



University of Kerbala
College of Computer Science & Information Technology
Computer Science Department

THERMAL IMAGES SEGMENTATION AND RECOGNITION USING DEEP LEARNING TECHNIQUES

A Thesis

Submitted to the Council of the College of Computer Science & Information
Technology / University of Kerbala in Partial Fulfillment of the Requirements
for the Master Degree in Computer Science

Written by

Haider Ali Muften

Supervised by

Assist. Prof. Dr. Ali Retha Hasoon

٢٠٢٤ A.D.

١٤٤٤ A.H.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ
وَحْيُهُ ۗ وَقُلْ رَبِّ زِدْنِي عِلْمًا)

صدق الله العلي العظيم

(سورة طه، آية ١١٤)

Supervisor Certification

I certify that the thesis entitled (**Thermal Images Segmentation and Recognition Using Deep Learning Techniques**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.

Signature:

Supervisor Name: **Assist. Prof. Dr. Ali Retha Hasoon**

Date: / /2024

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Thermal Images Segmentation and Recognition Using Deep Learning Techniques**" for debate by the examination committee.

Signature:

Assist. Prof. Dr. Muhannad Kamil Abdulhameed

Head of Computer Science Department


Date: / /2024


Certification of the Examination Committee

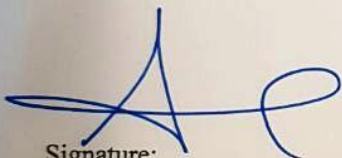
We hereby certify that we have studied the dissertation entitled (**Thermal Images Segmentation and Recognition Using Deep Learning Techniques**) presented by the student (**Haider Ali Muften**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Very good**) standing as a thesis for the Master degree in Computer Science.

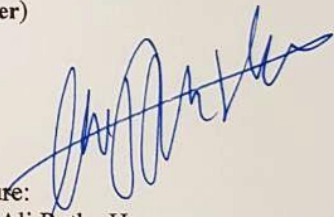
Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Thermal Images Segmentation and Recognition Using Deep Learning Techniques**) presented by the student (**Haider Ali Muften**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Very good**) standing as a thesis for the Master degree in Computer Science.

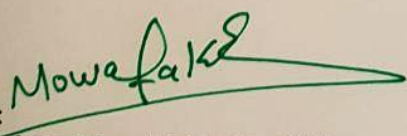
Signature: 
Name: Ashwan Anwer Abdul munem
Title: Assist prof. Dr
Date: / / 2024
(Chairman)

Signature: 
Name: Wathiq Laftah Abd Ali
Title: Assist prof. Dr
Date: / / 2024
(Member)

Signature: 
Name: Ayad Hamed Mousa
Title: Assist prof. Dr
Date: / / 2024
(Member)

Signature: 
Name: Ali Retha Hasoon
Title: Assist prof. Dr
Date: / / 2024
(Member and Supervisor)

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature: 
Assist. Prof. Dr. Mowafak khadom Mohsen
Date: / / 2024
(Dean of College of Computer Science & Information Technology)

Dedication

I devote my efforts first and foremost to the Iraqi martyrs, followed by my mother and all of my brothers, who have stood by me and supported me throughout my academic career and beyond. Lastly, the rest of my family and my friends, notably my brother Mustafa and buddy Muhammad, who have always been there for me.

Acknowledgement

First, I would want to give Allah, the Almighty, praise and thanks for his abundant blessings that enabled me to successfully complete my research.

I would like to sincerely thank Assistant Prof. Dr. Ali Retha Hasoon, my research supervisor, for his hard work, scientific guidance, and insightful counsel. His desire to become an expert in every aspect, no matter how tiny, really enhanced the core of my study and helped me deliver it at this level of science. It was an enormous pleasure and joy to work and study under him. I sincerely appreciate what he has done for me, and I pray that Allah rewards him abundantly.

My profound gratitude and regards go out to the dean of the College of Computer Science and Information Technology, the entire faculty, the head of the computer department, my colleagues and graduate students at the University of Karbala's College of Computer Science and Information Technology.

Abstract

Thermal images play a crucial role in object detection for various applications, including surveillance, security, industrial automation, and vehicle navigation, as they can capture thermal signatures. However, challenges such as low contrast, fluctuating thermal patterns, and the absence of specific thermal datasets require specialized algorithms to accurately identify the object and its location. This research explores the field of object detection in thermal images, which is a vital aspect for applications such as surveillance and autonomous systems.

The images used are from Kaggle's website, FLIR ADAS. We used it in two different ways for classification and segmentation models. This thesis used the Contrast Limited Adaptive Histogram Equalization (CLAHE) approach in image processing to improve contrast. This pre-processing phase attempts to enhance the performance of subsequent object recognition algorithms by addressing contrast related issues in thermal images

The first section of this study is to define a convolutional neural network model that can be used in classification situations. Using 2,906 thermal images, the study evaluates the accuracy, precision, F1 score, and recall of the MobileNetV2 and VGG19 architectures.

The results showed us that MobileNetV2 had an accuracy of 97% and F1 score of 96.8%, while VGG19 had an accuracy of 97% and an F1 score of 96.6%.

This study provides valuable insights into the application of transfer learning methods for thermal image classification and provides a comprehensive overview of the effectiveness of different algorithms in meeting the challenges posed by thermal imaging conditions.

In second section of this study, we take advantage of the YOLOv4 approach, which has been widely recognized for accurate real-time object detection, enhancing detection performance in the challenging region of thermal images. The technique was trained using a selected dataset contain 1000 images, of which 700 were used for training, 150 for validation, and 150 for testing. We used this approach to segment objects in thermal images. The average accuracy (mAP) of 82% of the algorithm shows how effective it is. These results confirm that YOLOv4 is suitable for thermal imaging applications, which are new areas that require accurate object detection under extremely hot conditions.

Declaration Associated with this Thesis

University of Kerbala
Mail

Haider Ali muftin - حيدر علي مافتن <haider.ali@s.uokerbala.edu.iq>

[SETCAC'23] Your paper #1570960973 ('Object segmentation in thermal images using YOLOv8')

2 messages

CoCoNet Conference (coconet.conference@gmail.com)
<coconet.conference@gmail.com@edas.info>
Reply-To: CoCoNet Conference <coconet.conference@gmail.com>
To: Haider Ali Muftin <haider.ali@s.uokerbala.edu.iq>

Sat, Nov 4, 2023 at
6:58 PM

Symposium on Emerging Topics in Computing and Communications (SETCAC'23)
Paper-ID and Title: 1570960973 - Object segmentation in thermal images using YOLOv8
Authors: Haider Ali Muftin
Paper Category: Regular Paper (Oral Presentation)

Dear Mr. Haider Muftin

Congratulations! On behalf of the Technical Program Committee, we are pleased to inform you that your paper "1570960973" - "Object segmentation in thermal images using YOLOv8" has been accepted for oral presentation at the Fifth International Conference on Computing and Network Communications (CoCoNet'23) from December 18 to 20, 2023, in Bangalore, India. All accepted papers will be published by Springer in the Lecture Notes in Electrical Engineering series, and the proceedings will be accessible through the SpringerLink digital library. The proceedings of this series are indexed by ISI Proceedings, SCOPUS, Google Scholar, and SpringerLink. The conference is organized by PES University, Bangalore.

Publication and presentation of this paper at the conference are contingent upon:

1) Your completion and submission of the camera-ready final version of the technical manuscript for the paper. The camera-ready copy must be submitted according to the instructions provided at <https://coconet-conference.org/2023/?q=node/10>. Do not make any changes in the margins, font size, and line spacing of the Springer template.

The deadline for submission of the final manuscript, after incorporating reviewers' comments if any, is November 30, 2023. The registration deadline is November 30, 2023.

All final manuscripts of the paper should be submitted only through the EDAS portal (<https://edas.info/>).

All registered authors will have one more opportunity to revise their papers based on the feedback received during the presentation. The final PDF and corresponding source files should be uploaded to EDAS by a specified deadline. Further instructions will be communicated to authors via a separate email after the conference.

Before uploading your camera-ready paper (PDF) and source files (Latex/MS-Word files of your final PDF) on EDAS, please:

a) Check that title and abstract in EDAS match your camera-ready paper exactly. Please use Title Case for the paper title, i.e. capitalizing all main words.

b) Please compare all author names in EDAS with the author list in your paper. They MUST BE identical and in the same order. This is crucial as we will use the information provided in EDAS to generate the final conference program and the table of contents for the Proceedings.

c) Please ensure that the abstract and title should be readable stand-alone and not contain any references or unexplained acronyms. You are allowed to modify the abstract and author list yourself.

2) Receipt of the signed Springer copyright form through EDAS under your CoCoNet'23 Paper(s). Please upload the scanned copy of the copyright form (PDF) on EDAS (DIGITALLY SIGNED COPYRIGHT IS NOT ACCEPTED). Instructions are posted at <https://coconet-conference.org/2023/?q=node/10>

3) Your pre-registration for the conference. For inclusion of the paper in the conference proceedings, every accepted paper must be accompanied by at least one full registration. Where an author has more than one accepted paper, he/she has to register for each paper separately. The registration details are available at <https://coconet-conference.org/2023/?q=node/8>. The deadline to complete your registration is November 30, 2023.

Please review the attached comments from the reviewers, as they are meant to assist you in enhancing your paper for final publication. Kindly address the listed comments.

[SIRS'23] Your paper #1570964467 ('Comparison of MobileNetV2 and VGG19 for the categorization of thermal images')

1 message

Sirs Symposium (sirs.symposium@gmail.com) <sirs.symposium@gmail.com@edas.info>

Tue, Oct 31, 2023 at
1:31 PM

Reply-To: Sirs Symposium <sirs.symposium@gmail.com>

To: Haider Ali Mufftin <haider.ali@s.uokerbala.edu.iq>, "Ali R. Hasoon" <aliretha@uokerbala.edu.iq>

7th International Symposium on Signal Processing and Intelligent Recognition Systems (SIRS'23)
Paper-ID and Title: 1570964467 - Comparison of MobileNetV2 and VGG19 for the categorization of thermal images
Authors: Haider Ali Mufftin and Ali R. Hasoon
Paper Category: Short Paper(Oral Presentation)

Dear Mr. Haider Mufftin

Congratulations! On behalf of the Technical Program Committee, we are pleased to inform you that your paper "1570964467" - "Comparison of MobileNetV2 and VGG19 for the categorization of thermal images" has been accepted for oral presentation at the International Conference on Applied Soft Computing and Communication Networks (ACN'23) from December 18 to 20, 2023, in Bangalore, India. All accepted papers of ACN'23 main track and Co-affiliated Symposiums (SIRS'23 & SSCC'23) will be published by Springer in Lecture Notes in Networks and Systems(LNNS). LNNS is indexed in Scopus. The proceedings will be available via the SpringerLink digital library. The books of this series (proceedings) are submitted to ISI Proceedings, SCOPUS, Google Scholar and SpringerLink. The conference is organized by PES University, Bangalore.

Publication and presentation of this paper at the conference are contingent upon:

1)Your completion and submission of the camera-ready final version of the technical manuscript for the paper. The camera-ready copy must be submitted according to the instructions provided at <http://acn-conference.org/2023/submission>. Do not make any changes in the margins, font size, and line spacing of the Springer template.

The deadline for submission of the final manuscript, after incorporating reviewers' comments if any, is November 30, 2023. The registration deadline is November 30, 2023.

All final manuscripts of the paper should be submitted only through the EDAS portal (<https://edas.info/>).

All registered authors will have one more opportunity to revise their papers based on the feedback received during the presentation. The final PDF and corresponding source files should be uploaded to EDAS by a specified deadline. Further instructions will be communicated to authors via a separate email after the conference.

Before uploading your camera-ready paper (PDF) and source files (Latex/MS-Word files of your final PDF) on EDAS, please:

- a)Check that title and abstract in EDAS match your camera-ready paper exactly. Please use Title Case for the paper title, i.e. capitalizing all main words.
- b)Please compare all author names in EDAS with the author list in your paper. They MUST BE identical and in the same order. This is crucial as we will use the information provided in EDAS to generate the final conference program and the table of contents for the Proceedings.
- c)Please ensure that the abstract and title should be readable stand-alone and not contain any references or unexplained acronyms. You are allowed to modify the abstract and author list yourself.

2) Receipt of the signed Springer copyright form through EDAS under your ACN'23 Paper(s). Please upload the scanned copy of the copyright form (PDF) on EDAS (DIGITALLY SIGNED COPYRIGHT IS NOT ACCEPTED). Instructions are posted at <http://acn-conference.org/2023/submission>

3) Your pre-registration for the conference. For inclusion of the paper in the conference proceedings, every accepted paper must be accompanied by at least one full registration. Where an author has more than one accepted paper, he/she has to register for each paper separately. The registration details are available at <https://acn-conference.org/2023/reg>. The deadline to complete your registration is November 30, 2023.

Please review the attached comments from the reviewers, as they are meant to assist you in enhancing your paper for

Table of Contents

Dedication	iii
Acknowledgement	iv
Abstract	v
Table of Contents	ix
List of Tables	xii
List of Figures	xiii
List of Abbreviations	xv
CHAPTER ONE	1
1.1 Overview	1
1.2 Problem statement	3
1.3 Research aim and objective	3
1.4 Challenges	3
1.5 Contributions	5
1.6 Thesis Organization	5
CHAPTER TWO	7
2.1 Overview	9
2.2 Machine learning	9
2.3 Supervised Learning	10
2.4 Deep Learning	10
2.5 Convolutional neural networks (CNN)	11
2.5.1 Convolutional Layer	14
2.5.2 Pooling Layer	15
2.5.3 Global Average Pooling Layer (GAP)	16
2.5.4 Fully Connected Layer	17
2.5.5 Activation function	18
2.5.5.1 Rectified Linear Unit (ReLU)	19
2.5.5.2 Sigmoid Function	20
2.5.6 Optimizer Selection	21

2.6	Transfer learning.....	22
2.7	MobileNetV2	23
2.8	Visual Geometry Group (VGG19)	25
2.9	You Only Look Once (YOLO)	25
2.10	Contrast Limited Adaptive Histogram Equalization (CLAHE)	27
2.11	Evaluation Matrices	28
2.11.1	Accuracy	28
2.11.2	Precision.....	28
2.11.3	Recall (Sensitivity)	29
2.11.4	Average Precision (AP) and Mean Average Precision(MAP)	29
2.11.5	F1 measure	30
2.11.6	Confusion Matrix	30
2.12	Related work	31
CHAPTER THREE.....		44
3.1	Overview.....	45
3.2	Models Implementation	45
3.3	Proposed System.....	45
3.3.1	Classification Model	47
3.3.2	Segmentation Model	47
3.4	Dataset.....	48
3.4.1	Dataset for classification	50
3.4.2	Dataset for segmentation	50
3.5	Dataset preprocessing	50
3.5.1	Contrast Limited Adaptive Histogram Equalization (CLAHE).....	51
3.6	Dataset splitting	53
3.7	Dataset Augmentation.....	54
3.8	Normalization	55
3.9	Building the classification Model	55
3.10	Training the segmentation	57
CHAPTER FOUR.....		59
4.1	Overview.....	60

ξ.۲ System Requirement	۶۰
ξ.۲.۱ System Requirement to classification	۶۰
ξ.۲.۲ System Requirement to segmentation	۶۰
ξ.۳ Proposal classification Models	۶۱
ξ.۳.۱ Classification Model based on MobileNetV۲	۶۱
ξ.۳.۲ Classification Model based on VGG-۱۹	۶۵
ξ.۴ Test Results of the Segmentation Model	۶۸
ξ.۵ Summary	۷۲
CHAPTER FIVE	۷۳
۵.۱ Overview	۷۵
۵.۲ Conclusion	۷۵
۵.۳ Future work	۷۶
REFERENCES	۷۷

