figure 4.19: UAV object recognition in light fog	91
Figure 4.20: UAV object recognition in medium fog	91
Figure 4.21: UAV object recognition in thick fog	92
Figure 4.22: UAV object recognition in medium fog	92
Figure 4.23: UAV object recognition in light fog	93
Figure 4.24: Setup for UAV object recognition at varied distances in fog	94
Figure 4.25: Recognition at $\approx 6\text{m}$ in light fog	95
Figure 4.26: UAV object recognition at $\approx 9\text{m}$ in fog	95
Figure 4.27: UAV object recognition at $\approx 6\text{m}$ in fog	96
Figure 4.28: UAV object recognition at $\approx 9\text{m}$ in light fog	96
Figure 4.29: UAV object recognition at $\approx 9\text{m}$ in thick fog	96
Figure 5.1: Results for UAV visibility improvement in fog densities using DCP	101
Figure 5.2: Results for UAV light fog removal	102
Figure 5.3: Results for UAV medium fog removal	103
Figure 5.4: Results for UAV thick fog removal	104
Figure 5.5: Performance summary of DCP for UAV fog removal	105
Figure 5.6: Results - object visibility improvement at varied distances	107
Figure 5.7: Results for fog removal at average distances	108
Figure 5.8: Results for fog removal at further distances	110
Figure 5.9: Results for fog removal at furthest distances	111
Figure 5.10: Performance summary of DCP fog removal at varied distances	112
Figure 5.11: Results - image defogging using different methods	114
Figure 5.12: Loss graph for the CNN model	116
Figure 5.13: Accuracy graph for the CNN model	117
Figure 5.14: UAV Object recognition at $\approx 4\text{m}$	118
Figure 5.15: UAV human recognition at $\approx 6\text{m}$	118
Figure 5.16: UAV human recognition at $\approx 10\text{m}$	119
Figure 5.17: UAV ball recognition oat $\approx 12\text{m}$	124
Figure 5.20: UAV object recognition in light fog	125
Figure 5.21: UAV object recognition in medium fog	126
Figure 5.22: UAV object recognition in thick fog	127
Figure 5.23: Performance summary of recognition accuracy in fog	129

Figure 5.24: UAV object recognition at average distance	130
Figure 5.25: UAV object recognition at further distance	131
Figure 5.26: UAV object recognition at furthest distance	132
Figure 5.27: Performance summary of recognition accuracy of proposed method method at varied distances	134
Figure 5.28: Summary of computation time for the proposed method for object recognition in fog using AR-drone	134
Figure 5.29: Summary of computation time (%) for the proposed method executed on AR-drone for object recognition in fog	135

List of appendices

Appendix 1: CNN programming code.....	158
Appendix 2: DCP programming code.....	162

List of abbreviations

UAV	Unmanned Aerial Vehicles
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DCP	Dark Channel Prior
ROS	Robotic Operation System
RGB	Red Green Blue color scheme
GPU	Graphical Processing Unit
R-CNN	Region Convolutional Neural Network
YOLO	You Only Look Once
CDR	Correct Detection Rates
CLAHE	Contrast Limited Adaptive Histogram Equalization

Abstract

Object recognition using Unmanned Aerial Vehicles (UAVs) is increasingly becoming more useful. Tremendous success has been achieved on UAV object recognition in clear weather conditions where adequate illumination makes it easier for UAVs to recognize objects in the scene. Unfortunately, for outdoor applications, there is no escape from bad weather moments such as haze, fog, dust, smoke and smog. These weather nuisances occur due to suspended particles in the atmosphere, ultimately resulting in degraded visibility. Thus, these weather nuisances cause unsatisfactory performance in UAV object recognition. Current UAV object recognition algorithms do not guarantee satisfactory performance in bad weather conditions. Therefore, this study was motivated by the need for UAV object recognition systems that can perform robustly despite the state of the weather.

Several state-of-the-art methods exist for object recognition and image dehazing/defogging. Nonetheless, the performance of these methods is dependent on the scenarios where they are used. In this study, a novel method that deployed the Dark Channel Prior (DCP), for scene dehazing/defogging; and Convolutional Neural Network (CNN) for object recognition; was proposed and investigated in the context of UAV for object recognition in bad weather.

The aim of the study was to investigate the proposed method for enabling the UAV to efficiently recognize objects in bad weather conditions such as fog, haze, smoke and smog. The proposed method was experimented to determine the extend at which it can enable the UAV recognize objects in fog/haze weather. The objective of the experiments was to investigate the performance of the proposed method for addressing UAV object recognition in bad weather by observing two independent variables, namely; (1) fog density, which is the measure of fog present in the scene and (2) distance of object from the UAV, in fog.

Analysis of results demonstrated that the DCP method effectively addresses UAV visibility improvement in bad weather conditions. On varied densities of haze/fog, the DCP method enables the UAV to effectively dehaze/defog scenes and improve

visibility of objects present in the scene. Additionally, analysis of results illustrated that the constructed CNN model can enable the UAV to accurately recognize objects from the dehazed/defogged scenes with a confidence accuracy of 94.3%.

Chapter 1: Introduction

Achieving a robust visual object recognition for UAVs is a challenging task, especially in bad weather conditions typically haze, fog, smoke, volcanic dust and smog. These conditions make it difficult for the UAV camera to cope with object recognition tasks because images captured suffer some degradation. These weather conditions can greatly reduce visibility, thus causing obscurity of the objects present in the scene. “This occurs mainly due to existence of small particles in the atmosphere; these particles deviate the light reflected off an object from its path of propagation”, (Santra and Chanda, 2015). “Also the environmental illumination incident on these particles gets scattered in the direction of the camera creating a semi-transparent layer, thus images lose contrast and their color shifts towards the color of the atmospheric light”, (Santra and Chanda, 2015). Images captured in these conditions are low quality and as a result are ineffective for object detection and recognition tasks.

Object recognition in bad weather conditions is highly desired for various computer vision applications such as surveillance (Kanellakis and Nikolakopoulos, 2017), search and rescue missions (Waharte and Trigoni, 2010), inspection (Morgenthal and Hallermann, 2014), detection (Bhaskaranand and Gibson, 2011), (Tomic et al., 2012), (Merino et al. 2016), (Sa et al., 2015) to detect and recognize target objects from a given imagery scene. However, there is lag in UAVs that can perform object recognition in bad weather conditions; therefore this study exist to address this problem.

1.1 Problem Statement

Like many computer vision applications, UAVs are faced with the challenge of alleviated visibility in bad weather conditions that cause obscurity of the objects present in the scene. In reality, it is desirable to have UAVs execute object recognition missions robustly despite conditions such as fog, haze and smog. Currently, in fog, haze, and smog weather conditions, UAVs are incapable of doing a range of

applications; on these conditions, UAVs cannot be sent out to execute missions that require object recognition such as search and rescues during disasters, poaching detection, detection of oil spills in dams, surveillance of borders, to mention a few. This is because images presented by the UAV camera in these weather conditions usually suffer degradation thus affecting the accuracy of information collected by the UAV camera (Forest Compass 2017).

Therefore, to date, the use of UAVs for object recognition is wholly dependent on the state of the weather conditions despite the fact that UAVs were primarily invented to perform tasks that are beyond human capability.

Object recognition using UAVs have been investigated in research by many studies, There is vast literature, algorithms and methods on UAV object recognition , for example (Lee et al., 2017) and (Xue et al., 2018). . Nonetheless, experiments done in these studies were conducted in clear weather conditions; they were mostly tested indoors where there is controlled illumination/lighting and certainly these experiments did not have to cope with the problem of dealing with poor visibility imagery scenes captured in bad weather conditions.

Precisely, all current UAV object recognition literature is under the assumption that the UAV camera feed is always captured in clear visibility conditions. In essence, these studies did not address UAV object recognition in bad weather conditions such as haze, fog and smog.

This study is driven by the desire to have UAV object recognition systems that find target objects in the real world from an image of the world and interprets these images the way human beings do by classifying the detected object in bad weather conditions. Achieving such object recognition systems for UAVs would widely extend the possibility.

The motivation for this study was inspired by limitations in prior studies for addressing UAV object recognition in bad weather conditions; a high prevalence of similar

paradigms reveals literature that is towards UAV object recognition and classification in clear visibility weather conditions rather than in bad weather.

1.2 Aims of the study

Traditionally, UAVs were not to be flown in adverse weather conditions such as rain, fog, mist to mention a few. Nonetheless, recent UAV technologies enable them to be operated in these weather conditions. For example, DJI revealed the Phantom 4 can be operated in good to moderate weather condition except in severe and extreme weather conditions such as extreme winds, snowy conditions, rain and hail (dji.com, 2016). Meanwhile, with further advances in UAV technology, contemporary UAVs innovations such as an all-weather DJI's Matrice 200 can withstand hostile weather conditions such as rain, strong winds and snow (dji.com, 2018).

This is a breakthrough for many research opportunities and the interest of this study therefore, was to leverage these technological advances in UAVs by proposing a method that enables UAVs to perform object recognition in bad weather conditions such as fog and haze.

The main aim of this study was to address the limitations in UAV object recognition in bad weather conditions by proposing a method that addresses the problem in context. The aim of the study was to propose a method that adopts the DCP method as the state-of-the-art for image enhancement in conditions such as haze and fog. The DCP method has achieved phenomenal benchmarks on numerous studies marking its efficiency to address the problem of visibility enhancement in haze/fog conditions.

Additionally, the aim of this study is to exploit CNNs because they are appealing frameworks for image recognition. This is because they are biologically inspired tools for learning complex mappings from a set of large examples of datasets. CNNs are simple but very precise architectures and this makes them perfect tools for solving difficult pattern and image recognition tasks (O'Shea and Nash, n.d.). With the immense growth in their computation power, it is feasible to exploit CNNs for the task of UAV object recognition. CNNs, by far, have proven to be the state-of-the-art in the

object recognition domain (Krizhevsky et al., 2012) and (Hinton et al., 2006). “Since their emergence in the early 1960s, neural networks have achieved greater performance on a number of image recognition benchmarks, including the ImageNet Large-Scale Visual Recognition Challenge” (Hinton et al., 2006).

With the proposed the method for UAV object recognition in bad weather conditions, the study further aimed to experiment and present the results of the proposed method. The aim of the study was to discuss and analyse the proposed method whilst presenting its limitations and noting its prospect further work of improvement.

1.3 Objectives and research questions

1.3.1. General Objective

The general objective of this study is to propose a method for UAV object recognition in bad weather conditions using the DCP method and CNN.

1.3.2 Specific Objectives

The specific objectives of this study were:

1. To propose and evaluate a DCP method for enhancing the visibility of UAV scenes in bad weather conditions.
2. To implement and evaluate a CNN for recognition of objects using UAV
3. To propose and evaluate a method for UAV object recognition in bad weather using the DCP method and CNN
4. To determine the performance efficiency of the proposed method in terms of computation time.

1.3.3 Research questions

The research questions for this study were as follows:

- 1. How effective is the DCP method for enabling the UAV enhance scenes captured in bad weather conditions?***

This research question was designed to determine the effectiveness of the DCP method in addressing the problem of UAV visibility improvement in bad weather conditions.

2. How effective, in terms of accuracy, is the developed CNN model for UAV object recognition?

This research question was intended to determine the ability of the developed CNN to enable the UAV to accurately recognize objects.

3. How effective is the proposed method for addressing UAV object recognition in bad weather conditions?

This research question was intended to determine the effectiveness of the proposed method for addressing the problem of UAV object recognition in bad weather conditions.

4. How efficient is the proposed method for enabling UAV object recognition in bad weather conditions in terms of computation time?

This research question was intended to determine the efficiency of the proposed method for achieving results for UAV object recognition in bad weather conditions in terms of computation time.

1.4 Contribution of the study

The aim of this study was to make a contribute to the body of knowledge in research in the following ways:

Firstly, the study seeks to propose a method that addressed the problem of UAV object recognition in bad weather conditions; methods for solving UAV recognition in bad weather conditions are seldomly investigated and researched, marking the study possibly the first of its kind to address the problem in context. Currently, there is limited evidence of any literature or research work that attempted to address the

problem of UAV object recognition in bad weather conditions such as fog, haze, volcanic dust and smog.

Secondly, the study experimented the dehazing method, the DCP in this case, for enhancing the camera scene captured in bad weather conditions using the UAV. This was experimented to observe the efficiency of the DCP method for enabling the UAV improve visibility in conditions such as fog, haze and smog. In the interest of this study, this is contribution to the body of knowledge because there is no evidence of the prior work for investigating a scene dehazing method for the problem of UAV object recognition in bad weather conditions.

Thirdly, the proposed method, which integrated the DCP method and CNN, was experimented on the UAV environment for the task of visibility improvement on images taken on fog/haze/dust/smog weather conditions; the specialty here lies in that the UAV environments are rather unique; they operate in the Robotic Operating Systems, they use wi-fi to communicate and they are run on batteries. Hence, the proposed method was tested and analysed for its performance on the in terms of its ability to enhance visibility on UAV scenes captured in bad weather, accuracy to recognize objects, computation time; thereby contributing all these profiles to the body of new knowledge.

Ultimately, the proposed method will benefit many UAV applications which are currently impossible in bad weather because if practically implemented in UAVs, applications such as object recognition, object tracking, search and rescue missions, surveillance, to mention a few, will become possible in bad weather. Government departments and commercial industries could use this method to extend the use of UAVs for various missions such as boarder patrols, wildlife monitoring, poaching patrols, criminals tracking, fire rescues, aerial photography and filming, to mentions a few, in bad weather moments such as haze, fog, smoke and smog.

1.5 Research approach for the study

The research approach used in this study is as follows:

1. Review of the state-of-the-art

The two state-of-the-art in this study were the DCP method and CNN. Therefore, a comprehensive literature review of the two state-of-the-art aimed at reviewing the major theories, backgrounds and related works.

A review of the state-of-the-art was to ultimately give insight on this study towards establishing its proposed approach with systematic methods for addressing the problem in context.

2. Implementation of the proposed method

The proposed method was presented and implemented. The proposed method which deployed the DCP method for image enhancement and CNN for object recognition was implemented by executing program codes on the UAV for object recognition in fog/haze.

3. Experimentation and testing

Subsequently, the proposed approach was experimented and tested to observe its performance for object recognition in bad weather conditions. Sets of procedures for experiments were identified and followed to execute experiments which eventually led to establishing answers to the research questions of this study.

4. Analysis of the results

Results of the experiments were presented and analyzed qualitatively and quantitatively to extract answers to the research questions of the study.

5. Conclusions

Based on the analysis of the results of the proposed method, conclusions that provided answers to research questions were drawn about the performance of the proposed method.

1.6 Organization of the thesis

This thesis is organized in the following approach:

Chapter 1: Introduces the topic in depth, presenting the problem, aims and objectives of the study.

Chapter 2: Presents an extensive literature review for the state-of-the-art (DCP and CNN), by discussing the current works, and similar studies and ultimately identifying the strengths and weaknesses of the prior related studies.

Chapter 3: Outlines the proposed method for addressing the problem in context.

Chapter 4: Discusses the set of procedures followed to execute the experiments used as methods to establish answers to research questions of the study.

Chapter 5: Presents and discusses the results of the experiments.

Chapter 6: Discusses the conclusions drawn from the study and recommendations for further work.

Chapter 2: Literature review

2.1 Introduction

This chapter presents the review of the literature of the state-of-the-art for this study. In this chapter, the Dark Channel Prior (DCP) method for image de-fogging/de-hazing is introduced and discussed. Some experimental results of this approach are demonstrated.

Similarly, this chapter also presents literature review on CNNs, as the state-of-the-art for object recognition by presenting their background in object recognition, their motivation in object recognition problems as well as their structure.

Existing literature on related works on de-hazing and de-fogging methods for image visibility enhancement to achieve object recognition in bad weather conditions were reviewed and evaluated by identifying the practical achievements and limitations of these current methods and algorithms.

2.2 Dark Channel Prior

2.2.1 Introduction

Images taken outdoors are ordinarily degraded by phenomena such as fog, haze and smog due to the atmospheric absorption and scattering. Therefore “this causes the irradiance received by the camera from the scene point to be reduced along the line of

sight, and also the incoming light is blended with air-light which contains some particles/ water droplets”, (Geethu et al., 2016). Therefore the degraded image loose contrast, making scene visibility difficult for cameras and even human beings.

The main aim of image dehazing/de-fogging is to recover the scene radiance to achieve visually pleasing haze-free images. Numerous image dehazing/defogging methods are anchored on the model of formation of hazy image depicted on figure 2.1. However, “many studies have demonstrated the challenge in the model-based haze removal schemes to appropriately estimate the model parameters, such as the global atmospheric light and the transmission map”, (Hsieh et al., 2018).

Therefore the DCP, now considered state-of-the -art scene dehazing scheme, was proposed. Since its introduction, the DCP has achieved remarkable results in image dehazing and defogging problems. This is mainly attributed to that the DCP method has enabled an efficient way to achieve the transmission map estimation. The DCP method of dehazing uses some statistics information depicted by haze-free outdoor images. The DCP theory lies in that “in most of the local patches or regions which do not cover the sky, haze-free outdoor images often contain some pixels (called ”dark pixels”) have very low intensity in at least one RGB channel and the minimum intensity in such a patch should have a very low value”, (He et al., 2011). He et al., (2011) discovered that the dark channel of an image can be calculated by the equation:

$$\mathbf{J}^{\text{dark}}(x) = \min_{c \in \{r,g,b\}} (\min_{y \in \Omega(x)} (\mathbf{J}^c(y))), \quad (1)$$

Where:

$\mathbf{J}^c$ denotes a color channel of $\mathbf{J}$

$\Omega(x)$ denotes a local patch centered at x

The DCP is based on “that except for the sky regions, if $\mathbf{J}$ is a haze/fog free outdoor image, then intensity of $\mathbf{J}^{\text{dark}}$ is low and tends to be zero; and $\mathbf{J}^{\text{dark}}$ is called the dark channel of $\mathbf{J}$ and this statistical observation is called the dark channel prior”, (He et al., 2011). The DCP is also anchored on the attribute that “in haze images, the

intensity of these dark pixels in that channel is mainly contributed by the air-light, therefore using this prior, is it possible was able to make an estimation of the amount of haze or the thickness of haze and eventually recover scene radiance of the image” (He et al., 2011). It is this attribute about the DCP that makes it outstanding for addressing image dehazing problems.

2.2.2 Background

The idea behind dark channel prior was inspired by a particular technique widely used in multi-spectral remote sensing called the dark-object subtraction. For example, Chavez (1988) removed “spatially homogeneous haze by subtracting a constant value corresponding to the darkest object in the scene”. Therefore based on this, He et al., (2011) used this idea and proposed the DCP algorithm for natural image de-hazing.

The following equation is used to describe the atmospheric scattering model that depicts how a fog image is formed:

$$\mathbf{I}(x) = \mathbf{J}(x) t(x) + \mathbf{A}(1-t(x)) \quad (2)$$

(Narasimhan et al., 2002) and (Tan, n.d.)

Where;

x denotes scene pixels

$\mathbf{I}$ denotes the hazy image

$\mathbf{J}$ denotes haze-free image

t denotes the medium transmission

$\mathbf{A}$ denotes the global atmospheric light.

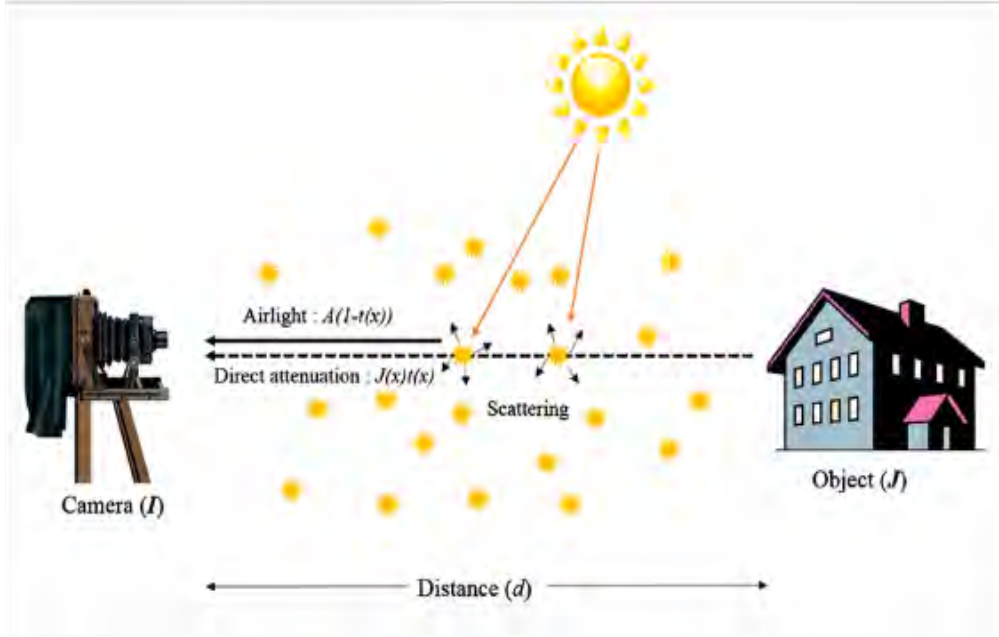


Figure 2. 1: Diagram depicting how a hazy image is formed (Lee et al., 2016)

The main purpose of scene dehazing using equation in (2) is to recover $\mathbf{J}$, $\mathbf{A}$ and t from $\mathbf{I}$ using the following equation by Tao et al., (2011):

$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}}{t(x)} + \mathbf{A} \quad (3)$$

In the equation, “ $\mathbf{J}(x) t(x)$ is called direct attenuation, which describes the scene radiance and its decay in the medium; and $\mathbf{A}(1-t(x))$ is called air-light, which results from previously scattered light and leads to the shift of the scene color”, (Narasimhan and Nayar, 2002).

Tao et al., (2011) further expressed that the transmission $\mathbf{t}$, can be obtained by the following equation in (4) for when the atmosphere is homogeneous:

$$t(x) = e^{-\beta d(x)} \quad (4)$$

“where β is the scattering co-efficient of the atmosphere - it indicates that the scene radiance is attenuated exponentially with the scene depth d ”, (Tao et al., 2011)

2.2.3 Image dehazing using Dark Channel Prior

He et al., (2011) explains that for image dehazing, the DCP algorithm first constructs a dark channel from the given input/hazy image using the equation (1). Figure 2.2 shows the construction of a dark channel from a hazy image.

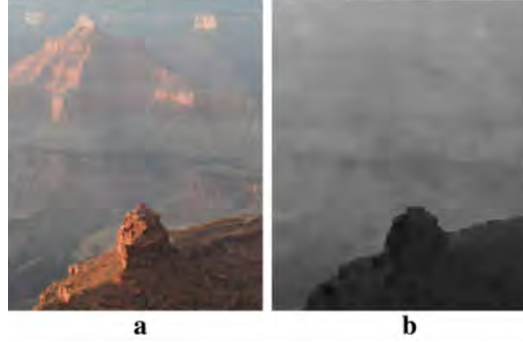


Figure 2. 2: dark channel of a hazy image, (He et al., 2011)
(a) hazy image (b) dark channel of (a)

This operation is then followed by obtaining the atmospheric light and transmission map from the dark channel. The transmission map is further refined and subsequently the haze-free image is reconstructed using equation (3).

According to Lee et al., (2016), given the degradation model of:

$$\mathbf{I}(x) = \mathbf{J}(x) t(x) + \mathbf{A}(1-t(x)), \quad (5)$$

The minimum intensity in the local patch of each color channel is taken after dividing both sides of equation (5) by A^c as follows:

$$\min_{y \in \Omega(x)} \frac{I^c(y)}{A^c} = \tilde{t}(x) \min_{y \in \Omega(x)} \frac{J^c(x)}{A^c} + (1-\tilde{t}(x)) \quad (6)$$

Here, He et al., (2011) assumes the transmission in the local patch $\Omega(x)$ is to be constant and is represented as $\tilde{t}(x)$. Therefore, the min operator of the three color channels can be applied to equation (6) as follows:

$$\min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left(\min_c \frac{J^c(y)}{A^c} \right) + (1-\tilde{t}(x))$$

(7)

According to the DCP approximation of dark channel values are approximately zero, where $J^{\text{dark}} \approx 0$, $\tilde{t}(x)$ can be represented as:

$$\tilde{t}(x) = 1 - \min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right)$$

(8)

Lee et al., (2016) explained that in order to obtain the transmission map $\tilde{t}$, it is necessary to first estimate the atmospheric light A .

In real images, the brightest pixel represents white objects. However, the DCP algorithm uses haze-opaque pixels obtained after converting a hazy image into the dark channel to estimate the atmospheric light A . For example Xiao and Gan (2012)

used the pixel with highest intensity as the atmospheric light. “The pixel value of the dark channel is highly correlated with haze density” (Lee et al., 2016). He et al., (2011) selected the top 0.1% of the brightest pixels first, and used the color with the highest intensity value among the selected pixels as value for the atmospheric light intensity, A .

Figure 2.3 demonstrates the process used to estimate the atmospheric light, A as proposed by He et al., (2011).

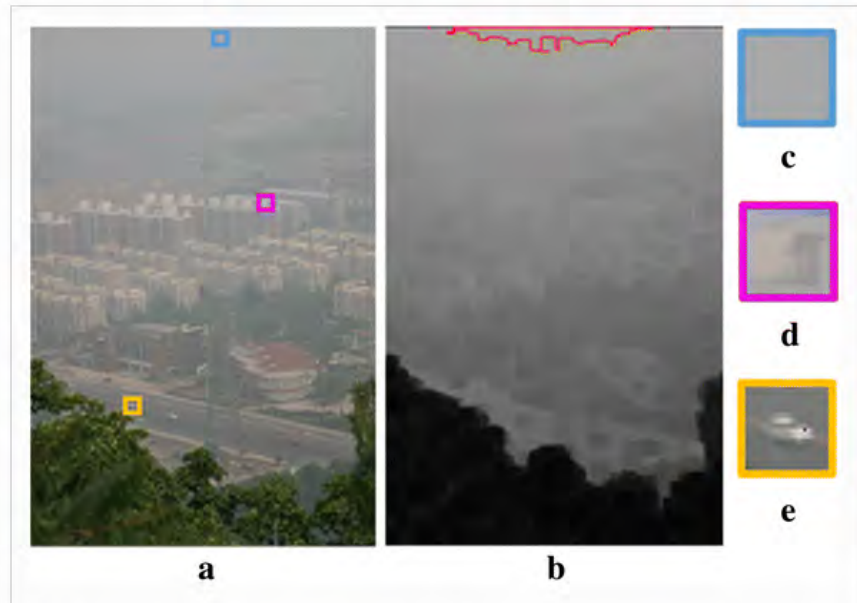


Figure 2. 3: “(a) Hazy image.(b) Dark channel: the region inside the *red boundary lines* corresponds to the most haze-opaque region. (c) Patch used to determine the atmospheric light. (d)

and (e) Patches that contain intensity values higher than that of the atmospheric light.” (He et al., 2011)

The DCP is highly uncertain in the regions of the sky as claimed by He et al., (2009). He et al., (2009) further claims that “fortunately, the color of the regions of the sky resembles the atmospheric light $\mathbf{A}$, in hazy images, and therefore we can have”:

$$\min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right) = 1 \text{ and } \tilde{t}(x) = 0 \quad (9)$$

The above equation in (9) “corresponds to the definition of $\mathbf{t}(x)$ because $\mathbf{d}(x)$ approaches infinity for the sky region”, (He et al., 2011). Therefore no special consideration is required for sky regions to estimate the transmission map since $\tilde{t}(x)$ can be achieved as equation (8). He et al., (2011) revealed that given the values of atmospheric light $\mathbf{A}$, transmission map $\mathbf{t}$, and the observed radiance/haze image $\mathbf{I}$, the haze-free image can be expressed using the following equation as:

$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A \quad (10)$$

Where:

t_0 denotes the transmission map lower bound

2.2.4 Limitations of the Dark Channel Prior

The DCP is an approached based on a strong assumption that images captured in clear weather conditions, and are haze-free, tend to contain pixels in them with low color intensities in at least one RGB channel (He et al., 2009). Therefore with this prior information about the haze image model, it is feasible to precisely make an estimate of the haze thickness and thus recover dehazed images. Nonetheless, like any computation based on a specific assumption, the DCP has its own limitations which are often identified as follows;

- Artifacts may occur in sky regions: The DCP may not be valid “when the scene object is inherently similar to the atmospheric airlight such as the sky regions”,(He et al., 2009)
- “Halos may occur in large depth discontinuities”,(He et al., 2009)
- Color distortions is highly likely; extreme color saturation and high contrast colors are often found in recovered images. The DCP may be invalid on some extreme cases of haze(Lee et al., 2016).
- “The soft matting algorithm used to refine the initial transmission map usually results in a high computational cost”,(He et al., 2009)

2.2.5 Studies on advances of the DCP image dehazing

He et al., (2011)’s DCP image dehazing scheme, as discussed earlier, has received great attention by studies for image dehazing. It has been used by many studies as the template scheme for image haze and fog removal. However, because of the limitations of the DCP method discussed in section 2.2.5, numerous studies have prevailed; demonstrating a lot of variations to improve the DCP image dehazing method.

DCP image dehazing using guided image filtering

In contrast to their original DCP approach, He et al., (2013) replaced the soft matting algorithm with the guided image filter for the transmission map refinement. The guided image filter had two parameters; (1) window size used for filtering and (2) window for smoothing the details of the image. The study pointed out that though this approach was efficient for computational time, in some cases the recovered images by this variation may reveal some halo and color distortions in them. Results of this method variation using the guided image filter are presented in figure 2.4.



Figure 2.4: Results of DCP image dehazing using guided image filter. Left: original image. Right: Output image (He et al., 2013)

Weighted Guided Image Filtering

Li et al., (2015) proposed a weighted guided image filter to improve the results by DCP image dehazing using guided image filtering by He et al., (2013) in figure 2.5. In this method, the weighted guided image filter was used in a similar way to the DCP approach for single image haze removal, however varied in the following ways: (1) the guide image was used as the luminance of the input hazy image and (2) a power function was used for an adaptive adjustment the final transmission map. However, the method achieved results which contained some blurry details as shown in figure 2.5 and is considered ineffective compared to results by He et al., (2013) using the guided image filter in figure 2.4.



Figure 2.5: Results of DCP image dehazing using weighted guided image filter by Li et al.,(2015)

A Modified DCP Based Dehazing Algorithm

Hsieh et al., (2018) proposed a modified DCP scheme that followed similar steps for the DCP by He et al., (2013) except modifications on both the atmospheric light estimations and transmission map estimations were made. “For atmospheric light estimation A , a pixel-based dark channel is employed with an adaptive scaling factor”. To estimate the transmission map, the method used an “adaptive scaling factor to replace the fixed scaling factor to find the initial transmission map”, (Hsieh et al., 2018). Additionally, contrary to the DCP method by He et al., (2013), where the guiding image was based on the input hazy image, the proposed method employed

“the pixel-based dark channel as the guide image in the guide image filtering”, (Hsieh et al., 2018). Figure 2.6 presents the results of the modified DCP method in comparison with DCP by He et al., (2013).

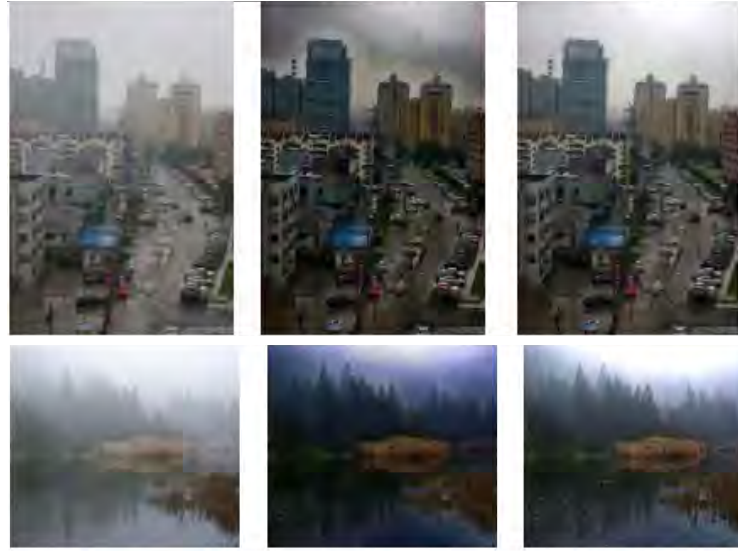


Figure 2.6: Results of the modified DCP versus DCP

Left: Hazy images **Center:** DCP dehazing **Right:** Modified DCP

Results in figure 2.6 verified that the DCP achieved better quality images compared to modified DCP images.

2.3 Convolutional Neural Networks

2.3.1 Overview of convolutional neural networks

“Convolutional neural networks are a specialized kind of neural network for processing data that has a known, grid-like topology”, (Goodfellow et al., 2016), as shown in figure 2.8. “CNNs are considered powerful image processing and artificial intelligence (AI) tools that use deep learning to perform both generative and descriptive tasks, often used in machine vision tasks in image and video recognition”, (searchenterpriseai.techtarget.com 2018).

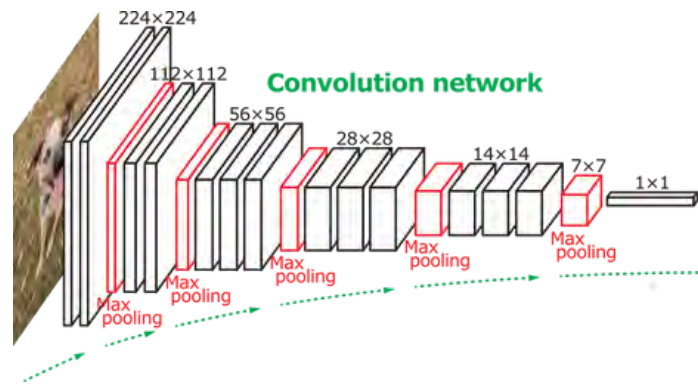


Figure 2.8: CNN structure/topology (Singh et al., 2016)

“The term convolution in CNNs indicates that the network employs a mathematical operation called convolution which is a specialized kind of linear operation and these are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers”, (Goodfellow et al., 2016). “CNNs are considered deep models in which trainable filters and local neighborhood pooling operations are applied alternately on the raw input images, resulting in a hierarchy of increasingly complex features”, (Mobahi et al., 2009).

CNNs have been remarkably successful in object recognition and computer vision applications; they have gained tremendous recognition in domains such as image processing, achieving remarkable accuracy performance as compared to other frameworks (Kim, 2014); particularly due to their capability to be invariant to certain variations presented in the images such as illumination, object clutter as well as pose (Mobahi et al. 2009).

Essentially, CNNs have achieved the best possible correct detection rates also known as CDRs in image recognition applications. For famous challenges such as MNIST database of handwritten characters, CNNs scored a CDR of 99.77% (Cireşan et al., 2012), a CDR of 97.47% with the NORB dataset of 3D objects (Ciresan et al., 2011); while a “CDR of 97.6% on ~5600 images of more than 10 objects”, (Lawrence et al., 1997). According to Russakovsky et al., (2015) CNNs are so powerful that they are even capable of outperforming the human brain in object classification such as “classifying objects into fine-grained categories such as the particular breed of dog or species of bird”.

Therefore, CNNs are remarkable at learning hierarchical representations. In vision, low-level features reflect basic visual concepts such as edges; and mid-level features represent composite visual elements including circles and simple textures; while high-level features represent visual objects (Liang, 2017). These hierarchical feature representations can be learned based on end-to-end training of the CNN (Ma et al., 2017)

2.3.2 CNNs as state-of-the-art in object recognition

Ideally, CNNs are also inspiring for object recognition task because of their capacity to leverage important ideas important for machine learning systems; sparse interactions, parameter sharing and equivariant representations (Ma et al., 2017). These ideas are discussed as follows:

- **Sparse interaction**

Sparseness is a concept of efficiency in neural networks, and exists in two variants in the following context ; “the sparse activity property means that only a small fraction of neurons is active at any time”, (Laughlin and Sejnowski, 2003) while sparse connectivity refers to sparse connections between consecutive layers within the network (Ma et al., 2017). Sparse connectivity/interaction and the sparse activity have been observed in the brains of mammals (Hubel and Wiesel, 1962), (Olshausen and Field, 2004), (Mason et al., 1991), (Markram et al., 1997) and since then, it has motivated various machine learning algorithms such as neural networks.

Whilst “traditional neural network layers use the matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input layer and each output unit, this means that every output unit interacts with every input unit”, (Goodfellow et al., 2016). The sparse interaction property in CNNs however, means that each neuron in the structure is linked to only certain other neurons, which are limited in number (Laughlin and Sejnowski, 2003). CNNs typically have sparse interactions which are made possible by making the filter relatively smaller than the input image (Goodfellow et al., 2016). According to Goodfellow et al.,(2016) CNNs can detect small yet detailed information such as kernels that cover hundreds of pixels

out of the millions of pixels in the input image; this illustrates just how the network only uses fewer parameters, and the benefits of this include the following:

- “Memory requirements are reduced; thus, the statistical efficiency of the network is improved”
- “Computing the output requirements requires fewer operations”

The graphical illustration of sparse connectivity is shown in figure 2.9

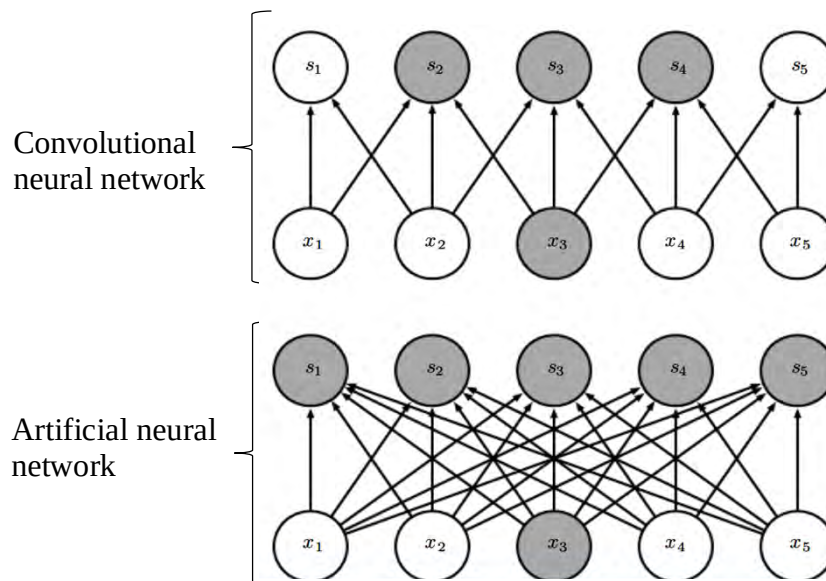


Figure 2.9: Sparse connectivity in CNN versus regular neural network (i.e. MLP) (bottom), (Goodfellow et al., 2016)

As depicted in figure 2.9, for both networks (CNN and ANN) an input neuron, x_3 , is highlighted, also the affected neurons in the output layer denoted by s are highlighted. Figure 2.9 depicts that in a CNN, not all neurons in the next layer are connected to the neuron in the input layer, this demonstrates the sparsity attribute of the CNN. Nevertheless, “in a typical ANN, when s is formed by matrix multiplication, connectivity is no longer sparse but nodes are fully connected and so all of the outputs are affected by x_3 ”, (Goodfellow et al., 2016).

- **Parameter sharing**

A single function in a CNN model can share and use the same parameter (Goodfellow et al., 2016). Convolutional layers have shared parameters (or tied

weights), in every output neuron in a given slice uses the same set of parameters in the filter (Hubel and Wiesel 1962). Parameter sharing is used to control the number of parameters used in different network layers; if it turns out that the number of parameters are too high, CNNs can “dramatically reduce the number of parameters by making one reasonable assumption: that if one feature is useful to compute at some spatial position (x,s) , then it should also be useful to compute at a different position (x_2,s_2) ”, (Quora.com, n.d). Figure 2.10 (top) depicts the idea of parameter sharing in a CNN.

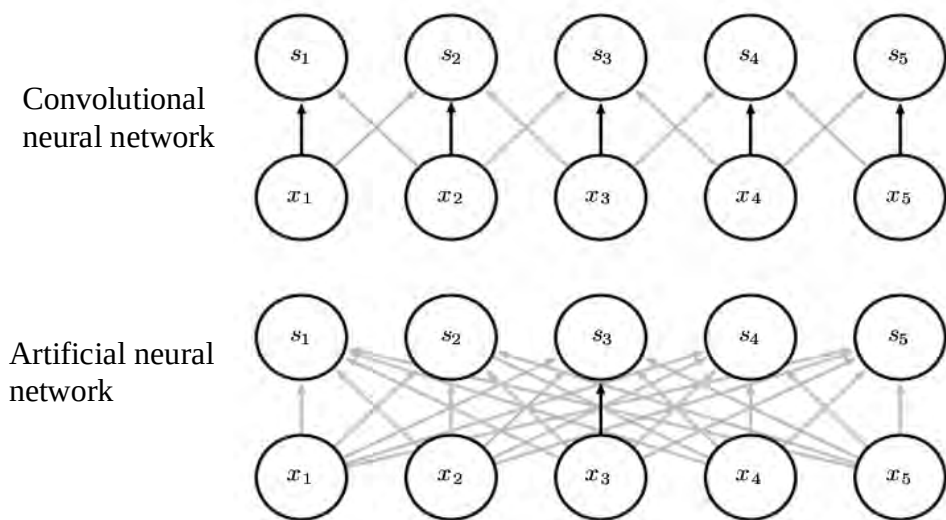


Figure 2.10: Parameter sharing in CNN versus in ANN (Goodfellow et al., 2016)

In figure 2.10, the darker arrows illustrates the inter-links that use a certain parameter .

In an ordinary artificial neural network, each weight matrix is triggered or activated once and never reused for calculating the output of another layer in the network (Goodfellow et al., 2016), as demonstrated in figure 2.10 (bottom). “As a synonym for parameter sharing, it can be said that a network has tied weights, because the value of the weight applied to one input is tied to the value of a weight applied elsewhere”, (Goodfellow et al., 2016). In contrary to ANNs, CNNs have filters and each value within the filter is applied at all pixel points of the input image (Goodfellow et al., 2016), see figure 2.11. Therefore this means that the network can learn only one set of parameters.

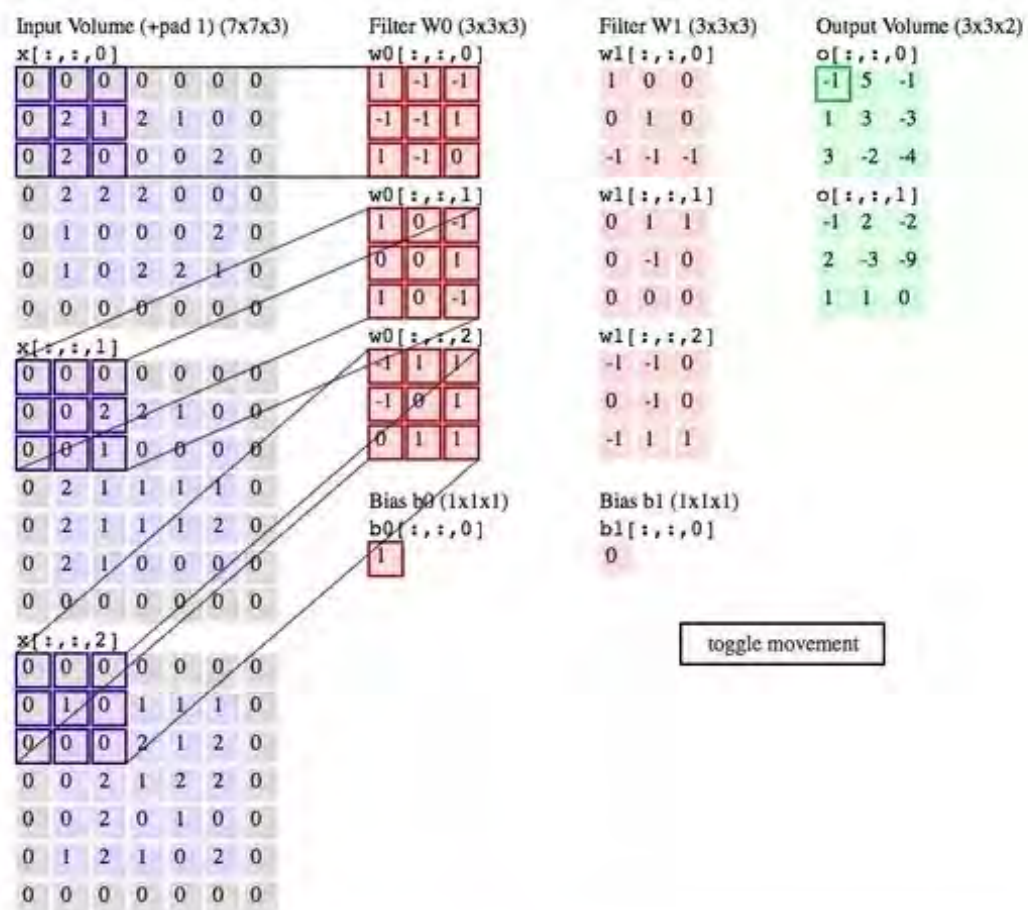


Figure 2.11: The filter operation in CNN: input layer is x, The “Filter W0” is the “weight shared” (Quora 2018)

Ideally, CNNs for object recognition leverage parameter sharing idea for the following advantages:

- Shared parameters reduce the storage requirements of the CNN model. “In the case where a fully connected layer is used to extract the features, the input image of size 32x32 and a hidden layer having 1000 features will require an order of 106 coefficients, a huge memory requirement”, (Hijazi et al., n.d.).
- In the convolutional layer, the same coefficients are used across different locations in the space, so the memory requirement is drastically reduced (Hijaz et al., n.d.). Convolution is thus dramatically more efficient than dense matrix multiplication used by the traditional neural networks, in terms of the memory requirements and statistical efficiency (Goodfellow et al., 2016).

- “Parameter sharing can dramatically improve the efficiency of a linear function for detecting edges in an image”, (Goodfellow et al., 2016).
- Furthermore, “using the traditional neural network that would be equivalent to a CNN, because the number of parameters would be much higher, the training time would also increase proportionately (Hijaz et al., n.d.). In a CNN, since the number of parameters is drastically reduced, training time is proportionately reduced”, (Hijaz et al., n.d.).

- **Equivariant representations**

“Natural images have many statistical properties that are invariant to translation”, (Srihari, n.d.). For example, an image of a cat remains an image of a cat even if it translated to the left. By leveraging parameter sharing across multiple image locations, CNNs handle this exceptionally well (Srihari, n.d.). CNNs have a certain technique of parameter sharing; thus it is through this parameter sharing technique that CNNs have the property of translation to equivariance; this means that CNNs deploy special functions that ensure that the changes in the input will result in same changes being made to the output (Hijaz et al., n.d.). “This is accomplished by shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects)”, (Cohen and Welling, 2016). In other words, “the symmetry (translation) is preserved by each layer, which makes it possible to exploit it not just in the first, but also in higher layers of the network”, (Cohen and Welling, 2016). “Ideally, a function $f(x)$ is equivariant to a function g if $f(g(x)) = g(f(x))$, thus in the convolution operation, if g is any function that translates the input, then the convolution function is equivariant to g ”, (Goodfellow et al., 2016).

2.3.3 CNN Structure

In this study, CNN structure primarily focuses on the input being a series of image scenes. CNNs are made up of three distinct layers; the convolutional layers, pooling

layers and fully-connected layers, respectively. According to (O'Shea and Nash, n.d.), a CNN architecture or structure is built when the convolutional, polling and fully-connected layers stacked in that respect. The structure of a CNN is depicted by figure 2.12.

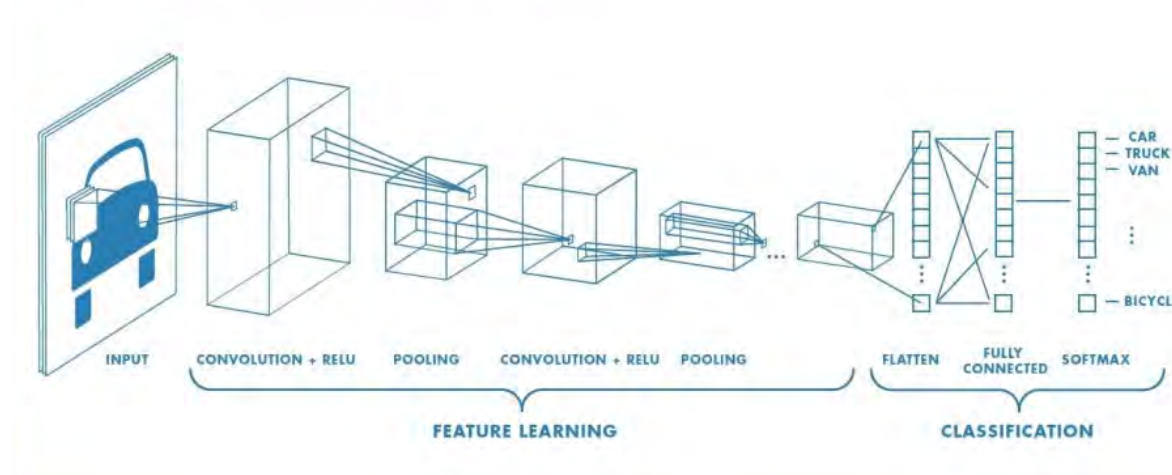


Figure 2.12: Structure/architecture of a convolutional neural network (O' Shea and Nash, n.d.)

The CNN structure is classified into three segments which are the input layer, the feature-learning layers and the classification layers, (figure 2.12). According to Lin and Shen, (2018), the input layer consists of the input data (usually image pixels for vision purposes); the feature learning part consists of two layers; convolution + activation layer and pooling layer; while the classification layer consists of the fully connected layer. The functions of the layers are discussed in the below section;

● Input layer

The input layer is the initial and basic layer in the CNN structure. The input layer accepts and holds raw pixel values of the input image. The dimensions of the image will be **length** x **width** x **depth** for color images where depth = 3 representing the R,G,B color channels in the image (O'Shea and Nash, n.d.) and (Goodfellow et al., 2016). While the dimensions for black and white images will be **length** x **width** (O'Shea and Nash, n.d.).

● Convolution layer

In its general form, “convolution is a mathematical operation on two functions of a real-valued argument”, (Goodfellow et al., 2016). The convolutional layer forms the

pivotal role in the way CNNs work. In this layer, kernel/filters are defined. “These kernels are usually small in spatial dimensionality, but spreads along the entirety of the depth of the input”, (O’Shea and Nash, n.d.) . This is depicted by figure 2.13. “When pixel data from the input reaches a convolutional layer, the layer convolves each filter across the spatial dimensionality of the input”, (O’Shea and Nash, n.d.). As the filter/kernel is spread across the receptive fields, the dot products’ between weights and inputs are “integrated” across channels (Anon 2018) as demonstrated in figure 2.13.

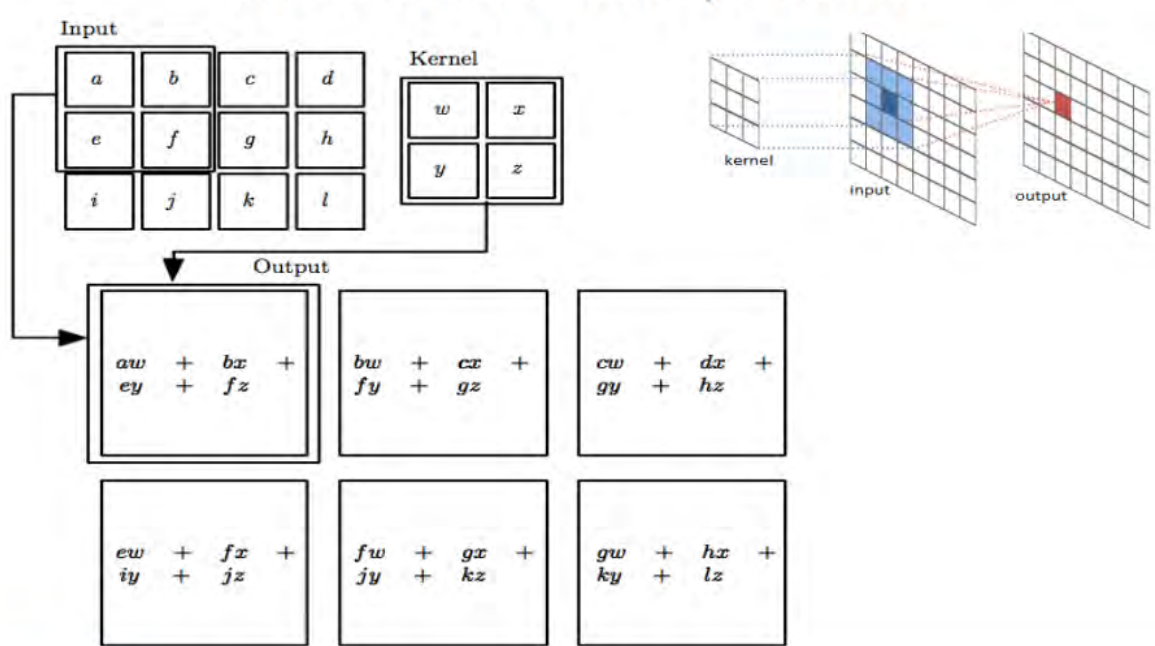


Figure 2.13: The CNN convolution operation (Goodfellow et al., 2016)

According to Goodfellow et al., (2016), the general function for the convolution operation is:

$$s(t) = (x * w)(t)$$

Where:

s denotes convolved feature

t denotes time

x denotes the matrix of receptive field

w denotes the filter weights

“As the filter is spread across the input, the scalar product is calculated for each value in that kernel. From this the network will learn kernels that ‘fire’ when they see a specific feature at a given spatial position of the input. These are commonly known as activations. Every kernel will have a corresponding activation map, of which will be

stacked along the depth dimension to form the full output volume from the convolutional layer”, (O’Shea and Nash, n.d.)

The result of convolution is called convolved feature/ feature map (Goodfellow et al., 2016). The volume of feature map/convolved feature is usually smaller as compared the input image (Goodfellow et al., 2016). After convolving the image with several filters, the produced convolved features are fed into the consecutive layer in the model (LeCun et al., 1998)

Activation

“The activation function is an important part of the convolution neural network. In the three stages of convolution neural network, convolution, pooling and full-connection, a non-linear activation function is usually used to map the calculated features, thus to avoid insufficient expression problem caused by linear operation”, (Lin and Shen, 2018).

“The activation function adds the non-linear factors to remove redundant data while preserving features, by retaining the active neuron features and maps out these features by non-linear functions” (Maas et al., n.d.). Precisely, the activation normalizes the pixel values by turning all negative pixel/weight values to zero and retaining all non-negative values (Maas et al., n.d.). Activation is in fact used to achieve edge detection in images (Maas et al., n.d.). Various activations functions have been applied in CNNs such as sigmoid (Iliev et al., 2017), Tanh (Hamidoğlu, 2016), softplus (Senior and Lei, 2014), ReLu (Senior and Lei, 2014), to mention a few. Sigmoid and Tanh are considered saturated non-linear functions and “thus these are susceptible to defects of slow convergence speed, gradient dispersion problem”, (Lin and Shen, 2018). Therefore non-linear functions such as ReLu, Softplus, Softsign are preferred activation functions in neural networks (Glorot and Bengio, 2010). The most widely used activation function is ReLu (Lin and Shen, 2018). “ReLu is easy to calculate, simple to achieve and has fast convergence speed, so it can effectively alleviate the gradient disappearance problems, and provide certain sparse characteristics for the neural networks after training , more in line with the nature of biological neuron activation”, (Lin and Shen, 2018). “RELU will apply an element-wise activation

function, such as the max (0,x) thresholding at zero”, (Lin and Shen, 2018). This results in a series of images turning into images with non-negative values (Goodfellow et al., 2016).

● **Pooling layer**

The aim of the pooling layer is to reduce the dimension of the weights (O’Shea and Nash, n.d.) The pooling layer also reduces the parameters and the complexity of the computation; this also helps control the model overfitting (LeCun et al., 1998), “by replacing the representation at a certain location with a summary of statistics of the nearby outputs”, (Goodfellow et al., 2016). This layer is often between adjacent convolution layers in the structure as shown in figure 2.12. The pooling layer scales down the input size using special functions; the max pooling and average pooling functions (O’Shea and Nash, n.d.) & (Kalogeiton, n.d.).

The max-pooling function returns the highest pixel value within the pixel neighbourhood (Zhou and Chellappa, 1988). The results of this operation is achieved by the ‘max’ activation over the rectangular regions which do not overlap each other with dimensions (K_x, K_y) (Nagi et al., 2011). The Max-pooling operation enables invariance especially in wider local regions and “down-samples the input image by a factor of K_x and K_y along each direction”, (Scherer et al., 2010). “Max-pooling results in faster convergence rate by selecting superior invariant features which improves generalization performance”, (Nagi et al., 2011). Figure 2.14 demonstrates max-pooling operation.

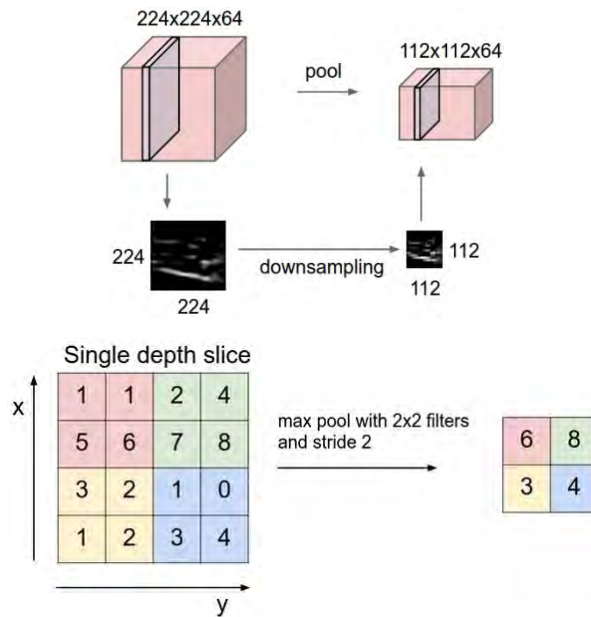


Figure 2.14: The pooling operation in CNN (LeCun et al., 1998)

The average pooling function, however reports the average of a rectangular neighborhood (Goodfellow et al., 2016).

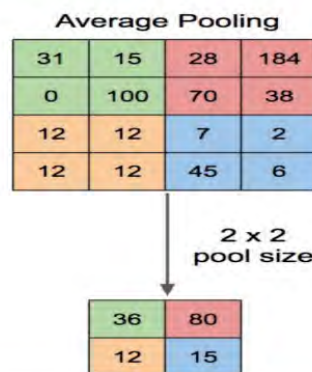


Figure 2.15: Average pooling in CNN

“In all cases, the most common form of pooling is a filter with 2×2 dimension applied with a stride of 2 along the spatial dimensions of the input. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size”, (O’Shea and Nash, n.d.). Additionally, “pooling helps in making the representation to become invariant to small translations of the input such that if the input is translated slightly, then the values of the most pooled outputs do not change”,

(Goodfellow et al., 2016). This also helps control the for handling inputs of varied sizes.

● Fully-connected layer

Once the stages convolution and pooling have been completed, the network would need to give output results in the form of a class. Since the convolution and pooling operations would have extracted important features and reduced a number of parameters from the original image, the network would need to produce the image classification. This is accomplished by the fully-connected layer. Goodfellow et al., (2016) discusses the fully-connected layer as a “traditional Multi-Layer-Perceptron that uses a softmax activation function in the output layer”. This layer implies that, in the network, all neurons found in previous layer are directly linked to all neurons on the proceeding layer (O’Shea and Nash, n.d.). “Precisely, neurons in the fully-connected-layer have full connections to all activations in the previous layer”, (LeCun et al., 1998), as it is the case in regular neural networks

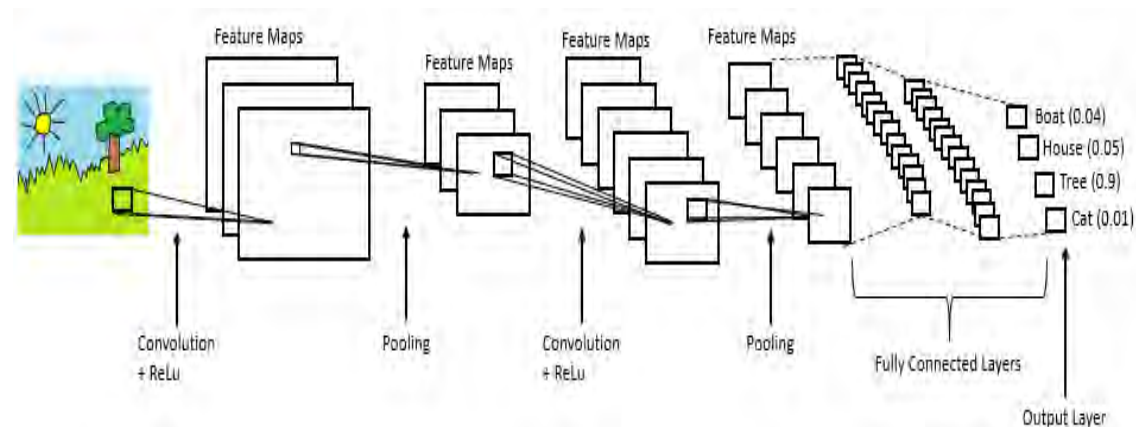


Figure 2.16: Image classification task to perform four possible outputs. The best output would be a tree (O’Shea and Nash, n.d.)

Precisely, the fully-connected layer extract products obtained from the convolution and pooling layers to assign objects present in images to their specific classes, based on the network’s training (O’Shea and Nash, n.d.). The fully-connected maps the visual features which have been extracted (from the convolution and pooling layers) onto their desired output or classes. “The common output is a vector, which is then passed through a softmax activation to represent the classification confidence”, (O’Shea and Nash, n.d.), which is usually represented by probabilities for each digit (figure 2.16).

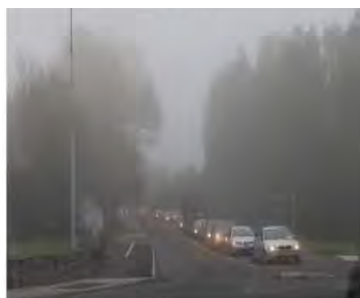
2.4 Prior studies on object recognition in bad weather

Adaptive object recognition of objects in varied weather conditions is an essential task in visual recognition. Improving visibility on images captured in bad weather conditions, will contribute greatly in higher processing of object recognition. This domain remains an active research topic in computer vision, vast literature exist object recognition in bad weather moments. The methods used for object recognition in unclear climates can be divided into three perspectives which are identified and discussed as follows.

Visibility enhancement using image processing

This is a common method of object recognition in unclear climate conditions where weather degraded images are improved for visibility using image processing techniques.

Subathra et al., (n.d.) presented “A murk removal and object detection in foggy image using CLAHE and CURVELET transform”. The proposed model involved pre-processing where the RGB images were converted into grayscale images and the the image was filtered with median filter. A “Contrast Limited Adaptive Histogram Equalization (CLAHE)” was used for enhancing the filtered image, then segmented where parameters such as mean square error, standard deviation, etc have been extracted from the segmented image. Finally objects are detected based on those parameters by applying CURVELET Transform. “The CURVELET transform is a multi-scale directional transform that allows an almost optimal non adaptive sparse representation of objects with edges”,(Subathra et al., n.d.). The following images presents the results of the work by authors:



Input image



Output image

Figure 2.17: Image dehazing using CLAHE and CURVELET transform by Subathra et al., (n.d.)
In another case, Singh et al., (2016) presented a technique of vehicle detection in fog. The authors presented a method where the blurry image captured by the camera in fog condition is restored using “Adaptive Gaussian Thresholding Technique”. In this method, “the threshold value is the weighted sum of the neighbourhood pixel values (Singh et al., 2016).

In addition, Gibson and Nguyen, (2013) presented a “fast single image fog removal using the adaptive wiener filter.” In this work, the authors presented a method of calming the estimation of a measure of fog noise present in the observed image using the “LAWF (Local Adaptive Wiener Filter)”. According to Gibson and Nguyen, (2013), this method “provided a solution to the estimation of fog performance parameters for the filter when the image and fog are co-related by de-correlating with the estimated fog-free image”.

Visibility enhancement using atmospheric scattering models (DCP)

Narasimhan and Nayar, (2000) presented a method for improving vision in bad weather. The authors analyzed the variations of color in identical scenes under various weather conditions based on the “dichromatic atmospheric scattering mode”¹ which they presented in their earlier work by Nayar and Narasimhan, (1999). Narasimhan and Nayar, (2002) computed 3D structures, using constraints on changes of scene colors, to restore natural scene colors based on two or more images taken in bad weather. Results demonstrated that this method can significantly enhance visibility, however, the drawback is that it is not practically to restore visibility for scenes that have never been encountered before.

Also, Dong et al., (2011), presented some work on “Adaptive Object Detection and Visibility Improvement in Foggy Image”. The work solved the problem of extracting objects of interest in a fog image, by proposing an algorithm for detection of objects on the foreground and improve the visibility of these objects. Here, the background depth is computed from two distinct images taken on both clear and unclear weather

conditions using the atmospheric scattering model. Based on the new observed imaged the current depth map can be calculated. According to the authors, Dong et al., (2011) “objects present in the current image are extracted by identifying and calculating the difference between the background depth map and current depth map and finally only regions with objects of interest are enhanced”. The merit of this approach is that it significantly reduces the computational time. While the limitation of this approach is that it only solves the problem of defogging of still scene such as surveillance where the camera is constantly recording one scene and is not practical to various scenes.

Lu et al., (2015) presented a study on “Single Image Dehazing through Improved Atmospheric Light Estimation”. The study proposed an optimized single image haze removal approach for estimating the atmospheric light and thus removing haze from the image through estimating a “semi-globally adaptive filter”. First the ambient light is estimated, followed by scene transmission estimation, then “semi-globally adaptive filter” is used to refine the scene. “The enhanced images contains little noise and good exposure in dark regions, textures and edges of the processed images are also enhanced significantly”,(Lu et al., 2015). The following demonstrates the results of the authors’ proposed approach of image dehazing. As seen in figure 2.18, this method only extracts the edges of high contrast objects and omit other important features for object recognition.