List of Tables

Table Title	Page
Table ٢.١: Summary of the Related Work	٤٠
Table ٣.١: Divide the image data into three groups. For both segmentation and classification	٥٤
Table ٣.٢: The utilized Data Augmentation hyper-parameter values	٥٥
Table ٤.١: Hyperparameter of the MobileNetV٢ Model in Implementingy.....	٦٣
Table ٤.٢: Specifics of the Outcomes attained During Model Testing.....	٦٠
Table ٤.٣: Classification report for each object in the MobileNetV٢.....	٦٤
Table ٤.٤: Hyperparameter of the VGG١٩ Model in Implementation	٦٦
Table ٤.٥: Specifics of the Outcomes attained While Testing the Model	٦٨
Table ٤.٦: Classification report for each object by using VGG١٩	٦٨
Table ٤.٧: Information on the Outcomes of the YOLO Model Testing	٦٨

List of Figures

Figure Title

Page

Figure 2.1 Traditional Programming Concepts and Machine Learning	9
Figure 2.2: General Deep Learning Architecture	11
Figure 2.3: Building blocks of a CNN	13
Figure 2.4 CNN Convolution Process	15
Figure 2.5 Process for Maximum and Average Pooling.....	16
Figure 2.6 Process for GAP	17
Figure 2.7: Architecture of Fully Connected Layers	18
Figure 2.8: Formats for activation functions	18
Figure 2.9: A ReLU function plot illustration	19
Figure 2.10. Sigmoid Activation Function Response Characteristics	20
Figure 2.11 (a) Standard convolution is divided into two layers: (b) depthwise convolution and (c) pointwise convolution, to create a depthwise separate filter	23
Figure 2.12: Architecture of MobileNetV2	24
Figure 2.13 Architecture of VGG-19	25
Figure 2.14: The classification model's confusion matrix for two classes	29
Figure 2.15-2.22: Sample of the dataset used	32
Figure 2.15-2.22: Sample of the dataset used	33
Figure 2.15-2.22: Sample of the dataset used	34
Figure 2.15-2.22: Sample of the dataset used	35
Figure 2.15-2.22: Sample of the dataset used	36
Figure 2.15-2.22: Sample of the dataset used	37
Figure 2.15-2.22: Sample of the dataset used	38
Figure 3.1: Diagram for the workspace	40
Figure 3.2: Examples taken from the dataset.....	49
Figure 3.3: Sample pictures before and after using CLAHE.....	50
Figure 3.4: : Architecture of The Modified MobilNetv2 model.....	57

Figure 3.4: : Architecture of The Modified VGG-19 model	57
Figure 4.1: Proposed MobileNetV2 Models.....	58
Figure 4.2 The suggested architecture's accuracy and loss function	59
Figure 4.3: The proposed architecture's confusion matrix.....	59
Figure 4.4: Proposed VGG-19 Models.....	61
Figure 4.5 The proposed architecture's accuracy and loss function	62
Figure 4.6: Confusion Matrix for the Suggested Design.....	62
Figure 4.7: Sample of Test Dataset.....	65
Figure 4.8: The proposed architecture's confusion matrix.....	66
Figure 4.9: Average Precision (AP) of Detection Model	67
Figure 4.10: depicts the F1 Confidence of the YOLOv4 algorithm.....	67
Figure 4.11 Model training result	67

List of Abbreviations

Abbreviation	Description
AdaDelta	Adaptive Learning Rate
AdaGrad	Adaptive Gradient Algorithm
AI	Artificial Intelligent
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
FC	Fully Connected
FN	False Negative
FP	False Positive
GAP	Global Average Pooling
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
VGG	Visual Geometry Group
IOU	
AP	Average precision
DDSSD	Dilation and Deconvolution Single Shot Multibox Detector

CHAPTER ONE

INTRODUCTION

1.1 Overview

In recent years, thermal cameras have grown more prevalent in surveillance systems[1]. Due to its ability to locate objects in low-light and nighttime conditions, thermal imaging is a helpful addition to visual imaging [2]. Visible imaging differs slightly from thermal imaging in that the former is created by the object's reflection of light, whereas the latter is created when infrared rays are focused on a specific area by the thermal camera. Thermal images of that specific area are created as color maps of images, which vary from thermal camera to thermal camera with respect to the intensity of various colors[3].

Object detection has been attracting increasing amounts of attention in recent years due to its wide range of applications and recent technological breakthroughs[4]. Object detection and identification are critical components for most computer vision systems, determining the effectiveness of many applications such as tracking, video surveillance, and images captioning. The performance of object detection and recognition is greatly impacted by the quality of the extracted features and the robustness of the classifiers. This is because there are many factors that can affect how an image appears, such as lighting, pose, object reflectance, and intrinsic camera characteristics [5].

In computer vision, object detection is the process of locating an item by scanning an image or video [6]. Among the various applications for object recognition and classification that have garnered attention are pedestrian detection and surveillance. Standard imaging equipment for classic detection procedures consists of cameras that operate in the visible area because to the strong silhouettes and dramatic contrast in the pictures. Visual cameras are

more sensitive to changes in light than thermal cameras, which may operate in complete darkness[¹⁷].

Classification and object localization are the two main elements of object detection. Object localization defines the position and size of an object instance or instances by defining a bounding box around them[¹⁸]. The objective of object detection is to ascertain whether an image includes any objects of the provided classes and to pinpoint their locations within the picture given an image and a collection of object classes, picture classification, which solely considers the existence or absence of these objects. Thus, object identification is a subtask of semantic segmentation, which is a subtask of image classification[¹⁹]. Although the detectors provide trade-offs in terms of their speed, accuracy, and granularity of the outcomes of detection, the decision about the object detection approach is still based on the issue that has to be resolved[²⁰].

In the past decade, artificial intelligence (AI) has a significant influence on every aspect of human existence. Deep learning is one area of AI that uses Artificial Neural Networks(ANNs) for representation learning. In the wider family of deep learning architectures, CNNs which are composed of a collection of neural network layers, are used for image processing and computer vision[²¹]. A set of hidden layers, an input layer, and an output layer make up the structure. After receiving an image as input, CNN analyzes and categorizes it. The CNN object detection application was applied in years when an arbitrary number of hidden layers were employed for face detection [²²].

١.٢ Problem statement

The central research problem is to detect and classify objects within thermal images. Thermal imaging presents unique challenges, including limited texture information, low spatial resolution, and variations in temperature intensity. Consequently, current image analysis techniques, primarily designed for visible light images, often struggle to provide reliable results in thermal imaging applications.

١.٣ Research aim and objective

The main aim to develop deep learning based model. For object detection and recognition on thermal images to achieve this aim the following objectives should be achieved.

- a. To classify the selected dataset into four main object (person, car, None, Mixed (person and car)).
- b. To segment the select data person and car.

١.٤ Challenges

The study explores the future possibilities and obstacles in object detection in thermal imaging, highlighting the unique challenges compared to visual pictures:

- ١- Low resolution: Thermal cameras frequently struggle with object detection, especially when objects are small and far away. The process can also be made more difficult by hazy or unclear boundaries in low-quality photos.
- ٢- Limited color information: Thermal images use heat signatures in place of color information, and since color information is missing, it may be

challenging to distinguish between similar items or minute variations within an object class.

- ƴ- Variability in thermal signatures: Various materials have variable thermal properties, which lead to diverse thermal signatures for the same thing. This makes developing algorithms that can reliably identify and classify objects across a broad temperature range difficult.
- ξ- Background noise: It can become more challenging to correctly identify items against their backdrop when using object recognition algorithms since thermal photos show heat signatures in their backgrounds.
- ο- Environment and lighting effects: Thermal cameras can be affected by environmental factors such as changes in the surrounding temperature, humidity, and air quality. Accurately identifying and classifying objects can be challenging due to noise and distortions caused by these elements in thermal images.
- ϒ- Data Scarcity: Lack of big, well-annotated thermal imaging datasets is a major obstacle. To build strong machine learning models for thermal imaging object detection and segmentation, access to such datasets is essential [13],[14].
- ϒ- Cross-Modality Adaptation: Since thermal images record temperature changes rather than color, effective methods for using existing algorithms designed for visible light images to the thermal domain are required [14].
- ∧- Object Detection and Localization: Accurate object localization and detection in thermal photos pose considerable challenges due to the fluctuations in the appearance of objects and the presence of thermal noise [15],[16].

9- Semantic Segmentation: Accurate semantic segmentation in thermal pictures is a difficult [16],[14].

1.5 Contributions

This thesis focused on improving object detection (classification and segmentation) using two separate models to address these problems. These barriers include:

- Classification of objects using MobileNetV2 and VGG19
- Segmentation of objects using Yolov4

1.6 Thesis Organization

Chapter 2 provides a literature review and an overview of the entire argument. The theoretical underpinnings of the working model, which explain the organization of the models used in this thesis, an explanation of stacked layers and pooling approaches is provided, and the system is assessed using the metrics. Chapter Three provides specifics on the proposed technique for automatically classifying and segmenting thermal pictures to identify people and cars. Chapter Four reports the experimental results of the human and automobile segmentation and classification models. The thesis conclusion and future initiatives will be discussed in chapter five.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Overview

The object detection system is covered in this chapter. It also explains the concept of the techniques used in this system, which are the pre-trained convolutional neural network model is (MobileNetV2 and VGG19) and (yolov4), in addition to problems with network training. Lastly, the proposed model methodology, software testing, and metrics will be discussed.

2.2 Machine learning

A branch of artificial intelligence known as "machine learning" is used to simulate or replace human behavior in order to solve issues or implement automation [13]. When handling and forecasting huge data, machine learning uses learning algorithms to describe the data [14]. Machine learning is distinct from traditional programming, where the computer requires input in the form of data and instructions in order to create an output. In contrast, for machine learning, computers require inputs and outputs in order to create a program [15].

The process of learning, or training, is what sets machine learning apart from other forms of artificial intelligence. In machine learning algorithms, three common forms of data that are used are training, validation and test data. When use testing data, algorithms that have already been trained are tested against the newly discovered training data to assess how well they perform [16].

Machine learning uses the classification technique to categorize items according to specific traits. Based on the data that was learned, prediction or regression is used to forecast outputs from input data [17]. Machine learning may be applied in a variety of situations, including supervised learning, unsupervised

learning. When training data includes outputs that have been recognized is tagged, the learning is supervised [٢١].

٢.٣ Supervised Learning

The input and output characteristics are predetermined and labeled in supervised machine learning [٢٢]. This strategy is simpler to utilize in learning processes since it handles categorize data in this manner. This technology's ability to produce data outputs based on current data is most advantageous feature. When using this strategy, there is a chance that the classification method will be overly rigorous if the training collection does not contain instances that are typical of the target class [٢٣].

٢.٤ Deep Learning

Deep learning is a branch of machine learning that develops algorithms with inspiration from the architecture of the human brain[٢٤]. Alternatively said, deep learning is a novel approach to machine learning where a model learns to do categorization tasks directly from data (pictures, text, or voice). The term "deep" refers to the amount of layers in the neural network, and a neural network design is typically utilized to execute deep learning. While deep networks, also known as convolutional neural networks (CNN), might contain hundreds of layers, standard neural networks typically only have two or three [٢٥].

Deep learning can offer accurate results by automatically extracting features [٢٦]. Due to the fact that they may run on specialized computer hardware, deep learning algorithms are able to handle vast volumes data and is capable of continuously enhanced through adding new data. Applications like product categorization, software for translation, and natural language processing all require

a substantial amount of data [24]. Input layer, hidden layer, and output layer are the three layers that make up deep learning, as shown in Figure 2.2. Nodes in the input layer hold the input value throughout training and are only able to alter when a new input value is supplied. Hidden layers are used for all training and recognition tasks, and the number of layers relies on the architecture used to discover the best algorithmic mix and reduce output mistakes. In the meanwhile, a hidden layer activation function in the output layer shows the outcomes of system computations depending on input received [25].

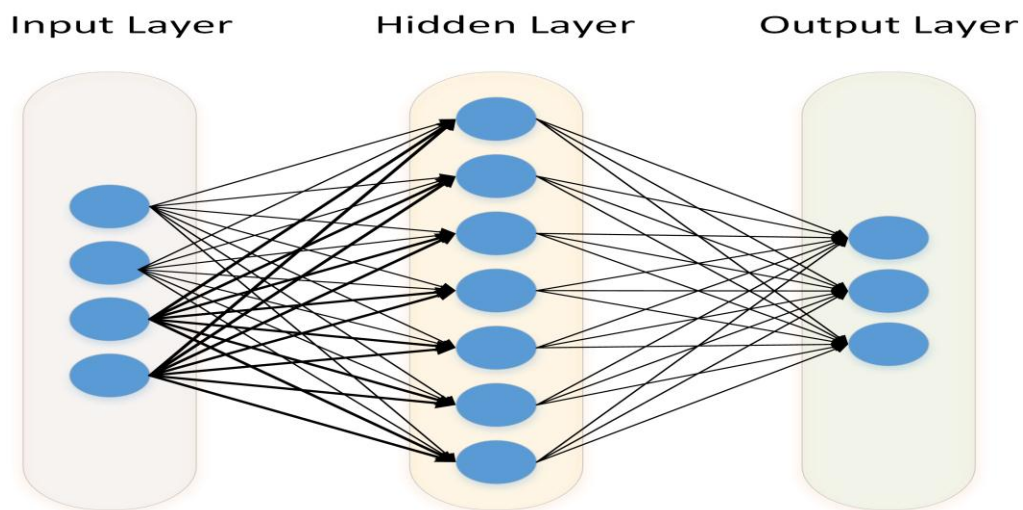


Figure 2.2: General Deep Learning Architecture [26].

2.2 Convolutional neural networks (CNN)

Convolutional neural networks is the term used to describe networks that employ the convolution mathematical methods [27]. CNN is one of the most popular deep learning subcategories, is particularly useful for high-dimensional data, such photos and movies [28],[29]. CNNs are neural networks that, used processing input, give the impression of having a grid-like structure. It includes, for instance, data in time series, which depicts a one-dimensional network that is

periodically sampled, and digital image data, which displays a pixels-in-two dimensions network [30]. As with the formation of the human brain's neural communication pattern, the CNN's architecture is influenced by how the visual cerebral cortex is organized [33]. Both supervised and unsupervised learning methods employ the CNN [34]. The channels, width, height, and number of filters are the dimensions in which neurons are put in the layers of CNNs in the most fundamental 2D case. A convolution neural network is composed of a number of layers, each of which employs a differentiable function to modify the activations or outputs of the layer below [35].

In order to overcome image processing challenges when the computer recognizes the object in the image, a CNN is utilized. CNN modeling may be applied to identification, recognition, classification, and image processing. Data is transmitted by the software to patterns that replicate how the nervous system in humans works. One kind of deep learning architecture is the CNN, which are extensively used in many real-world applications, including image classification and pattern recognition [36].

As shown in Figure 4.3, the supervised learning model maps characteristics, such as picture categorization, between inputs and outputs that are known to the model.