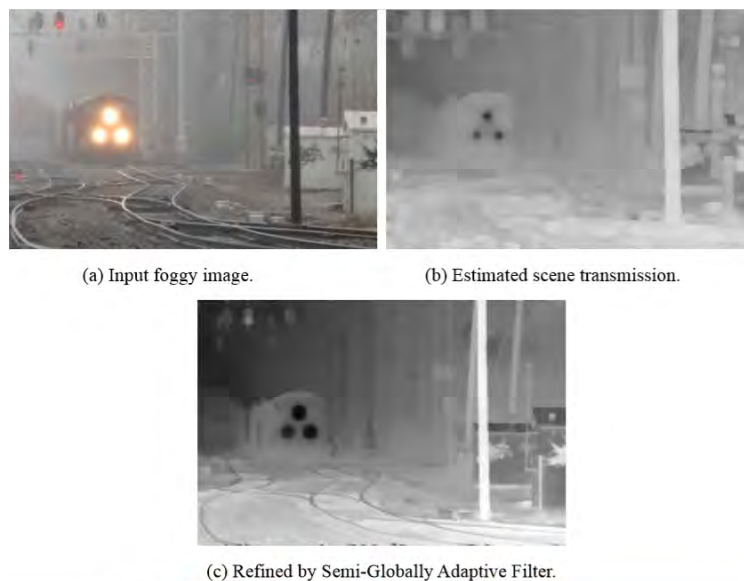


Figure 2.18: Single image dehazing using atmospheric light estimation by Lu et al.. (2015)

Visibility enhancement using deep learning

Yeh et al., (2018) presented a method for “Single Image Dehazing via Deep Learning-based Image Restoration”. The authors presented a method based on “deep learning architecture for single image de-hazing” using image restoration. “Instead of learning an end-to-end mapping between each pair of hazy image and its corresponding haze-free images, this method transforms the problem into the restoration of the image base component”, (Yeh et al., 2018). According to Yeh et al., (2018), this method “first decomposes the input hazy image into two parts: the base and the detail components, a learning CNN is deployed for haze removal by applying it to map between the hazy and haze-free base components while the detail component is further enhanced”. Finally, haze-free image is achieved by an approach that integrates the components of both the haze-free base and the enhanced detail. This method is summarized by figure 2.19.

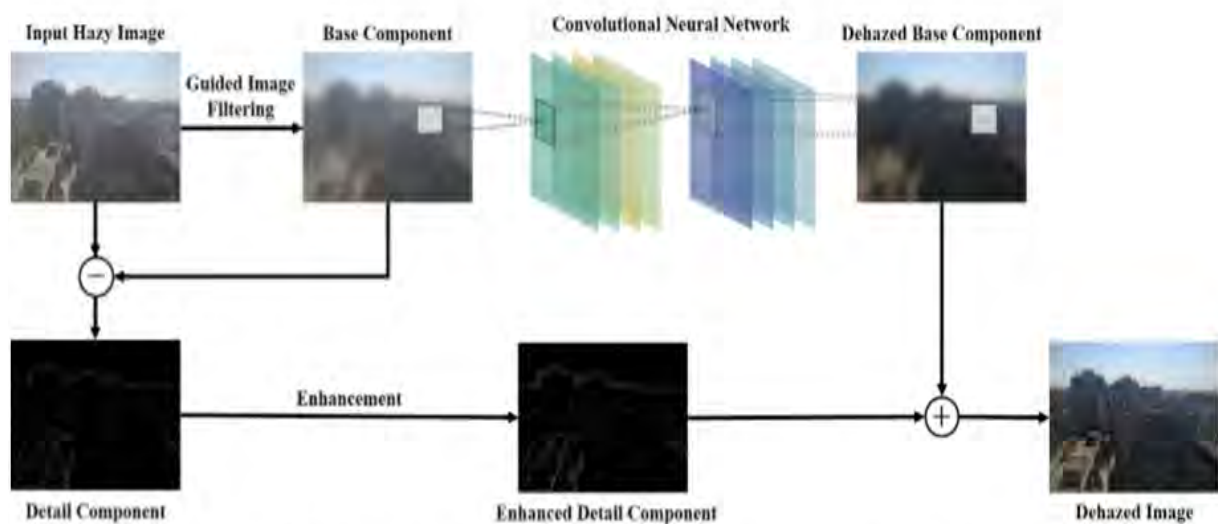


Figure 2.19: De-hazing using deep learning-based image restoration by Yeh et al., (2018)

Then the CNN is trained to remove haze from images using the corresponding haze-free images. Therefore it causes the limitation in de-hazing scenes that the network was not trained for. Furthermore, the dataset used for training the CNN

consists of synthesized haze images with light haze. This does not account for varied densities of haze such as medium to thick haze. Additionally, the proposed method achieves images with low contrast thereby making recognition of objects a challenge.

Li et al., (2018) presented a “Cascaded Convolutional Neural Network for Single Image Dehazing”. Li et al., (2018) observed that “existing learning-based methods usually predict the medium transmission using CNNs, but ignore the key global atmospheric light”. Therefore, this method uses a cascaded CNN, to consider the “medium transmission and global atmospheric light jointly by two task-driven sub-networks”, Li et al., (2018). According to the authors Li et al., (2018), “the medium transmission estimation sub-network was inspired by the densely connected CNN while the global atmospheric light estimation sub-network is a lightweight CNN”. Li et al., (2018) revealed that these two sub-networks used in this scenario are “cascaded by sharing the common features and subsequently the estimated model parameters enable the method to obtain the haze-free image, by the atmospheric scattering model inversion”. The following figure summarizes the proposed method by Li et al., (2018).

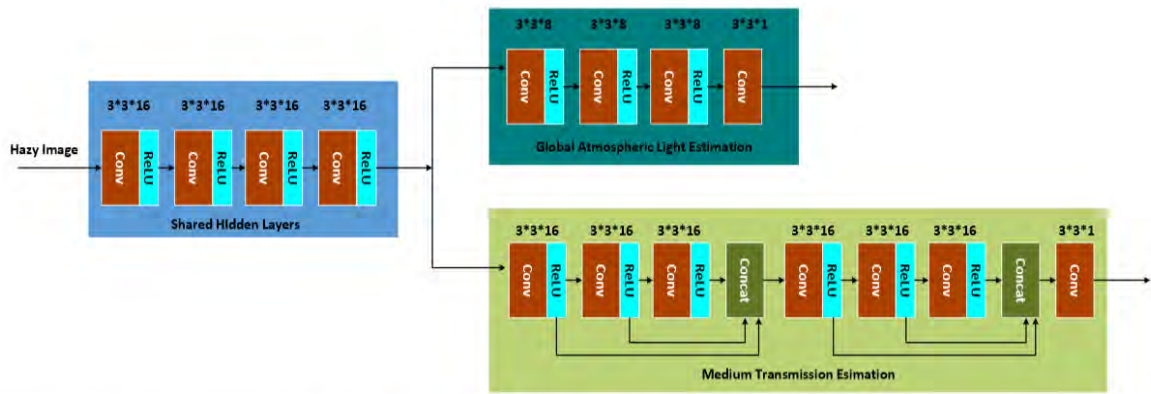


Figure 2.20: Dehazing using cascaded CNN by Li et al., (2018)

Results demonstrated that the proposed method is effective haze removal and outperforms most dehazing methods. However, the limitation of this method is that images used for training the CNNs contained synthesized fog. Synthesized fog may not appear to natural fog and the proposed method may fail to dehaze/defog natural images. Also, the method was tested on light fog and may not be applicable for medium or dense fog scenes.

Discussion of limitations in current visibility enhancement methods

The literature presented in this chapter demonstrated that vast research is focused towards enhancing the visibility of poor imagery scenes under bad weather conditions such as fog, rain, hazy and smoke. The results of current method are very substantial in the area of computer vision. However, certain limitations and gaps in the current research approaches in vision in bad weather conditions are as follows:

- Vast results of current literature on bad weather imagery scene enhancement are obtained from offline experimentation rather than real time experimentation. These experiments use still images with poor visibility due to bad weather. The results of these experiments are only limited to offline still images. Additionally, literature presents very few research where experiments are done in real time such as using surveillance cameras. However, with vast applications nowadays moving into the use of UAVs in computer vision, there is limited work on the use of UAVs in bad especially for the task of object recognition.
- Additionally, current methods are limited to improving the quality of imagery scenes captured in bad weather conditions. In modern computer vision applications, there is a need to go beyond the problem of improving image quality to address the problem of object extraction from the scene and recognize objects. Whilst current methods of bad weather image enhancement yield good results, there is a need to experiment capability of a whole object recognition methods on UAV imagery scenes captured in bad weather moments. Therefore, a research area on the use of real time computer vision systems such as UAVs for object extraction and recognition in bad weather is very key to today's applications
- Moreover, there is limited literature that specifies the measurement of the variables used to achieve the results of visibility enhancement in bad weather

such as dense, medium to light haze/fog/smoke, to show the performance of the methods towards each variable. Discussion of results based on variables such as depth or distance of objects in the imagery is also limited. Most literature only exists on the subject of haze/fog/smoke removal without considering the measure of thickness or the distance of the object from the camera source.

2.5 Chapter Summary

Chapter 2 has presented the literature review of the two “state-of-the-art” used in this study, namely; the DCP method for image defogging/dehazing and CNN for object recognition. Literature has demonstrated how the generic DCP, as originally introduced by He et al (2011), works for image dehazing/defogging. Literature has demonstrated how the DCP has been improved and used by other studies for image dehazing. Also literature has revealed that the DCP remain the best method for image scene fog/haze removal and studies has experimented by making variations and modification to the DCP for research.

Additionally, chapter 2 has demonstrated literature on how CNN are built and how models are developed for object recognition and classification tasks.

This chapter concluded by demonstrating works on studies related to the subject of fog removal using various method. These methods were analyzed to illustrate their strengths and limitations in comparison with the DCP method.

Chapter 3: Proposed Method

3.1 Introduction

Chapter 2 has presented a series of discussions that demonstrate the effectiveness of the DCP method for addressing the problem of image visibility improvement in unfavorable weather conditions such as haze and fog. Therefore, the focal of this study was to investigate the DCP method for improving visibility of the images captured using the UAV on hazy and bad weather conditions, for object recognition.

Additionally, chapter 2 has evidently demonstrated that CNNs are the “state-of-the-art” tools for solving object recognition problems. Thus, this study employed these “state-of-the-art” in the proposed method for addressing the problem of UAV object recognition in bad weather moments.

Therefore, the aim of this chapter, was to present and discuss in-depth discussion of the proposed method to address the problem. This chapter illustrated the step-by-step procedures taken in developing the algorithms for the proposed methods of the DCP and CNN. The chapter also demonstrate the code snippets that were used in implementing the method.

3.2 Proposed method

As discussed in chapter 1, the general objective of this study was to propose a method for UAV object recognition in bad weather conditions. To achieve the general objective of this study, research questions (Table 1) had to be addressed. Therefore, the aim of the proposed method was for investigating and finding the answers to the research questions of the study, which are presented in table 1 as follows.

Table 1: Research questions of the study

Research question	Objective of the research question
1. How effective is the DCP method for enabling the UAV enhance scenes captured in bad weather conditions?	This research question was designed to determine the effectiveness of the DCP method in addressing the problem of UAV visibility improvement in bad weather conditions.
2. How effective, in terms of accuracy, is the developed CNN model for UAV object recognition	This research question was intended to determine the ability of the developed CNN to enable the UAV to accurately recognize objects.
3. How effective is the proposed method for addressing UAV object recognition in bad weather conditions?	This research question was intended to determine the effectiveness of the proposed method for addressing the problem of UAV object recognition in bad weather conditions.
4. How efficient is the proposed method for enabling UAV object recognition in bad weather conditions in terms of computation time?	This research question was intended to determine the efficiency of the proposed method for achieving results for UAV object recognition in bad weather conditions in terms of computation time.

The proposed method mainly consisted of two main tasks which are presented and discussed as to how they address the study's research questions as follows:

1. To implement a DCP method for the task of improving visibility on images captured by the UAV in weather conditions such as haze and fog.

- A DCP with a guided image filter for UAV fog removal was developed and implemented. The implemented DCP method was used for improving visibility on scenes taken by UAV on bad weather. The method was evaluated for enabling the UAV improve visibility in bad weather; to address research question 1 of the study. The DCP method implementation for the task of UAV camera scenes visibility improvement was eventually used to address research question 3; for addressing the task of object recognition and research question 4, which seeks to determine the efficiency of the proposed method for achieving results for UAV object recognition in bad weather conditions in terms of computation time, when

executed wholly with the developed CNN as a method illustrated in figures 3.1 and 3.2 (section 3.2.1).

2. To develop a CNN model for recognizing objects on images captured using the UAV

- The CNN model was implemented for the task of recognizing objects of camera scenes captured by the UAV. Therefore, the implemented model was used to address research question 2 of the study by evaluating the performance of the model for UAV object recognition in terms of recognition accuracy. The implemented CNN model was also integrated in the proposed method (shown in figure 3.1 and 3.2) to subsequently address research question 3- for the task of object recognition in bad weather conditions, and research question 4- which seeks to determine the efficiency of the proposed method for achieving results for UAV object recognition in bad weather conditions in terms of computation time, when executed wholly with the developed DCP, as a method illustrated in figures 3.1 and 3.2 (section 3.2.1).

3.2.1 Diagrams of the proposed method

Essentially, this study integrated the DCP method and CNN for addressing the problem of UAV object recognition in visibility-obstructed weather conditions. The following diagrams in figures 3.1 and 3.2 depicts the flowchart of the proposed method and the visual diagram of the proposed method, respectively.

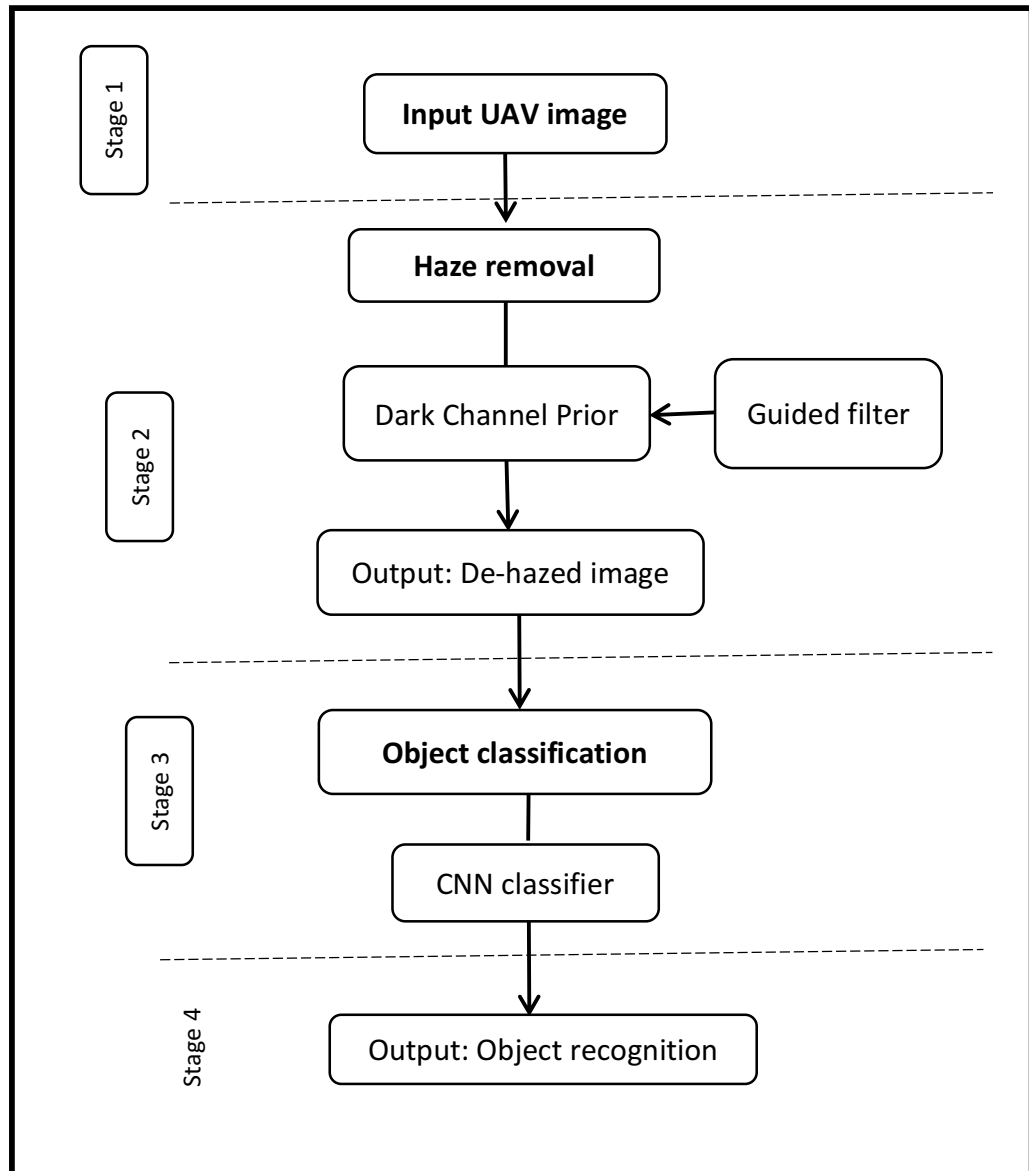


Figure 3.1: Flowchart of the proposed method for UAV object recognition in bad weather conditions

The flowchart in figure 3.1 illustrates the high level representation of the proposed method for object recognition in bad weather conditions. The diagram depicted here is discussed in section 3.2.2

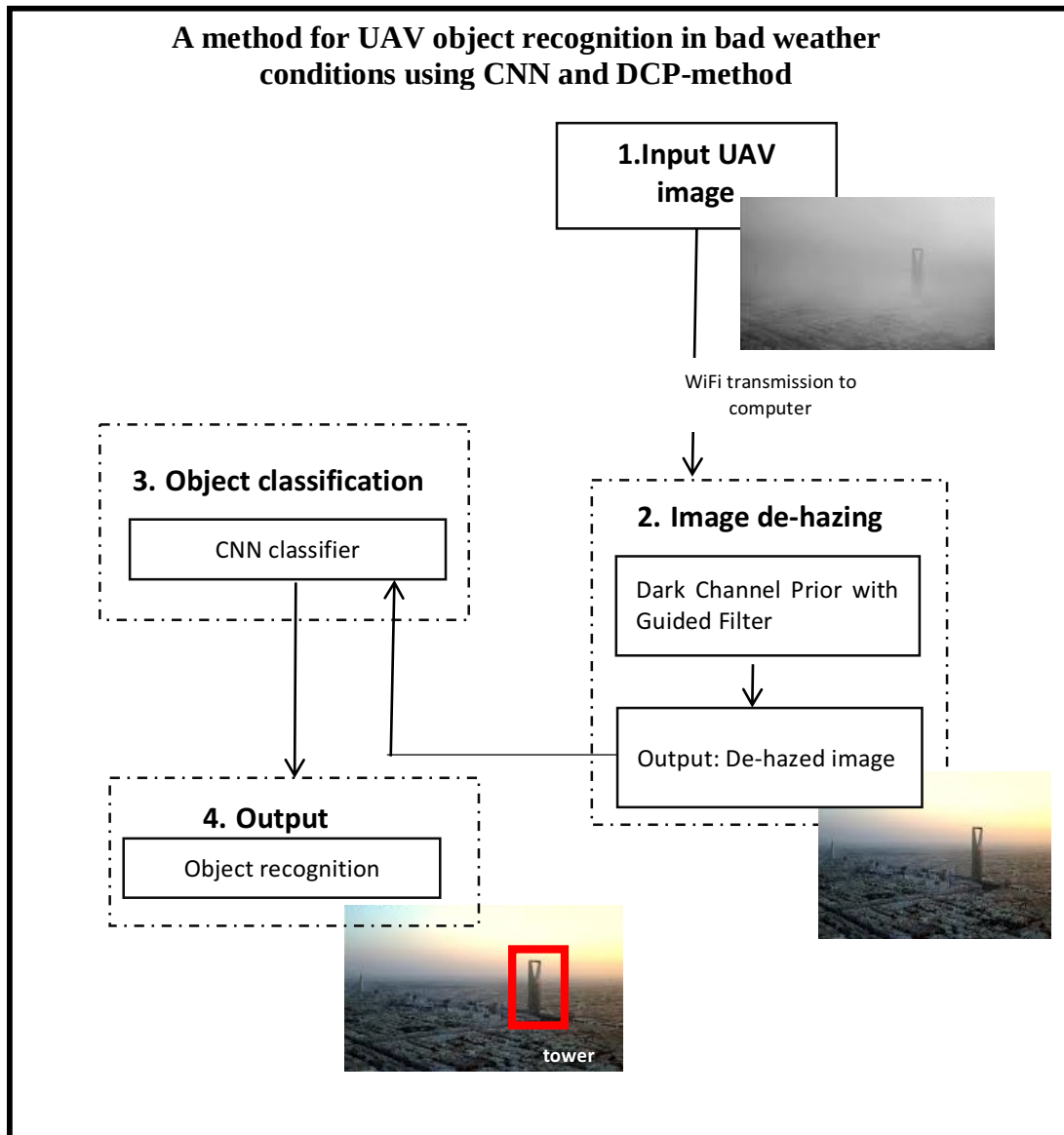


Figure 3.2: Visual diagram of the proposed method for UAV object recognition in bad weather conditions

The visual diagram in figure 3.2 illustrates the high level representation of the proposed method for object recognition in bad weather conditions. The diagram depicted here is discussed in section 3.2.2

3.2.2 Discussion of the proposed method

The proposed method consists of four essential operations as illustrated in figure 3.1 and figure 3.2. The operations in the proposed method all served to ultimately address the research questions of the study. These operations are discussed as follows:

Input UAV image

The method receives a UAV camera feed with limited visibility due to fog/haze. The camera feed contains objects of interest but they appeared visibly obstructed due to the bad weather condition with fog/haze. The UAV image scenes are sent to the computer via a wireless network.

Image de-hazing

In this stage, the DCP method removes the UAV fog/hazy image. The goal of the DCP method is to achieve a fog/haze free imagery scene. The output of this is an imagery scene with improved visibility. This operation would ultimately address research question 1 of the study (table 1), by determining the effectiveness of the DCP method for addressing the problem of UAV visibility improvement in bad weather conditions.

Object classification

In this stage, the CNN receives a fog/haze free scene as an input to the model. The CNN detects the object present in the image and classifies that object by assigning it to a class of object label, thus achieving object classification. This operation would ultimately address research question 2 of the study (table 1), by determining the ability of the developed CNN to enable the method to accurately recognize objects on UAV scenes.

Output

The last operation is the display of the output of the method. The output of the method is haze-free image with objects recognized in the images. This operation would ultimately address research question 3 and 4 of the study (table 1).

In essence, the proposed method in figure 3.1 and 3.2 ultimately addresses the task of UAV object recognition in bad weather conditions such as fog and haze.

3.3 Method development

This section discusses the development of the two state-of-the-art that was used for the development of the proposed method.

3.3.2 DCP method development

Development tools

The DCP algorithm was implemented using the following development tools:

- Programing environment: Anaconda Jupyter notebook (Python 3)
- Processor: CPU-intel Core i3
- Libraries: Numpy, opencv

Proposed DCP approach

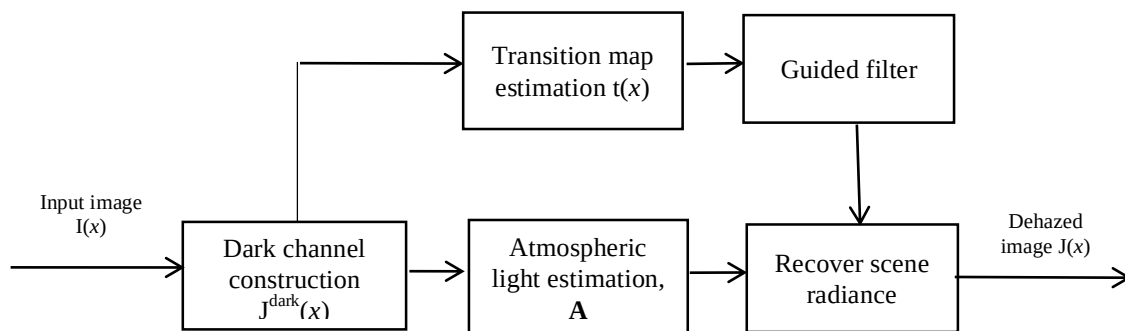


Figure 3.3: Block diagram of the DCP method

The following is a discussion of the DCP method illustrated in figure 3.3

Input image

The implemented DCP algorithm accepts a weather degraded image as the input. Literature has demonstrated that weather degraded images are formed when haze/fog/dust occludes the scene as the distance between the camera and object increases as demonstrated in figure 3.4. Therefore, for the modelling of foggy/hazy images in this DCP approach, the following assumptions were made:

- The medium does not spontaneously emit light on its own
- Light in the scene has gotten enough multiple scattering effect therefore the color blended into one color and this color is uniform and is tropic in space.

In essence, the method uses the standard knowledge that a hazy/fog image is formed when rays get transmitted in the object into the camera over the distance in between.

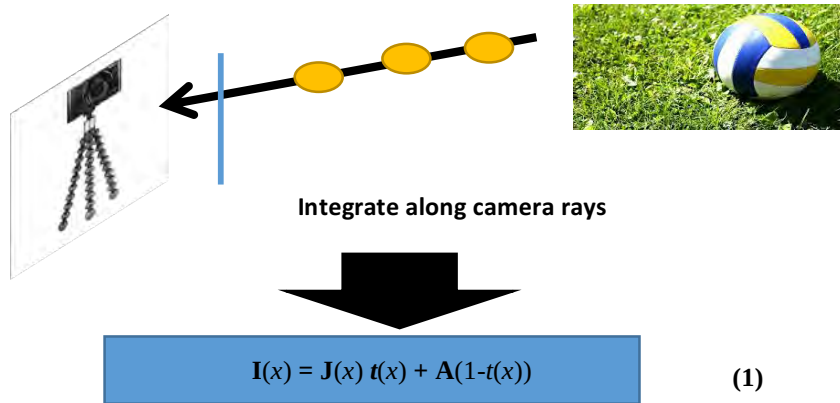


Figure 3.4: Formation of a hazy/fogged image

The formula shown in figure 3.4 mathematically expresses the how haze images are formed where;

I is the observed intensity. **I**(*x*) is the image at pixel *x*, which is a vector of *x*,*y*. **I** itself is an RGB vector

J is the scene radiance

t is the scalar, the transmission of the scene, describing the portion of the light that is not scattered. Therefore, at every pixel there is different fraction of ability to see the true color of the surface behind.

A is the global atmospheric light.

The goal of image dehazing/defogging using this equation is to recover variable $\mathbf{J}(x)$.

Dark Channel Construction

The second step in the DCP method was the construction of the “dark pixels” or the dark channel in the input image. The dark channel construction involves finding the pixels with the lowest intensities in the weather degraded image. The dark channel construction used in this method followed He et al., (2011)’s investigation on haze-free images. He et al., (2011) discovered the property that most outdoor haze free images contains at least one color channel whose pixel intensity is close to zero (low intensity). These low intensity pixels are called the dark channels. Therefore the mathematical description of the dark channel is expressed as:

$$\mathbf{J}^{\text{dark}}(x) = \min_{c \in \{r, g, b\}} (\min_{y \in \Omega(x)} (\mathbf{J}^c(y))), \quad (2)$$

where $\mathbf{J}^c$ is an intensity for a color channel $c \in \{r, g, b\}$ of the RGB image and $\Omega(x)$ is a local patch centered at pixel x . According to this equation, the minimum value within the RGB channels and all pixels in $\Omega(x)$ was chosen as the dark channel $\mathbf{J}^{\text{dark}}(x)$. Based on this discovery by Zhu et al., (2015), the dark channel is also expressed as:

$$\mathbf{J}^{\text{dark}} \approx 0 \quad (3)$$

To construct the dark channel of the hazy/fog image, minimum pixel intensity in the local patch of each RGB channel was extracted by using the equation presented in (4) as follows.

$$\min_{y \in \Omega(x)} \frac{I^c(y)}{A^c} = \tilde{t}(x) \min_{y \in \Omega(x)} \frac{J^c(x)}{A^c} + (1 - \tilde{t}(x)) \quad (4)$$

The equation for extracting the dark channel in each RGB channel in (4) was obtained by dividing both sides of the equation in (1) in figure 3.4 by A^c , the global atmospheric light.

Here, the method follows He et al., (2011)'s assumption that the transmission in the local patch $\Omega(x)$ is constant and is represented as $\tilde{t}(x)$. In this case, the min operator is applied to the RGB channels and this is mathematically expressed as:

$$\min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right) = \tilde{t}(x) \min_{y \in \Omega(x)} \left(\min_c \frac{J^c(y)}{A^c} \right) + (2 - \tilde{t}(x)) \quad (5)$$

Using the equations in (2), (3) and (4) the dark channel on the input weather degraded image was constructed. The following code snippet was implemented for the construction of the dark channels in the input hazy/foggy images:

```
#The dark channel construction
def DarkChannelConstruct(img, sz):
    b, g, r = cv2.split(img)
    dc = cv2.min(cv2.min(r, g), b);
    sz = 15
    mask = cv2.getStructuringElement(cv2.MORPH_RECT, (sz, sz))
    DarkChannel = cv2.erode(dc, mask)
    return DarkChannel
```

Figure 3.5: Python implementation of dark channel construction

The size of the local patch used is 15 x 15. This size is sufficient to estimate the dark channel, thus resulting in less DCP computation time. A morphological operation called erosion, was applied to decrease white regions in the image and retain only the dark channels, thereby creating a significant contrast of whites and blacks in the hazy/foggy image.

Atmospheric light estimation

Once the dark channel have been obtained, the next step was to estimate the atmospheric light, **A**. To retain the atmospheric light estimation, the haze-opaque pixels obtained after converting a hazy/fog image into the dark channel were used as the value for **A**. This method was followed He et al., (2011) who used the pixel with highest intensity as the atmospheric light. According to Lee et al., (2016), the pixel value of the dark channel is highly correlated with haze density; therefore this claim was used in the estimation of the atmospheric light. In this method, 0.1% of the brightest pixels were selected first, and the color with the highest intensity value

among the selected pixels were used as the value for the atmospheric light estimation, **A**. The following mathematical equation in (6) summarizes the operation for obtaining **A**, the atmospheric light estimation:

$$A = I(\arg\max_x (I^{\text{dark}}(x))) \quad (6)$$

```
#Finding the estimate of the atmospheric light
def AtmosphereLight(img,DarkChannel):
    [h,w] = img.shape[:2]
    imgsz = h*w
    numpx = int(max(math.floor(imgsz/1000),1))
    darkvec = DarkChannel.reshape(imgsz,1);
    imvec = img.reshape(imgsz,3);
    indices = darkvec.argsort();
    indices = indices[imgsz-numpx::]
    atmosum = np.zeros([1,3])
    for ind in range(1,numpx):
        atmosum = atmosum + imvec[indices[ind]]
    AtmoEst = atmosum / numpx;
    return AtmoEst
```

Figure 3.6: Python implementation of atmospheric light estimation

Transmission map estimation

Once the atmospheric light **A**, had been estimated, based on the haze/fog imaging model equation shown in figure 3.4, and the dark channel construction equation shown in (2), (3) and (4), an estimation of the transmission map was derived as follows:

$$\tilde{t}_D(x) = 1 - \omega_D \min_{y \in \Omega(x)} \left(\min_c \frac{I^c(y)}{A^c} \right) \quad (7)$$

He et al., (2011) argues that if the haze is removed completely, the image may seem unnatural. There a constant parameter ω_D ($0 < \omega_D < 1$) is introduced to calculate the estimation of transmission $\tilde{t}_D(x)$ in equation (7). Thus ω_D is fixed to 0.95.

The following code was implemented to to calculate the transmission estimation map as expressed by the equation.

```
#Finding the estimate of the transmission map
def TransmissionEstimate(img,AtmoEst,sz):
    omega = 0.95;
    im3 = np.empty(img.shape,img.dtype);
    for ind in range(0,3):
        im3[:, :, ind] = img[:, :, ind]/AtmoEst[0,ind]
    transmission = 1 - omega*DarkChannel(im3,sz);
    return transmission
```

Figure 3.7: Python implementation of transmission map estimation

Guided filter

The guided filter was used to refine the transmission map. Contrary to refining transmission map using soft matting as was used by He et al., (2011), the guided filter has been proven to be computationally inexpensive and more feasible (Pang et al., 2011). The guided filter is an operator used for preserving edges and smoothing which filters the input image under the guidance of another image. The Guided filter denotes:

Input image as p ,
Guidance image as I , and
Filtering output as q .

According to Pang et al., (2011), the local linear model of guided filter assumes that q is a linear transform of the guidance I in a window w_k centered at pixel k , so that mathematically it becomes:

$$q_i = a_k I_i + b_k, \quad \forall i \in w_k,$$

The guided filter was used to refine the coarse transmission map obtained by dark channel prior. This was implemented by the code as follows:

```
#implementation of the guided filter
def Guidedfilter(img,p,r,eps):
    mean_I = cv2.boxFilter(img,cv2.CV_64F,(r,r));
    mean_p = cv2.boxFilter(p, cv2.CV_64F,(r,r));
    mean_Ip = cv2.boxFilter(img*p,cv2.CV_64F,(r,r));
    cov_Ip = mean_Ip - mean_I*mean_p;

    mean_II = cv2.boxFilter(img*img,cv2.CV_64F,(r,r));
    var_I = mean_II - mean_I*mean_I;
    a = cov_Ip/(var_I + eps);
    b = mean_p - a*mean_I;
    mean_a = cv2.boxFilter(a,cv2.CV_64F,(r,r));
    mean_b = cv2.boxFilter(b,cv2.CV_64F,(r,r));
    q = mean_a*im + mean_b;
    return q;
```

Figure 3.8: Python implementation of guided filter

Recover scene radiance

The final scene radiance $J(x)$ is recovered by the following equation in (8)

$$J(x) = \frac{I(x) - A}{\max(t_D(x), t_0)} + A \quad (8)$$

Where t_0 is a user-defined lower bound of $t_D(x)$, and a typical value of t_0 is 0.1. The scene radiance was recovered using the following code:

```
#recovering the scene radiance
def RecoverRadiance(img,t,AtmoEst,tx = 0.1):
    res = np.empty(img.shape,img.dtype);
    t = cv2.max(t,tx);
    for ind in range(0,3):
        restore[:, :,ind] = (img[:, :,ind]-A[0,ind])/t + A[0,ind]
    return restore
```

Figure 3.9: Python implementation of radiance recovery

3.3.2 CNN development

Development tools

The CNN was built using the following development tools:

- Programing environment: - Anaconda (Jupyter notebook using python 3)
- Keras 2.0 with tensorflow backend,
- Processor: Intel Core i3
- Libraries: Scikit learn, Numpy, Opencv

Dataset description

The dataset used for training the CNN consisted of 4000 images with 4 equally distributed classes. The dataset contained images of objects such as airplanes, horses, balls, and human beings (figure 3.10). The images were in color with RGB components.

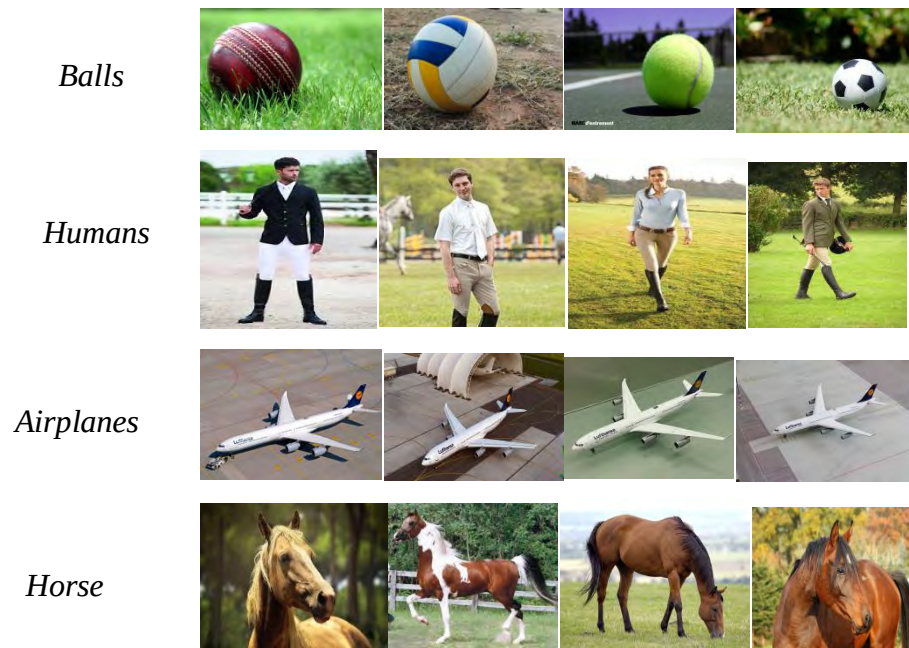


Figure 3.10: Dataset used for training the CNN

Dataset pre-processing

This section describes the pre-processing steps that were followed to prepare the dataset to be fed into the CNN model. For pre-processing of the dataset, a scikit-learn library was used. “Scikit-learn is an open source Python library mainly used providing many unsupervised and supervised learning algorithms and it is built on NumPy, SciPy and Matplotlib”, (codecademy n.d.).

- **Grayscale conversion**

The dataset used in this study were in RGB channel. The images were converted to the grayscale channel. The following code snippet was used for this operation.

```
#conversion of images into grayscale
input_img=cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
```

Figure 3.11: Python implementation of grayscale conversion

- **Image re-sizing and flattening**

The dataset used in this study contained images of various sizes. Yet, CNNs usually expect input images which are equally sized, therefore the images in the dataset were

re-sized to the dimension of 128x128 pixels. This size was selected randomly based on that most CNN architectures use images with a square shape.

Additionally, images in the dataset were flattened. The results of the flattening step is a long vector of input data that is then passed through the CNN model. The following line of code was used to perform the flattening operation.

```
#image resizing and flattening
input_img_flatten=image_to_feature_vector(input_img,(128,128))
```

Figure 3.12: Python implementation of image resizing

This operation accepts the images size of 128x128 and flattens it into a vector of its products $128 \times 128 = 16384$ pixels. The goal of flattening the data was to reshape it into a format that can be fed into the CNN model.

- Convert images into array, floating values

The dataset was converted into an array of images and images were cast into floating points values.

```
#array conversion of images
img_data = np.array(img_data_list)
img_data = img_data.astype('float32')
```

Figure 3.13: Implementation of array image conversion

- Scaling of data (zero mean and unit variance)

The goal of scaling was to standardize the dataset. It is a requirement for datasets used in machine learning implemented in scikit-learn to be standardized. According to scikit-learn.org (n.d.) if the data is not standardized, then the learning model might behave badly. To achieve this, a function called *scale* in the pre-processing library was used to achieve unit variance together with the zero mean for the used dataset. The zero mean operates by “removing the mean value of each feature and then scale it by dividing non-constant features by their standard deviation”, scikit-learn.org (n.d.). The following line of code was used to ‘standardize the dataset using the zero mean and unit variance;

```
#scaling of dataset
img_data_scaled = preprocessing.scale(img_data)
print (np.mean(img_data_scaled)) #zero mean of the dataset
```

```
print (np.std(img_data_scaled)) #unit variance of dataset
```

Figure 3.14: Implementation of dataset scaling

- Reshaping the dataset format

The dataset had to be reshaped into a format that can be fed into the CNN model. Initially, the dataset was in the format (4000,128,128). The dataset was reshaped by expanding its dimensions to include the color channel of the dataset. The dataset format became (4000,128,128, 1). In this format, the 1 indicates that the dataset is in the grayscale channel. The following lines of code were used to compute this operation.

```
#reshaping the dataset format for CNN
img_data_scaled=img_data_scaled.reshape(img_data.shape[0],img_rows,img_cols,num_channels)
```

Figure 3.15: Implementation of dataset reshaping

In this shape the dataset was ready to be fed into the developed CNN model

Assigning labels to samples

- Defining the classes

The dataset used in this study consisted of four classes of objects (balls, horses, humans, airplanes). The four classes were defined by the following lines of code.

```
#assigning labels to the training samples
num_classes = 4
num_of_samples = img_data.shape[0]
labels = np.ones((num_of_samples,),dtype='int64')
labels[0:999]=0
labels[999:1999]=1
labels[1999:2999]=2
labels[2999:]=3
names = ['ball', 'horse',
         'human', 'aeroplane']
```

Figure 3.16: Assigning labels to datasets

- Converting class labels to one-hot encoding

“One Hot encoding is the most widely used coding scheme in machine learning. It operates by comparing each level of the categorical variable to a fixed reference level; it transforms a single variable with n observations and d distinct values, to d binary variables with n observations each”, (Potdar 2017, 7).

Each observation indicates the presence, denoted by 1 or absence denoted by 0 of the binary variable. For the CNN implementation, one-hot encoding was used to convert input variables into a binary matrix for classifying objects according to classes. Four classes of objects were used in this study; and since the number of classes is four, then the CNN has 4 output layer. The following line of code was used to achieve this

```
#one hot encoding
Y = np_utils.to_categorical(labels, num_classes)
```

Output :

```
[1 0 0 0] = class_label0<-----ball
[0 1 0 0] = class_label1<-----horse
[0 0 1 0] = class_label2<-----human
[0 0 0 1] = class_label3<-----aeroplane
```

Figure 3.17: Converting class labels to one-hot encoding

Shuffle and splitting the dataset

To improve the learning of the CNN model, the shuffle module in the scikit learn was used to shuffle the dataset in a random state of 2. Then the data was divided into groups of training and testing datasets. The following lines of code were used to achieve this.

```
#Shuffle the dataset to improve learning
x,y = shuffle(img_data,Y, random_state=1)
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
```

Figure 3.18: Shuffling and splitting the dataset

CNN architecture

The architecture of the proposed CNN for UAV image recognition is summarized in Figure 3.6. The proposed architecture consists of 4 convolutional layers and 4 maximum pooling layers. The only fully connected layer is the last layer in the network. In this architecture, batches of images of sizes 128x128 pixels were considered as inputs to the network. ReLu was used as an activation function in between the hidden layers. The following code depicts the CNN model used.

```
#building the CNN model
model = Sequential()
model.add(Conv2D(32, (3,3),input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))

```

Figure 3.19: Implementation of the CNN model

Summary of the CNN model

A summary of the CNN model that was used for object recognition in the dataset is depicted in figure 3.20 as follows:

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 126, 126, 32)	320
activation_1 (Activation)	(None, 126, 126, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_6 (Conv2D)	(None, 61, 61, 64)	18496
activation_2 (Activation)	(None, 61, 61, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_7 (Conv2D)	(None, 28, 28, 128)	73856
activation_3 (Activation)	(None, 28, 28, 128)	0
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_8 (Conv2D)	(None, 12, 12, 256)	295168
activation_4 (Activation)	(None, 12, 12, 256)	0
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 256)	2359552
activation_5 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 2)	514
activation_6 (Activation)	(None, 2)	0
Total params: 2,747,906		

Figure 3.20: Summary of the Proposed CNN model

Figure 3.6 illustrates that the model used total parameters of 2 747 906, within various operations and functions. The operations and functions used in the CNN model are discussed as follows:

The first convolutional layer consist of 3x3 kernel. Then an activation function (ReLU) is applied. Then a max pooling operation with size 2x2 followed.

The second convolutional layer has 64 kernels of size 3x3. Similarly, the ReLU activation function is passed. Then a max pooling operation with the pool size of 2x2 was applied.

The third convolutional layer consists of 128 kernels of size 3x3 and has the ReLU activation function passed. Then a max pooling operation with the pool size of 2x2 was applied. In the last convolutional layer, there are 256 3x3 kernels. Similarly, the ReLU activation function is passed and a max pooling operation with a pool size 2x2 was applied.

Subsequently, the images were flattened in one dimensional vector which became the feature vector. This feature vector was then fed into the fully connected layer which executed the classification task. An activation function (ReLU) was applied. A dropout was also applied to prevent over fitting.

The output layer has four output nodes; denoting that there are four classes of objects. And lastly a softmax activation function was used to map the feature vectors into probability distributions of the estimated output classes.

Training the CNN model

During training, the input image was fed into the developed CNN model. The input travelled by feedfoward method through the layers of the CNN. Then the input backpropagated the errors, and the weight filters got updated based on the error and in this way the model was able to learn. All the images in the training dataset were used for training the model until the learning was improved.

```
#training the CNN model for recognition
hist = model.fit(X_train, y_train, batch_size=16, nb_epoch=nb_epoch, verbose=1,
validation_data=(X_test, y_test))
```

Figure 3.22: Implementation of model training

For training the developed model, the *fit* command was used as shown in code above. The batch size is 16, the number of iteration for training is 20 while the validation data is as it was split using the scikit-learn in the previous section.

Evaluating the model

The developed model was evaluated for its accuracy and loss using the following lines of code:

```
#Evaluating the CNN model
score = model.evaluate(X_test, y_test, verbose=0)
print('Test Loss:', score[0])
print('Test Accuracy:', score[1])
```

Figure 3.23: CNN model evaluation

Predicting a test image

The developed CNN was also used to test the class value of a new image. A single test image from an array of x images was fed into the *model.predict* command to test its class value. The softmax function applied in the output layer was used to give a series of probability values for each of the four classes and the class of the test image is given by the probability with the highest value. The *model.predict_classes* command was used to output the predicted class of the test image. This operation is summarized by the following code:

```
#Predicting a test image
test_image = X_test[0:1]
print (test_image.shape)
print(model.predict(test_image))
print(model.predict_classes(test_image))
print(y_test[0:1])
```

Figure 3.24: Predicting recognition on test images

Predicting new images

To predict the class of a new image, the images were first pre-processed into a format that was used to train the model. Images were converted from RGB to grayscale, resized to 128 x 128 pixels, converted into an array and normalized. The new shape of the images were 128 x 128 and this had to be converted into a format that can be predicted using the model. To achieve this, the dimension of the image shapes were expanded into number of sample, image rows, image columns and channel in this format: (1,128,128,1). The new images were then passed into the model to be predicted. The output of this was the predicted class value of each new image. The following code implements this operation:

```
# Predicting recognition on new images
test_image=cv2.resize(test_image,(128,128))
test_image = np.array(test_image)
test_image = test_image.astype('float32')
test_image /= 255
print (test_image.shape)
test_image= np.expand_dims(test_image, axis=3)
test_image= np.expand_dims(test_image, axis=0)
print (test_image.shape)
test_image= np.expand_dims(test_image, axis=0)
print (test_image.shape)
# Predicting the test image
print((model.predict(test_image)))
print(model.predict_classes(test_image))
```

Figure 3.25: Predicting recognition on new images

3.3 Chapter summary

Chapter 3 has presented the proposed method, both at high level and low level, for UAV object recognition in bad weather conditions such as haze and fog. It was revealed in this chapter that the proposed method consisted two state-of-the-art; DCP method and CNN. The DCP method was implement however approaches to each step were modified to suit the training data and implemented to enable better results for UAV object recognition in bad weather.

Similarly, a CNN model was developed and trained to recognize objects using the UAV. The training data set and object classes were demonstrated. Coding approaches to each step in the development of the two state-of-the-art was also demonstrated.

Chapter 4: Method experiments

4.1 Introduction

In this chapter, method experiments used as a method to determine the performance of the proposed method are discussed. First, an overview of the experiment as a method of data collection was discussed. This was followed by the justification for using method experiments as an investigative tool to provide findings for research questions of the study.

Subsequently, this chapter presents and discusses a series of experiments and procedures that were prepared and conducted for testing the developed method in an endeavor to establish answers to the research questions of the study.

4.2 Overview of the experiment method

The nature of this study is an experimental study. This is because research questions presented in the study requires the researcher to manipulate conditions. This is confirmed by Ayash, (n.d.) by arguing that “experimental method in computer science is used in several different fields like artificial neural networks”. Tierney, (2008), explains that experimental methods “are developed to answer hypotheses, or testable statements, formulated by researchers to address specific questions by setting up an experimental study and collect and analyse data, which will support or disprove the hypothesis”. This allows the researcher to manipulate variables, and to control and measure effects in other variables (explorable.com, n.d.) Therefore this study

conducted a series of experiments to collect data about the proposed method that will answer the research questions of the study.

4.3 Justification for using experimental method

In this study, experimental method was relevant to collect data about the response of the constructed method towards investigating the DCP method on UAV for fog removal from image scenes as well as towards investigating the CNN for object recognition from image scenes that have been dehazed/defogged using the DCP method. The following literature supports the justification of using experiment as a data collection method for this study:

Ayash, (n.d) explains that “experimental shows the experiments that will occur in order to extract results from real world implementations”. Experimental study includes a hypothesis (in this study, the research questions), a variable that can be manipulated by the researcher, and variables that can be measured, calculated and compared (Harland, 2011). Robson (2002) argues that “experimental methods are said to be the fundamental approach for obtaining information about causal relationships, thereby enabling researchers to assess the correlation or relationship between one variable and another”.

“The experimental method is the only method of research that can truly test hypotheses concerning cause-and-effect relationships. It represents the most valid approach to the solution of educational problems, both practical and theoretical, and to the advancement of education as a science”, (Gay, 1992, 298).

(Moore, 2007) further argues that “the best method of establishing causation is to conduct a carefully designed experiment in which the effects of possible lurking variables are controlled; to experiment means to actively change x and to observe the response in y ”. Therefore, in this study, conducting an experimental method was useful for collecting data about the performance of the constructed method under various controlled conditions such as the performance of the proposed method for

UAV recognition of objects under various fog densities at varied distances while observing variables such as computation time. The empirical observations based on the experiments of the constructed method provided the strongest arguments for the study. Through experiments, data was recorded and analyzed to further establish answers to the research questions of the study.

4.4 Discussion of experiments

In this study, experiments were conducted in four different phases as follows:

1. UAV haze/fog removal using DCP method
2. UAV object recognition using CNN
3. UAV object recognition in haze/fog using the proposed method (a method collective of 1 and 2)
4. Observing the computation time of the proposed method in 3

The following section provides an in-depth discussion of how the experiments were conducted to extract data/results for the study.

4.4.1 Experiments on UAV haze/fog removal using DCP method

The first phase of the experiments was to explore the UAV fog/haze removal using the DCP method.

Objective of the experiments

The objective of the experiments conducted in this section was to test and assess the performance of the implemented DCP method for UAV image dehazing/defogging. Series of experiments performed for UAV image dehazing/defogging using the DCP method were aimed at extracting findings that was used to address research question 1 of the study that reads:

How effective is the DCP method for enabling the UAV enhance scenes captured in bad weather conditions?

This research question was designed to determine the effectiveness of the DCP method for addressing the problem of UAV visibility improvement in bad weather conditions; in the context of haze/fog.

Experimented variables

Two variables were being investigated on the experiments: independent and dependent variables. “The independent variable, manipulated by the researcher, describes what is expected to influence the outcomes”, (Leroy, 2011). Independent variables that were being experimented are:

- Fog density
- Distance to object

However, “a dependent variable, often called an outcome or response variable, represents the outcome of a treatment”, (Leroy, 2011). The dependent variable in this study is the level of visibility improvement. “In experimental methods, it is expected that the dependent variable will present different outcomes for the different experimental conditions”, (Leroy, 2011). In the context of this study, it was expected that visibility improvement in hazy/foggy images would vary based on the varied fog densities and distance of the UAV to the object. Therefore the goal of the experiments conducted was to assess the following:

- Level of visibility improvement against varied fog/haze densities
- Level of visibility improvement against varied object distance depth in the scene

Experimental setup

Apparatus: the following set of apparatus were used:

- An Intel Core I3 personal computer which was used to develop the programs
- The Parrot AR drone 2.0 was used as the UAV (using its front camera)
- And the fire extinguisher (CO_2) was used to simulate fog/haze
- Objects - humans, balls, donkey as a substitute for horse were used as target objects

Environment: experiments were conducted outdoors



Figure 4.1: Apparatus used for experiments on haze/fog removal using DCP method

Configurations

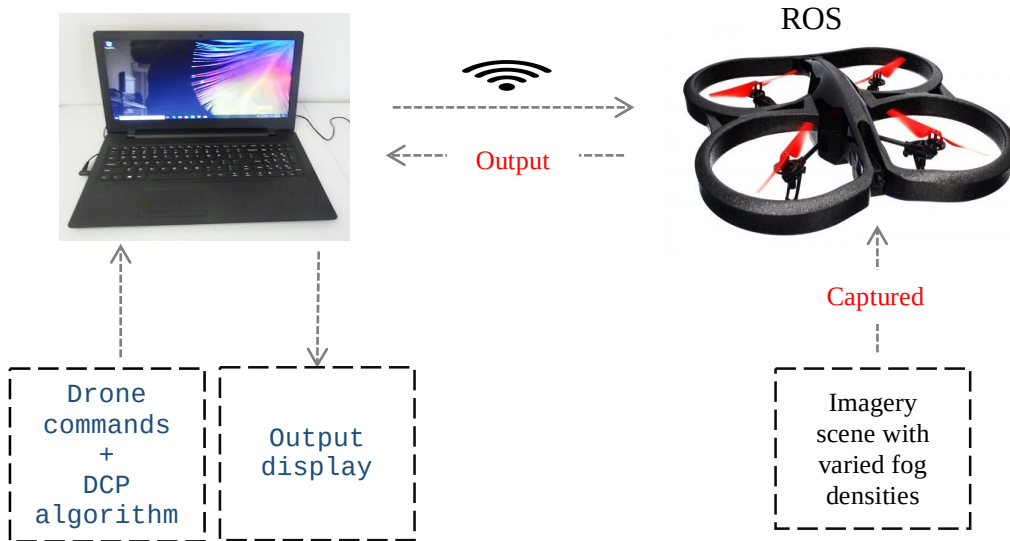


Figure 4.2: Configurations of apparatus used for experiments on UAV fog removal using the DCP

The configuration in figure 4.2 illustrates that the image dehazing/defogging algorithm was implemented in the intel core i3 processor. The DCP algorithm program was transmitted to be executed on the AR drone 2.0's ROS via the WiFi. Therefore, the AR drone captures the fogged/hazy scene and receives the DCP

program and executes it for the task of scene dehazing/defogging. The output of the image scene dehazing or defogging is transmitted back to the personal computer and displayed.

Experiments on level of visibility improvement against varied fog/haze densities

To investigate the performance of the constructed DCP method on improving visibility on hazy/fogged images using the UAV, varied fog/haze densities were applied to the scenes. The objective was to observe the performance of the method on the UAV for improving visibility in varied fog/haze densities.

For experiments, CO² was used to simulate fog and haze under the assumption that these variables (haze/fog) all contain the similar properties and give similar visual effects. Varied fog simulation densities ranging from low, medium to thick densities were applied to the UAV camera feed.

The following discusses the steps of procedures which were followed to perform an experiment to observe the degree of visibility obtained by the UAV through varied haze/fog densities using the DCP method.

Experimental procedure

- The experimental apparatus were set up and configured as shown in figure 4.2. Then samples of objects were placed on the ground and fog was produced and spread over the objects on the ground as shown in Figure 4.3.

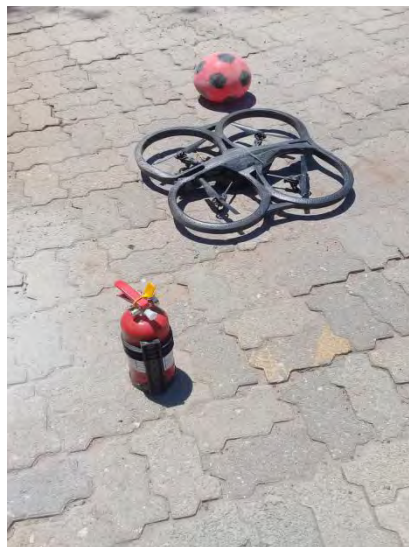


Figure 4.3: Setup of the scene to experiment visibility improvement in light,medium and thick fog

- A set of drone commands were executed to fly the drone over the objects on the ground in the fog simulated environment, over the WiFi and to run the DCP algorithm on the drone's ROS. Various fog densities were simulated on the scene to investigate the ability of the AR drone to improve the visibility of the images in these varied fog densities. Figures 4.4, 4.5 and 4.6 shows the experiments of the drone being flown to improve visibility in light, medium and thick fog respectively.

Input image



Output image



Figure 4.4: Experiment for visibility improvement in light fog

Figure 4.4 shows the input image scene with the object (ball) obstructed by light fog. The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

Input image



Output image



Figure 4.5: Experiment for visibility improvement in medium fog

Figure 4.5 shows the input image scene with the object (ball) obstructed by medium fog. The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

Input image



Output image

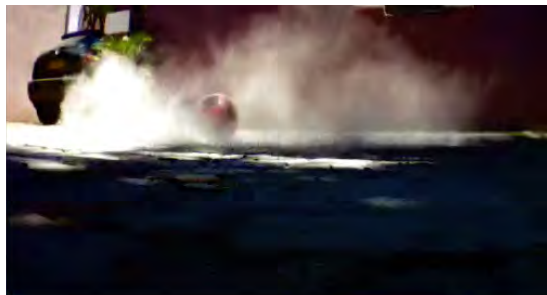


Figure 4.6: Experiment for visibility improvement in thick fog

Figure 4.6 shows the input image scene with the object (ball) obstructed by thick fog. The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

The performance of the UAV for improving the degree of visibility in varied fog densities as shown in figures 4.4, 4.5 and 4.6 was observed and results were recorded in a graph of degree of visibility against fog/haze density (depicted in chapter 5)

Experiments on level of visibility improvement against varied UAV depth distance from objects

Similarly, a series of experiments were conducted to investigate the performance of the constructed DCP method on UAV for improving visibility of objects over varied depth distance from the target object, in hazy/fog conditions. In these experiments fog (CO^2) was spread to the scenes with the objective to investigate the extent to which the DCP method can efficiently improve visibility over varied depth distances to the object using the UAV.

Practically, the nearer the object, to the camera, the more visible the object is and the further the object is the less visible it is in clear weather conditions. In essence, the goal of experiments here was to test and investigate the extent to which the DCP method can enable the UAV to achieve visibility improvement over further depths distances in images captured on a hazy/fog weather. In these experiments, the UAV was flown in distances ranging from 6 meters to furthest distances within line of sight, to the target objects on hazy/fog conditions.

The following discusses the experimental setup and the steps of procedures which were followed to perform an experiment to observe the degree of visibility obtained by the UAV through varied depth distances from the target objects on hazy/foggy conditions.

Experimental procedure

- The experimental apparatus were set up and configured as shown in figure 4.2. Then samples of objects were placed on the ground. Fog was produced and spread over the objects on the ground.

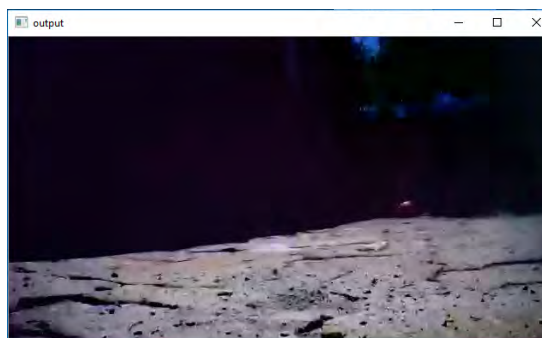


Figure 4.7: Scene setup for UAV fog removal at varied distances

The drone was flown in various distances from the target objects. Figures 4.8, 4.9 and 4.10 demonstrates the various distances to target objects at which the drone was flown and tested for its ability to improve visibility in fog in a given distance from object.



Input image



Output image

Figure 4.8: AR-drone flying at $\approx 6\text{m}$ from the object in light fog

Figure 4.8 shows the input image scene with the object (ball) placed at a distance of $\approx 6\text{m}$ from the Ar-drone camera source and obstructed by light fog . The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

Input image



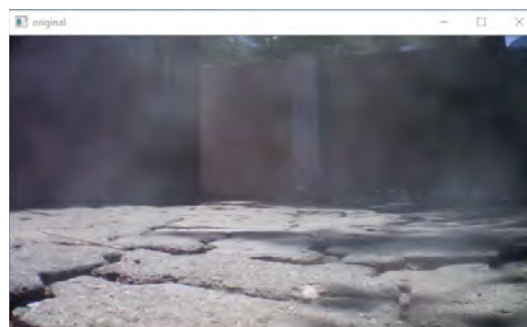
Output image



Figure 4.9: AR-drone flying at $\approx 9\text{m}$ from the object in medium fog

Figure 4.9 shows the input image scene with the object (ball) placed at a distance of $\approx 9\text{m}$ from the Ar-drone camera source and obstructed by medium fog . The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

Input image



Input image

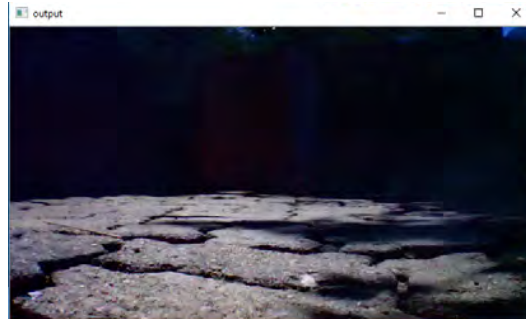


Figure 4.10: Drone flying at $\approx 11\text{m}$ from the object in light fog

Figure 4.10 shows the input image scene with the object (ball) placed at a distance of $\approx 11\text{m}$ from the Ar-drone camera source and obstructed by light fog . The output image shows the scene image after the DCP has been executed to clear the fog using the Ar-drone.

The performance of the UAV for improving the level of visibility over varied depth distances from objects in fog as shown in figures 4.8, 4.9 and 4.10 was observed and results were recorded in a graph of visibility level against the depth distance in fog from object (Graph depicted in chapter 5)

4.4.2 Experiments on UAV object recognition using CNN

The second phase of the experiments was to explore the UAV object recognition using the developed CNN.

Objective of the experiments

The objective of the experiments conducted in this section was to test and assess the performance of the constructed CNN model for object recognition using the UAV. Series of experiments conducted in this section were aimed at extracting results to address research question 2 of this study which is re-stated as follows:

How effective, in terms of accuracy, is the developed CNN model for UAV object recognition?

This research question was intended to determine the ability of the developed CNN to enable the UAV to accurately recognize objects.

Experimented variables

In this section, the constructed CNN was being investigated for its accuracy to recognize objects at different distances. Therefore, two variables were being explored here; the independent variable being the distance of object from the UAV's line of sight; and the dependent variable was the recognition accuracy.

Experimental setup

Apparatus: the following set of apparatus were used:

- An Intel Core I3 personal computer which was used to develop the programs
- The Parrot AR drone 2.0 was used as the UAV (using its front camera)
- Objects - humans, balls, donkey as a substitute for horse were used as objects to be recognized

Environment: experiments were conducted outdoors in clear weather

Configurations

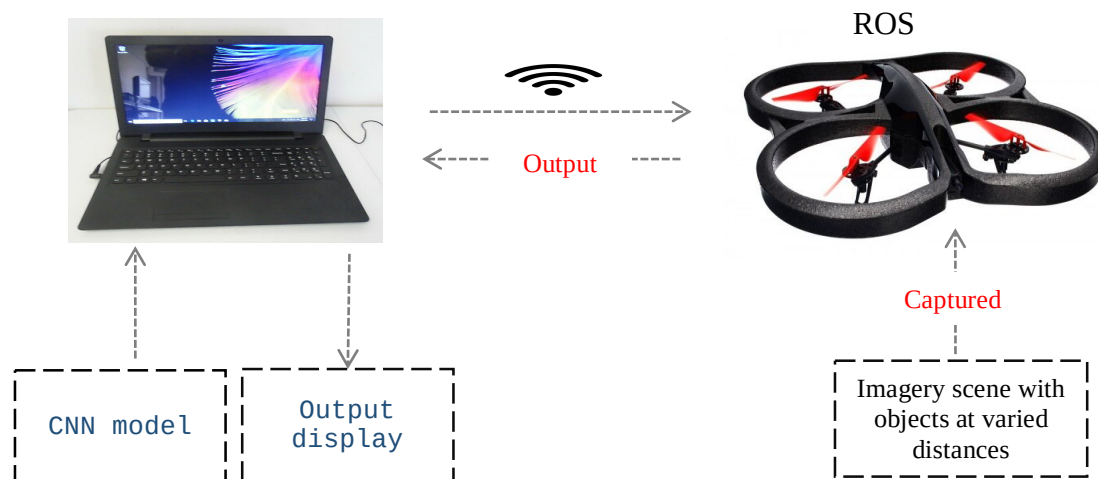


Figure 4.11: Configurations of apparatus used for experiments on UAV object recognition using CNN

Figure 4.11 illustrates apparatus configuration for UAV object recognition using CNN. The CNN was implemented in the intel core i3 processor. The CNN program was transmitted to be executed on the AR drone 2.0's ROS via the WiFi. Therefore, the AR drone captures the scene with target objects to be recognized and receives the CNN program and executes it for the task of object recognition. The output of the CNN program is transmitted back to the personal computer and displayed.

To investigate the performance of the developed CNN for object recognition at using the UAV, objects were placed on the ground at varied distances from the UAV. The objective of the experiments performed here was to investigate the performance of the developed CNN for recognizing objects at different distances in varied appearance sizes.

The following discusses the steps of procedures which were followed to perform an experiment to observe the accuracy of the developed CNN for recognizing objects at various distances.

Experimental procedure

- The experimental apparatus were set up as shown in figure 4.12 and configured as shown in Figure 4.11.



Figure 4.12: Setup of the scene for UAV object recognition experiment (apparatus: human, ball,donkey/horse)

Objects were placed at various distances within the drone's line of sight for recognition to investigate the capability of the developed CNN to accurately recognize these objects at varied distances and appearance sizes. Figures 4.13, 4.14, 4.15 and 4.16 shows the experiments of the drone being flown to recognize objects at different distances and thus appearance sizes.

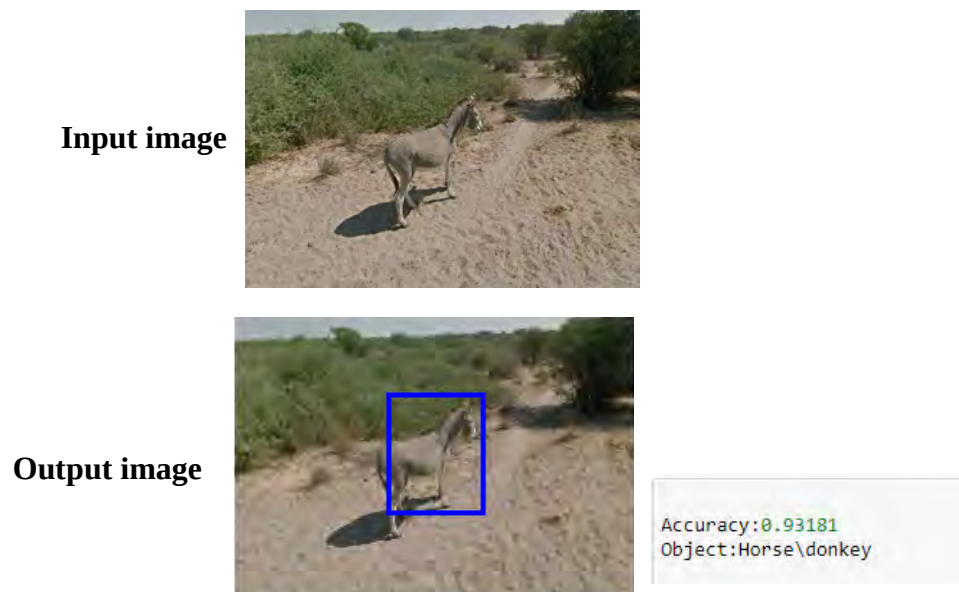


Figure 4.13: Experiment for UAV recognition at ≈ 6 meters from object (horse/donkey)

Figure 4.13 shows the input image scene with the object (donkey) at a distance of ≈ 6 m from the Ar-drone camera. The output image shows the recognition accuracy of donkey by the CNN model after being executed on the Ar-drone.