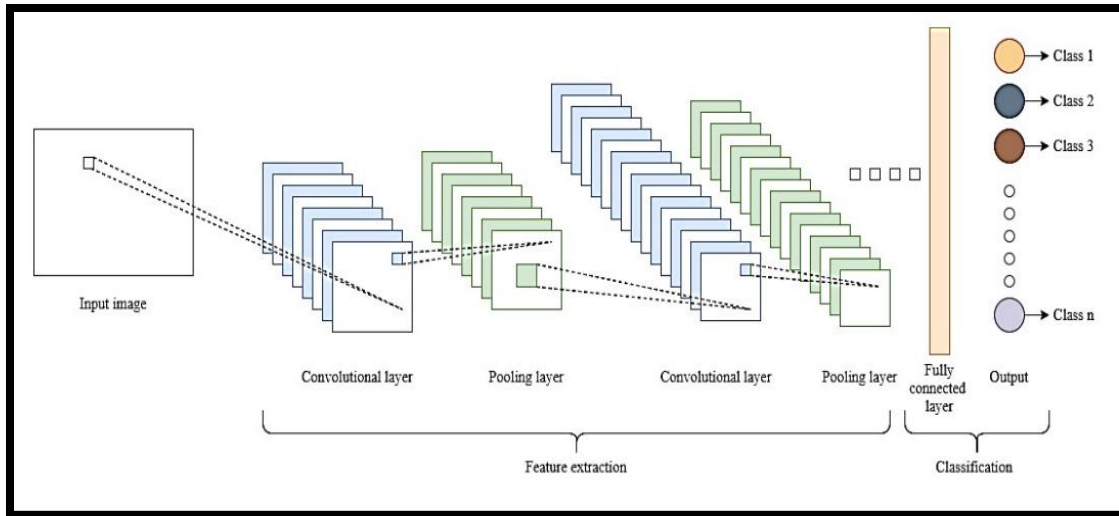


Figure 1.1: Building blocks of a CNN [17]

When a computer has trouble identifying an object in a picture, it uses a CNN to address the problem. Image processing, categorization, recognition, and identification might all be done using CNN modeling. The program sends data in patterns that are similar to how the nervous system functions in humans. Pattern recognition and picture classification are two examples of practical applications where CNNs, a kind of deep learning architecture, have been frequently used [18]. It is a neural network that, for at least one layer, employs a particular "convolutional" layer as opposed to a fully connected layer [19]. A special kind of linear process is convolution. Convolutional networks are basic neural networks with at least one layer that employ convolution instead of conventional array multiplication. With convolutional networks, inputs may be processed at various spatial scales. Traditional neural networks based on matrix multiplication are unable to represent these forms of input [20],[21]. Convolutional layers are regarded as the fundamental components that convolutional neural networks possess. These layers are made up of a number of filters, whose settings must be

learnt. Depending on the input picture, each filter's (kernel's) weights are either ν -dimensional or ν -dimensional $[\xi^1], [\xi^2]$. The filter's dimension is less than the input dimension. In order to calculate every filter is applied to the image from left-top to right-bottom by taking the dot products between the kernel and the input image at each spatial position. Each kernel is calculated formally as $[\nu^2]$,

$$y = \sum_{n=1}^N I_n \cdot K_n \quad (\nu.1)$$

And this means, y represents the output, $\sum_{n=1}^N$ denotes the summation over N terms, I_n is the input image at the n -th spatial location, K_n is the filter (kernel) at the n -th spatial location.

$\nu.5.1$ Convolutional Layer

The first layer, referred to as the convolutional layer, extracts information from the input image by applying convolution operations to the output of the preceding layer $[\xi^3]$. The most crucial primary layers in CNN are convolutional layers. Convolution is the process of repeatedly applying one function to the results of another. When applying a filter to an $\wedge^* \wedge$ picture, for example, the $\nu^* \nu$ filter, the filter first calculates at the current pixel position, then shifts to the right and calculates once again until all pixel locations are computed $[\xi^4]$. As a result of the convolution, the input data are transformed linearly in accordance with the spatial information included in the data. The convolution kernel utilized is specified by the layer's weights so that the CNN may train it using input $[\xi^5]$. Figure $\nu.5$ depicts the convolution process.

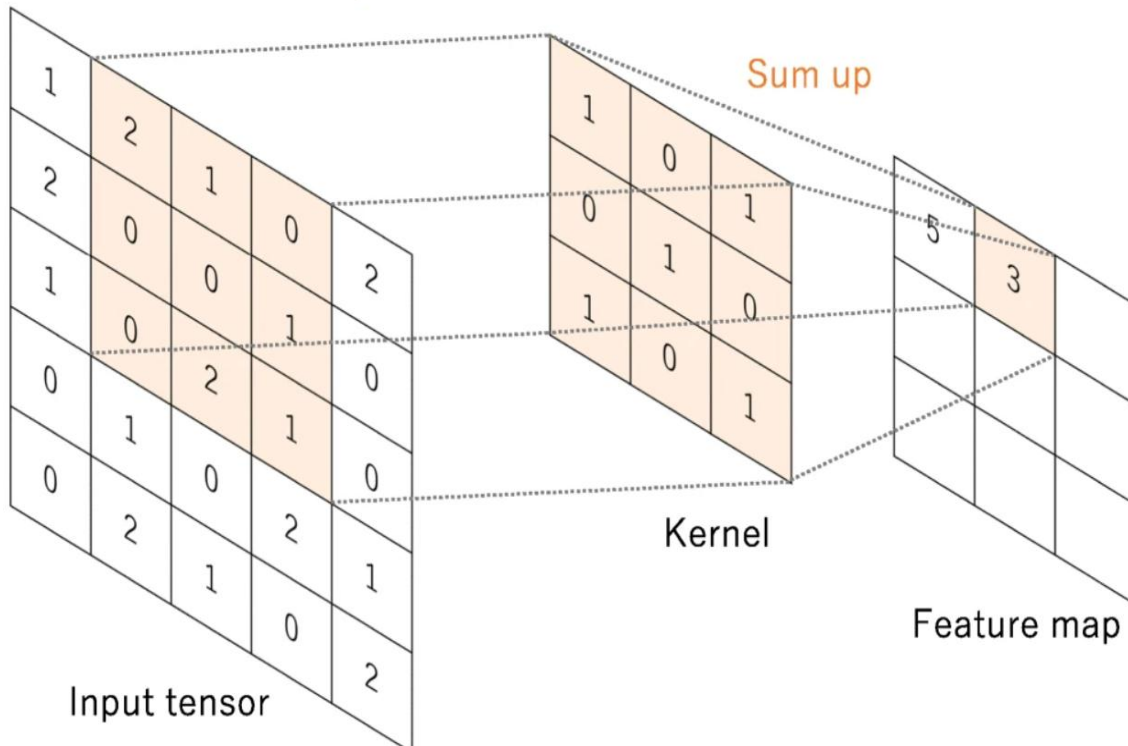


Figure 2.4 CNN Convolution Process [47]

2.5.2 Pooling Layer

The convolutional layer's excessively vast picture data will be reduced in size by the use of pooling layers, making the data smaller, more manageable, and easier to maintain [43]. The input will be divided into many grids by this layer. There are several types of pooling layers, including both average and maximum pooling. The most popular strategy is max pooling. Average pooling takes the average value of the individual pixel sections in the image, whereas max pooling takes the maximum value from each grid [44]. Figure 2.5 illustrates the distinction between the maximum pooling and average pooling methods.

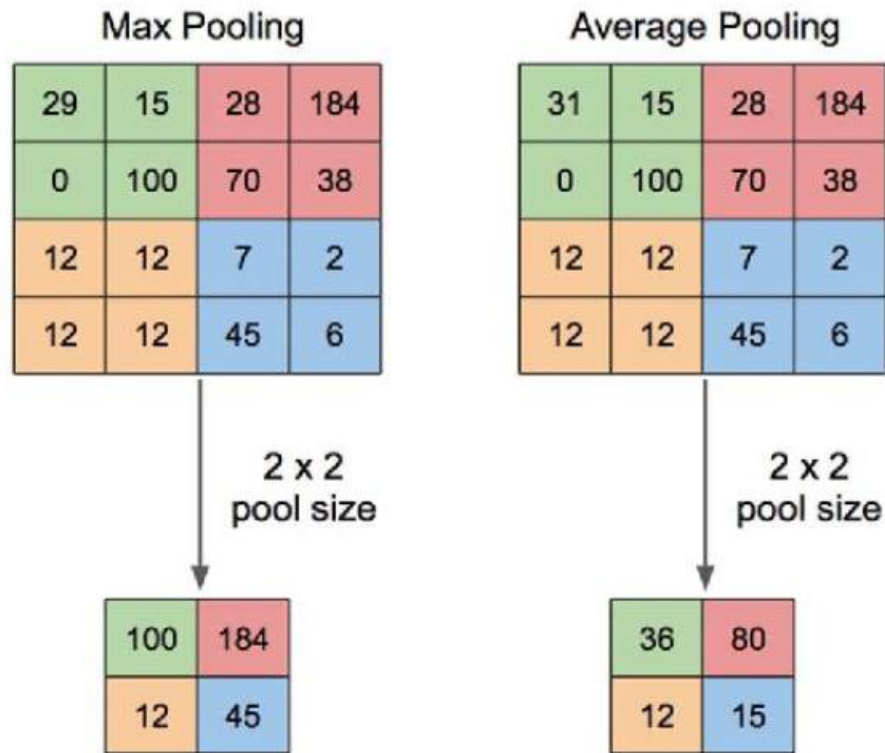


Figure 2.5 Process for Maximum and Average Pooling [14].

2.5.3 Global Average Pooling Layer (GAP)

Global average pooling is average pooling that can convert a tensor of size $w \times w \times d$ into the form $1 \times 1 \times t$ where the average value is taken in each pool [14]. After the final convolutional layer, the GAP layer is applied. The feature map's average values are calculated using the global average pooling technique. To lessen the storage needed by the matrix and the substantial weight of the fully linked layer, GAP can replace it [10]. Figure 2.6 illustrate the Global Average pooling methods.

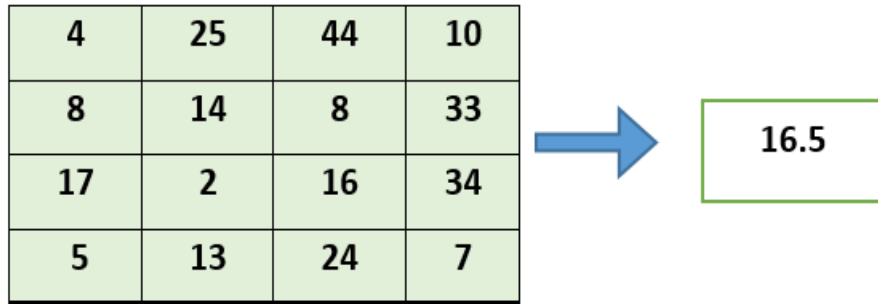


Figure 2.7 Process for GAP.

2.5.4 Fully Connected Layer

In classification issues, a few (often one or two) fully-connected layers are put on top of a CNN and utilized as the final layers. This is accomplished by flattening the CNN output and seeing it as a single vector [30]. Usually, the CNN has a number of entirely connected layers. Dense layers are created by adding more neural network layers and coupling each input and output with a learnable weight [31]. Features extracted down sampled by pooling layers and convolution layers are transported regarding the network's ultimate outputs, using a portion of the fully connected layers to calculate probabilities for every category in classification tasks [32]. In the fully-connected layer, neurons are grouped similarly to how they are in a conventional neural network. As a result, as illustrated in Figure 2.7, every node in a completely linked layer is directly connected to every node in both the preceding and subsequent layers [33].

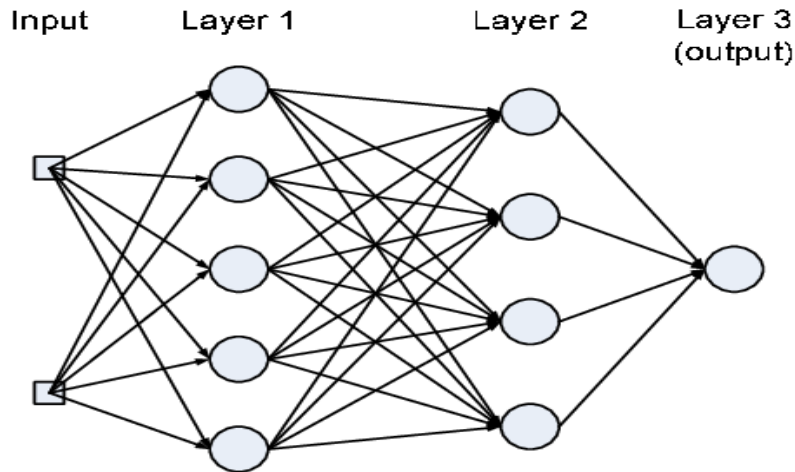


Figure 1.7: Architecture of Fully Connected Layers [14]

1.5.5 Activation function

In artificial neural networks, activation functions are utilized specifically to convert input signals into output signals which are sent as input to the subsequent layer in the stack. In an artificial neural network, we add up the inputs and their corresponding weights, then apply an activation function to generate the layer's output, which will be used as the input for the subsequent layer [15]. ReLU and SIGMOID were the two methods we employed in this investigation.

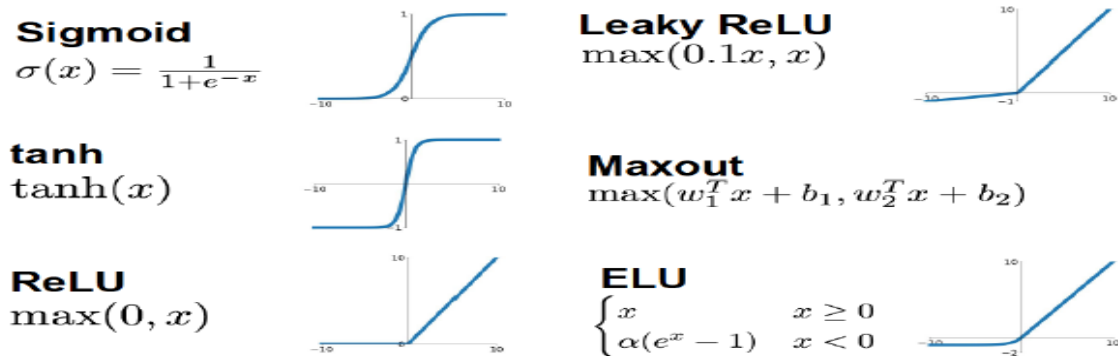


Figure 1.8: Formats for activation functions [16].

2.2.2.1 Rectified Linear Unit (ReLU)

One of the mechanisms for activation employed in this work is the rectified linear unit (ReLU), which is one of the most well-known AF in deep learning. A rapid activation function called ReLU seeks to provide cutting-edge performance and superior outcomes. With reference to the efficiency and deep learning's flexibility [10]. Gradient-descent optimization algorithms are made significantly more straightforward to apply given that the function is almost linear, and preserves the features of linear models [11]. The ReLU function has the benefit of preventing simultaneous activation of all neurons [12]. The most significant advantage of using the ReLU function in the calculation is that it increases computation speed by eliminating the need of compute divisions and exponents, which eventually results in a speedier total computation [13],[14]. Because the negative portion is always zero, the ReLU function also maintains that the output mean value of the function is higher than zero [15]. It can be formally defined as:

$$f(x) = \max(0, x) \quad (2.2)$$

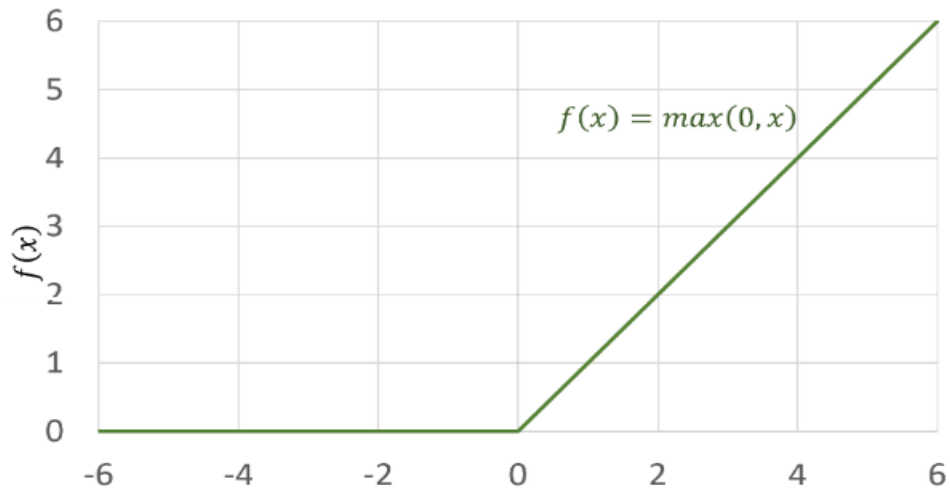


Figure 2.2: A ReLU function plot illustration [16].