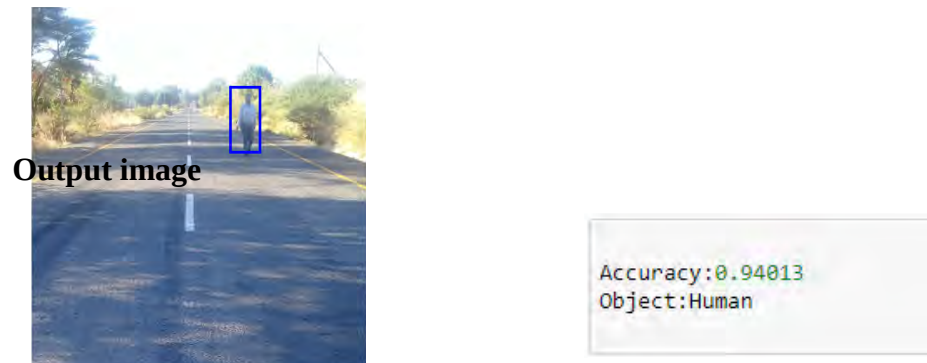


Figure 4.14: Experiment for UAV recognition at ≈ 10 meters from object (human)

Figure 4.14 shows the input image scene with the object (human) at a distance of ≈ 10 m from the Ar-drone camera. The output image shows the recognition accuracy of human by the CNN model after being executed on the Ar-drone.



Figure 4.15: Experiment for UAV recognition at ≈ 15 meters from object (human)

Figure 4.15 shows the input image scene with the object (human) at a distance of ≈ 15 m from the Ar-drone camera. The output image shows the recognition accuracy of human by the CNN model after being executed on the Ar-drone.

Input image



Output image



Accuracy:0.89882
Object:Ball

Figure 4.16: Experiment for UAV recognition at ≈ 14 meters from object (ball)

Figure 4.14 shows the input image scene with the object (ball) at a distance of ≈ 14 m from the Ar-drone camera. The output image shows the recognition accuracy of ball by the CNN model after being executed on the Ar-drone.

The performance of the UAV object recognition accuracy over varied depth distances from objects as shown in figures 4.13, 4.14, 4.15 and 4.16 were observed.

4.4.3 Experiments on UAV object recognition in haze/fog using the proposed method

The third phase of the experiments was to investigate the performance of the developed method for UAV object recognition in haze/fog (combination of experiments in phases 1 and 2).

Objective of the experiments

The developed method for UAV object recognition deploys the DCP method for fog/haze removal and CNN for object recognition. By far, the experiments on DCP

fog/haze removal using UAV have been discussed; similarly, experiments for CNN for object recognition using UAV have also been discussed. Therefore the objective of the experiments conducted in this section was to test and investigate the performance of the proposed method for UAV object recognition in haze/fog weather conditions. Series of experiments performed for UAV object recognition in haze/fog were aimed at extracting results that were used to address research question 3 of the study that reads:

How effective is the proposed method for addressing UAV object recognition in bad weather conditions?

This research question was intended to determine the extent to which the developed method can effectively address the problem of UAV object recognition in bad weather conditions such as haze/fog. This was measured by the accuracy rate of the method to recognize objects in fog/hazy conditions.

Experimented variables

a. Independent variables that were being experimented here are:

- Fog density (measure of fog available in the scene)
- Distance to object (measure of object distance from the UAV in fog)

b. Dependent variable in this study is:

- Detection and recognition accuracy

Therefore, the goal of the experiments conducted was to assess the following:

- Recognition accuracy against varied fog/haze densities
- Recognition accuracy against varied object distance in fog scene

Experiment setup

Apparatus: the following set of apparatus were used:

- An Intel Core i3 CPU was used to develop the programs
- The Parrot AR drone 2.0 was used as the UAV (using front camera)
- And the fire extinguisher (CO²) was used to simulate fog/haze

- Objects - humans, balls, donkey as a substitute for horse were used as target objects



Environment: experiments were conducted outdoors

Configurations

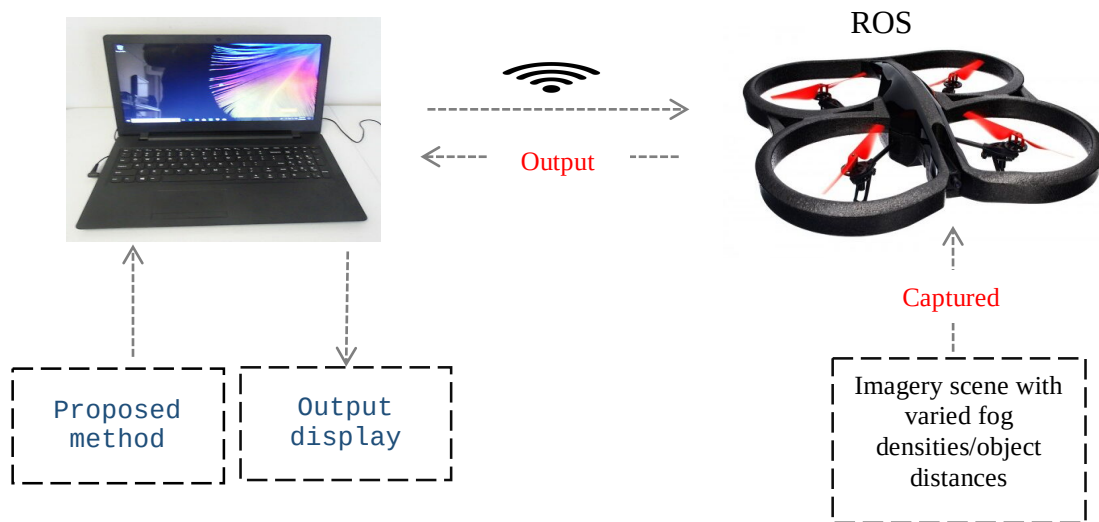


Figure 4.17: Configurations of apparatus used for experiments on UAV object recognition in varied fog densities

The developed method was implemented in the intel i3 core processor. The method was transmitted to be executed on the AR drone 2.0's Robotic Operation System via the WiFi, as demonstrated in figure 4.17.

Experiments on UAV object recognition on varied fog densities

To investigate the performance of the developed method for UAV object recognition in bad weather conditions, varied fog/haze densities were applied to the scenes with

the aim to observe the performance of the method to improve visibility and subsequently enable the UAV to recognize objects on the scene.

For experiments, CO² was used to simulate fog and haze and applied to the scenes in varied quantities to simulate light, medium and thick fog. The aim was to investigate the extent to which the developed method can tackle UAV object recognition in these conditions.

The following discusses the steps of procedures which were followed to perform experiments to observe the performance of the proposed method for UAV object recognition in bad weather conditions with fog simulations.

Experimental procedure

The experimental apparatus were set up and configured as shown in figure 4.18. Then samples of objects were placed on the ground and fog was produced and spread over the objects on the ground.



Figure 4.18: Apparatus setup to experiment UAV object recognition in fog

A set of drone commands were executed to fly the drone over the objects on the ground in the fog simulated environment, over the WiFi and to run the developed method on the drone's ROS. Various fog densities were simulated on the scene to investigate the ability of the AR drone to improve the visibility of the scene and to recognize the objects in the scene. Figures 4.19, 4.20, 4.21, 4.22 and 4.23 shows the experiments of the AR drone for recognizing objects in light, medium and thick fog respectively.

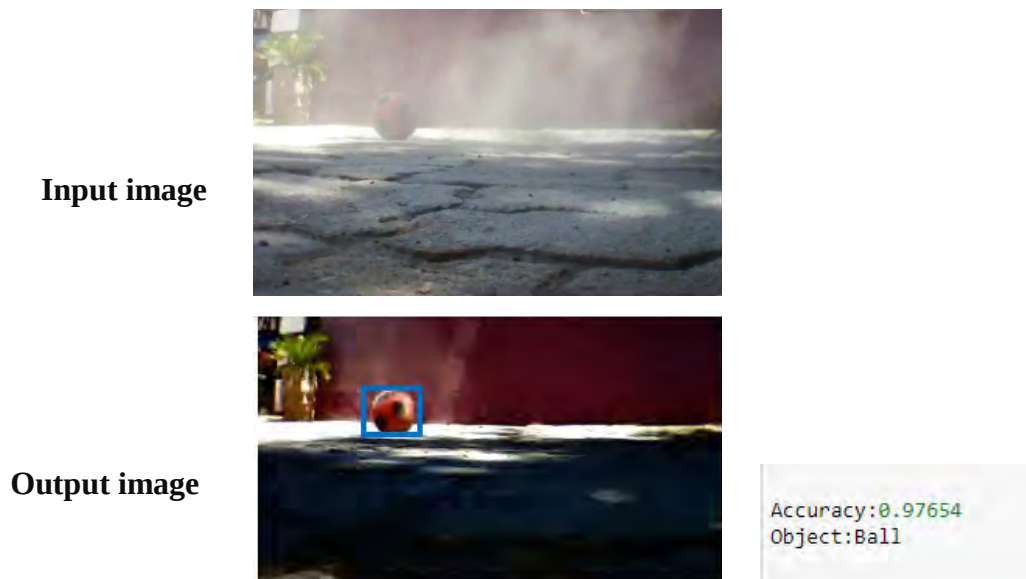


Figure 4.19: Experiment for UAV object recognition in light fog

Figure 4.19 shows the input image scene with the object (ball) captured by the Ar-drone camera in light fog. The output image shows the de-hazed image and recognition accuracy of ball in light fog by the CNN model after being executed on the Ar-drone.



Figure 4.20: Experimentation for UAV object recognition in medium fog

Figure 4.20 shows the input image scene with the ball captured by the Ar-drone in medium fog. The output image shows the de-hazed image and recognition accuracy of ball in medium fog by the CNN model after being executed on the Ar-drone.



Figure 4.21: Experimental for UAV object recognition in thick fog

Figure 4.21 shows the input image scene with the ball captured by the Ar-drone camera in thick fog. The output image shows the dehazed image and recognition accuracy of ball in thick fog by the CNN model after being executed on the Ar-drone.



Figure 4.22: Experimental for UAV object recognition in medium fog

Figure 4.22 shows the input image scene with the object (human) captured by the Ar-drone camera in medium fog. The output image shows the dehazed image and recognition accuracy of the human in medium fog by the CNN model after being executed on the Ar-drone.

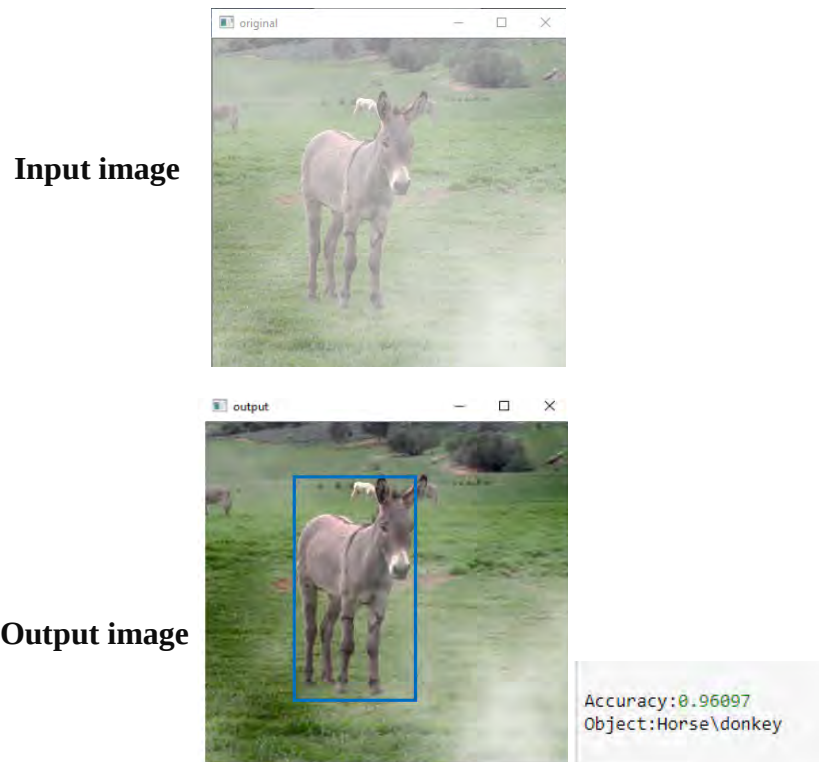


Figure 4.23: Experimental for UAV object recognition in light fog

Figure 4.23 shows the input image scene with the object (donkey) captured by the Ar-drone camera in light fog. The output image shows the dehazed image and recognition accuracy of donkey in light fog by the CNN model after being executed on the Ar-drone.

The performance of the proposed method for UAV object recognition in varied densities of fog as observed during experiments was recorded in a graph of recognition accuracy against fog/haze density (graph depicted in chapter 5).

Experiment on UAV object recognition on varied object distances in fog

Similarly, a series of experiments were conducted to investigate the performance of the developed method for UAV object recognition of objects at different distances in fog. In these experiments fog (CO_2) was spread to the scenes with the objective to investigate the extent to which the proposed method can accurately recognize objects at various distances in fog scenes. The closer the objects, the larger in size and the more distant the object is from the camera, the smaller it appears in size. Experiments conducted here were aimed at experimenting the ability of the proposed method to recognize objects in further distances in fog conditions.

The following discusses the experimental setup and the steps of procedures which were followed to perform an experiment to observe the UAV object recognition accuracy at further distances in fog.

Experimental procedure

- The experimental apparatus were set up and configured as shown in figure 4.20. Then samples of objects were placed on the ground. Fog was produced and spread over the objects on the ground.



Figure 4.24: Scene setup for experiments on UAV object recognition in varied distances in fog.

The drone was flown in various distances from the target objects. Figures 4.25, 4.26, 4.27, 4.28 and 4.29 demonstrates the various distances to target objects at which the drone was flown and investigated for its ability to recognize objects using the proposed method.

Input image



**Object
recognition**



Accuracy:0.96097
Object:Horse\donkey

Figure 4.25: Experiment for recognition with drone flying at $\approx 6\text{m}$ from the object in light fog

Figure 4.25 shows the input image scene with the object (donkey) captured by the Ar-drone camera in light fog at $\approx 6\text{m}$. The output image shows the recognition accuracy of donkey in light fog by the CNN model after being executed on the Ar-drone.

Input image



**Object
recognition**



Accuracy:0.95444
Object:Horse\donkey

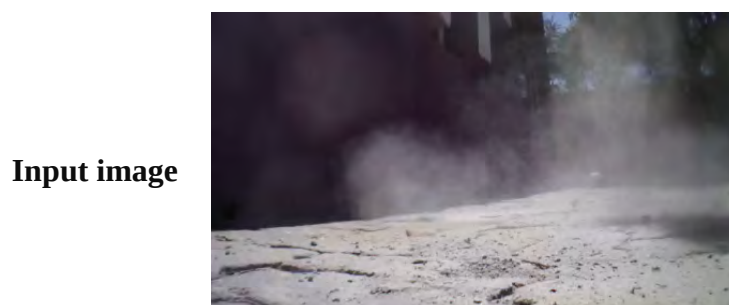
Figure 4.26: Experiment for recognition with drone flying at $\approx 9\text{m}$ from the object

Figure 4.26 shows the input image scene with the object (donkey) captured by the Ar-drone camera in light fog at $\approx 9\text{m}$. The output image shows the recognition accuracy of donkey in light fog by the CNN model after being executed on the Ar-drone.



Figure 4.27: Experiment for recognition with drone flying at $\approx 6\text{m}$ from the object

Figure 4.27 shows the input image scene with the object (ball) captured by the Ar-drone camera at $\approx 6\text{m}$ distance. The output image shows the recognition accuracy of ball in light fog by the CNN model after being executed on the Ar-drone.



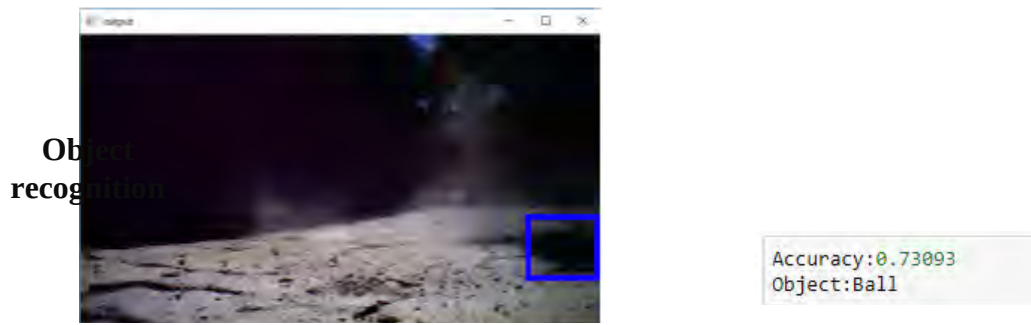


Figure 4.29: Experiment for recognition with drone flying at $\approx 9\text{m}$ from the object in dense fog
 Figure 4.29 shows the input image scene with the object (ball) captured by the Ar-drone camera in dense fog at $\approx 9\text{m}$. The output image shows the recognition accuracy of the ball in thick fog by the CNN model after being executed on the Ar-drone.

The performance of the UAV recognition of objects at further distances in fog were observed and results were recorded in a graph of recognition accuracy against the distance (graph depicted in chapter 5)

4.4.4 Computation time of the proposed method

The fourth phase of the experiments was to determine the run time profiles for the proposed method. Whilst experiments for the proposed method for UAV object recognition in fog/haze were conducted as discussed in chapter 4.4.4, response time was measured to determine how much time each step in the proposed method was consumed. The experimentation of time was aimed at providing answers to research question 4 of the study that reads:

How efficient is the proposed method for enabling UAV object recognition in bad weather conditions in terms of computation time?

A stop watch was used to measure the response time of each operation in the proposed method. A series of computation time results were recorded in *ms* and the mean/average computation time was calculated to determine the average run time for the method.

4.5 Chapter summary

Chapter 4 has discussed experiments that were conducted to investigate the proposed method for UAV object recognition in bad weather. The chapter has presented a series of steps and procedures followed to conduct experiments on all levels of the proposed method. As previously discussed, the proposed method consisted mainly of two main tasks which are; fog/haze removal to improve visibility of the scene using the DCP method and object recognition. It was revealed that these two form a proposed method for UAV object recognition in bad weather conditions. Therefore the chapter illustrated experiments on each of the proposed state-of-the-art and ultimately experiments on the proposed method as a whole were conducted. The proposed method was also experimented for its computation time.

Chapter 5: Presentation and Discussion of Results

5.1 Introduction

Chapter 4 has presented and discussed a series of experiments that were conducted to collect results of the proposed method for UAV object recognition in bad weather conditions. Chapter 5, thus, presents and discusses the analysis of the results obtained during experimentation of the proposed method.

In chapter 4, experiments of the proposed method was done in four phases as follows:

- UAV fog/haze removal using the DCP method; aimed at addressing research question 1
- UAV object recognition using CNN; aimed at addressing research question 2
- UAV object recognition using the proposed method; aimed at addressing research question 3
- Observing the computation time of the proposed method; aimed at addressing research question 4

The discussions in this chapter presented both quantitative and qualitative analysis of the results achieved in the above listed four phases of the experiments conducted in chapter 4.

5.2 Discussions on UAV fog/haze removal using the DCP

method

As discussed in the experiments chapter in 4.4.1, the DCP method was executed on the UAV (AR-drone) to observe its ability and efficiency for fog/haze removal to improve the visibility of the UAV camera scene in bad weather conditions. The goal of the results obtained here was to provide answers to research question 1 of the study that reads:

How effective is the DCP method for enabling the UAV enhance scenes captured in bad weather conditions?

In this section, the two independent variables which are being discussed and analyzed are:

- Fog density (ranging from light, to medium and heavy/thick)
- Object distance in fog scene (measure of object distance from the UAV in fog)

These variables were evaluated against the dependent variable, which is the level of visibility enhancement. The following section presents and discusses and analyses the results of the DCP for haze/fog removal using the UAV whilst the two variables aforesaid (fog density and object distance in fog scene) were being observed.

5.2.1 Results and discussion of UAV haze/fog removal using the DCP method

The proposed DCP method was tested and experimented for UAV fog removal in various densities ranging from light to thick fog. The aim was to investigate the performance of the DCP for fog/haze removal in these varied conditions while observing the level of scene visibility improvement.

Results of the DCP method for fog/haze removal

The following images in figure 5.1 illustrates the results of the DCP method for fog removal in different densities of fog ranging from light to thick, using the UAV (AR drone 2.0)

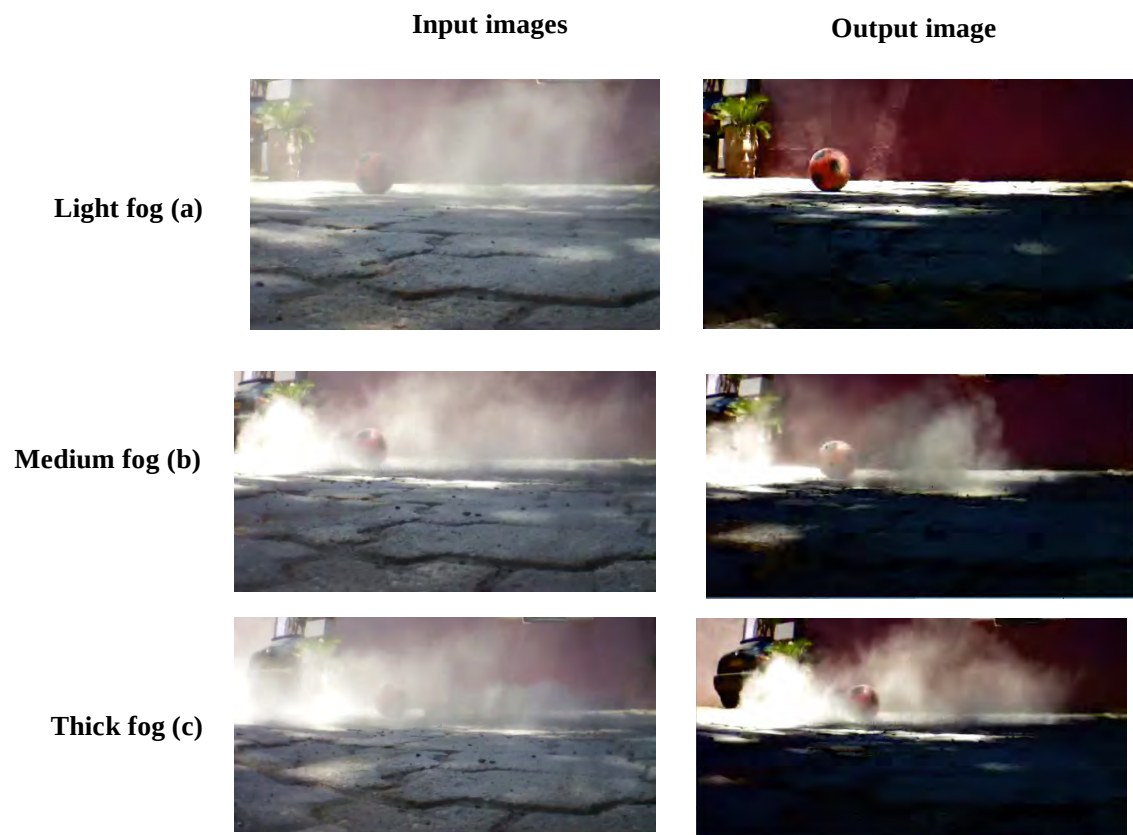


Figure 5.1: Results of the UAV scene visibility improvement in various fog densities using the implemented DCP method. (a) shows the light fog and results of light fog removal, (b) shows the medium fog and results of medium fog removal (c) shows the thick fog and results after thick fog removal

Analysis metrics of the DCP method for UAV fog/haze removal

To analyse the DCP method for UAV fog removal, two evaluation metrics for analysing the quality of dehazed/defogged images were used, namely: the Q_e metric and the Q_o metric.

Q_e metric

The Q_e metric is the most widely used method of evaluating the quality of dehazed images. The Q_e metric is a measure of the ratio of visible edges between the dehazed/fogged and hazy/fog images and was used in studies such by Zhang et al.,(2012), Tarel and Hautière, (2009) and Hautière et al., (2008). The dehazed image tends to have details which appear more sharp than hazy images which tends to have blurry details. In this case, it can be said that dehazed images have a higher value of Q_e .

Q_o metric

The Q_o metric is another metric used to evaluate the quality of dehazed images given by a measure of pixels which turns absolutely black or white (reported in percentage) after applying the dehazing operation. The Q_o metric was famously used in similar studies by Zhang et al.,(2012), Tarel and Hautière, (2009) and Hautière et al., (2008). Smaller values of Q_o result in better dehazed images while bigger values of Q_o result in over-enhancement of hazy images.

5.2.2.1 Object visibility level at varied fog densities

Analysis of the DCP method for light fog/haze removal using the Q_e metric



Therefore, using the Q_e evaluation metric, experimental results have demonstrated that in light fog/haze, the DCP method can remove fog/haze efficiently (figure 5.2), using the UAV. As can be observed, the output image scored a higher value of Q_e . This is supported by the observation that the output image has high visibility; the edges of the objects in the scene such as the ball/horse and its background appear sharper and the ratio of sharper edges in the output image versus the input image is higher. The Q_e value in this case is accounted for by a guided filter that was used in the DCP method as discussed in chapter 3. A guided filter is an “edge-preserving smoothing operator, which filters the input image under the guidance of another image”, (Pang et al., 2011). Guided filter was applied to sharpen images after the transmission refinement method as discussed in chapter 3. Additionally, the transmission estimate and refinement decreases atmospheric light scattering in the scene thereby resulting in improved scene recovery, as discussed in chapter 3. Therefore, these results indicate that in light fog/haze, the DCP method is efficiently capable of enabling the UAV to remove fog and restore quality images.

Analysis of the DCP method for light fog/haze removal using the Q_o metric

As can be observed in figure 5.2, in light fog, the DCP approach for fog/haze removal using UAV unveiled the details and recover vivid pixel color information by restoring the lower Q_o value. The contrast of the restored images are consistent with the input images. This demonstrates that the dark channel construction method discussed in chapter 3.2.2 was able to accurately locate all the darkest regions in the images; where shadows and dark pixel regions were computed and detected in contrast to the light pixel regions, of which the brightest, other than the sky regions, were used to estimate the atmospheric light using the method discussed in chapter 3.

Analysis of the DCP method for medium fog/haze removal using the Q_e metric



Figure 5.3: Images before and after medium fog removal using the DCP Top:
medium fogged image. Bottom: defogged image

In contrast, in medium fog as shown in figure 5.3, it is observed that the implemented DCP method achieves a good value of Q_e . More precisely, the DCP method yields a significant count of sharper edges in the image as compared to the original fogged image. Information about object is more visible although in some regions where the fog seemed concentrated, there are less edges visible. The results indicate that the transmission estimate, as discussed in chapter 3, is well capable in medium fog because the depth map of the image is well recovered and the edges were well refined using the guided filter. All, in all, it can be said that in medium fog/haze, the DCP is capable of enabling the UAV achieve substantial image dehazing/defogging except in the regions where the fog elements seemed too concentrated and formed a heavy fog.

Analysis of the DCP method for medium fog/haze removal using the Q_o metric

The results of the DCP method for medium fog removal as presented in figure 5.3 illustrated that, in medium fog, the DCP approach for fog/haze removal using UAV can unveil more than average of the details of the scene after fog/haze removal and recover vivid pixel color information by restoring the lower Q_o value. Also, it is observed that the contrast of the restored images is consistent with the input images. This is made possible by the ability of the dark channel construction method as discussed in chapter 3 to be able to accurately locate all the darkest regions in the images; where shadows and dark pixel regions were computed and detected in contrast to the light pixel regions, of which the brightest, other than the sky regions, were used to estimate the atmospheric light as discussed in chapter 3.

Analysis of the DCP method for thick fog/haze removal using the Q_e metric



Figure 5.4: Images before and after thick fog removal using the DCP

Left: thick fogged image. Right: defogged image

In thick fog, the results demonstrated that the DCP method for UAV fog removal achieves an average value of Q_e . Figure 5.4 demonstrated that in thick fog, the DCP method for UAV fog removal does not completely remove fog. However, the results revealed that the DCP method can slightly remove fog in regions which are nearer to the camera. These results are substantial when compared to the natural vision in dense or thick fog that visibility becomes crucially limited. However, the interesting yet important factor is that though fog was not completely removed, the DCP was able to recover a slight scene radiance where a few edges in some regions are visible. In computer vision, this is a remarkable achievement because the edges contain some information about the content of the image and this is useful for object recognition tasks. All-in-all, it can be said that the constructed DCP method is not efficient for fog/haze removal in extreme cases except where the objects of target are closer to the camera.

Analysis of the DCP method for thick fog/haze removal using the Q_o metric

The results of the DCP method for removal of thick fog is presented in figure 5.4. The results presented demonstrate that in thick fog, the DCP method for fog/haze removal using UAV only unveils substantial details of the scene after fog/haze removal. However, it is notable that there is high contrast in the dark channel pixels; that is the darker pixels becomes too dark (figure 5.4) thereby denoting a higher Q_o value. Also, it is observed that the contrast of the restored images is not consistent with the input images.

Summary of results of the DCP method performance in varied fog densities

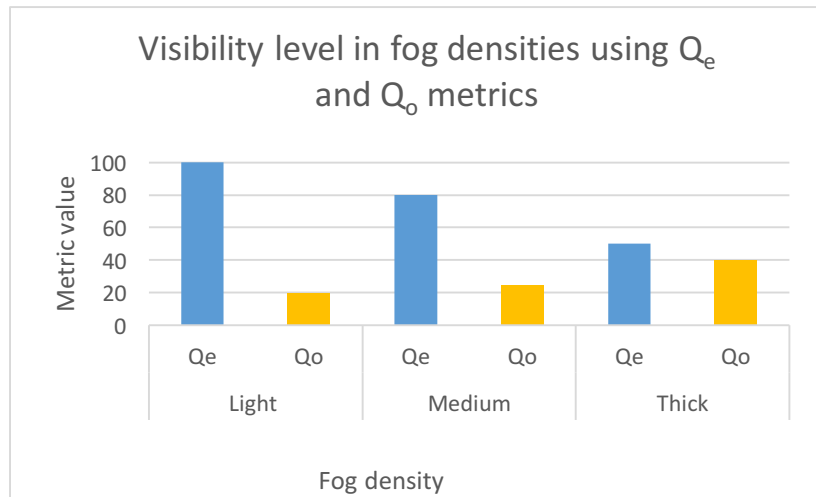


Figure 5.5: Summary of performance of DCP for UAV fog removal in varied fog/haze densities

Figure 5.5 illustrates results of the DCP method for fog removal using UAV (AR drone) in varied densities of fog ranging from light density to thick density of fog. Based on the results, it is revealed that the DCP method for UAV fog removal works best in light fog. Results presented in the graph depicted that in light fog, the DCP method achieves high Q_e value; denoting a high visibility of sharp edges of objects in the image and objects appear clearer after image defogging. Additionally, based on the results, the method achieved a lower Q_o value denoting that the output colors are consistent with the natural colors before fog. In light fog, results demonstrate a slight decrease in the Q_e value and a slight increase in the Q_o value; denoting a slight decrease in the quality of the output images. However, in thick fog, the DCP achieved an average score of the Q_e value and an increase in the Q_o value; denoting substantial image quality. Based on the results, the study concluded that the performance of the DCP method is best in light fog and performance decreased with the increase in fog density.

5.2.2.2 Object visibility level at varied object distance in fog/haze

The DCP method was also tested and experimented for fog removal at various object distances from the UAV camera in fog scenes. Here, target objects were placed at varied further distances from the UAV with the aim to investigate the ability of the DCP method for enabling the UAV remove fog to enable visibility of target objects.

Here the independent variable - object distance, was measured against the dependent variable - visibility improvement.

During experiments, objects were placed at varied distances from the UAV camera but not in distances that are beyond the line of sight by the UAV. The following distances were considered based on the line sight by the AR drone 2.0;

- Average distance $\approx$ 6 meters/less
- Far distances $\approx$ 9 meters
- Further distance $\approx$ 11 meters
- Beyond sight $\approx$ 12 meters

Results for DCP method for fog removal in varied object distance using UAV

The following figure in 5.6 illustrates the results of the DCP method for fog removal in varied object distance using AR drone 2.0.



Figure 5.6: Results of object visibility improvement at varied object distances from AR-drone 2.0 camera in fog using the DCP method. Distances ranged from ≈ 6 meters (average distance), ≈ 9 meters (far) and ≈ 11 meters (further).

Analysis of visibility improvement by the UAV at varied object distances using the DCP method

Similarly, to analyse the DCP method for UAV visibility improvement at varied object distances, the Q_e and the Q_o metrics (as introduced in section 5.2.1) were used to evaluate the quality of de-hazed/defogged image for enabling objects to appear visible on the scene.