2.2.2 Sigmoid Function

The logistic function or squashing function are other names for the sigmoid activation function that have been used in certain publications [12]. Additionally, the sigmoid function is not symmetric around zero, which implies that all of the output values from the neurons will have the same signs. By scaling the sigmoid function, this problem can be resolved [10]. The input values are mapped by this activation function into the range of 0 to 1. Equation 2.3 displays the equation for the activation function, while Figure 2.9 displays the typical response [11].

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.3)$$

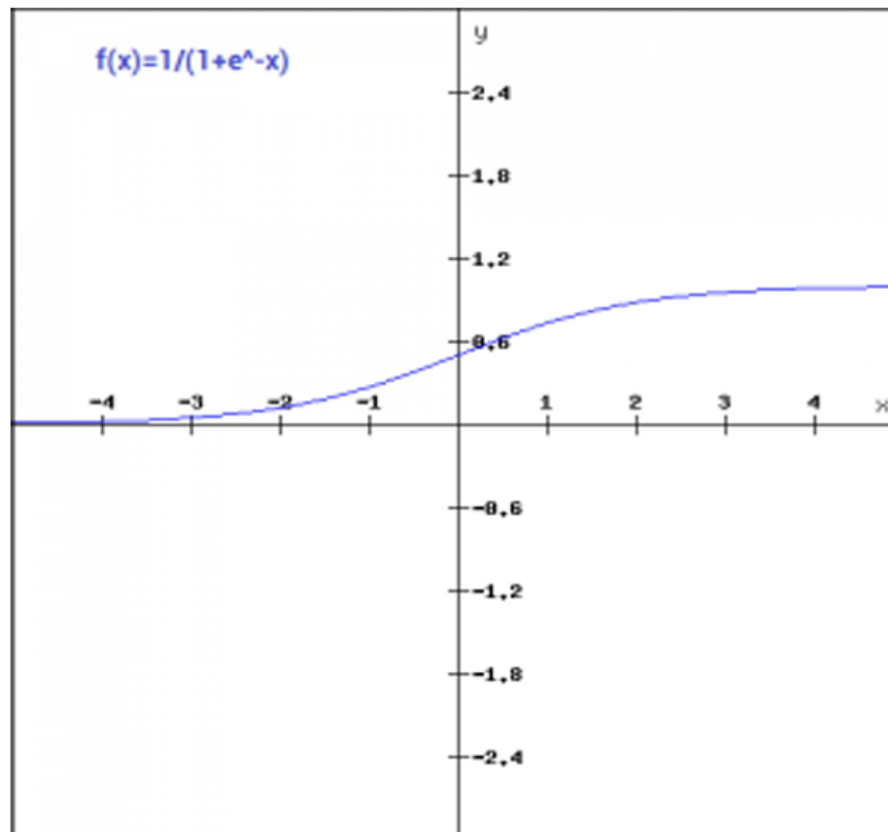


Figure 2.9. Sigmoid Activation Function Response Characteristics [11]

2.5.6 Optimizer Selection

CNN optimizer components keep track of the convergence process and improve classification precision. Nowadays, researchers use RMSProp, Adaptive Moment Estimation (Adam), and Stochastic Gradient Descent (SGD) as optimizers [14]. By frequently modifying learnable network parameters, such as weights and kernels, gradient descent is a well-known optimization technique that aims to reduce loss (the discrepancy between the predicted and actual output) [10]. Using a hyperparameter known as the learning rate, the learnable parameters are impacted by the gradient's backpropagation in the opposite manner. An illustration of a single parameter modification is as follows:

$$W = W - \alpha * \partial L / \partial W \quad (2.4)$$

Where W stands for "learnable parameters," L for "loss function," and α for "learning rate." It's vital to note that one of the most crucial hyperparameters that must be chosen before starting training is the learning rate [16].

In reality, memory constraints are imposed on the parameter changes and the gradients of the parameterized loss function are produced using the mini-batch, an portion of the dataset used for training [17]. One of the hyper parameters is referred to as the Mini-Batch Gradient Descent (MBGD) process, also known as the SGD. The parameters of the model are continuously updated during each training cycle, and throughout each training epoch, the model repeatedly searches for the optimal local solution. The most well-known learning algorithms or optimization techniques include SGD, Adam, RMSProp, AdaGrad, AdaDelta, Momentum, Batch Gradient Descent, and MBGD. RMSProp, Adam, and SGD have been widely employed with great momentum. [18].

2.6 Transfer learning

Learning transfer [19] is a kind of machine learning that picks up new skills by utilizing the information from an earlier model that was employed for different purposes. Using a base, a model that has already been trained using a substantial dataset, this approach leverages the model to perform additional activities that require the usage among several datasets. Because this technique permits precise classification must be performed out with a little dataset, transfer learning is commonly employed. This is due to the fact that the trained deep learning model entirely starting from begin with a tiny dataset finds it very challenging in order to attain great precision because it does not get enough information about the variances in the data. It cannot thus learn the important data points. The process by which a pre-trained model gathers data from a large dataset and uses it to represent it as weights in a neural network is called transfer learning. Next, a second network with a different dataset and job is given these weights. Consequently, rather of training the second network from scratch, we "transfer" the first network's learned features to the second network, frequently with a minimal dataset [20],[21].

When it comes to computer vision, the initial layers of a deep convolutional network are typically used to train the network to recognize common characteristics in pictures, encompassing forms and lines. When the network reaches its final layers for example, picture classification it learns about the exact attributes required for a certain task. The model's or base network's weights are thus often not that modified (frozen) during transfer learning. The network's last layer, which is often a completely linked layer, comes next subjected to learning so that it can acquire the precise properties needed to categorize the new dataset [20].

2.2 MobileNetV2

The model, originally known as MobileNet, was developed into MobileNetV2 [12]. The second version of MobileNet, was made available in 2018 [13]. A key distinction between the MobileNet design and CNN architectures generally is the use of a convolution layer whose filter thickness corresponds to the input image's thickness, allowing the MobileNet architecture to conserve the size of the generated model [14]. Figure 2.11 illustrates how MobileNet splits the convolution layer into depthwise convolution and pointwise convolution using the Depthwise Separable convolution (DSP) approach [15]. This layer's goal is to decrease computation (parameters) so that the model is smaller [16]. The first layer to carry out light filtering is the deep convolution layer, which applies one convolutional filter to each input channel. Concurrently, the second layer, known as pointwise convolution or 1×1 convolution, computes linear combinations of the deep convolution's output to create additional features [14].

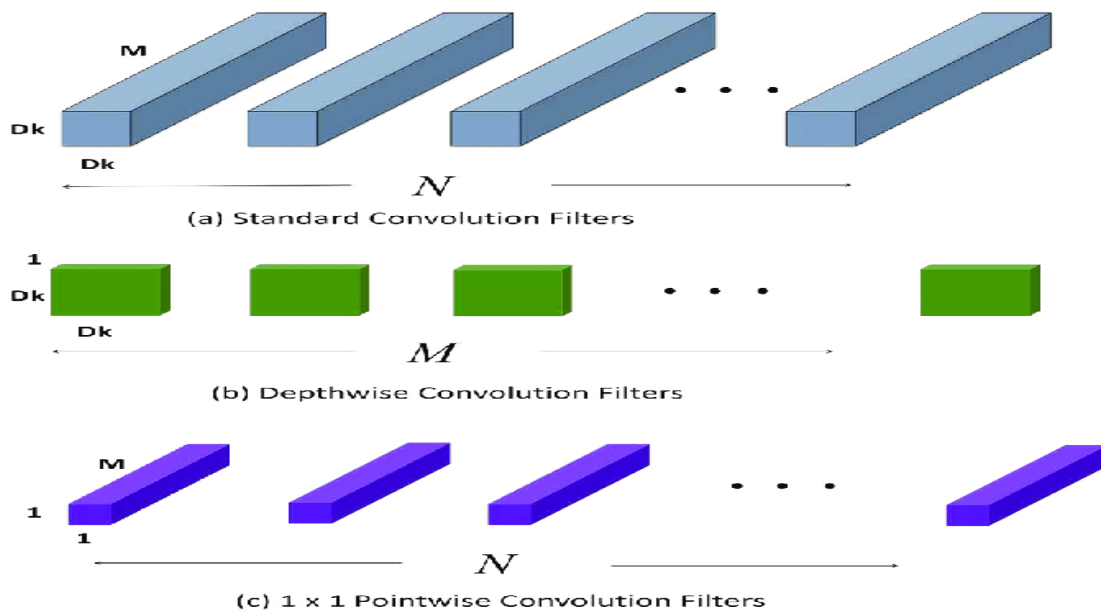


Figure 2.11 (a) Standard convolution is divided into two layers: (b) depthwise convolution and (c) pointwise convolution, to create a depthwise separate filter [17]

Depthwise Separable Convolution serves as the primary layer in MobileNetV2 as well. Linear bottlenecks and connection shortcuts between bottlenecks are two new features that set MobileNetV2 apart from the previous version [19]. Inputs and outputs are located between the model's bottlenecks, and the inner layers include the mechanism that converts inputs from lower-level ideas (pixels) to higher-level descriptors (picture categories). While bypass links allow for quicker training and increased accuracy, linear bottlenecks guard against information tampering [19]. The architectural features, which are based on the MobileNetV2 building blocks, include 3x3 filters in complete convolutional layers, 1x1 bottleneck layers left, and ReLU as a non-linear variable because to its cheap processing overhead. A typical 3x3 kernel is utilized; in addition, batch normalization is applied during the training phase [19]. Figure 2.11 demonstrates the fundamental design of MobileNetV2.

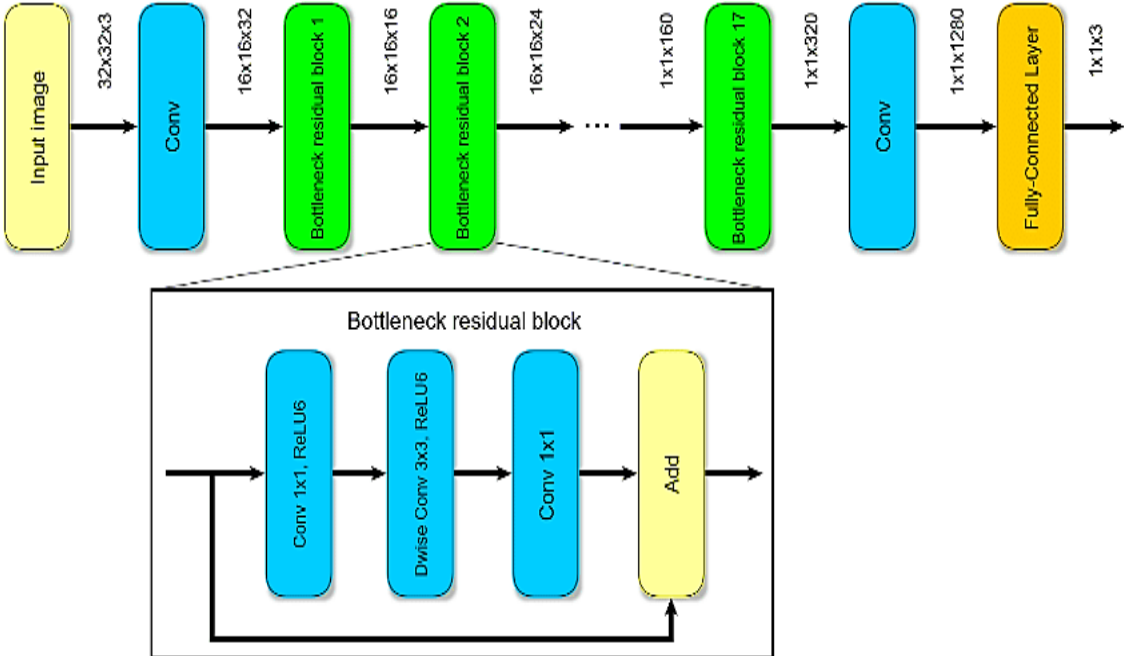


Figure 2.11: Architecture of MobileNetV2 [19].

2.8 Visual Geometry Group (VGG-19)

Applying multilayer operations to picture data sets, VGG or VGGNet (visual geometry group network) is a deep network learning technique based on the CNN model [10]. Because 3×3 convolution layers are used in the top convolution layer to improve the network depth, VGG-19 is a basic model. To minimize noise, VGG-19 employs a max-pooling layer [10]. VGG-19 comprises three completely linked layers and sixteen convolution layers. The VGG architecture is composed of layers that are arranged sequentially. A max pooling layer and a 3×3 convolution layer will be applied repeatedly to the input picture. Afterward, the picture will be categorized under the Fully Connected Layer.

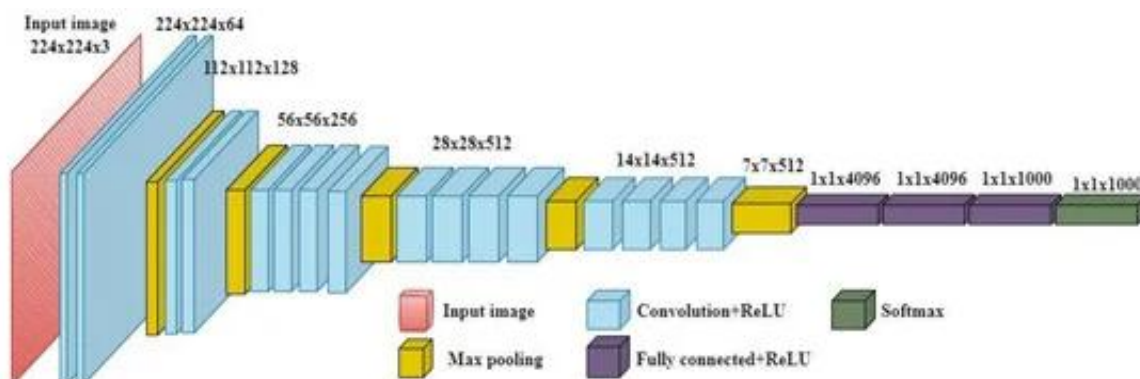


Figure 2.17: Architecture of VGG-19 [11]

2.9 You Only Look Once (YOLO)

Often used multi-object detection technique YOLO, which stands for "You Only Look Once," is described [12]. By evaluating the image just once, YOLO seeks to offer detection findings, as its name indicates. Many multi-object detection algorithms used several steps before YOLO was introduced in order to reliably identify the position and class of the item [13],[14]. Nevertheless, the requirement for several phases made these techniques unsuitable for real-time

applications. YOLO transformed object detection by employing a single neural network to concurrently identify an item's position and class. YOLO has been acknowledged from its beginnings for its quick speed of inference and great accuracy in comparison to other object identification algorithms. From YOLOv1 through YOLOv8, the most recent cutting-edge version, it has undergone evolution [10],[11].

YOLO's capabilities have grown beyond its initial goal of identifying the 80 items that MS-COCO described. It is extensively used in many detection disciplines, such as remote sensing, as a foundation or benchmarking model. And the discovery of minute flaws [12]. It has significantly influenced study in a variety of fields. We thus propose a network in this study that is according to YOLOv8, the most recent cutting edge one-stage multi-object identification algorithm. This builds the strong foundation of Yolo's successes in a variety of detection tasks. The first object detection network to integrate the tasks of class label identification and bounding box drawing into a single end-to-end differentiable network was the YOLO model. Furthermore, different variations of the algorithm have been produced, including YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, and YOLOv7 [13].

In January 2023, YOLOv8 [14] was formally launched. The YOLO collection of algorithms has gained interest in computer vision. Because it maintains a high level of accuracy while maintaining a small model size, YOLO is quite popular. YOLOv8, the most advanced and newest YOLO technique, can be used for segmentation, object recognition, and image classification, among other applications. Yolo v8 was created by Ultralytics, the same company that created the important YOLOv5 model that helped to define the market. There are various architectural upgrades and improvements in YOLOv8 over YOLOv5 [15].

The YOLOv[^] model does not use anchors. It suggests that it directly guesses the object's center rather than estimating the item's distance from a known anchor box. By reducing the quantity of box predictions, anchor-free detection speeds up Non-Maximum Suppression (NMS), a difficult post-processing step that filters through possible detections after inference. YOLOv[^]_n, YOLOv[^]_s, YOLOv[^]_m, YOLOv[^]_l, and YOLOv[^]_x are the five models for identification, segmentation, and classification, respectively. YOLOv[^] Nano is the tiniest and fastest of them all, while YOLOv[^]_x is the most accurate but slowest of them all [90].

2.10 Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE, which stands for Contrast-Limited Adaptive Histogram Equalization, addresses the limitations inherent in traditional histogram equalization. Serving as an enhanced variant of Adaptive Histogram Equalization (AHE), CLAHE offers a solution to challenges such as preserving clear field edges, minimizing noise, and maintaining high spatial frequency information. It achieves this through a combination of adaptive contrast-limited holographic equalization methods, incorporating CLAHE median filtration and edge sharpening techniques [91].

The fundamental improvement of CLAHE over AHE lies in its approach to image processing. The algorithm divides an image into tiles, treating them as contextual regions. Within each contextual region, CLAHE constructs a histogram and performs clipping at a predetermined value. The clipped portion is then redistributed among the histogram bins, resulting in a modified histogram compared to the original. This approach effectively mitigates the edge-shadowing effect associated with AHE and addresses the problem of over-enhancement [92].

2.11 Evaluation Matrices

Numerous categorization metrics are utilized to perform a robust and effective analysis, evaluation, and comparison of the results, as well as to determine how well the model executes the goal of the test dataset that is new to the model of the network. Measures of categorization accuracy, for instance, are employed. Loss is connected to both sensitivity and accuracy [93].

2.11.1 Accuracy

It is often characterized as a measuring unit that forecasts a machine learning model's accuracy. Accuracy is defined as the proportion of properly categorized testing data, and it is computed as [94]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (2.5)$$

With a dataset that is unequal in class, accuracy is useless. In order to prevent this issue, the accuracy is calculated for each class separately before determining the mean accuracy.

2.11.2 Precision

The precision measures how many predicted values turned out to be true. Afterward, equation (2.6) was used to compute it.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.6)$$

Additionally, when the main score is high and the majority of ground truth boxes are accurately recognized, the recall score falls within ranges (0, 1).

2.11.3 Recall (Sensitivity)

It is also known as sensitivity and is another crucial parameter utilized in object detection. This ratio, which is the primary true positive of all the positives in the ground truth, may be computed using [90]:

$$Recall = \frac{TP}{TP+FN} \quad (2.7)$$

Also, the recall score falls between ranges (0, 1), when high main score main most ground truth boxes are correctly detected [72].

2.11.4 Average Precision (AP) and Mean Average Precision(MAP)

A measure of the model's performance is its Average Precision (AP). Object detection is one of its frequent uses. Additionally, the average precision is the result of combining recall and precision [96]. Average precision (AP) is the area under the precision–recall curve and is calculated as shown in Equation (2.8).

$$AP = \sum_{k=1}^{n-1} [Recall(k) - Recall(k + 1)] \times Precision(k) \quad (2.8)$$

MAP^{0.5}: The model's accuracy in predicting item positions is evaluated using the Average Precision (AP) at an Intersection over Union (IOU) criterion of 0.5, which is set at 0.5 when the predicted bounding boxes match the ground truth boxes.

$$IoU = \frac{\text{Object} \cup \text{Detected box}}{\text{Object} \cap \text{Detected box}} \quad (2.9)$$

2.11.5 F1 measure

The F1-score is a metric used to evaluate a model's accuracy on a dataset. It combines recall and Precision, resulting in a weighted average. A low ratio of true negatives to false positives results in a better outcome, closer to 1. A high F1-score indicates good object detection and minimal false alarm impact, with a range of 0 to 1. The measure is computed in this manner:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (2.10)$$

2.11.6 Confusion Matrix

A crucial metric for classification that is used to encapsulate the effectiveness the model's confusion matrix. When there are disparities in the amount of samples in each class and when the dataset contains many classes, the classification accuracy of any model may be deceptive [94]. Furthermore, the confusion matrix computation provides a more accurate understanding of the classification model's performance [34]. For every class, the actual and anticipated values are shown in Figure 2.13. Within the framework of this study, the following definitions apply to the entries in the confusion matrix:

- TP is the quantity of accurate predicate indicating a good instance.
- FP is the quantity of inaccurate predicate that a positive case has.
- FN is the quantity of false predicate that a given instance is negative.
- TN is the quantity of accurate predicate indicating a bad incident.

		Predicted class		
		Positive	Negative	
Actual class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Recall $\frac{TP}{TP + FN}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{TN + FP}$
		Precision $\frac{TP}{TP + FP}$	Negative predictive value $\frac{TN}{TN + FN}$	Accuracy $\frac{TP + TN}{TP + TN + FN + FP}$

Figure 2.14: The classification model's matrix of confusion for two classes [94].

2.12 Related work

Previous studies have utilized various techniques to study thermal images, aiming to identify crucial components for object detection. This study will explore conventional and traditional methods for recognizing and classifying thermal pictures.

Rodin, C.D., et al [99], the study used experimental data to classify and identify items at the water's surface for research on the use of unmanned aerial systems in maritime Search And Rescue (SAR) operations. The items include boats, pallets, people, and buoys. Thermal imaging data is used to differentiate foreground items from the background using a Gaussian Mixture Model (GMM). A Convolutional Neural Network (CNN) is trained using bounding boxes and an estimation of the object's observed area. The k-fold technique with 10 folds results in an average accuracy of 92.5%. Images of boats from different datasets were evaluated using CNN, achieving 100% certainty of being boats.

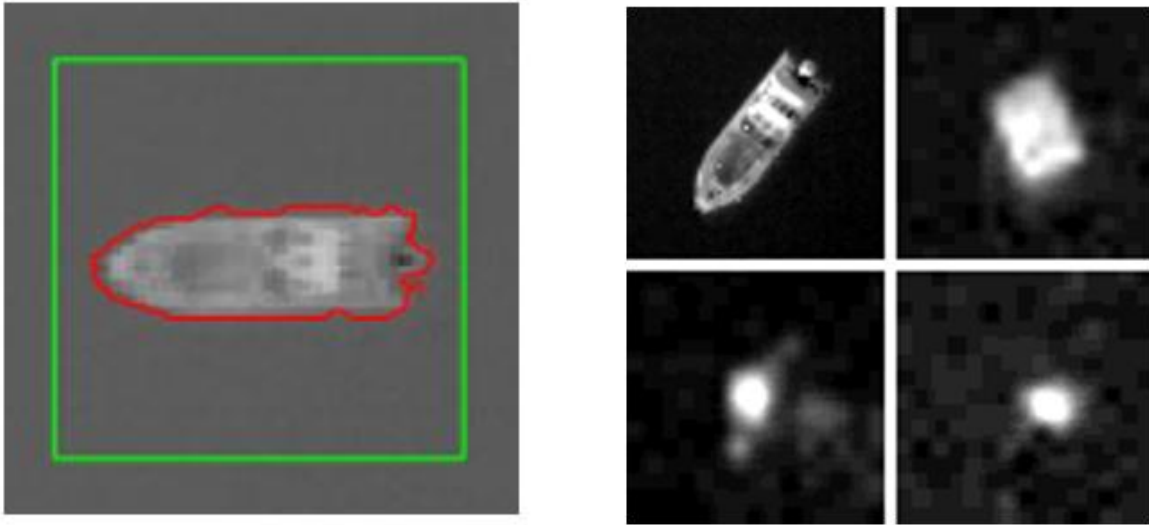


Figure 2.10: Sample of the dataset used[99]

Nam, Y. and Y.-C. Nam [100], A study has proposed new methods for identifying and categorizing automobiles using thermal and visible light images. These strategies can be applied to in intelligent systems for real-time surveillance. The researchers removed headlight and grill areas from the images and used textural characteristics to classify different types of vehicles. They assessed texture, energy, entropy, homogeneity, and contrast in front view photographs. The study found that Classifiers using visible light and thermal pictures have an accuracy of 92.7 and 60.8% when divided into six types and 90.9 and 70.0% when divided into three groups. However, the accuracy of thermal images at night is lower than visual images due to inadequate resolution and fewer similar characteristics.

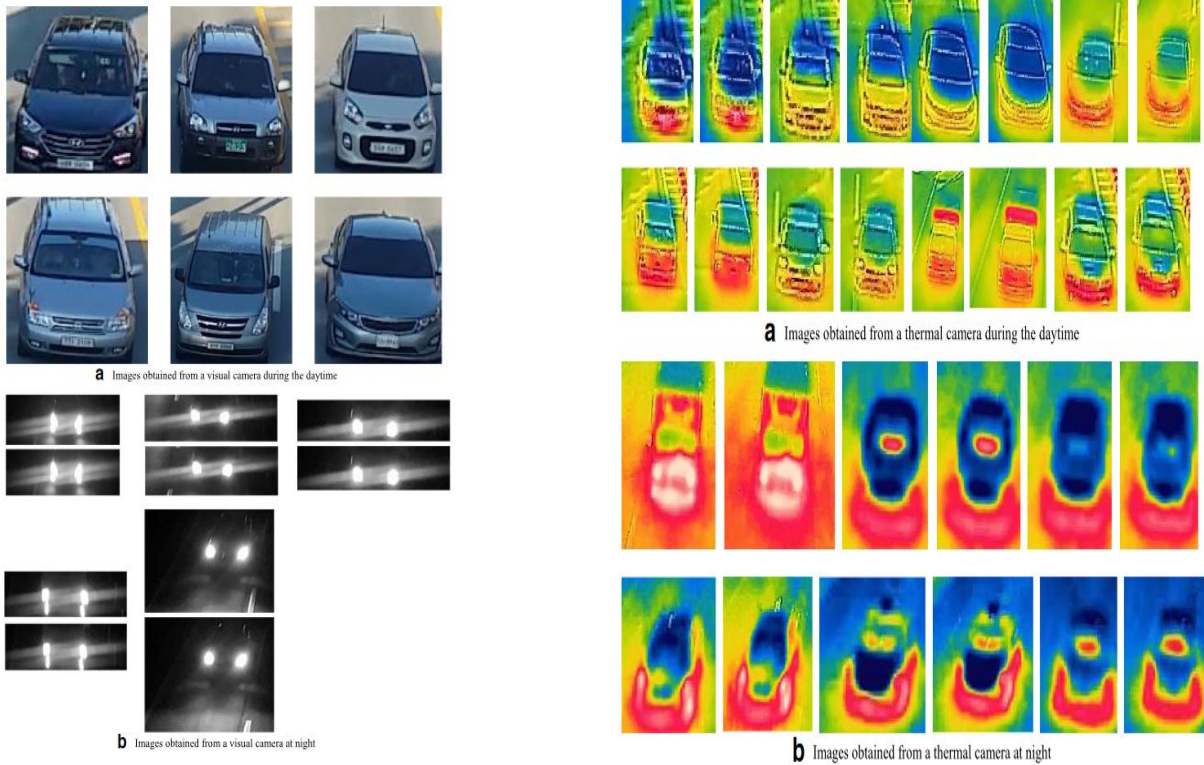


Figure 2.17: Sample of the dataset used [100]

Khalid, B., M.U. Akram, and A.M. Khan [10], The authors propose a method for semantically segmenting each person in an image by combining thermal and visual images using a CNN. They used an encoder-decoder architecture for fusion, and the Resnet-102 architecture for image classification. The Mask-RCNN architecture localizes objects using the Resnet-101 architecture. The multispectral data set from KAIST is used to train CNNs, with a beam splitter aligning LWIR and RGB picture data. The dataset consists of 90,000 optical and thermal image pairs, each 640 by 512 in size. A 70:30 split of the data set is used for training and testing. The results show that the fused model performs better for object localization than the visible model and offers promising results for human identification in surveillance applications. The proposed model's miss rate of 0.20% is significantly lower than the prior state-of-the-art method applied to the KAIST dataset.

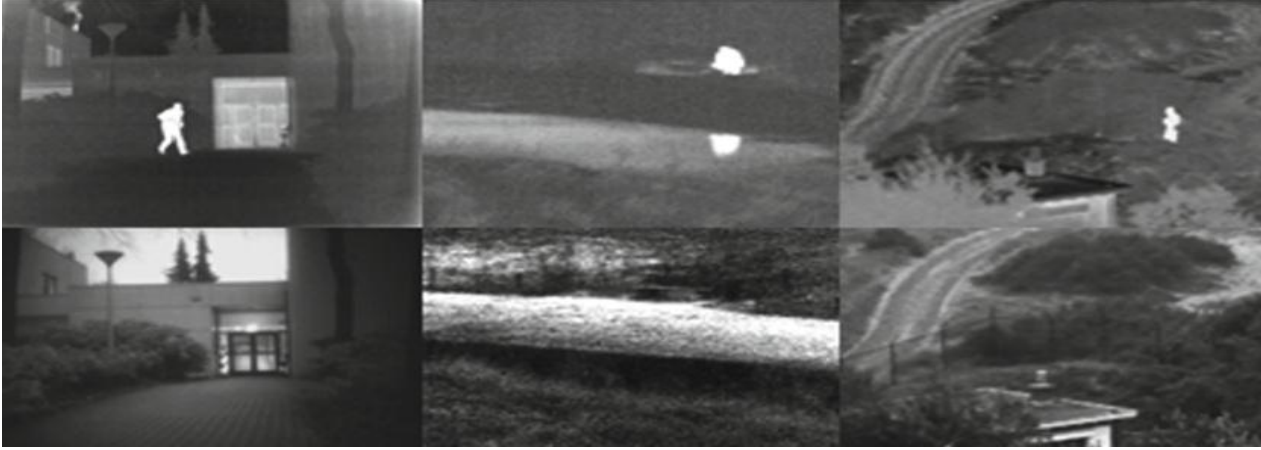


Figure 2.17: Sample of the dataset used [10]

Bhatarai, M. and M. Martinez-Ramon [11], The Santa Fe Firefighting Facility in New Mexico is using infrared cameras to improve situational awareness among firefighters by classifying objects from thermal imagery. A deep convolutional neural network is trained to classify standing, sitting, and crawling stances. The results show that the network's accuracy ranges from 97 to 99%, but if the number of layers is increased to 8, confusion matrices show a misunderstanding of 6 to 11%. A single-layer network achieves 70% accuracy, while a four-layer network achieves over 97% accuracy.

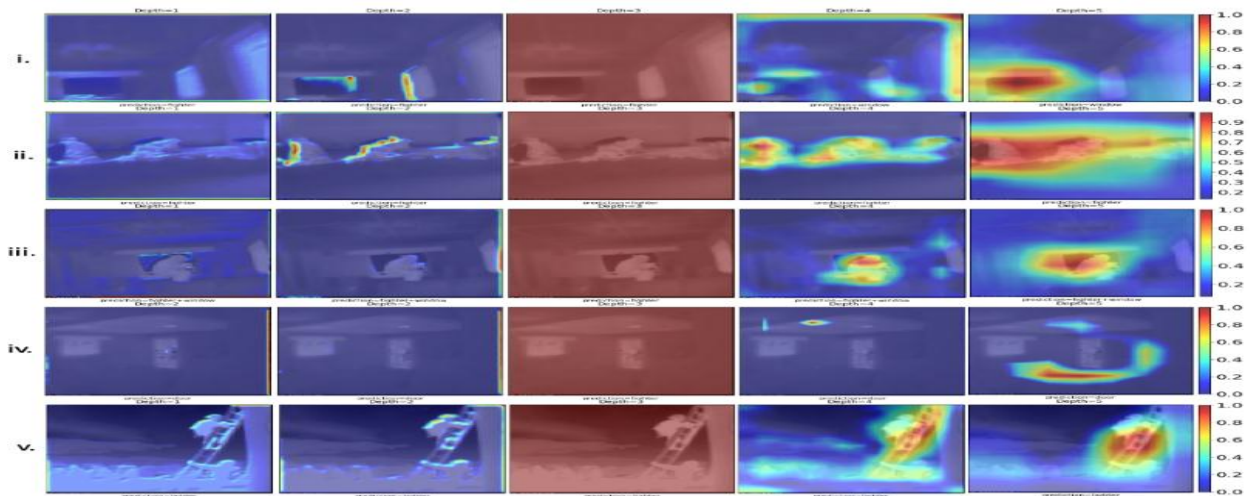


Figure 2.18: Sample of the dataset used [11]

Ivašić-Kos, M., M. Krišto, and M. Pobar [102], Researchers have developed a convolutional neural network called YOLO to automate person detection in thermal recordings and photos. They trained the YOLO network on a portion of their dataset, using a dataset of thermal movies that can show up to three people. The researchers compared the performance of the YOLO network, pre-trained on the COCO image dataset of RGB photos, with the YOLO network after extra training on thermal images from their dataset. The bYOLO model received training using thermal pictures from their proprietary dataset to improve performance. The average precision (AP) metric was employed to evaluate the models' performance, the bYOLO model has an accuracy of 97% and a recall of 19%, but the tYOLO model has an accuracy of 97% and a recall of around 70%. This implies that the tYOLO model finds a lot more individuals in the photographs with the same accuracy.