Analysis of visibility improvement for objects at average distance from the UAV using the Q_e metric

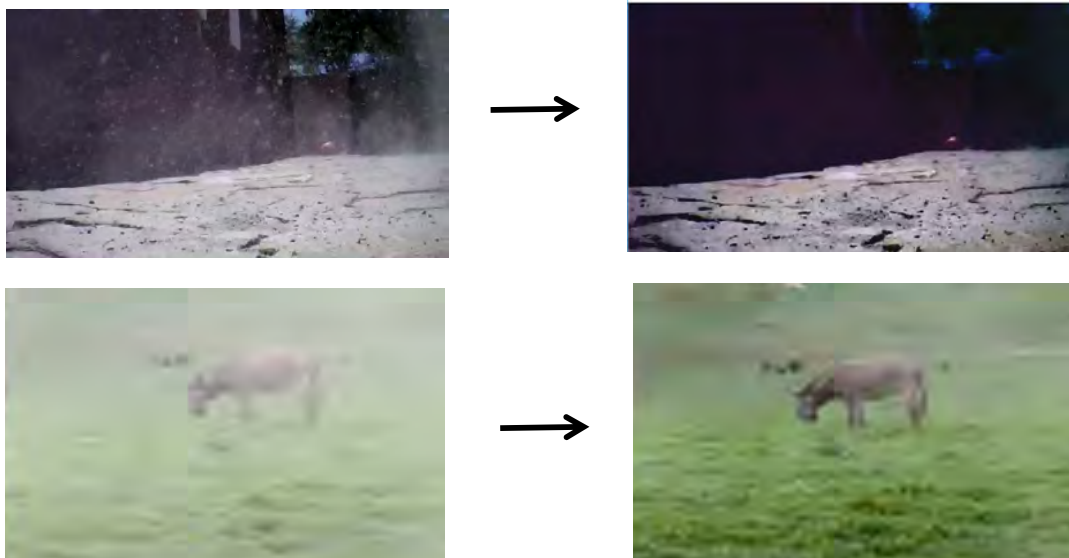


Figure 5.7: Images before and after fog removal at average distances from UAV
 Left: fogged image with object at $\approx 6\text{m}$. Right: defogged image with object at $\approx 6\text{m}$

At average distances ($\approx 6\text{m}$) results in figure 5.7 demonstrated that the DCP method for UAV fog removal achieves a high value of Q_e in well illuminated areas (bottom images) while a good Q_e value is achieved in areas covered by shadows and dark surfaces (top images). Experiments demonstrated that at ≈ 6 meters for both illuminated and shadowy areas, before fog removal by the UAV, there was limited visibility of objects in the scene. Results of fog removal revealed that visibility was improved; fog in the scene is thoroughly removed because the transmission estimate and refinement decreases atmospheric light scattering in the scene as discussed in chapter 3.2.2; a significant count of edges became defined, and the object (ball) is visible. The good Q_e value in this case is accounted for by a guided filter that was used in the DCP method as discussed in chapter 3.2.2. The guided filter was used to preserve edges by sharpening images after the transmission refinement method as discussed in chapter 3.2.2.

Based on the results, the study concludes that the DCP method can enable the UAV reveal objects at average distances (6m) in fog. Nonetheless, results demonstrated that the DCP achieves a high visibility of objects in average distance ($\approx 6\text{m}$) from the UAV in fog scenes with a significant amount of light as compared to scenes obstructed by shadows. On areas with dark surfaces such as shadows there tend to be darker patches on surfaces which were initially darker such as shadows. This indicates that the DCP method has limitations for handling extremely dark areas using AR drone.

Analysis of visibility improvement for objects at average distance from the UAV using the Q_o metric

As can be observed on the results presented in figure 5.7, at average distances (6m) the DCP method for UAV fog removal achieved a lower value of Q_o in well illuminated areas (bottom images); results revealed that the color of white and black pixels was well

restored, white pixels (fogged pixels and bright pixels) were neither completely white nor the dark pixels were completely darker. The contrast of the restored images are consistent with the input images. This demonstrates that the dark channel construction method discussed in chapter 3.2.2 was able to accurately locate all the darkest regions in the images. The results demonstrated in well illuminated areas(shadow-free) denoted that a good (lower) value of Q_0 was achieved for fog removal using AR drone.

Nonetheless, in areas with darker surfaces such as shadows (top images), results demonstrated that the DCP achieves a substantial Q_0 value, with slightly higher value; the white pixels (with fog) are well restored while the darker pixels appeared more darker. The achieved color information in not vivid at distance depth. Regardless of the reduction in size of the balls results demonstrated that objects at a distance appear but with limited visibility because they become obstructed by the darker regions using AR drone. Results denoted that in dark regions, the implemented DCP method cannot effectively improve visibility using AR drone.

Analysis of visibility improvement for objects at far distance from the UAV using the Q_e metric

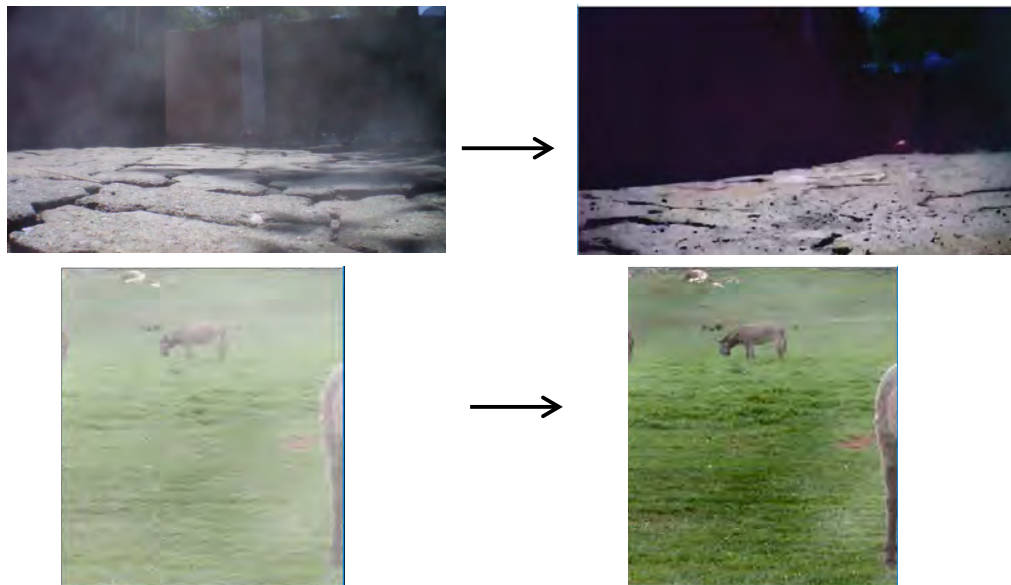


Figure 5.8: Images before and after fog removal at far distances from UAV

Left: fogged images with object at $\approx 9m$. Right: defogged images with object at $\approx 9m$.

Top: objects in dark regions. Bottom: objects in illuminated regions

At far distances ($\approx 9m$), in illuminated regions (bottom images), results in figure 5.8 demonstrated that the DCP method for UAV fog removal achieves a high Q_e value

using AR drone. Results of fog removal in illuminated regions (bottom images) revealed that visibility was improved; fog is thoroughly removed; a significant count of edges became sharper and defined; objects in the scene are adequately visible. The high Q_e value in this case is accounted for by a guided filter sharpening images after the transmission refinement method.

However, a substantial Q_e value is achieved in areas covered by shadows and dark surfaces (top image). Results illustrated that the DCP method can effectively remove fog at a distance depth of ≈ 9 meters using AR drone. However, results further demonstrated that at ≈ 9 meters small objects such as a ball, have limited visibility in darker regions. Objects' edges are not well defined, they appear blurry and not vivid. Based on the results, the study concludes that the dark channel construction does not appropriately estimate the dark pixels; dark pixels appear to have more color contrast and saturation.

Analysis of visibility improvement for objects at far distance from UAV using the Q_o metric

Based on the results presented in figure 5.8, the DCP method achieves a good value value of Q_o in well illuminated regions (bottom images); results demonstrated that the color of white and black pixels was well restored, white pixels (fogged pixels and bright pixels) were neither completely white nor the dark pixels were completely darker. The contrast of the restored images are consistent with the input images. Based on results, the dark channel construction effectively located all the darkest regions in the images. However, in areas with dark regions (figure 5.8, top images), results revealed that the DCP achieves a lower value of Q_o . White pixel areas are restored well while dark pixel regions appear too dark and the colors in the dark regions are not vivid. Results demonstrate that in dark regions, there is limited visibility of objects using AR drone, denoting that the DCP has limitation in dark regions.

Analysis of visibility improvement for objects at further distance from UAV using the Q_e metric

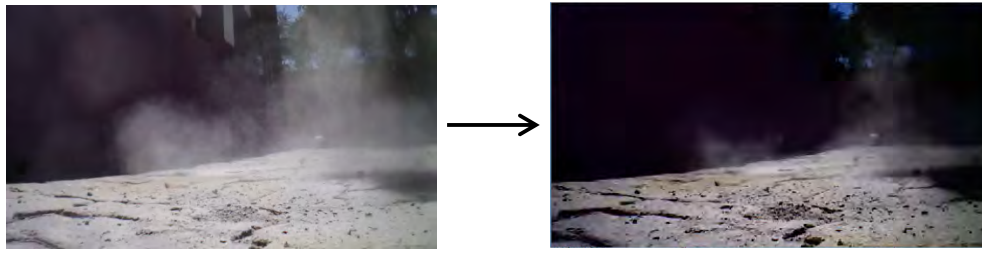


Figure 5.9: Images before and after fog removal at furthest distances from UAV

Left: fogged images with object at $\approx 11\text{m}$. Right: defogged images with object at $\approx 11\text{m}$.

Based on results presented in figure 5.9, at further distances (11m), the DCP method for UAV fog removal achieves a substantial value of Q_e using AR drone. Results demonstrated that a substantial amount of fog was removed from the scene. The output image showed objects which were visually obstructed by fog. Although, results demonstrate that edges of objects are not well defined, they appear thin and blurry. Also results reveal a relatively lower percentage of image details. Therefore, based on the results, the study concludes that the DCP method for fog removal using AR-drone achieves average Q_e value.

Analysis of visibility improvement for objects at further distance from UAV using the Q_o metric

Based on the results presented in figure 5.9, the DCP method achieves an average value of Q_o in illuminated regions using AR drone; results demonstrated that the color of white pixels due to fog was not completely but substantially removed, revealing the objects previously obstructed by fog. Additionally, results however demonstrated that the black pixels were not well restored, rather, they have attained more contrast, making them appear even more darker. These results denoted that the dark channel construction was not effective for improving visibility in darker regions of the scenes. Based on the results, the study concludes that the DCP method for fog removal achieves average quality images when using the Q_o as an evaluation metric.

Summary of results of the DCP method performance at varied object distances

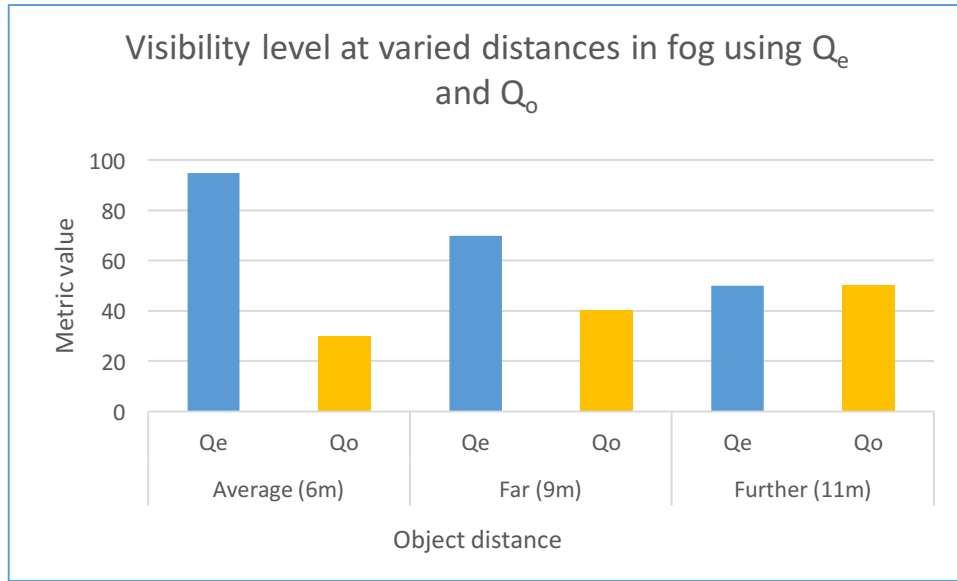


Figure 5.10: Summary of performance of DCP for fog removal at varied distances of objects from the UAV

Figure 5.10 illustrates results of the DCP method for fog removal using UAV (AR drone) at varied object distances from the UAV. Based on the results, in average distances and less, the DCP performed best; this is denoted with a high value of Q_e and lower value of Q_o . In far distances, results demonstrate a decrease in the Q_e value and an increase in the Q_o value; denoting a decrease in the quality of the output images, the output images are of average quality. However, in further object distances, the DCP achieved lower Q_e value and a high Q_o value; resulting in image quality below average. Based on the results presented on figure 5.10, the study concluded that the performance of the DCP method for fog removal in various object distances decreased with increase in distance. Based on the graph results, it is revealed that DCP method performs best in objects nearer or in average distances from the UAV camera in light as compared to when objects are further deep in the fog.

Analysis of DCP method for fog removal by comparison of results for related works

Results of the DCP method were compared with the performance of other methods to remove the scattered light from natural images. While comparison was done, the following were considered:

- In contrary to the method, similar fog removal methods discussed here were experimented on images with synthesized fog instead of physical and more natural fog simulation.
- Additionally, methods used for comparison were experimented only on light fog/medium density thus omitting dense fog.
- And lastly, on the contrary, experiments for these methods were done offline on still images.

Therefore for experimentation to compare results the “*fog train*” image was used as the input into the DCP and and the proposed method was tested offline on the intel i3 CPU. Figure 5.11 as follows demonstrates the results obtained by various defogging methods against the proposed method in this study using the “*fog train*” image.



Original image



(a) He et al., (2011)'s



(Tarel and Hautière, 2009)



(c) Fattal, (2014)'s method



(d) Gibson et al., (2012) method



Figure 5.11: Results of image defogging using different methods.

Figure 5.11 illustrates an example to comparison of scene radiance under different methods. Figure 5.11 illustrated that the results for methods developed by Tarel and Hautière, (2009) (b), Lu et al., (2015) (e) and Gibson et al., (2012) (d) remained with some fog on both the foreground and background. Colors of the objects in the image scene are not well defined and the edges appear blurry. Result images contain a lot of white pixels (fog pixels). Additionally, results achieved by He et al., (2011)'s method in (a) contain a lot of dark pixels; the color of output image after fog removal is not consistent with the input image; the output image appeared too dark, denoting that the dark channels were overestimated. Similarly, the results of the method by Fattal (2014) in (c) presented a lot of dark pixels after defogging. Details of the images especially those in depth appeared darker and less visible in the scene.

However, compared with the other state-of-the-art method, the implemented DCP method (f) achieved a better scene radiance. Details of the objects appeared more clarified than other methods in figure 5.11(a), (b), (c), (d) and (e). The method estimated the transmission map better and the edges of objects appeared more sharper than in other methods. Fog was completely removed on the foreground and significantly on the background. The train details at a far depth are more well defined than in other methods. However, the noticeable limitation, compared with other methods, is that the implemented DCP method slightly overestimated the dark channel, more contrast is added to the dark surfaces. Therefore, based on analysis by comparison to other methods, the study concludes that the DCP method can effectively remove fog than other state-of-the-art methods. The method achieved smooth and vivid color information on the images in light and medium fog densities.

5.3 Discussion on object recognition using CNN

As discussed in the experiments chapter 4.4.2, the developed CNN model was executed on the UAV (AR-drone) to observe its ability to recognize objects in different distant objects. The goal of the results obtained here was to provide answers to research question 2 of the study that reads:

How effective, in terms of accuracy, is the developed CNN model for UAV object recognition?

The constructed CNN was being investigated for its accuracy to recognize objects. The CNN model was evaluated and analyzed for its performance using two metrics namely:

- Loss and accuracy graphs of the model
- Performance accuracy of the model

These two performance metric are discussed as follows:

Loss and accuracy curves/graphs

The loss and accuracy graphs were used as metrics to evaluate the performance of the CNN during the training phase.

Loss graph

The loss graph was used to find the best weight for the model by updating the weights in order to optimize the CNN training. Figure 5.12 shows the loss graph.

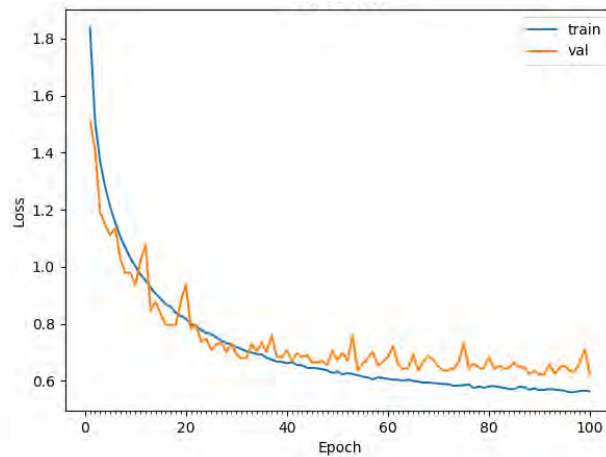


Figure 5.12: The loss graph for the CNN model

The loss is calculated during the training and validation of the model and in contrary to accuracy, it is not a percentage but a summation of errors made for each epoch in training and validation set and its interpretation. In CNN training, the lower the loss, the better the model and loss values implies how well the model behaves after each iteration of optimization. The loss graph presented in figure 5.12 illustrates lower loss for the CNN model. Figure 5.12 demonstrates that the CNN model used for recognition performs better. This demonstrates that the model is able to perform learning consistently by minimizing the loss functions; value with the model's weight vectors by the back propagation method, thus maximizing the accuracy rate.

Accuracy graph

The accuracy graph was used during training from an applied perspective to evaluate how accurate the model's prediction when compared to the actual data. Figure 5.13 demonstrates the accurate curve for the CNN model used in the study.

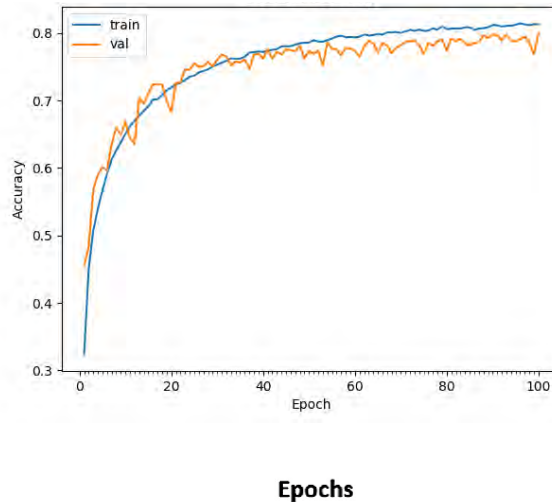


Figure 5.13: The accuracy graph for the CNN model

Once the model parameters such as weights are adjusted and learned as indicated in the loss graph (figure 5.12), and no learning is taking place, then the accuracy of the model can be determined. Accuracy of the CNN model was obtained by feeding the test samples to the model and the frequency of mistakes made by the model and compared to the true targets and recorded and plotted on the graph (figure 5.12).

The accuracy of the CNN model used in the study was plotted as shown in figure 5.13. The values presented in the accuracy graph are in percentage and indicate the percentages at which the model can accurately do the classification of the test sample. The accuracy curve in figure 5.13 presents high accuracy rates demonstrating that the model is accurate for recognition of objects in the test sample. The model scores an accuracy rate of **94.3%** in average.

Performance accuracy of the CNN model

Another metric used to evaluate the performance of the CNN model presented and used in the study is accuracy. In [5] accuracy is described as the ability to do something without making mistakes. In this study, accuracy for the task of classification is the number of correct predictions made by the CNN model. Accuracy is a good measure of performance of the CNN model used in this study because the target variable classes in the data are balanced.

The constructed CNN was being investigated for its accuracy to recognize objects at different distances. Therefore, two variables were being explored here; the independent variable being the distance of object from the UAV's line of sight; and the dependent variable was the recognition accuracy.

Results of the CNN for object recognition using the UAV

The following illustrates the results of the CNN for object recognition using the AR-drone at different object distances. The CNN was experimented for its accuracy to recognize objects such as horse, humans and balls and this was experimented by observing the results.

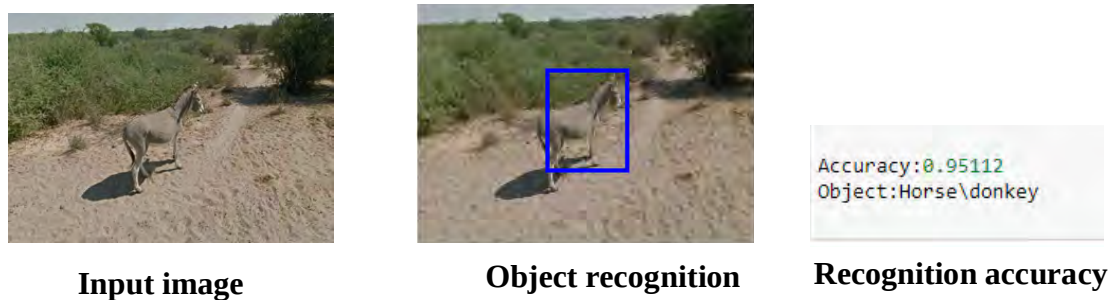


Figure 5.14: Recognition of a horse donkey at $\approx 4\text{m}$ away from the UAV

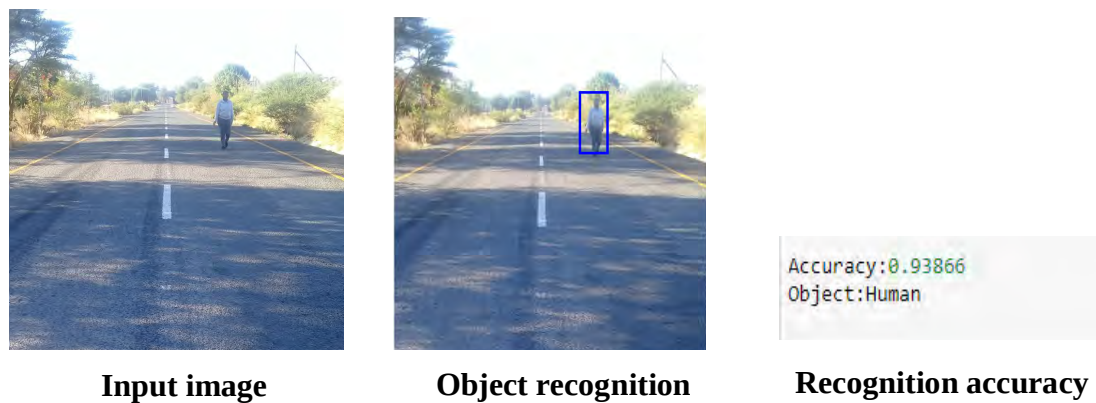


Figure 5.15: Recognition of a human being at $\approx 6\text{m}$ away from the UAV

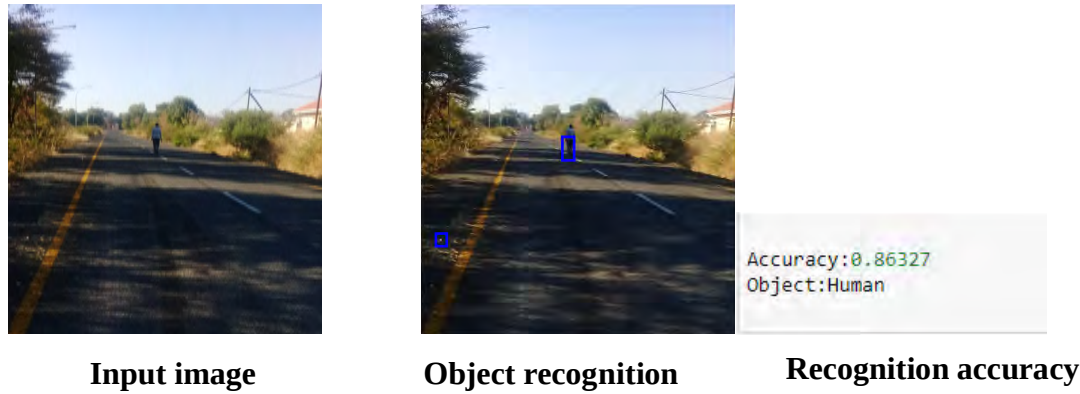


Figure 5.16: Recognition of a human being at $\approx 10\text{m}$ away from the UAV

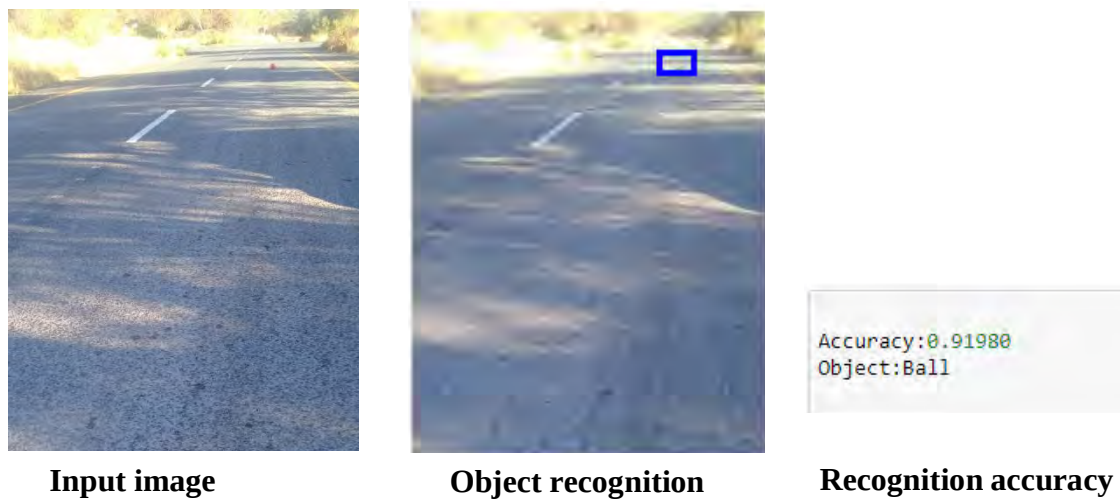


Figure 5.17: Recognition of a ball at $\approx 12\text{m}$ away from the UAV

Discussion and analysis of the CNN for UAV object recognition using performance accuracy

The results presented in figures 5.14, 5.15, 5.16 and 5.17 above illustrate the performance accuracy of the CNN for recognition of objects of different classes which were used to train the CNN. In figure 5.14, the results presented that the model can accurately classify horse/donkey using the UAV with an accuracy score of more than 90%, and this is if the object is nearer (4meters) to the UAV camera as shown in the figure.

Similarly, in figure 5.15, the model was experimented to recognize a human being at a distance of approximately 6meter from the UAV, and results presents that the model is able to accurately classify human beings with a confidence score of more than 90%. Practically, the further the distant the object is from the camera, the smaller it appears in size. This demonstrates the extent to which the model effective is for classifying objects at certain further distances where the objects appear more full.

In figure 5.16, the results for experimenting the model for classifying objects (humans) at a distance of approximately 10meters from the UAV are presented. The results demonstrate that the model accurately classifies the object. However, results in figure 5.16 demonstrates a decrease in accuracy confidence with a score lower than the average **94.3% score**. The lower confidence score of **86.3%** posed a very significant point as it illustrates that somehow the model is *guessing* but not certain of what the object is. The recognition error made by the model is accounted for by a few factors which are identified as follows:

- Naturally, the more distant the object from the camera, the smaller the object appears and and losses its shape. Therefore, this could justify why the CNN model had a decrease in accuracy score. However, appearance could also depend on the camera type that is being used. The more powerful the camera, the more line of sight it can reach. The results presented in figure 5.16 is that of the AR-drone camera, which is not so powerful compared to other UAV cameras.
- Additionally, it is noted that in figure 5.16, the CNN model estimated two detected regions. The developed model was for single object recognition rather than a multiple object recognition. This is because the further the distance the object is places, the wider the scene coverage it presents in the image captured. Other objects come into the scene, some of which its pixel representation possibly resemble other objects which are in the training dataset but not the target in the scene.

Analysis by comparison of the CNN with related works

The study compared and analyzed the accuracy of the CNN used with other CNN for UAV object recognition as follows:

Table 2: CNN performance accuracy for related UAV object recognition studies

Study	Author	Method description	Accuracy
Object Recognition in Images using Convolutional Neural Network	(Duth and Raj, 2018)	Study presented object classification and detection using cifar-10 data set with intended classification and detection of airplane images. Study used CNN on keras with tensorflow	96%
Towards real-time people recognition on aerial imagery using convolutional neural networks	(Oliveira and Wehrmeister, 2016)	Study presented a step towards real-time people recognition on aerial imagery. The study proposed a new approach based on CNN for real-time people recognition on aerial images obtained from cameras attached to small UAVs.	95%
Cattle detection and counting in UAV images based on convolutional neural networks	Shao et al., (2019)	Study proposed a cattle detection and counting system based on Convolutional Neural Networks (CNNs) and YOLO using aerial images taken by an Unmanned Aerial Vehicle (UAV).	95.2%
Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3	Benjdira et al., (2018)	Study investigated the performance of two state-of-the art CNN algorithms, namely Faster R-CNN and YOLOv3, in the context of car detection from aerial images	99.66% for Faster R-CNN and 99.73% for YOLOv3
Proposed CNN model in this study			94.3%

Table 2 presented a performance accuracy of other CNN studies for UAV object recognition. Based on the accuracy by other related studies as shown on the table, it is demonstrated that recognition using YOLO (You Only Look Once) surpasses all the other methods, scoring highest accuracy of over 99%. Yolo is an open source CNN-based object recognition used for in real time videos and tracking. “Unlike region-based approaches, YOLO passes the n by n image only once in a fully convolutional neural network (FCNN), which makes it quite fast and real-time”, (Radovic et al., 2017). A prevalence of object recognition using YOLO achieved high recognition accuracies compared to any CNN solely used for object recognition. Compared to other CNNs presented in table 2, the proposed CNN performs slightly lower by at most 1%. Based on this, the study concludes that the proposed CNN is competitive compared to results by other CNNs and therefore is efficient for UAV object recognition.

5.4 Discussion on UAV object recognition in fog/haze using the proposed method

As discussed in the experiments chapter in 4.4.3, the proposed method was experimented and tested to investigate its performance for UAV object recognition in haze/fog. The goal of the results obtained here was to provide answers to research question 3 of the study that reads:

How effective is the proposed method for addressing UAV object recognition in bad weather conditions?

a. Independent variables that were being experimented are:

- Fog density (measure of fog available in the scene)
- Distance to object (measure of object distance from the UAV in fog)

Dependent variable in this analysis was:

- Detection and recognition accuracy

Therefore, the goal of the experiments conducted was to assess the following:

- Recognition accuracy against varied fog/haze densities
- Recognition accuracy against varied object distance in fog scene

The following section presents and discusses the results of the proposed method for UAV object recognition in haze/fog.

Results and discussion of the proposed method for UAV object recognition in haze/fog

In this section, results for the performance of the proposed method for (a) UAV object recognition in varied fog densities and (b) UAV object recognition in varied object distances in fog are discussed and analyzed. To analyze results, the recognition accuracy was used as an evaluation metric. Therefore, the goal of analysis here is to determine the effectiveness of the proposed method to accurately recognize objects in fog/haze using the UAV.

5.4.1 Discussion of results for UAV object recognition in varied fog densities

In chapter 4, the proposed method was experimented and tested for object recognition in varied fog densities. This section presents a discussion and analysis of results of the performance of the proposed method for object recognition in varied fog densities ranging from light, medium to thick fog.

Discussion of results for UAV object recognition in light fog

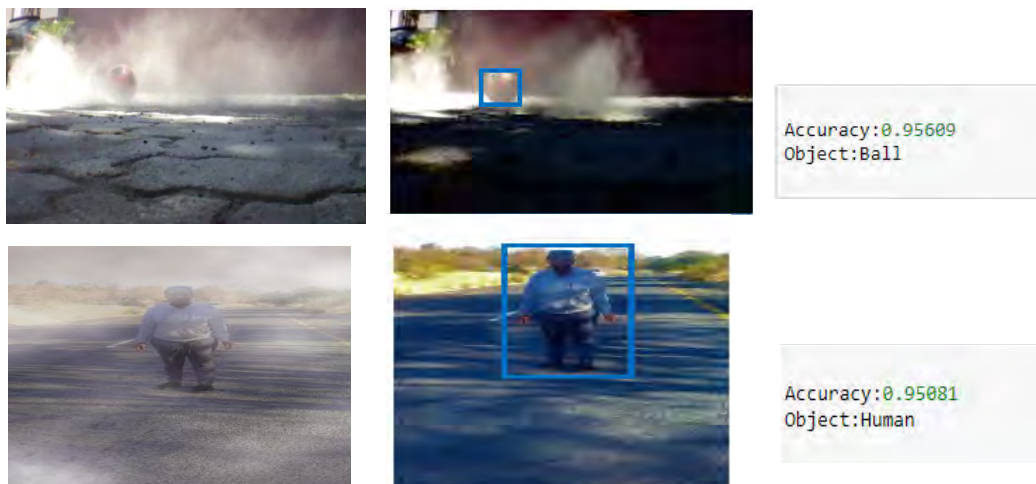


Input image	Object recognition	Recognition accuracy
-------------	--------------------	----------------------

Figure 5.20: UAV object recognition in light fog

The results presented in figure 5.20 demonstrate the recognition accuracy of the proposed method for UAV object recognition in light fog scenes. Results demonstrate recognition accuracy of various objects (balls, horse/donkey) which the proposed method was trained to recognize using the CNN. Results in figure 5.20 demonstrate high recognition accuracy for the objects in light fog. It is noted in figure 5.20 that objects in the input images (before fog removal) appear less visible. However the developed CNN was trained using dataset which contained images taken on clear weather without fog and results for the performance accuracy tested in clear images are demonstrated in section 5.3. As a result, in comparison of recognition accuracy scores on clear images (in section 5.3) against recognition accuracy scores on images defogged by the DCP method (in figure 5.20), the study concludes that the proposed method achieved similar results; therefore this indicates that recognition the implemented DCP is effective for improving visibility by image defogging to enable the task of object recognition by the CNN. Thus, results demonstrated in figure 5.20, illustrated that the analysis and conclusion made by the study for the performance of the DCP method for light fog removal (section 5.2) is accurate. Therefore based on results of the method experimentation in figure 5.20, the proposed method is effective for enabling the UAV recognize objects in light fog.

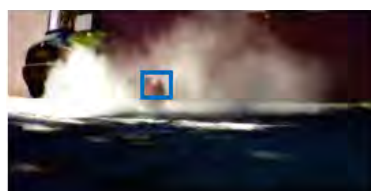
Discussion of results for UAV object recognition in medium fog



Results presented in figure 5.21 demonstrate the recognition accuracy of the proposed method for UAV object recognition in medium fog scenes. Results demonstrate recognition accuracy of various objects used in the training dataset. Results in figure 5.21 demonstrated high recognition accuracy of over the average recognition accuracy score of ~94.3% for the objects in medium fog. Figure 5.21, demonstrated that the input image with medium fog had objects details that appeared obscure; colors of the image were not vivid, edges of object appeared blunt and the scenes were not well defined. Nonetheless, after fog removal by the DCP, scenes in the image appeared more visible and defined; objects information became more defined and the colors appeared more vivid except in regions where fog seemed too concentrated/thick. The results demonstrated that the DCP implemented for the proposed method is effective for improving visibility in medium fog conditions using the UAV.

Results in figure 5.21 demonstrated a high recognition accuracy score for UAV object recognition in bad weather using the proposed method. A comparison of recognition accuracy results for recognition in clear weather in section 5.3 revealed that both UAV object recognition in medium fog weather (figure 5.21) and clear weather (section 5.3) achieved consistent results with an accuracy score of over 90%. These results demonstrated the effectiveness of the implemented DCP for improving visibility by image defogging to enable the task of object recognition by the CNN. Thus, results demonstrated in figure 5.21, illustrated that the analysis and conclusion made by the study for the performance of the DCP method for medium fog removal (section 5.2) is accurate. Therefore these results demonstrated the effectiveness of the proposed method for object recognition in medium fog weather using the UAV.

Discussion of results for UAV object recognition in thick fog



Accuracy:0.90132
Object:Ball

In thick fog, results presented in figure 5.22 demonstrate the recognition accuracy of over 90%. Results demonstrated that the input image scene was completely obscured by fog; the target object for recognition is the ball but its visibility is extremely low in the dense fog. After fog removal, majority of the regions in the scene became visible and defined and therefore the proposed method can detect and recognize the object. However, the output image did not reflect a complete removal of fog; regions which high concentration of fog were still obscured and visibility is limited. The proposed method only revealed just substantial information about the details of the image such as edges, background and colors. In computer vision, this is a remarkable achievement because the proposed method can reveal at least some information about the content of the image and this is useful for object recognition tasks. As a result, the proposed method was able to use the revealed information in the image such as edges and color to recognize the object, at a fairly high accuracy despite other regions of the objects not being completely visible. These results demonstrated the effectiveness of the implemented DCP for improving visibility by image defogging to enable the task of object recognition by the CNN. Based on these results, the study concludes that the proposed method for UAV object recognition in thick fog is fairly good because in extreme cases of dense fog, it is not effective to improve visibility of the scene.

Summary of results of the UAV object recognition in varied fog densities

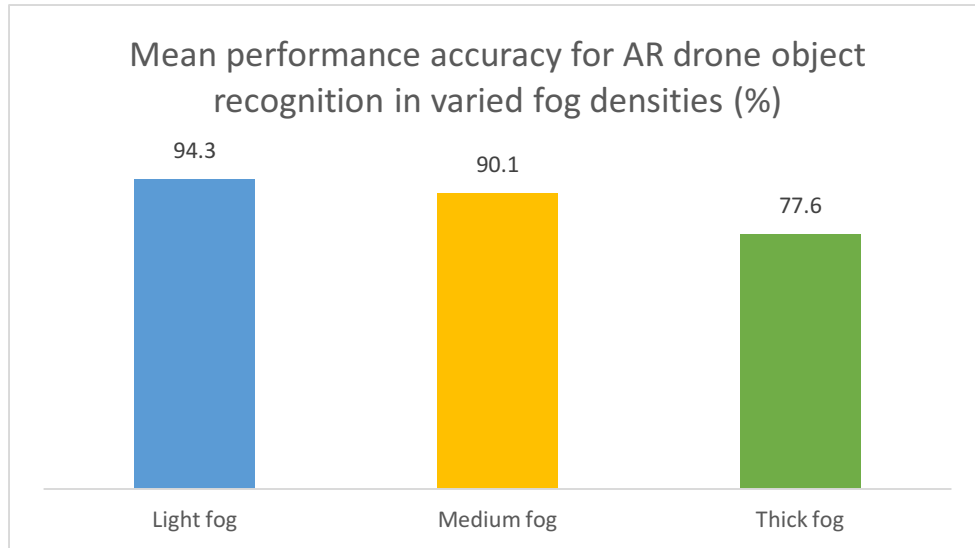


Figure 5.23: Summary of mean performance accuracy results of proposed method for UAV object recognition in varied fog densities using AR drone

Figure 5.23 illustrates results of the proposed method for UAV object recognition in varied densities of fog (light, medium and thick densities) evaluated by the mean performance recognition accuracy. Results in figure 5.23 demonstrated that the proposed method achieves a high performance recognition accuracy in light fog, followed by in medium fog. This is because, in light and medium fog, the proposed method is effective in fog removal to make object recognition possible by the method. As discussed earlier, the DCP for fog removal using UAV achieved good quality images in light and medium fog; therefore the implemented CNN was able to recognize objects on the output images of the DCP method, because the objects in the image scene appeared more visible after fog removal.

However, results in figure 5.23 demonstrate a significant decrease to 77.6% of the mean accuracy for object recognition by the proposed method in thick fog. This accuracy value was accounted for by that occasionally, the proposed method inaccurately detect objects in thick especially in extreme dense fog. On frequent experiments however, the proposed method was able to accurately recognize objects in thick fog because although the proposed method does not work effectively in thick and dense fog, as previously discussed, but it was able to remove significant amount of thick fog from the image; thus revealing important object information such as the

shapes, edges and color and the method used this information to make a prediction about the object.

Based on these results, the line of best fit for the graph depicted that the recognition accuracy of the proposed method for UAV object recognition in fog decreased with increase in fog density. Therefore, the study concludes that the proposed method is effective for object recognition using UAV in light and medium fog densities using the AR drone.

5.4.1 Discussion of results for UAV object recognition at varied distances in fog

In chapter 4, the proposed method was experimented and tested for object recognition at varied distances in fog. This section presents a discussion and analysis of the proposed method for object recognition at varied object distances from the UAV in fog. The results discussed in this section are in the following range based on the line of sight by the AR drone camera:

- Average distance $\approx$ 6 meters/less
- Far distances $\approx$ 9 meters
- Further distance $\approx$ 11 meters
- Beyond sight $\approx$ 12 meters

Discussion of results for UAV object recognition at average distances (5-6 meters)

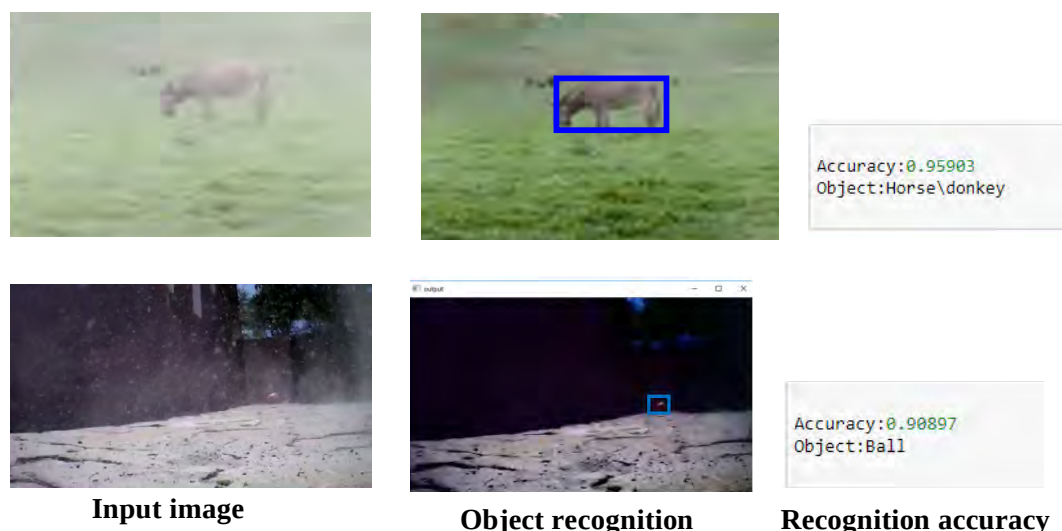


Figure 5.24: UAV object recognition at average distance ($\approx$ 6meters) in light fog (top) and medium fog (bottom)

Results presented in figure 5.24 demonstrate the recognition accuracy of the proposed method for UAV object recognition at average distances of 6 meters from the UAV in light and medium fog. Results demonstrated high recognition accuracy (above 90%) for the objects in both light and medium fog. It is depicted in figure 5.24 that objects in the input images (before fog removal) had less visibility level because of the presence of fog in the scene. However, after fog removal by the proposed method, objects became more visible, together with the background. As a result, the proposed method was able to accurately detect and recognize the target objects in the scene with a high confidence accuracy of over 90%. Additionally, results in figure 5.24 demonstrate how effective the implemented DCP method is for fog removal is for removing fog at depth distances and background in light and medium fog scene to enable the CNN to accurately locate and recognize objects in average distances of 5-6 meters in both medium and light fog.

Therefore, results presented in figure 5.24 illustrate the accuracy of the conclusion made by the study for DCP fog removal at average distance of 5-6m (section 5.2), that the DCP method is effective for improving visibility in light and medium fog to enable the task of object recognition by the CNN. Based on the results and analysis of figure 5.24, the study concludes that the proposed method is effective for enabling the UAV (AR-drone) recognize objects at average distances of ~5 6m in light and medium fog.

Discussion of results for UAV object recognition at far distances (~9 meters)



Results presented in figure 5.25 demonstrate the recognition accuracy of the proposed method for UAV object recognition at far distances of ≈ 9 meters from the UAV (AR drone) in light (top images) and medium (bottom images) fog. In light fog, results demonstrated high recognition accuracy (above 90%) for recognition of objects at ≈ 9 meters; this illustrates that the proposed method for fog removal (DCP) is effective for removing light fog at deeper scenes of ≈ 9 meters in light fog. After fog removal operation, objects appeared more visible with well-defined edges, background, and the vivid color information. Therefore, the effectiveness of the DCP method for removing fog at 9 meters is accounted for the ability of the CNN to accurately detect and recognize objects at such a distance depth in the scene.

However, based on results presented in figure 5.25, the proposed method did not achieve a high accuracy score in medium fog, neither did the method achieve accurate recognition of the object in the scene at a distance of ≈ 9 meters (bottom images). Results demonstrate that after medium fog removal at ≈ 9 meters by the proposed method, the background became too dark while only the foreground had clear visibility. Nonetheless, the darkness in the output image could be accounted for by the dark regions on the background, proving the conclusion by the study in section 5.2 that the DCP does not handle dark regions effectively; they appeared even more darker after the DCP operation. In medium fog with shadowy background, the proposed method does not accurately recognize and the recognition accuracy is low.

Based on the results present in figure 5.25, the study concludes that the proposed method is effective for recognition of big objects in well-illuminated spaces at AR drone far distances (≈ 9 meters), while the method performs poorly for recognition of small-sized objects at far distances (≈ 9 meters) where the spaces are not well-illuminated and have more dark regions in the background.

Discussion of results for UAV object recognition at furthest distances (≈ 11 meters)



Figure 5.26: UAV object recognition at furthest object distance (≈ 11 meters) in medium fog

Results presented in figure 5.26 illustrate the recognition accuracy of the proposed method for UAV object recognition at further distances of ≈ 11 meters from the AR drone in medium fog. In medium fog, results demonstrate that the proposed method does not completely remove fog from the scene, rather only the foreground regions are without fog after applying the method while the background regions contains traces of fog. As a results, objects placed at a further distance (~ 11 meters and beyond) in the scene become obscured by depth fog after applying the method. Beyond the fog in the output image, objects details such as edges, color information are almost completely undefined and appear less visible.

Therefore, the method did not accurately locate the objects. Results in figure 5.26 depicted that the proposed method only locates *possible lookalike* objects in regions which have been defogged and inaccurately recognizes these object. As can be observed in results presented in figure 5.26, the method inaccurately detected the ball in at the furthest distance, rather it predicted the dark circular shadow (at average distance) to be the ball with a confidence accuracy of $\approx 70\%$. The recognition of circular-like shadow in figure 5.26 could be because the pixel representation of that region resembles that of a ball, hence this is why the proposed method make such a precision, with a confidence level of over 70%.

Therefore, based on these results, the study concludes that at furthest AR drone distances (≈ 11 meters and beyond), the proposed method does not effectively recognize objects in medium fog.

Summary of results for UAV object recognition at varied object distances

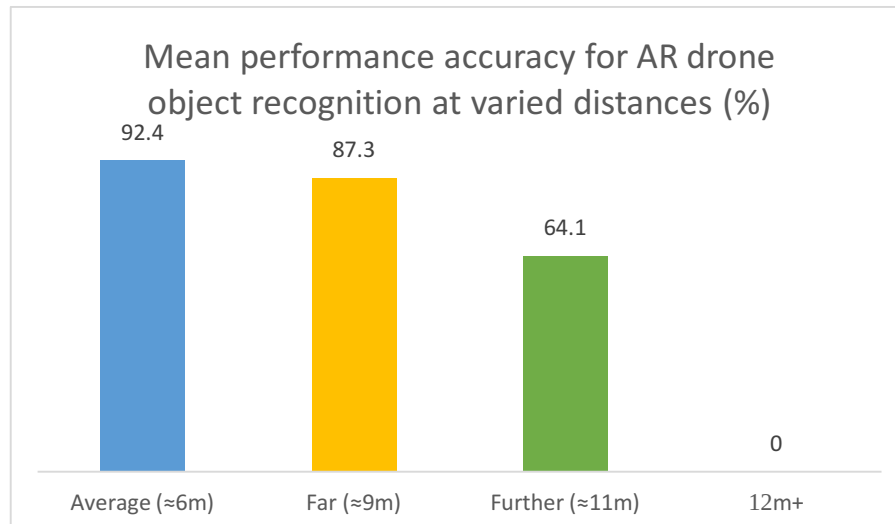


Figure 5.27: Summary of mean performance accuracy of proposed method for UAV object recognition in fog at varied distances of objects using AR drone

Figure 5.27 illustrates results of the proposed method for UAV object recognition at varied object distances from the UAV (AR drone) in fog; evaluated by the mean performance recognition accuracy. Results in figure 5.27 demonstrated that the proposed method achieves a high performance recognition accuracy in distances range not further than ≈ 10 meters from the UAV (AR drone); that is $> \approx 10$ meters. At such distance range, the proposed method was able to score a mean accuracy of over 87% using the AR drone. The accuracy score of more than 87% applied to recognition in both light and medium fog and all objects sizes ranging from small (balls), humans (medium) and large (donkey/horse). The at least 87% recognition accuracy demonstrated in figure 5.27 is accounted for by that occasionally the proposed method would not accurately small objects such as balls, in shady areas, because, as previously discussed, in dark regions such as shades and high contrast colors, the DCP method developed was not effective for fog removal.

Nonetheless, results in figure 5.27 also depicted that at furthest distances range of $\approx 10\text{m} < \text{distance} > \approx 11\text{m}$, the method achieved substantial mean accuracy of at least 64% for object recognition in fog. This is because, as previously discussed, the DCP method used in the study was not adequately effective for fog removal at furthest depths in the images; it just performed good enough to remove the foreground fog especially light and medium fog scenes. In this way objects did not appear fully in the output images because other parts would still be obscured by fog thus not revealing all the object details. As a result, occasionally, the proposed method did not accurately recognize objects with a high confidence accuracy.

Based on these results, the line of best fit for the graph depicted that the recognition accuracy of the proposed method for UAV object recognition in fog decreased with increase in distance. Therefore, the study concludes that the proposed method is effective for object recognition using UAV in distances not further than 10 using the AR drone.

However, performance analysis of the proposed method by comparison was not made because of lack of studies in UAV object recognition in bad weather.

5.5 Discussion on computation time of proposed method

As discussed in the experiments chapter in 4.4.4, the proposed method was experimented to investigate its computational time or response. Series of time recording was done and their average/mean was calculated and profiled. The goal of the results of time profiles obtained was to provide answers to research question 4 of the study that reads:

How efficient is the proposed method for enabling UAV object recognition in bad weather conditions in terms of computation time?

This section presents and discusses the results for computation time of the proposed method for UAV object recognition in bad weather (fog) with the goal to determine the efficiency of the proposed method.

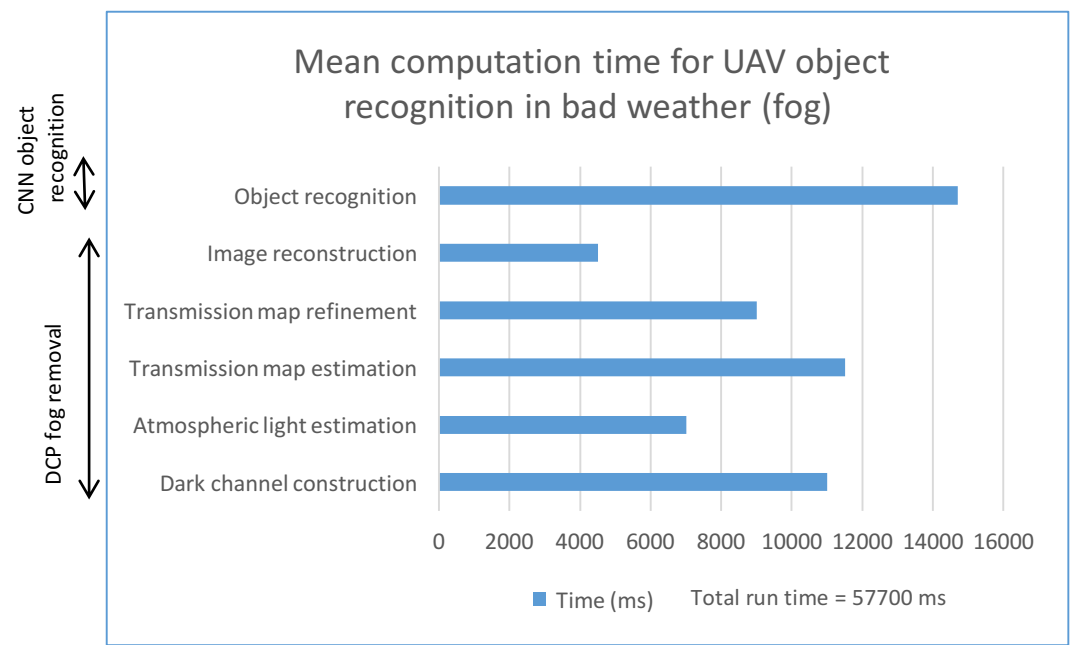


Figure 5.28: Summary of mean computation time (*ms*) for the proposed method executed on AR-drone for object recognition in fog

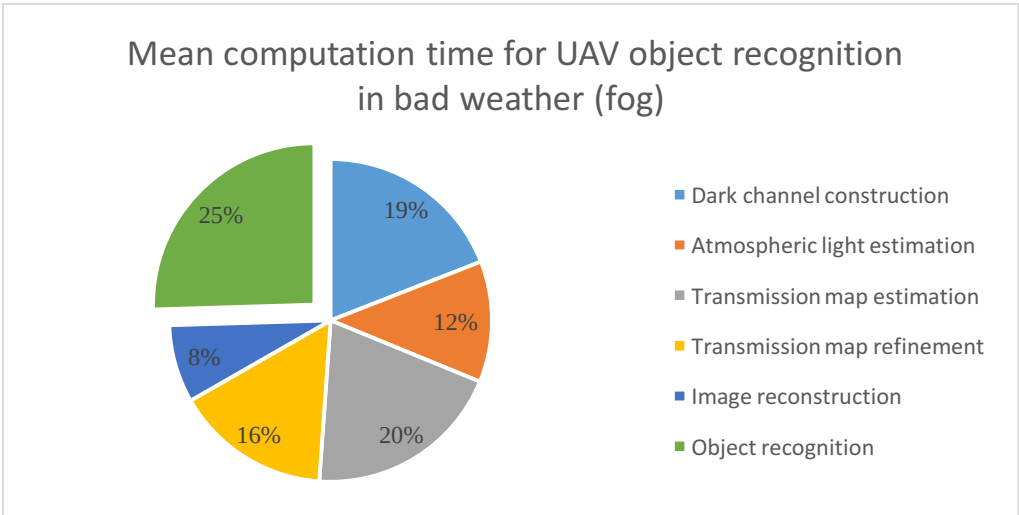


Figure 5.29: Summary of mean computation time (%) for the proposed method executed on AR-drone for object recognition in fog

Figures 5.28 and 5.29 depicts the average run time profiles of the proposed method for UAV object recognition in bad weather using AR-drone 2.0. Figure 5.28 depicts the time consumption, in ms, for each step in the proposed method for object recognition in fog; while figure 5.29 depicts the statistical presentation, in percentage, of run time profiles for each steps in the proposed method.

The graph in figure 5.28 demonstrated run time profiles for each steps in the two main activities in the proposed method which are: (a) DCP method for fog removal and (b) CNN object recognition. These are discussed as follows:

- **Run time profiles for image fog removal using DCP method**

Results presented in table illustrate computation time for each sub-task in the DCP method for fog removal. The following table depicts the following average computation time profiles for the DCP sub-operations as depicted in table 3:

Table 3: Computation time (ms) of the DCP sub-operations

Time profiles for the DCP sub-operation	
Sub-operation	Time (ms)
Dark channel construction	11000
Atmospheric light estimation	7000
Transmission map estimation	11500
Transmission map refinement	9000
Image reconstruction	4500

Total DCP run time	43000
--------------------	-------

According to computation time statistics presented in figure 5.29, more time is constituted in the computation of transmission map estimation (20%), followed by the dark channel construction (19%), while transmission map refinement constitutes 16%, and the atmospheric light estimation constitutes 12%. The least time is on the image reconstruction with a computation time constituting 8% of the proposed method. Based on results, the average run time profile of fog removal by DCP method consumes 43000ms of time, denoting 75% of the total run time of the proposed method.

● **Run time profiles for object recognition using CNN**

Results presented in figure 5.28 illustrate computation time for object recognition with an average total of 14700 ms. This illustrated a consumption of 25% of the proposed method using AR drone 2.0 as shown in figure 5.29. Results depicted that the developed CNN model for object recognition consumed the least computation time in comparison with the fog removal operation using the DCP method.

Therefore, based on results, the study concluded that the implemented DCP method consumed more time for the proposed method than the recognition task by the CNN.

Analysis of run time profiles of the proposed method

The average run times of the proposed method presented in figure 5.28 demonstrated a total of 57700 ms. In this study, analysis by comparison of run time profiles for the proposed method was limited because there were no available results for related works. However, there are apparent latency factors that contributed to the computation times presented in figure 5.28 and these were identified as follows:

- The method was executed on an intel core i3 CPU which is less powerful processor for image processing applications as compared to GPU processor

- The method was executed by a wireless means via wifi between the AR drone and the computer, therefore latency in communication of devices was expected
- The proposed method is a linear approach, therefore latency was expected as one operation depended on the output of the other (queuing is expected)
- AR drone is generally considered slower than other powerful drones, especially for real time applications such as video recording

There, based of these factors contributing to the run time profile, the study concluded that the proposed method is fairly time efficient, however best resources such as powerful processors could be used to improve computation time.

5.6 Summary of findings

Chapter has presented the results of experiments of the proposed method for UAV object recognition in bad weather conditions. Results were presented, discussed and analyzed in stages of the method as follows:

- UAV fog removal using DCP

Based on the results of the DCP method for image defogging and their evaluation using the dehazed/defogged image quality metrics, the study found out that the DCP used for scene fog removal using UAV is effective. The DCP for UAV fog removal is more effective in removal of light and medium fog. However, in extreme cases of thick fog, the study found that the DCP method achieved just substantial results where other regions, especially the foreground regions are well defogged and the background still contained fog in the background. Based on result for UAV fog removal using the DCP, the study also discovered that good results are achieved in well illuminated areas than in shadowy areas. In shadowy areas, the DCP suffers color over-saturation, the dark regions become even more darker therefore making the scene even more invisible. Results of the used DCP were also compared to other image defogging/dehazing methods by running the used DCP on similar hazy/foggy images. Results for comparison demonstrated that the DCP used in this study achieved quality images in light and medium fog compared to previous methods. In

conclusion, the study found that the DCP used for UAV object recognition is effective in light and medium fog removal especially in distances that are not too far from the object; denoting that the method can enable the UAV achieve improved visibility in a radius of ~11m as it flies. However, in extreme cases of fog and in dark regions, the used DCP suffered is limited and could be improved.

- Object recognition

Based on the results of UAV object recognition using the developed CNN were and the evaluation of the model using the performance accuracy, the study found that the implemented CNN is effective for UAV object recognition. The CNN was analysed for its ability to accurately detect objects of various sizes in various distances from the UAV. Based on this, results demonstrated that the model achieved a high accuracy of 94.3% on average. Analysis demonstrated that at longer distances, with the UAV's line of sight, the CNN achieved good recognition accuracy. Recognition accuracy of the results was compared to accuracy of related studies. Comparison demonstrated that the CNN model used in this study performed slightly lower, by at most ~1% than other CNNs for object recognition using UAVs. Comparison demonstrated that the CNN used in this study was outperformed by the YOLO model, which scored over 99% accuracy. Based on this, the study concluded that the used CNN is effective for object recognition using UAV, however, it can still be improved to achieve even higher recognition accuracy.

- Analysis on the proposed method

The proposed method was evaluated and analyzed for its ability to: (1) recognize objects in different densities of fog and (2) recognize objects in distances away from the UAV in fog. Based on the results and analysis, the study discovered that in light and medium fog, the proposed method can effectively recognize objects. This was determined based on the high accuracy scored by the method for recognition of objects in light and medium fog. However, in thick/dense fog, results and analysis revealed that the proposed method is not effective. In dense fog, the proposed method achieved decreased recognition accuracy. Therefore, based on this, the study concludes that on medium and light fog density, the proposed method is effective. However, more improvement is required for object recognition in dense fog.

Additionally, for object recognition at various distances, analysis of results demonstrated that in light and medium fog, the proposed method is effective for object recognition at all ranges of distances that are in the line of sight by the UAV. However, the proposed method struggled to accurately objects at further distances in thick fog. Based on this, the study determined that the proposed method is effective for UAV object recognition at all distances within the line of sight in only medium and light fog densities.

- Computation time of the proposed method

Lastly, results for the run time profiling of the proposed method was also discussed and analyzed. Based on the analysis of run time profile, the study determined that the response time was slow. Although, this finding was not made in comparison to any apparent method, due to lack of available results on studies similar, the results can still be further argued. However, in comparison to UAV object recognition in clear weather methods, the study determined that this proposed method is not time efficient.

Chapter 6: Conclusions and recommendations

This study sought to propose and investigate the performance a method for UAV object recognition in bad weather. In bad weather conditions such as fog and haze, UAVs are often challenged to perform object recognition tasks due to limited visibility. Numerous studies have been done on UAV object recognition in clear weather, while not focusing on UAV object recognition on bad weather. However,

vast state-of-the-art methods and algorithms exist for image dehazing/de-fogging and object recognition. Hence, the focus of this study was proposing and analyzing a method that employed the DCP for fog/haze removal and CNN for object recognition in bad weather using UAVs.

In summary, the focus of this study was particularly based on determining the effectiveness of the proposed method for addressing UAV object recognition in bad weather by asking:

1. How effective is the DCP method for enabling the UAV enhance scenes captured in bad weather conditions?
2. How effective, in terms of accuracy, is the developed CNN model for UAV object recognition?
3. How effective is the proposed method for addressing UAV object recognition in bad weather conditions?
4. How efficient is the proposed method for enabling UAV object recognition in bad weather conditions in terms of computation time?

To establish answers to the research questions, the study implemented and employed the experiment methods, to investigate the proposed method for object recognition, in bad weather (fog simulation), using the UAV (AR- drone).

In the first phase of investigation, the DCP algorithm was experimented for fog removal using the UAV. The goal of DCP method was to achieve improved scene visibility by the UAV in bad weather. Results of the series of experiments conducted here were discussed and analyzed using the dehazed/defogged scene quality metrics. The DCP was investigated for effective fog removal in (a) varied fog densities in the range of light, medium and thick fog, and; (b) varied distances in the ranges within the line of sight by the UAV.

Results of fog removal in varied densities of fog demonstrated that DCP method achieve better scene visibility in light and medium fog. Scenes became more visible after fog removal by the UAV using the DCP method. However, in thick fog, certain

regions where fog seemed extreme, the DCP could perform better to improve scene visibility, thus indicating a limitation.

Moreover, results in fog removal in varied depth distances depicted that the DCP is effective in object distances up to ~11m in light and medium fog; beyond that the results demonstrated a limitation in fog removal. Results of the DCP method for fog removal were compared with other studies that implemented the DCP for fog removal; based on comparison, the proposed method outperformed results by other studies.

Based on analysis of results, the study determined the answer for research question 1; the DCP used for UAV fog removal performs effectively in light and medium fog and in distances not further than ~11m. However, the method has limitations in thick fog removal therefore further work is recommended to enable the UAV to improve scene visibility in extreme densities of fog and minimize color over-saturation in dark regions.

In the second phase of investigation, a CNN was developed and experimented for object recognition using UAV. Results of the series of experiments conducted here were discussed and analyzed using the performance accuracy as metrics to evaluate the performance of the CNN. The implemented CNN was investigated for performance accuracy for recognizing objects at varied distances and appearance in sizes. Results demonstrated that at all distances, within the line of sight of the UAV, the CNN achieved accuracy score of 94.3% on average. In comparison with other CNN for UAV object recognition, the accuracy of CNN used in this study was slightly lower by ~1%. However, a recognition accuracy of 94.3% was considered effective for the study because it posed a higher probability of detecting target objects on a given scene using the UAV.

Based on analysis of results, the study was able to establish an answer for research question 2 of the study; the implemented CNN is effective for UAV object recognition. However, to improve the recognition accuracy, further work using CNN with YOLO is recommended as results of YOLO outperforms most recognition problems as depicted on table 2.

In the third phase of investigation, the proposed method that deployed the DCP for fog removal and CNN for object recognition was experimented for UAV object recognition in bad weather. The goal of the proposed method was to enable UAV to accurately recognize objects in bad weather (haze/fog). Results of the series of experiments conducted here were discussed and analyzed using the recognition accuracy as evaluation metrics. The proposed method was investigated for accurate object recognition in: (a) varied densities of fog and, (b) varied object distances in fog, using UAV.

Results for UAV object recognition in bad weather demonstrated that the method achieves better object recognition in light and medium fog; as denoted by high accuracy scores for recognition. However, in dense fog scenes, the proposed method did not score high recognition accuracy; denoting that the method did not achieve visible scenes to reveal object details.

Additionally, results of the proposed method for object recognition at varied distances demonstrated accurate object recognition in distances within the radius of less than ~11m in light and medium fog, beyond this radius, the method could not accurately recognize the objects, especially small sized objects such as balls. In thick fog, the proposed method could recognize big sized objects such as donkey and humans within the radius not exceeding ~5m.

Based on analysis of results, the study was able to establish an answer for research question 3 of the study; the study determined that the developed method is effective for accurate object recognition, of all sizes, in medium and light fog within a distance radius of less than ~11m. In thick fog proposed method is likely unreliable especially for recognizing small objects and therefore further work is recommended to enable the method to achieve better visibility to allow recognition of objects in extreme fog densities.

In the last phase of the investigations, the the proposed method was observed for the response time. Run time by the proposed method were profiled and results were analyzed. Based on analysis of results, the study determined an answer for research

question 4; the study determined that the developed method is not time efficient as the method consumed more time. Therefore, further work to propose methods that are time efficient is would be essential to provide further insights.

In conclusion, this study has presented a systematic way to addressing the problem of UAV object recognition in bad weather and has provided insight on how the problem can be solved using the proposed method. Limitations of the study were discussed and it is hoped that this thesis lead to further insights on the subject.

Bibliography

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., Mougiakakou, S., 2016. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep

Convolutional Neural Network. IEEE Trans. Med. Imaging 35, 1207–1216.
<https://doi.org/10.1109/TMI.2016.2535865>

Ayash, E.M.M., n.d. Research Methodologies in Computer Science and Information Systems 4.

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., Ouni, K., 2018. Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3.

Bhaskaranand, M., Gibson, J.D., 2011. Low-complexity video encoding for UAV reconnaissance and surveillance. 2011 - MILCOM 2011 Mil. Commun. Conf. 1633–1638. <https://doi.org/10.1109/milcom.2011.6127543>

Ciresan, D., Meier, U., Masci, J., Gambardella, L., Schmidhuber, J., 2011. Flexible, High Performance Convolutional Neural Networks for Image Classification. Presented at the International Joint Conference on Artificial Intelligence IJCAI-2011, pp. 1237–1242. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>

Cireşan, D., Meier, U., Schmidhuber, J., 2012. Multi-column Deep Neural Networks for Image Classification. ArXiv12022745 Cs.