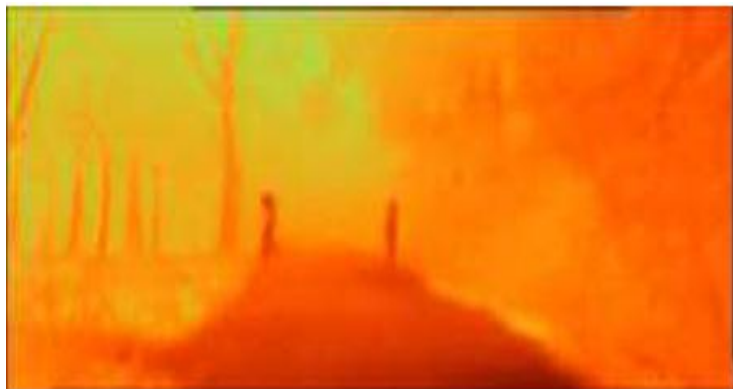


Figure 2.19: Sample of the dataset used [102]

Ippalapally, R., et al. [1], the study compared two models (SSDMobileNetV1 and SSDMobileNetV2) for object detection and classification in thermal camera pictures. Both models performed well in identifying objects from three classes: cars, bicycles, and people. Model 2 outperformed Model 1 in most assessment parameters, with 83% and 99% accuracy levels for automobiles and 98% and 99% for

people. Models 1 and 2 had 97% and 96% accuracy rates for bicycles, respectively. Model 2 had a detection rate of 99%, while Model 1 had 97% and 96% accuracy rates for bicycles. Despite the majority of items in photos being small and medium-sized, both models were able to accurately recognize objects with detection rates of 98% and 99%. Model 2 often had the best success rates in accurately recognizing items of all kinds.



Figure 2.2: Sample of the dataset used [1]

The study by Mittal, U., S. Srivastava, and P. Chawla [6], compared visible-spectrum and thermal images using a faster region-based convolutional neural network. The dataset consisted of 1,000 visible-spectrum photographs and 1,000 thermal images, divided into 800 training photos and 200 testing photos. The study found that visible spectrum images were as accurate as thermal camera images during the day, while thermal images were more accurate at night. Accuracy for thermal photos of a four-wheeler is 90.9%, compared to only 24.3% for visible spectrum photographs. Similar to this, the accuracy of thermal photos in scenarios like 2-wheeler, traffic-light, and person is 88.0%, 71.7%, and 77.1%, respectively, whereas the accuracy of visible spectrum images is 3.9%, 1.9%, and 12.9%, respectively. However, thermal pictures generally deliver better outcomes. The

study suggests that considering various times, seasons, and weather conditions can improve the system's overall usefulness.



Figure 2.21: Sample of the dataset used [7]

Zhang, H., X.-g. Hong, and L. Zhu [103], DDSSD, a feature-fusion module-enhanced SSD, has been proposed as a revolutionary solution for tiny item recognition. The technique involves adding semantic information to the shallow layer of SSD using a dilatation convolution module, also known as transpose convolution or up-sampling layer. DDSSD contains DE-convolution layers, which expand the receptive field of features from shallow layers. However, SSDs cannot save both local detailed characteristics and global semantic features. The detector must incorporate context, and feature maps are up-sampled at higher resolution using DE-convolution layers.

The FLIR ADAS dataset, which includes 10,228 frames and 9,219 images with MS COCO annotations, is used to test the model. The dataset benchmark includes 22,372 instances of people, cars, and bicycles, with 58% belonging to small objects under 32×32 . The training strategy is similar to PASCAL VOC. On the PASCAL

VOC2007 test, the network achieves 79.7% mAP, while on the MS COCO test-dev, it achieves 28.3% mmAP. Particularly for small items, DDSSD outperforms several cutting-edge object detection algorithms in terms of accuracy and speed, achieving 10.5% on MS COCO and 22.8% on FLIR infrared dataset.

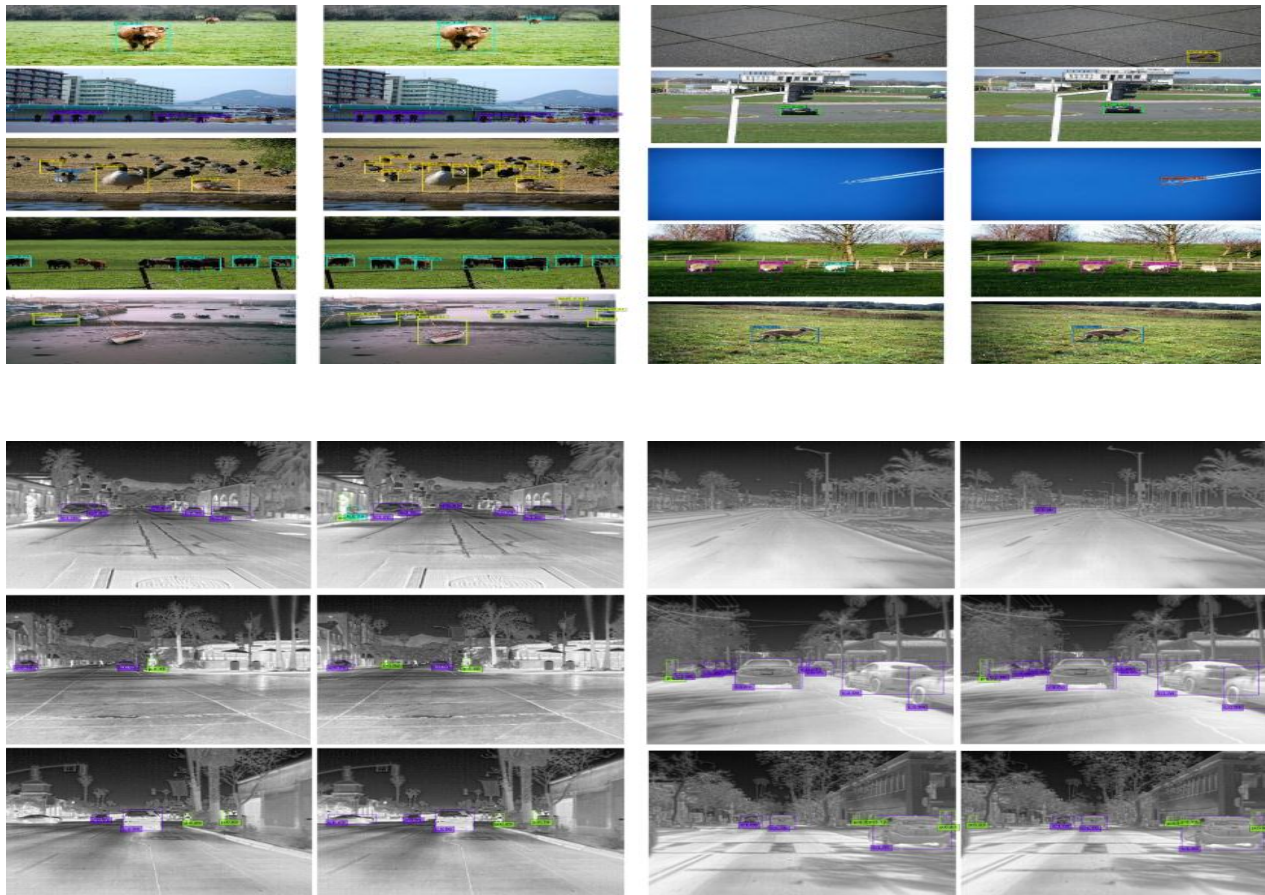


Figure 2.22: Sample of the dataset used [103]

Table 1.1: Summary of the Related Work.

An overview of some current deep learning-based object detection studies in thermal images					
Year	Study	Proposed solution	Dataset	object detection method	Results
2018	Rodin, C.D., et al [99]	The Gaussian Mixture Model (GMM) is utilized to distinguish between objects in the foreground and background. The CNN is trained using bounding boxes that enclose the object.	thermal camera images taken by Unmanned Aerial Systems (UAS)	segmentation	The average accuracy was 92.0% with a 0.50% standard deviation
2018	Nam, Y. and Y.-C. Nam [100]	From front-view images, they evaluated the energy, entropy, homogeneity, texture, and contrast. To extract features	they collected videos during the day and at night using FLIRONE.	Classification	When cars were categorized, the accuracy of the thermal image classifier was 70.8%, whereas that of the visual image classifier was 92.7%.
2019	Ivašić-Kos, M., M. Krišto, and M. Pobar [102]	YOLO	Their own dataset	segmentation	The bYOLO model has an accuracy of 97% and a recall of 99%, but the tYOLO model has an accuracy of 97% and a recall of around 70%.
2019	Mittal, U., S. Srivastava, and P. Chawla [6]	Faster R-CNN	FLIR (released in July 2018). The dataset contains the thermal images as well as visible spectrum images	segmentation	The accuracy for ϵ -Wheeler thermal imaging was 70.9%, compared to 24.3% for visible spectrum photos. Thermal pictures had 08.0%, 71.7%, and 77.1% accuracy for two-wheelers, traffic lights, and humans, compared to 3.9%, 1.9%, and 12.9% for visible spectrum photos, respectively.

Year	Study	Proposed solution	Dataset	object detection method	Results
2020	Khalid, B., M.U. Akram, and A.M. Khan [10]	The fusion of visible and thermal pictures is performed using CNN, and the resulting image is then fed into the Resnet-101 architecture for image classification. For the purpose of locating people, images from Resnet-101 are subsequently put into Mask-RCNN.	KAIST multi-spectral	semantic Segmentation	Compared to the prior state-of-the-art technique used on the KAIST dataset, the suggested model's miss rate of 0.20% is substantially superior.
2020	Bhatarai, M. and M. Martinez-Ramon [11]	Employ a trained CNN system to categorize and pinpoint interesting things.	Thermal images	Classification	A single-layer network achieves 70% accuracy, while a four-layer network achieves over 97% accuracy.
2020	Ippalapally, R., et al [1]	a comparative study of two different models SSDMobileNetV1 and SSDMobileNetV2	FLIR dataset	segmentation	Both models achieved 83% and 99% accuracy in the automobile class and 98% and 99% in the person class, respectively. Models 1 and 2 had 0% and 76% accuracy rates in the bicycle class, respectively. Model 2 had the highest detection rate at 09%, despite the majority of items being small and medium-sized
2021	H., X.-g. Hong, and L. Zhu [13]	DDSSD	VOC2007 and MS COCO	segmentation	On the PASCAL VOC2007 test, the network achieves 79.7% mAP, while on the MS COCO test-dev, it achieves 28.3% mmAP. Particularly for small items, achieving 10.0% on MS COCO and 22.8% on FLIR infrared dataset.

۲.۱۳ Summary

The theoretical underpinnings of detecting and recognizing objects are outlined in this chapter. It also describes other techniques, including convolutional deep learning and neural networks. The design and execution of the system will be covered in the upcoming chapter. The efficiency of algorithms in addition to the proposed model is also evaluated using a set of criteria, such as accuracy, recall, precision, and F¹-score, in order to gauge the rate method as a whole.

CHAPTER THREE

PROPOSED METHODOLOGY

3.1 Overview

The thesis' primary aim is to create an intelligence system for object detection and segmentation based on thermal images. Within this section, the presented suggested system is implemented training and test modes are the two primary modalities. Every mode has several phases and actions.

3.2 Models Implementation

The proposed model execution consists of two steps, or modes: training and testing:

- Training stage: The proposed model trains each training existing datasets for learning on samples with their class by feeding sample training samples with appropriate labels. This stage also includes the validation process.
- Testing stage: The proposed trained model is put into practice this stage is to simulate the use of the system by predicting the output class from unseen dataset.

3.3 Proposed System

The proposed system presented an object classification and segmentation system using deep neural networks. The primary objective of the suggested system is object detection in thermal images. Two different classification and segmentation systems have been built. The proposed system for classification is Transfer learning (MobileNetV2 and VGG19) and (yolov4) for segmentation, each one preceded by a

pre-processing step, as well as every stage contains several steps that perform different functions.

Phases involved in developing the model include preparing the data, building the model, training it, evaluating it, and testing it. The block diagram of the suggested deep neural networks models is seen in Figure 3.1) to classification model and to segmentation model.

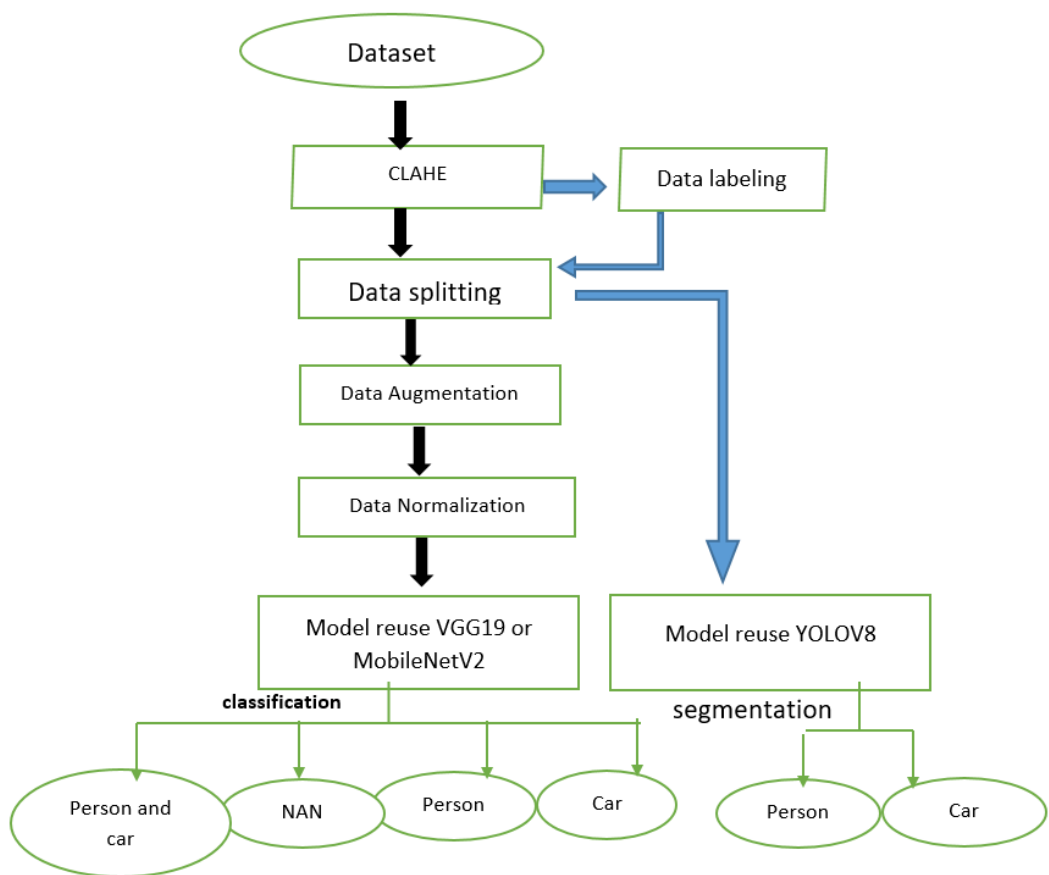


Figure 3.1): Diagram for the workspace.