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). CoRR abs/1511.07289.

Cohen, T.S., Welling, M., 2016. Group Equivariant Convolutional Networks. ArXiv160207576 Cs Stat.

DJI - The World Leader in Camera Drones/Quadcopters for Aerial Photography [WWW Document], n.d. URL <https://www.dji.com/> (accessed 9.9.19).

Dong, N., Jia, Z., Shao, J., Li, Z., Liu, F., Zhao, J., Peng, pei-yuan, 2011. Adaptive Object Detection and Visibility Improvement in Foggy Image. J. Multimed. 6, 14–21. <https://doi.org/10.4304/jmm.6.1.14-21>

Drones: the pros and cons for community-based monitoring | Forest Compass [WWW Document], n.d. URL <https://forestcompass.org/drones-pros-and-cons-community-based-monitoring> (accessed 9.9.19).

Duth, S., Raj, S., 2018. Object recognition in images using convolutional neural network. pp. 718–722. <https://doi.org/10.1109/ICISC.2018.8398893>

Experimental Research - A Guide to Scientific Experiments [WWW Document], n.d. URL <https://explorable.com/experimental-research> (accessed 9.10.19).

Fattal, R., 2014. Dehazing Using Color-Lines. *ACM Trans. Graph.* 34, 1–14. <https://doi.org/10.1145/2651362>

Gay, L.R., 1992. Educational Research: Competencies for Analysis and Application. Merrill.

Geethu, H., Shamna, S., Kizhakkethottam, J., 2016. Weighted Guided Image Filtering and Haze Removal in Single Image. *Procedia Technol.* 24, 1475–1482. <https://doi.org/10.1016/j.protcy.2016.05.248>

Gibson, K.B., Nguyen, T.Q., 2013. Fast single image fog removal using the adaptive Wiener filter. 2013 IEEE Int. Conf. Image Process. 714–718. <https://doi.org/10.1109/icip.2013.6738147>

Gibson, K.B., Vo, D.T., Nguyen, T.Q., 2012. An Investigation of Dehazing Effects on Image and Video Coding. *IEEE Trans. Image Process.* 21, 662–673. <https://doi.org/10.1109/TIP.2011.2166968>

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res. - Proc. Track 9*, 249–256.

Hamidoğlu, A., 2016. On general form of the Tanh method and its application to nonlinear partial differential equations. *Numer. Algebra Control Optim.* 6, 175. <https://doi.org/10.3934/naco.2016007>

Harland, D.J., 2011. STEM student research handbook. National Science Teachers Association, Arlington, Va.

Hautière, N., Tarel, J.-P., DIDIER, A., Dumont, E., 2008. Blind Contrast Enhancement Assessment by Gradient Ratioing at Visible Edges. *Image Anal. Stereol.* 27. <https://doi.org/10.5566/ias.v27.p87-95>

He, K., Sun, J., Tang, X., 2013. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1397–1409. <https://doi.org/10.1109/TPAMI.2012.213>

He, K., Sun, J., Tang, X., 2011. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. <https://doi.org/10.1109/TPAMI.2010.168>

He, K., Sun, J., Tang, X., 2009. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. <https://doi.org/10.1109/TPAMI.2010.168>

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE Int. Conf. Comput. Vis. ICCV 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>

Heaton, J., 2017. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. *Genet. Program. Evolvable Mach.* 19. <https://doi.org/10.1007/s10710-017-9314-z>

Hijazi, S., Kumar, R., Rowen, C., n.d. Using Convolutional Neural Networks for Image Recognition 12.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* 18, 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>

Hsieh, C.-H., Chen, J.-Y., Zhao, Q., 2018. A Modified DCP Based Dehazing Algorithm. <https://doi.org/10.1109/SMC.2018.00307>

Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>

Iliev, A., Kyurkchiev, N., Markov, S., 2017. On the approximation of the step function by some sigmoid functions. *Math. Comput. Simul., Biomath 2014 and Biomath 2015* 133, 223–234. <https://doi.org/10.1016/j.matcom.2015.11.005>

intro_cnn.pdf, n.d.

Kalogeiton, V., n.d. Introduction to Convolutional Neural Networks 109.

Kanellakis, C., Nikolakopoulos, G., 2017. Survey on Computer Vision for UAVs: Current Developments and Trends. *J. Intell. Robot. Syst.* <https://doi.org/10.1007/s10846-017-0483-z>

Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification. *ArXiv14085882 Cs*.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Commun ACM* 60, 84–90. <https://doi.org/10.1145/3065386>

Laughlin, S.B., Sejnowski, T.J., 2003. Communication in neuronal networks. *Science* 301, 1870–1874. <https://doi.org/10.1126/science.1089662>

Lawrence, S., Giles, C.L., Ah Chung Tsoi, Back, A.D., 1997. Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw.* 8, 98–113. <https://doi.org/10.1109/72.554195>

LeCun, Y., Bottou, L., Bengio, Y., Ha, P., 1998. Gradient-Based Learning Applied to Document Recognition 46.

Lee, J., Wang, J., Crandall, D., Šabanović, S., Fox, G., 2017. Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles, in: 2017 First IEEE International Conference on Robotic Computing (IRC). Presented at the 2017 First IEEE International Conference on Robotic Computing (IRC), pp. 36–43. <https://doi.org/10.1109/IRC.2017.77>

Lee, S., Yun, S., Nam, J.-H., Won, C.S., Jung, S.-W., 2016. A review on dark channel prior based image dehazing algorithms. *EURASIP J. Image Video Process.* 2016, 4. <https://doi.org/10.1186/s13640-016-0104-y>

Leroy, G., 2011. Designing User Studies in Informatics, Health Informatics. Springer-Verlag, London.

Li, C., Guo, J., Porikli, F., Fu, H., Pang, Y., 2018. A Cascaded Convolutional Neural Network for Single Image Dehazing. *ArXiv180307955 Cs*.

- Li, Z., Zheng, J., Zhu, Z., Yao, W., Wu, S., 2015. Weighted guided image filtering. *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* 24, 120–129. <https://doi.org/10.1109/TIP.2014.2371234>
- Lin, G., Shen, W., 2018. Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Comput. Sci.* 131, 977–984. <https://doi.org/10.1016/j.procs.2018.04.239>
- Lu, H., Li, Y., Nakashima, S., Serikawa, S., 2015. Single Image Dehazing through Improved Atmospheric Light Estimation. *Multimed. Tools Appl.* <https://doi.org/10.1007/s11042-015-2977-7>
- Ma, J., Wang, W., Wang, L., 2017. Irregular Convolutional Neural Networks. 2017 4th IAPR Asian Conf. Pattern Recognit. ACPR 268–273. <https://doi.org/10.1109/acpr.2017.108>
- Markram, H., Lübke, J., Frotscher, M., Roth, A., Sakmann, B., 1997. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *J. Physiol.* 500 (Pt 2), 409–440. <https://doi.org/10.1113/jphysiol.1997.sp022031>
- Mason, A., Nicoll, A., Stratford, K.J., 1991. Synaptic transmission between individual pyramidal neurons of the rat visual cortex in vitro. *J. Neurosci. Off. J. Soc. Neurosci.* 11, 72–84. <https://doi.org/10.1523/jneurosci.11-01-00072.1991>
- Mobahi, H., Collobert, R., Weston, J., 2009. Deep learning from temporal coherence in video. Presented at the Proceedings of the 26th International Conference On Machine Learning, ICML 2009, p. 93. <https://doi.org/10.1145/1553374.1553469>
- Moore, D.S., 2007. Introduction to the Practice of Statistics, Standard (Cloth) & CD-ROM Hardcover, 6 Har/Cdr edition. ed. W. H. Freeman.
- Movshon, J.A., Thompson, I.D., Tolhurst, D.J., 1978. Receptive field organization of complex cells in the cat's striate cortex. *J. Physiol.* 283, 79–99.
- Nagi, J., Ducatelle, F., Caro, G.A.D., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., Gambardella, L.M., 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition, in: 2011 IEEE International

Conference on Signal and Image Processing Applications (ICSIPA). Presented at the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 342–347. <https://doi.org/10.1109/ICSIPA.2011.6144164>

Narasimhan, S.G., Nayar, S.K., 2002. Vision and the Atmosphere. *Int. J. Comput. Vis.* 48, 233–254. <https://doi.org/10.1023/A:1016328200723>

Narasimhan, S.G., Nayar, S.K., 2000. Chromatic framework for vision in bad weather, in: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. Presented at the *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, pp. 598–605 vol.1. <https://doi.org/10.1109/CVPR.2000.855874>

Nayar, S.K., Narasimhan, S.G., 1999. Vision in bad weather, in: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Presented at the *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 820–827 vol.2. <https://doi.org/10.1109/ICCV.1999.790306>

Oliveira, D.C.D., Wehrmeister, M.A., 2016. Towards Real-Time People Recognition on Aerial Imagery Using Convolutional Neural Networks, in: *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*. Presented at the *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 27–34. <https://doi.org/10.1109/ISORC.2016.14>

5.3. Preprocessing data — scikit-learn 0.21.3 documentation [WWW Document], n.d. URL <https://scikit-learn.org/stable/modules/preprocessing.html> (accessed 9.10.19).

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., Mougiakakou, S., 2016. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. *IEEE Trans. Med. Imaging* 35, 1207–1216. <https://doi.org/10.1109/TMI.2016.2535865>

Ayash, E.M.M., n.d. *Research Methodologies in Computer Science and Information Systems* 4.

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., Ouni, K., 2018. Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3.

Bhaskaranand, M., Gibson, J.D., 2011. Low-complexity video encoding for UAV reconnaissance and surveillance. 2011 - MILCOM 2011 Mil. Commun. Conf. 1633–1638. <https://doi.org/10.1109/milcom.2011.6127543>

Ciresan, D., Meier, U., Masci, J., Gambardella, L., Schmidhuber, J., 2011. Flexible, High Performance Convolutional Neural Networks for Image Classification. Presented at the International Joint Conference on Artificial Intelligence IJCAI-2011, pp. 1237–1242. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210>

Cireşan, D., Meier, U., Schmidhuber, J., 2012. Multi-column Deep Neural Networks for Image Classification. ArXiv12022745 Cs.

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). CoRR abs/1511.07289.

Cohen, T.S., Welling, M., 2016. Group Equivariant Convolutional Networks. ArXiv160207576 Cs Stat.

DJI - The World Leader in Camera Drones/Quadcopters for Aerial Photography [WWW Document], n.d. URL <https://www.dji.com/> (accessed 9.9.19).

Dong, N., Jia, Z., Shao, J., Li, Z., Liu, F., Zhao, J., Peng, pei-yuan, 2011. Adaptive Object Detection and Visibility Improvement in Foggy Image. J. Multimed. 6, 14–21. <https://doi.org/10.4304/jmm.6.1.14-21>

Drones: the pros and cons for community-based monitoring | Forest Compass [WWW Document], n.d. URL <https://forestcompass.org/drones-pros-and-cons-community-based-monitoring> (accessed 9.9.19).

Duth, S., Raj, S., 2018. Object recognition in images using convolutional neural network. pp. 718–722. <https://doi.org/10.1109/ICISC.2018.8398893>

Experimental Research - A Guide to Scientific Experiments [WWW Document], n.d. URL <https://explorable.com/experimental-research> (accessed 9.10.19).

Fattal, R., 2014. Dehazing Using Color-Lines. ACM Trans. Graph. 34, 1–14. <https://doi.org/10.1145/2651362>

Gay, L.R., 1992. Educational Research: Competencies for Analysis and Application. Merrill.

Geethu, H., Shamna, S., Kizhakkethottam, J., 2016. Weighted Guided Image Filtering and Haze Removal in Single Image. *Procedia Technol.* 24, 1475–1482. <https://doi.org/10.1016/j.protcy.2016.05.248>

Gibson, K.B., Nguyen, T.Q., 2013. Fast single image fog removal using the adaptive Wiener filter. 2013 IEEE Int. Conf. Image Process. 714–718. <https://doi.org/10.1109/icip.2013.6738147>

Gibson, K.B., Vo, D.T., Nguyen, T.Q., 2012. An Investigation of Dehazing Effects on Image and Video Coding. *IEEE Trans. Image Process.* 21, 662–673. <https://doi.org/10.1109/TIP.2011.2166968>

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res. - Proc. Track 9*, 249–256.

Hamidoğlu, A., 2016. On general form of the Tanh method and its application to nonlinear partial differential equations. *Numer. Algebra Control Optim.* 6, 175. <https://doi.org/10.3934/naco.2016007>

Harland, D.J., 2011. STEM student research handbook. National Science Teachers Association, Arlington, Va.

Hautière, N., Tarel, J.-P., DIDIER, A., Dumont, E., 2008. Blind Contrast Enhancement Assessment by Gradient Ratioing at Visible Edges. *Image Anal. Stereol.* 27. <https://doi.org/10.5566/ias.v27.p87-95>

He, K., Sun, J., Tang, X., 2013. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1397–1409. <https://doi.org/10.1109/TPAMI.2012.213>

He, K., Sun, J., Tang, X., 2011. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. <https://doi.org/10.1109/TPAMI.2010.168>

He, K., Sun, J., Tang, X., 2009. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. <https://doi.org/10.1109/TPAMI.2010.168>

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE Int. Conf. Comput. Vis. ICCV 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>

Heaton, J., 2017. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. *Genet. Program. Evolvable Mach.* 19. <https://doi.org/10.1007/s10710-017-9314-z>

Hijazi, S., Kumar, R., Rowen, C., n.d. Using Convolutional Neural Networks for Image Recognition 12.

Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* 18, 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>

Hsieh, C.-H., Chen, J.-Y., Zhao, Q., 2018. A Modified DCP Based Dehazing Algorithm. <https://doi.org/10.1109/SMC.2018.00307>

Hubel, D.H., Wiesel, T.N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>

Iliev, A., Kyurkchiev, N., Markov, S., 2017. On the approximation of the step function by some sigmoid functions. *Math. Comput. Simul., Biomath 2014 and Biomath 2015* 133, 223–234. <https://doi.org/10.1016/j.matcom.2015.11.005>

intro_cnn.pdf, n.d.

Kalogeiton, V., n.d. Introduction to Convolutional Neural Networks 109.

Kanellakis, C., Nikolakopoulos, G., 2017. Survey on Computer Vision for UAVs: Current Developments and Trends. *J. Intell. Robot. Syst.* <https://doi.org/10.1007/s10846-017-0483-z>

Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification. ArXiv14085882 Cs.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Commun ACM 60, 84–90. <https://doi.org/10.1145/3065386>

Laughlin, S.B., Sejnowski, T.J., 2003. Communication in neuronal networks. Science 301, 1870–1874. <https://doi.org/10.1126/science.1089662>

Lawrence, S., Giles, C.L., Ah Chung Tsoi, Back, A.D., 1997. Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. 8, 98–113. <https://doi.org/10.1109/72.554195>

LeCun, Y., Bottou, L., Bengio, Y., Ha, P., 1998. Gradient-Based Learning Applied to Document Recognition 46.

Lee, J., Wang, J., Crandall, D., Šabanović, S., Fox, G., 2017. Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles, in: 2017 First IEEE International Conference on Robotic Computing (IRC). Presented at the 2017 First IEEE International Conference on Robotic Computing (IRC), pp. 36–43. <https://doi.org/10.1109/IRC.2017.77>

Lee, S., Yun, S., Nam, J.-H., Won, C.S., Jung, S.-W., 2016. A review on dark channel prior based image dehazing algorithms. EURASIP J. Image Video Process. 2016, 4. <https://doi.org/10.1186/s13640-016-0104-y>

Leroy, G., 2011. Designing User Studies in Informatics, Health Informatics. Springer-Verlag, London.

Li, C., Guo, J., Porikli, F., Fu, H., Pang, Y., 2018. A Cascaded Convolutional Neural Network for Single Image Dehazing. ArXiv180307955 Cs.

Li, Z., Zheng, J., Zhu, Z., Yao, W., Wu, S., 2015. Weighted guided image filtering. IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 24, 120–129. <https://doi.org/10.1109/TIP.2014.2371234>

Lin, G., Shen, W., 2018. Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Comput. Sci.* 131, 977–984. <https://doi.org/10.1016/j.procs.2018.04.239>

Lu, H., Li, Y., Nakashima, S., Serikawa, S., 2015. Single Image Dehazing through Improved Atmospheric Light Estimation. *Multimed. Tools Appl.* <https://doi.org/10.1007/s11042-015-2977-7>

Ma, J., Wang, W., Wang, L., 2017. Irregular Convolutional Neural Networks. 2017 4th IAPR Asian Conf. Pattern Recognit. ACPR 268–273. <https://doi.org/10.1109/acpr.2017.108>

Markram, H., Lübke, J., Frotscher, M., Roth, A., Sakmann, B., 1997. Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *J. Physiol.* 500 (Pt 2), 409–440. <https://doi.org/10.1113/jphysiol.1997.sp022031>

Mason, A., Nicoll, A., Stratford, K.J., 1991. Synaptic transmission between individual pyramidal neurons of the rat visual cortex in vitro. *J. Neurosci. Off. J. Soc. Neurosci.* 11, 72–84. <https://doi.org/10.1523/jneurosci.11-01-00072.1991>

Mobahi, H., Collobert, R., Weston, J., 2009. Deep learning from temporal coherence in video. Presented at the Proceedings of the 26th International Conference On Machine Learning, ICML 2009, p. 93. <https://doi.org/10.1145/1553374.1553469>

Moore, D.S., 2007. Introduction to the Practice of Statistics, Standard (Cloth) & CD-ROM Hardcover, 6 Har/Cdr edition. ed. W. H. Freeman.

Movshon, J.A., Thompson, I.D., Tolhurst, D.J., 1978. Receptive field organization of complex cells in the cat's striate cortex. *J. Physiol.* 283, 79–99.

Nagi, J., Ducatelle, F., Caro, G.A.D., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., Gambardella, L.M., 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition, in: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). Presented at the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 342–347. <https://doi.org/10.1109/ICSIPA.2011.6144164>

Narasimhan, S.G., Nayar, S.K., 2002. Vision and the Atmosphere. *Int. J. Comput. Vis.* 48, 233–254. <https://doi.org/10.1023/A:1016328200723>

Narasimhan, S.G., Nayar, S.K., 2000. Chromatic framework for vision in bad weather, in: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. Presented at the *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, pp. 598–605 vol.1. <https://doi.org/10.1109/CVPR.2000.855874>

Nayar, S.K., Narasimhan, S.G., 1999. Vision in bad weather, in: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Presented at the *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pp. 820–827 vol.2. <https://doi.org/10.1109/ICCV.1999.790306>

Oliveira, D.C.D., Wehrmeister, M.A., 2016. Towards Real-Time People Recognition on Aerial Imagery Using Convolutional Neural Networks, in: *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*. Presented at the *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 27–34. <https://doi.org/10.1109/ISORC.2016.14>

Olshausen, B.A., Field, D.J., 2004. Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* 14, 481–487. <https://doi.org/10.1016/j.conb.2004.07.007>

O’Shea, K., Nash, R., n.d. *An Introduction to Convolutional Neural Networks* 12.

Pang, J., Au, O.C., Guo, Z., 2011. Improved single image dehazing using guided filter.

Potdar, K., S., T., D., C., 2017. A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *Int. J. Comput. Appl.* 175, 7–9. <https://doi.org/10.5120/ijca2017915495>

Preprocessing data — scikit-learn 0.21.3 documentation [WWW Document], n.d. URL <https://scikit-learn.org/stable/modules/preprocessing.html> (accessed 9.10.19).

Quality Assessment of Unmanned Aerial Vehicle (UAV) Based Visual Inspection of Structures - G. Morgenthal, N. Hallermann, 2014 [WWW Document], n.d. URL <https://journals.sagepub.com/doi/abs/10.1260/1369-4332.17.3.289> (accessed 9.9.19).

Radovic, M., Adarkwa, O., Wang, Q., 2017. Object Recognition in Aerial Images Using Convolutional Neural Networks. *J. Imaging* 3, 21. <https://doi.org/10.3390/jimaging3020021>

Rectifier Nonlinearities Improve Neural Network Acoustic Models - Epistemo, n.d.

Robson, C., 2002. *Real World Research: A Resource for Social Scientists and Practitioner-Researchers* / C. Robson.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 211–252. <https://doi.org/10.1007/s11263-015-0816-y>

Sa, I., Hrabar, S., Corke, P., 2015. Outdoor Flight Testing of a Pole Inspection UAV Incorporating High-speed Vision, in: Mejias, L., Corke, P., Roberts, J. (Eds.), *Field and Service Robotics: Results of the 9th International Conference*, Springer Tracts in Advanced Robotics. Springer International Publishing, Cham, pp. 107–121. https://doi.org/10.1007/978-3-319-07488-7_8

Santra, S., Chanda, B., 2015. Single image dehazing with varying atmospheric light intensity, in: 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG). Presented at the 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), IEEE, Patna, India, pp. 1–4. <https://doi.org/10.1109/NCVPRIPG.2015.7490015>

Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, in: *Artificial Neural Networks – ICANN 2010*. Presented at the International Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, pp. 92–101. https://doi.org/10.1007/978-3-642-15825-4_10

- Senior, A., Lei, X., 2014. Fine context, low-rank, softplus deep neural networks for mobile speech recognition. pp. 7644–7648. <https://doi.org/10.1109/ICASSP.2014.6855087>
- Shao, W., Kawakami, R., Yoshihashi, R., You, S., Kawase, H., Naemura, T., 2019. Cattle detection and counting in UAV images based on convolutional neural networks. *Int. J. Remote Sens.* 0, 1–22. <https://doi.org/10.1080/01431161.2019.1624858>
- Singh, R., Singh, S., Kaur, N., 2016. A Review: Techniques of Vehicle Detection in Fog. *Indian J. Sci. Technol.* 9. <https://doi.org/10.17485/ijst/2015/v8i1/106793>
- Srihari, S.N., n.d. Parameter Tying and Parameter Sharing. *Deep Learn.* 12.
- Subathra, P., Rajasaranya, K., Suresh, E., n.d. Murk Removal and Object Detection in Foggy Image using CLAHE and CURVELET Transform. *Eng. Technol.* 6, 7.
- Tan, R.T., n.d. Visibility in Bad Weather from a Single Image 8.
- Tarel, J.-P., Hautière, N., 2009. Fast visibility restoration from a single color or gray level image. 2009 IEEE 12th Int. Conf. Comput. Vis. 2201–2208. <https://doi.org/10.1109/iccv.2009.5459251>
- Tierney, S., 2008. Experimental (quantitative) studies : An overview and general issues.
- Tomic, T., Schmid, K., Lutz, P., Dömel, A., Kassecker, M., Mair, E., Grix, I.L., Ruess, F., Suppa, M., Burschka, D., 2012. Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue. *IEEE Robot. Autom. Mag.* 19, 46–56. <https://doi.org/10.1109/mra.2012.2206473>
- Waharte, S., Trigoni, A., 2010. Supporting Search and Rescue Operations with UAVs. 2010 Int. Conf. Emerg. Secur. Technol. 142–147. <https://doi.org/10.1109/est.2010.31>
- What exactly is meant by shared weights in convolutional neural network? - Quora [WWW Document], n.d. URL <https://www.quora.com/What-exactly-is-meant-by-shared-weights-in-convolutional-neural-network> (accessed 9.10.19).

What is convolutional neural network? - Definition from WhatIs.com [WWW Document], n.d. . SearchEnterpriseAI. URL <https://searchenterpriseai.techtarget.com/definition/convolutional-neural-network> (accessed 9.10.19).

What is Scikit-Learn? | Codecademy [WWW Document], n.d. URL <https://www.codecademy.com/articles/scikit-learn> (accessed 9.10.19).

Xiao, C., Gan, J., 2012. Fast image dehazing using guided joint bilateral filter. *Vis. Comput.* 28, 713–721. <https://doi.org/10.1007/s00371-012-0679-y>

Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical Evaluation of Rectified Activations in Convolutional Network. *ArXiv abs/1505.00853*.

Xue, X., Li, Y., Shen, Q., 2018. Unmanned Aerial Vehicle Object Tracking by Correlation Filter with Adaptive Appearance Model. *Sensors* 18, 2751. <https://doi.org/10.3390/s18092751>

Yeh, C.-H., Huang, C.-H., Kang, L.-W., Lin, M.-H., 2018. Single Image Dehazing via Deep Learning-based Image Restoration. pp. 1609–1615. <https://doi.org/10.23919/APSIPA.2018.8659733>

Zhang, Y.-Q., Ding, Y., Xiao, J.-S., Liu, J., Guo, Z., 2012. Visibility enhancement using an image filtering approach. *EURASIP J. Adv. Signal Process.* 2012, 220. <https://doi.org/10.1186/1687-6180-2012-220>

Zhou, Y.T., Chellappa, R., 1988. Computation of optical flow using a neural network. *IEEE 1988 Int. Conf. Neural Netw.* 71–78 vol.2. <https://doi.org/10.1109/icnn.1988.23914>

Zhu, Q., Mai, J., Shao, L., 2015. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* 24, 3522–3533. <https://doi.org/10.1109/TIP.2015.2446191>

List of appendices

Appendix 1: CNN programming code

```
import keras
import os,cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.cross_validation import train_test_split
from keras import backend as K
K.set_image_dim_ordering('tf')

from keras.utils import np_utils
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers import BatchNormalization
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.optimizers import SGD,RMSprop,adam
from keras.utils.vis_utils import plot_model
from keras.preprocessing.image import ImageDataGenerator

img_rows=128
img_cols=128
num_channel=1
num_epoch=20

# defining the data path
data_path = r"C:\Users\Dr Topias\deffoged_scenes"
data_dir_list = os.listdir(data_path)

USE_SKLEARN_PREPROCESSING=True
if USE_SKLEARN_PREPROCESSING:
    from sklearn import preprocessing
    def image_to_feature_vector(image, size=(128, 128)):
        return cv2.resize(image, size).flatten()
    img_data_list=[]
    for dataset in data_dir_list:
        img_list=os.listdir(data_path+'/'+ dataset)
        print ('Loaded the images of dataset-'+ '{}\n'.format(dataset))
        for img in img_list:
            input_img=cv2.imread(data_path + '/' + dataset + '/' + img )
            input_img=cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)

input_img_flatten=image_to_feature_vector(input_img, (128,128))
img_data_list.append(input_img_flatten)

img_data = np.array(img_data_list)
img_data = img_data.astype('float32')
```

```

print (img_data.shape)

img_data_scaled = preprocessing.scale(img_data)
print (img_data_scaled.shape)

print (np.mean(img_data_scaled))
print (np.std(img_data_scaled))

print (img_data_scaled.mean(axis=0))
print (img_data_scaled.std(axis=0))

img_data_scaled=img_data_scaled.reshape(img_data.shape[0],img_row
s,img_cols,num_channel)
print (img_data_scaled.shape)

# prepare the data so that it can be fed into the CNN.
img_data_scaled=img_data_scaled.reshape(img_data.shape[0],img_row
s,img_cols,num_channel)
print (img_data_scaled.shape)

if USE_SKLEARN_PREPROCESSING:
    img_data=img_data_scaled

# Assigning Labels
# Define the number of classes
num_classes = 4
num_of_samples = img_data.shape[0]
labels = np.ones((num_of_samples, ), dtype='int64') #defines new array of
num of samples

labels[0:999]=0
labels[999:1999]=1
labels[1999:2999]=2
labels[2999:]=4

names = ['ball', 'horse',
        'human', 'aeroplane']

# convert class labels to one-hot encoding
Y = np_utils.to_categorical(labels, num_classes)

#Shuffle the dataset to improve learning
x,y = shuffle(img_data,Y, random_state=1)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=2)

### BUILDING THE CNN MODEL
# Defining the model
input_shape=img_data[0].shape

from keras.utils.vis_utils import plot_model
from keras import regularizers
from keras import callbacks

nb_epoch=10

```

```

model = Sequential()

#model for training on keras 2.0
model.add(Conv2D(32, (3,3),input_shape=input_shape))
model.add(Activation('relu'))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))

model.get_config()

model.compile(loss='categorical_crossentropy',
optimizer='rmsprop',metrics=["accuracy"])
hist = model.fit(X_train, y_train, batch_size=32,
nb_epoch=nb_epoch,verbose=1, validation_split=0.2)

#visualizing the accuracy and loss
train_loss=hist.history['loss']
val_loss=hist.history['val_loss']
train_acc=hist.history['acc']
val_acc=hist.history['val_acc']
xc=range(nb_epoch)

plt.figure(1,figsize=(7,5))
plt.plot(xc,train_loss)
plt.plot(xc,val_loss)
plt.xlabel('num of Epochs')
plt.ylabel('loss')
plt.title('train_loss vs val_loss')
plt.grid(True)
plt.legend(['train', 'val'])
plt.style.use(['classic'])

plt.figure(2,figsize=(7,5))
plt.plot(xc,train_acc)
plt.plot(xc,val_acc)
plt.xlabel('num of Epochs')
plt.ylabel('accuracy')
plt.title('train_acc vs val_acc')
plt.grid(True)
plt.legend(['train', 'val'], loc=4)

```

```

plt.style.use(['classic'])

# Evaluating the model
score = model.evaluate(X_test, y_test, verbose=0)
print('Test Loss:', score[0])
print('Test Accuracy:', score[1])

test_image = X_test[0:1]
print (test_image.shape)

# Testing a new image
test_image = cv2.imread("ballx.jpg",cv2.IMREAD_GRAYSCALE)

if test_image is not None:
    print ("image loaded")
else:
    print ('empty')

#test_image=cv2.imread(test_image, cv2.IMREAD_GRAYSCALE)
test_image=cv2.resize(test_image,(128,128))
test_image = np.array(test_image)
test_image = test_image.astype('float32')
test_image /= 255
print (test_image.shape)

    test_image= np.expand_dims(test_image, axis=3)
    test_image= np.expand_dims(test_image, axis=0)
    print (test_image.shape)

    test_image= np.expand_dims(test_image, axis=0)
    print (test_image.shape)

# Predicting the test image
print((model.predict(test_image)))
print(model.predict_classes(test_image))

```

Appendix 2: DCP programming code

```
import cv2;
import math;
import numpy as np;

#The dark channel construction
def DarkChannelConstruct(img,sz):
    b,g,r = cv2.split(img)
    dc = cv2.min(cv2.min(r,g),b);
    mask = cv2.getStructuringElement(cv2.MORPH_RECT,(sz,sz))
    DarkChannel = cv2.erode(dc,mask)
    return DarkChannel

#Finding the estimate of the atmospheric light
def AtmosphereLight(img,DarkChannel):
    [h,w] = img.shape[:2]
    imgsz = h*w
    numpx = int(max(math.floor(imgsz/1000),1))
    darkvec = DarkChannel.reshape(imgsz,1);
    imvec = img.reshape(imgsz,3);
    indices = darkvec.argsort();
    indices = indices[imgsz-numpx::]
    atmosum = np.zeros([1,3])
    for ind in range(1,numpx):
        atmosum = atmosum + imvec[indices[ind]]
    AtmoEst = atmosum / numpx;
    return AtmoEst

def TransmissionEstimate(img,AtmoEst,sz):
    omega = 0.95;
    im3 = np.empty(img.shape,img.dtype);
    for ind in range(0,3):
        im3[:, :, ind] = img[:, :, ind]/AtmoEst[0, ind]
    transmission = 1 - omega*DarkChannel(im3,sz);
    return transmission

def Guidedfilter(img,p,r,eps):
    mean_I = cv2.boxFilter(img,cv2.CV_64F,(r,r));
    mean_p = cv2.boxFilter(p, cv2.CV_64F,(r,r));
    mean_Ip = cv2.boxFilter(img*p,cv2.CV_64F,(r,r));
```

```

cov_Ip = mean_Ip - mean_I*mean_p;
mean_II = cv2.boxFilter(img*img,cv2.CV_64F,(r,r));
var_I = mean_II - mean_I*mean_I;
a = cov_Ip/(var_I + eps);
b = mean_p - a*mean_I;
mean_a = cv2.boxFilter(a,cv2.CV_64F,(r,r));
mean_b = cv2.boxFilter(b,cv2.CV_64F,(r,r));
q = mean_a*im + mean_b;
return q;

def TransmissionRefine(img,et):
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY);
    gray = np.float64(gray)/255;
    r = 60;
    eps = 0.0001;
    t = Guidedfilter(gray,et,r,eps);
    return t;

def RecoverRadiance(img,t,AtmoEst,tx = 0.1):
    res = np.empty(img.shape,img.dtype);
    t = cv2.max(t,tx);
    for ind in range(0,3):
        restore[:, :, ind] = (img[:, :, ind]-A[0,ind])/t + A[0,ind]
    return restore

if __name__ == '__main__':
    import sys
    try:
        fn = sys.argv[1]
    except:
        fn = cv2.imread("15.jpg")

    def nothing(*argv):
        pass
        fn = cv2.imread("03.png")
    if fn is not None:
        print ("image loaded")
    else:
        print ('empty')

    I = np.float64(fn,gray)/255;

    dark = DarkChannel(I,15);
    A = AtmLight(I,dark);
    te = TransmissionEstimate(I,A,15);
    t = TransmissionRefine(src,te);
    J = Recover(I,t,A,0.1);

    cv2.imshow("dark",dark);
    cv2.imshow("t",t);
    cv2.imshow('I',src);
    cv2.imshow('J',J);
    cv2.imwrite("J.png",J*255);

```

```
cv2.waitKey
```