3.3.1 Classification Model

The classifying model in thermal images using MobileNetV2 and VGG19 pre-trained. The central objective is to establish a robust classification system that remains effective despite fluctuations in thermal patterns. Prior efforts encountered challenges like limited data, artifacts, and one-way adaptability. Making use of the advancements in the VGG19 and MobileNetV2 architecture offers a workable option.

By capturing important thermal properties, the model seeks to accurately classify objects in thermal images, ensuring accurate classification of objects even in the presence of temperature fluctuations.

Using the capabilities of VGG19 and MobileNetV2, we modified the architecture to include thermal images in this section in order to build a model that accurately classifies objects and regulates temperature changes.

3.3.2 Segmentation Model

We applied the object segmentation technique using YOLOv4 in the second stage. This method's primary objective is to create a dependable system that can precisely identify and segment objects in thermal pictures. Previous segmentation approaches have struggled with border precision, class confusion, and model efficiency and accuracy.

This method's main objective is to precisely identify object boundaries in thermal pictures. To reduce class confusion, the model has to be extremely precise in segmenting and classifying objects.

We utilized YOLOv4's ability to segmentation objects in thermal pictures using this architecture. This application facilitates the process of

creating a model that is very proficient in identifying and segmenting object boundaries by tackling certain problems that are brought about by thermal imaging.

The model attempts to improve the precision and effectiveness of object segmentation in thermal photos by utilizing YOLOv^λ and tackling issues particular to thermal photography.

3.ξ Dataset

Data collection is an important stage in the process since contemporary deep learning approaches require less feature extraction and large amounts of data. Even when the training methodologies cannot completely clean data, we can make due with less-than-ideal data for model training.

We have used dataset from kaggle:

(<https://www.kaggle.com/datasets/albertofv/flir-thermal-images-dataset-reduced>). FLIR ADAS proposes 1024 images with 8000 annotations. These images have been acquired during a large period, 60% by day and 40% by night in various scenes. The images of the FLIR dataset are available under different formats: the raw images directly issued from the sensor and the modified and enhanced ones thanks to specific FLIR algorithms. Due to the difficulty of obtaining thermal images that are appropriate for our work, after obtaining this raw data set, we cut the images to obtain images of a separate person and car, as well as images that do not contain objects.

Original image

Images after crop

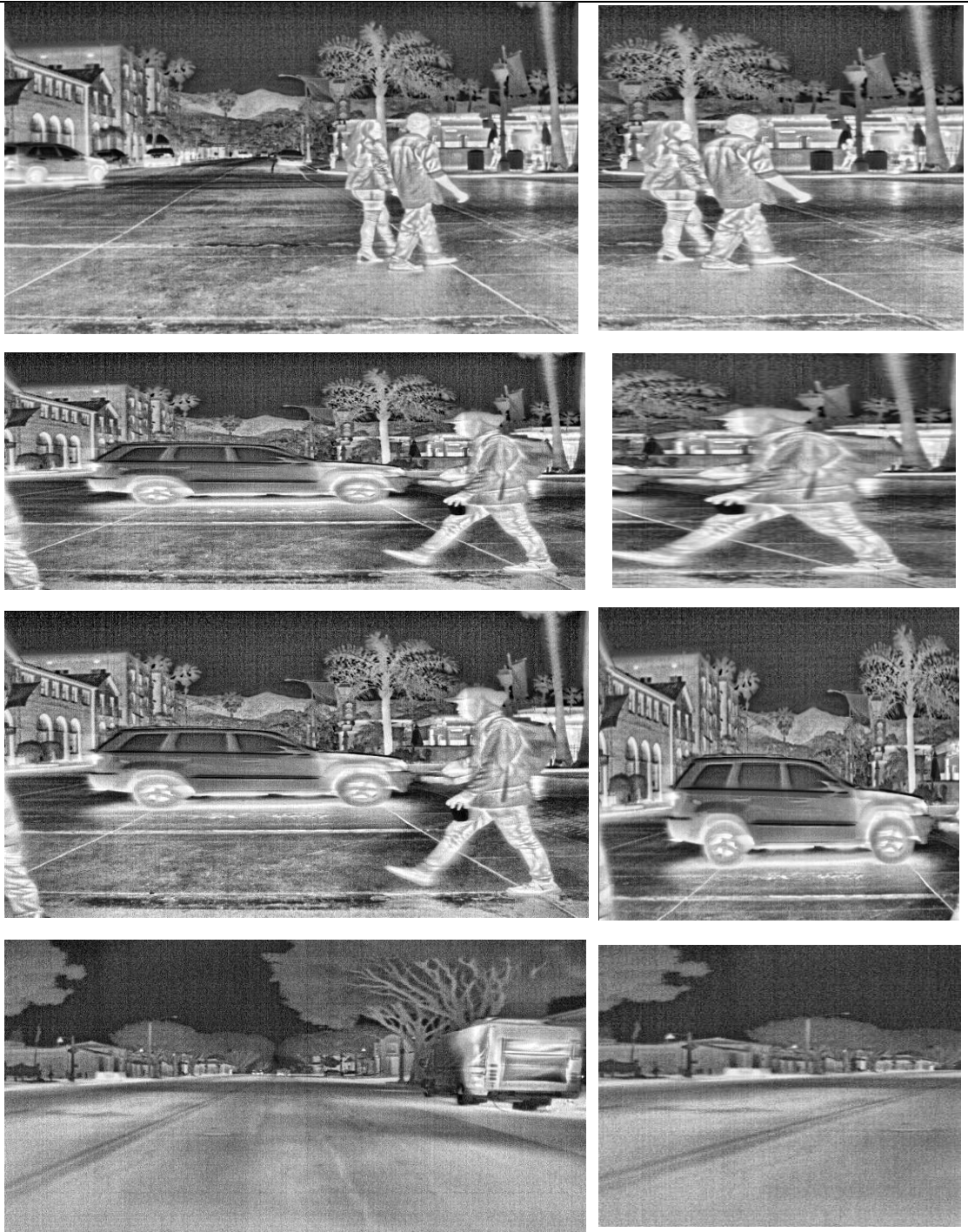


Figure 7.7: Examples taken from the dataset.

۳.۴.۱ Dataset for classification

Within this set, the dataset was created using part of the FLIR ADAS datasets, by manually segmenting and sorting the images, and the data contains two categories of objects, person and car, and contains four image files (person), car, mixed (person and car), None (None) Each of these files contains ۷۳۹ thermal images. The images in the classification dataset have different sizes. The total number of images used is ۲۹۵۶ images.

۳.۴.۲ Dataset for segmentation

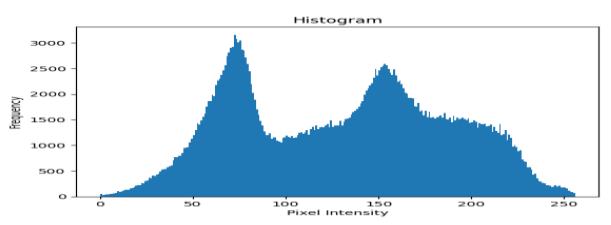
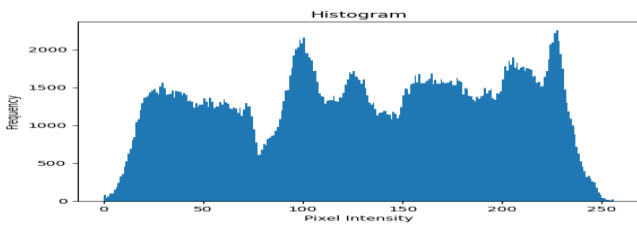
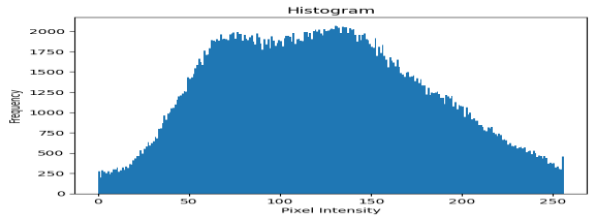
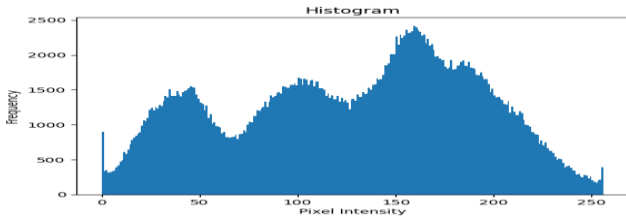
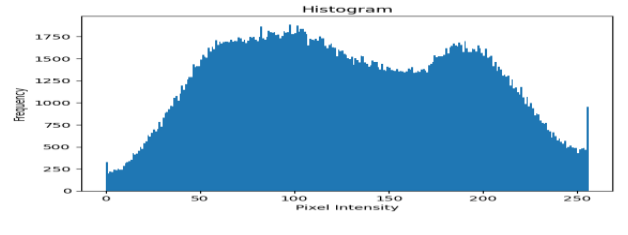
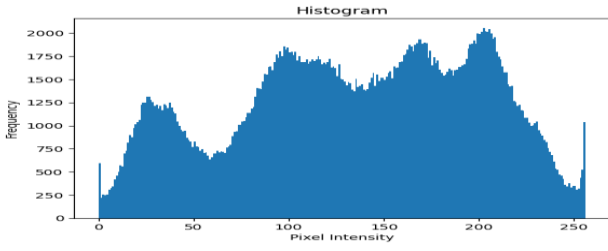
The second group used ۱,۸۹۸ thermal images, and the autodistill package released by roboflow in the June month of ۲۰۲۳ was used to generate annotations and associate them with two categories of people and cars.

۳.۵ Dataset preprocessing

Data preparation for neural network modeling is contingent upon the input requirements of the model. Initializing and converting raw data could be a crucial stage in the pre-processing procedure for efficient model training. These pre-processing steps include picture conversion to grayscale pixel grids, image values rescaling from ۰ to ۲۵۵ to [۰,۱] interval, and image enhancement. Since it's frequently required to change a picture's size, an important part of image processing is picture rescaling.

۳.۵.۱ Contrast Limited Adaptive Histogram Equalization (CLAHE)

We produced an improved image that is superior to the originals by applying an algorithm that was used especially for thermal photos. Using the CLAHE technique, each picture is subjected to histogram equalization after being divided into contextual areas. The distribution of gray values employed is produced by these equalizations, which enhances the visibility of the image's hidden characteristics. The enhanced algorithm for AHE is called CLAHE. The following outcomes, displayed in the pictures below, are the product of the optimization approach. The visual findings of the thermal pictures are shown in Figure ۳.۳. When compared to the adaptive histogram, the suggested approach enhances the image more precisely.



Before using CLAHE

After using CLAHE

Figure 3.3: Sample pictures before and after using CLAHE

A sample of images after using CLAHE. The results of the images had better resolution and clarity than before, as shown in the histogram.

3.6 Dataset splitting

In the machine learning process, datasets for testing, validation, and training are kept apart. Each set has a specific purpose during the model's training, fine-tuning, and assessment stages. While the training set helps identify patterns and correlations in the data, the validation set improves performance and optimizes hyper-parameters without affecting generalization to new data. The testing set is used for the final analysis as a reliable indicator of the model's ability to generalize. The model's efficacy in handling real-world events is increased by this three-way separation, which guarantees rigorous training, refinement, and assessment processes.

Three groups were identified in classification and segmentation of the data collection:

Table 3.1: Divide the image data into three groups. For both segmentation and classification

	Train	Validation	Testing
<i>Classification</i>	70%	10%	20%
<i>Segmentation</i>	70%	20%	10%

3.7 Dataset Augmentation

One of the main issues with training models for deep learning, such as convolutional neural networks (CNNs), is the scarcity of data. Stated differently, a sizable training dataset is required for deep learning networks.

Through altering the initial data's shape and getting additional unseen data augmentation techniques, training data is used to enhance the amount of the dataset during network training. It uses a number of techniques, the most common being zoom, rotation, translation, scale, and flip. These techniques are intended in order to lessen the possibility of overfitting and improve the display of the generated model. In certain cases, only a limited training dataset is available for the majority of critical real-life situations (e.g., medical datasets). This is what distinguishes more advanced machine learning algorithms from others.

Table 3.7: The utilized Data Augmentation hyper-parameter values

Parameter	Value
<i>rescale</i>	1/255
<i>rotation_range</i>	20
<i>width_shift_range</i>	0.2
<i>height_shift_range</i>	0.2
<i>shear_range</i>	0.2
<i>zoom_range</i>	0.2
<i>horizontal_flip</i>	True
<i>fill_mode</i>	'nearest'

3.8 Normalization

The intensity of each pixel in the picture data is represented by its pixel values, which are integer values between 0 and 255. When processing inputs, modest weight values are handled by neural network models. Large integer value inputs might cause disruptions or slow down the learning process. The process of normalizing modifies the pixel values' intensity range when seen properly, giving every pixel has a value between 0 and 1. No matter how wide the image's pixel values actually are, the procedure of normalizing the values of all the pixels is carried out by splitting all of the data across all channels of the pixels by the greatest pixel value, which is 255. When normalizing values of intensity within the range of (Min, Max), as follows:

The transformation from $I: \{X \subseteq \mathbb{R}^n\} \rightarrow \{\text{Min} \dots \text{Max}\}$ results in the new image IN, whose intensity values fall between (newMin, newMax).

$$IN = \frac{I}{255} \cdot \{X \subseteq \mathbb{R}^n\} \rightarrow \{0 \dots 1\} \quad (1)$$

A portion of the picture is shown in Figure (3.4) to explain the normalizing procedure.

3.9 Building the classification Model

In this architecture, pre-trained MobileNetV2 and VGG19 base models which have acquired features from ImageNet data are integrated to facilitate transfer learning. Since grayscale pictures only have one channel and the original MobileNetV2 and VGG19 architectures are made for three-channel (RGB) inputs, adapting these models for grayscale images is difficult. In order to solve this, the single-channel

grayscale pictures are transformed into 3-channel RGB images using a Lambda layer before being sent via the MobileNetV2 and VGG19 base models. This conversion guarantees that models built originally for color pictures can handle grayscale inputs efficiently.

Additionally, the final classification layers (fully connected layers) of MobileNetV2 and VGG19 are omitted by setting the `include_top` argument to `False` during model initialization. Excluded are the layers that are utilized for single-label classification on datasets like as ImageNet. Overlaying custom layers, the MobileNetV2 or VGG19 architecture is repurposed for a particular task: multi-label categorization of grayscale photos. Using pre-trained features from the base model and fine-tuning it for the new job is made possible by this technique, called transfer learning.

In order to minimize spatial dimensions and preserve important information, the design uses layers of Global Average Pooling. For high-level abstraction learning, Dense layers are used, and for multi-label prediction, the last two dense layers apply sigmoid activation. This design caters to grayscale images, and the multi-label setup accommodates scenarios where multiple class labels can be present simultaneously. Tables 3.3 and 3.4 show the layers and number of parameters used in MobileNetV2 and VGG19 respectively.

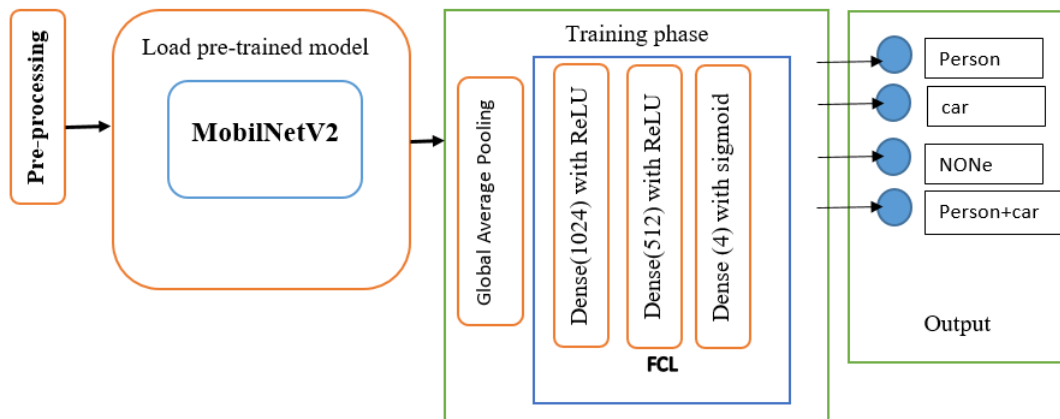


Figure 3.4: Architecture of The Modified MobilNet^{v2} model.

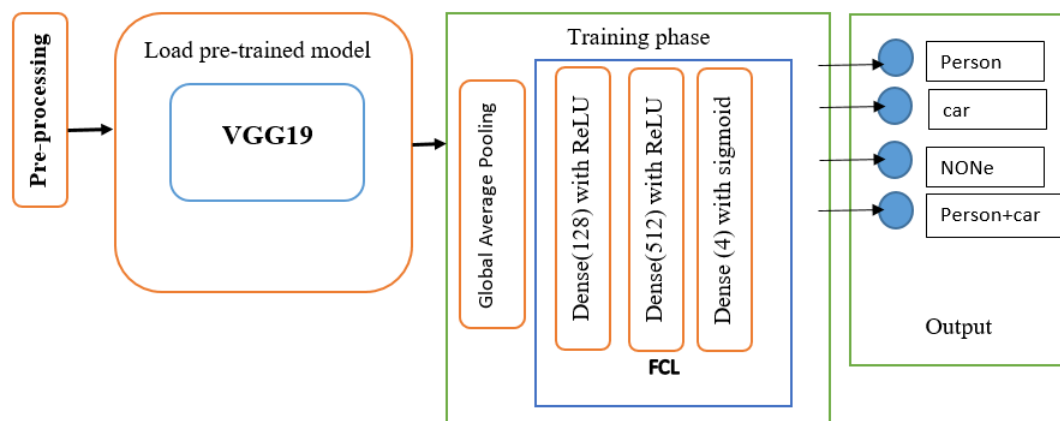


Figure 3.5: Architecture of The Modified VGG¹⁹ model.

3.1. Training the segmentation

YOLO^{v4} models were used in this study's transfer learning process to create a detection network that can make accurate detections in real-time and inference. There are five distinct models available in the YOLO^{v4} architecture namely (YOLO^{v4n}), (YOLO^{v4s}), (YOLO^{v4m}), (YOLO^{v4l}), (YOLO^{v4x}) . To ensure consistency, the trained base

model, a YOLOv⁴ model, was trained using the base models Yolov⁴. Twenty percent of the train dataset was randomly selected to serve as the validation dataset and five percent for testing, which was used to track training progress. The pretrained parameters officially given by YOLOv⁴ were used to determine the beginning values of each model's parameters. The trained base model was trained with the hyperparameters: The learning rate (Lr) of a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 was used, an image size of 720*720, and 100 epochs. The default settings for all other parameters were maintained.

The training function of YOLOv⁴ assesses the model's learning efficiency using the four loss functions. The outcomes of these three—Box loss, Classification loss, and Segmentation loss—will be shown in the section on training results. The difference between the ground truth bounding box and the anticipated bounding box is known as the box loss. The discrepancy between an object's real class and its anticipated class is known as classification loss. The Segmentation Loss quantifies the dissimilarity between the predicted segmentation mask and the ground truth mask.

CHAPTER FOUR

RESULTS AND DISCUSSION

ξ.1 Overview

Resulting of the two systems that are discussed in this chapter. Also the global accuracy of the segmentation and classification models on thermal images test dataset. On test data, however, each model's accuracy is shown as a confusion matrix. Additionally, the final report of the visualization model is shown, together with the data analysis and performance evaluation of the two systems.

ξ.2 System Requirement

ξ.2.1 System Requirement to classification

The following will be utilized in order to put the recommended approach into practice: Hardware/Central Processor Unit (CPU): 2.70GHz Intel(R) Core(TM) i7-12700H CPU. 32GB RAM and a 8GB GDDR6 NVIDIA GeForce RTX3050. The operating system is Windows 10 (64-bit), and the programming language is Python.

ξ.2.2 System Requirement to segmentation

We utilized Google's colab developed hosted runtime to conduct the research. The GPU utilized for training and testing was the Tesla T4 from Nvidia.

Our use of Google Colab was due to our computer's GPU being underpowered

4.3 Proposal classification Models

4.3.1 Classification Model based on MobileNetV2

Figure 4.1 illustrates the suggested method's process. The foundational model used in this study is the MobileNetV2 network. We augment the MobileNetV2 convolutional layers with many layers, referred to as the head model. The feature map, that is to say the base model's output, is the input for the global pooling layer, which is the first layer of the head model. The dimension of the data is greatly reduced by the global pooling layer, a pooling procedure that creates a vector of one-dimensional features. In the global pooling layer, the suggested approach makes use of an average pooling procedure.

Two fully-connected or dense layers come after the global pooling layer. The fully-connected layers used an activation function of Rectified Linear Unit (ReLU) and had 1024 and 512 nodes, respectively. The output layer comes next, when the dataset is divided into four classes: people, car, None, mixed (person and car). The activation function used in the output layer is sigmoid.

The pre-trained weights are used in the basic model, and the layers are frozen. As a result, only the head model's weights were trained throughout the training phase. Due to its rapid convergence, the SGD optimizer is the optimization technique employed during the training phase. As a result, we only employ 100 epochs, there are 24 in the batch size. Where network has to minimize a cost function called categorical cross-entropy during the training phase.

We did data augmentation on the training set in order to enhance the model's capacity for generalization. Data augmentation modifies the provided data—in this example, the training set's images—using one or more of the identified processes. The data augmentation procedure in this study involves the following operations: rotation up to 20° , flip, zoom, and shift. The network is then fed with the enriched data throughout every training cycle iteration to ensure that it doesn't always view the same data. As a result, the quantity of data that is sent into the network during each training cycle remains constant. But every repetition procedure will utilize different data, and the final model will be more capable of generalization. The ability of the model to handle testing data that has not been previously trained if it has strong generalization capabilities. As shown in figure 4.1.

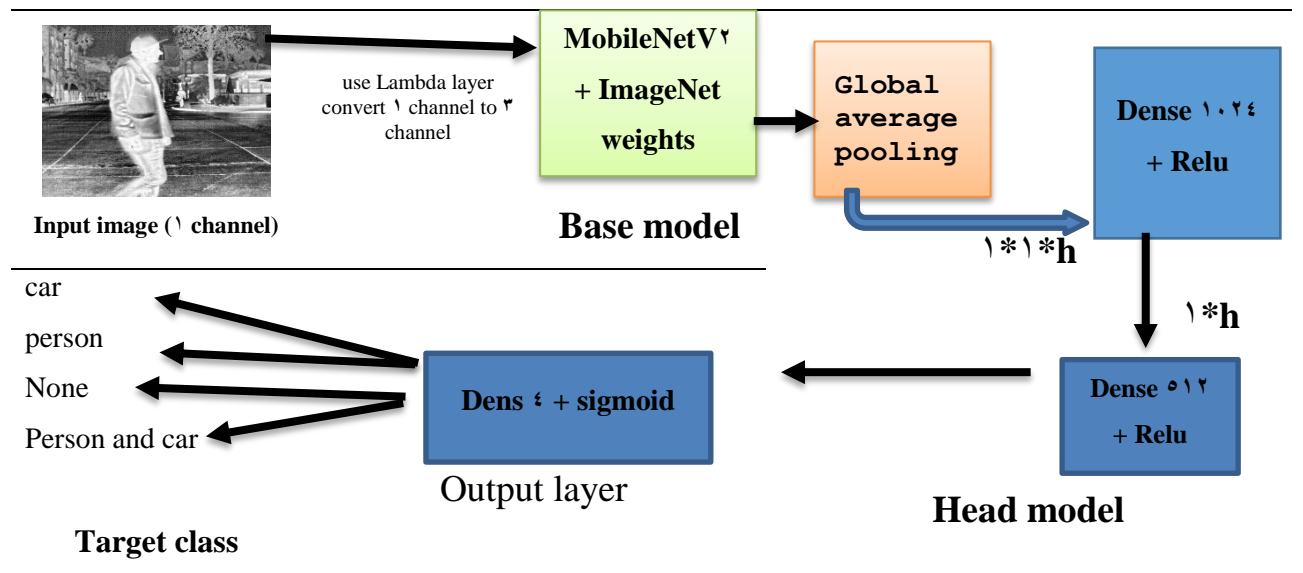
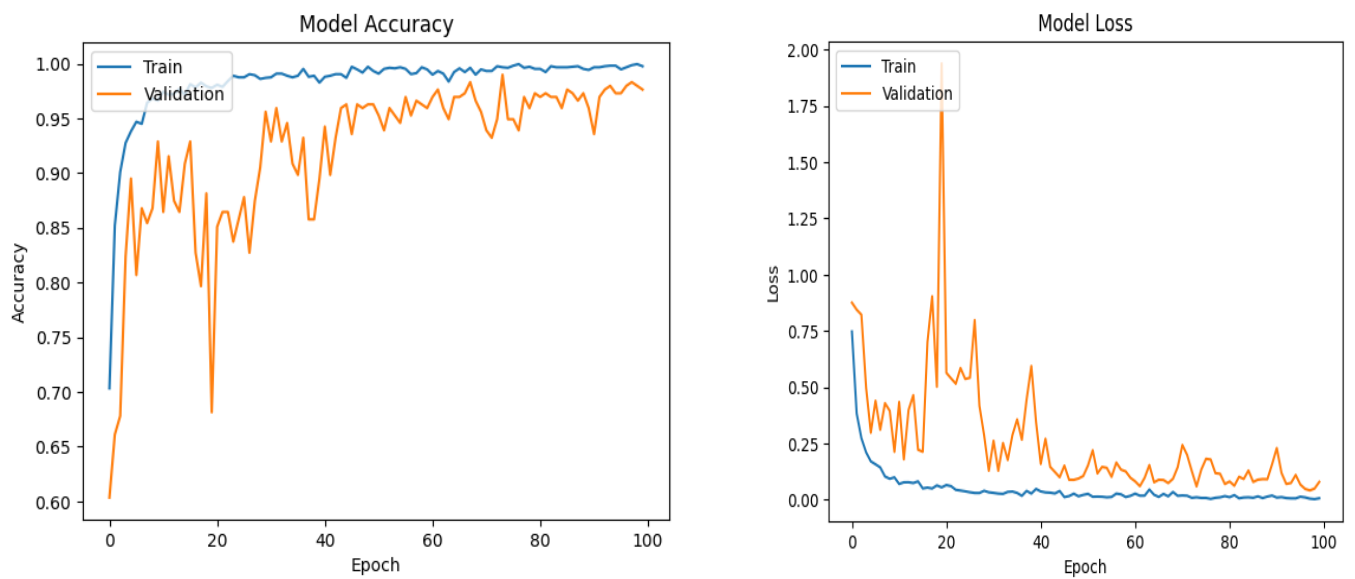


Figure 4.1: Proposed MobileNetV2 Models.

Table 4.1: Hyperparameter of the MobileNetV2 Model in Implementing.

Hyper parameter	Value architecture
Optimizer	SGD
Learning rate	0.0001
Batch size	24
Early Stopping	20
Epochs	100

Figure 4.2 shows a neural network's accuracy and loss after 100 epochs of training and validation, with Figure 4.2a illustrating model accuracy and Figure 4.2b illustrating model loss.



(a) Model accuracy

(b) Model loss

Figure 4.2: The suggested architecture's accuracy and loss function

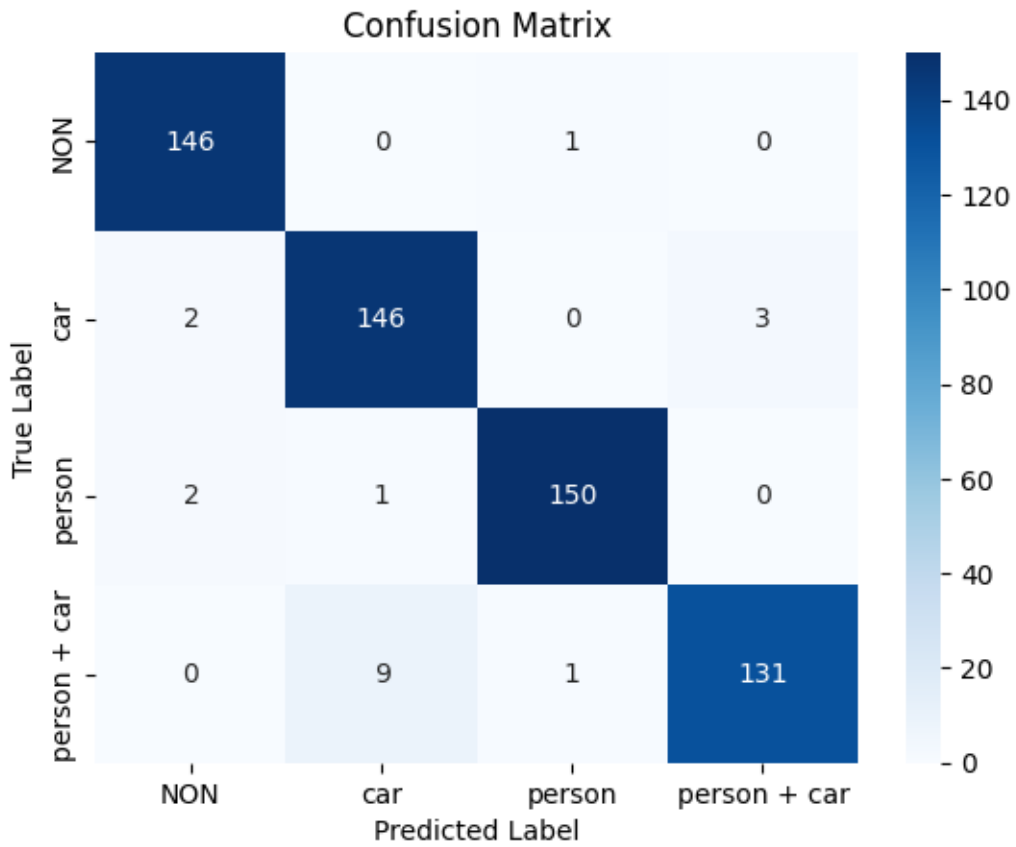


Figure 4.7: The Proposed Architecture's Confusion matrix

Table 4.7: Specifics of the Outcomes attained During Model Testing.

Type model	Accuracy	F ¹ -score	Precision	Recall
MobileNetV2	0.97	0.968	0.969	0.968

Table 4.7: Classification Report for Each Object in the MobileNetV2

Type class	precision	recall	f ¹ -score
Person	0.99	0.97	0.98
car	0.94	0.97	0.96
None	0.99	0.97	0.98
Person and car	0.90	0.96	0.96

4.3.2 Classification Model based on VGG-19

The top three dense (totally linked) layers of the original VGG-19 had been eliminated and added Global Average Pooling (GAP) layer in the proposed model. It transforms the input tensor into a $1 \times 1 \times d$ shape from $h \times w \times d$. Due to the much lower number of parameters, this layer not only helps to minimize dimensionality but also over-fitting in the model. Two more dense layers were added following the addition of the Global Average Pooling (GAP) layer. The sizes of these dense layers were 128 and 64, respectively. The suggested model's design is seen in Figure 4.2.

The Activation Function All of the convolutional layers and the dense layers—aside from the last dense layer—use the "Rectified Linear Unit (ReLU)". The sigmoid is employed as the activation function in the last dense layer. Since it provides the probability of each target class, the sigmoid function is typically included in the final layer of a neural network. The projected class has the highest likelihood of all. Basically, a range of numbers between 0 and 1 is what it outputs.

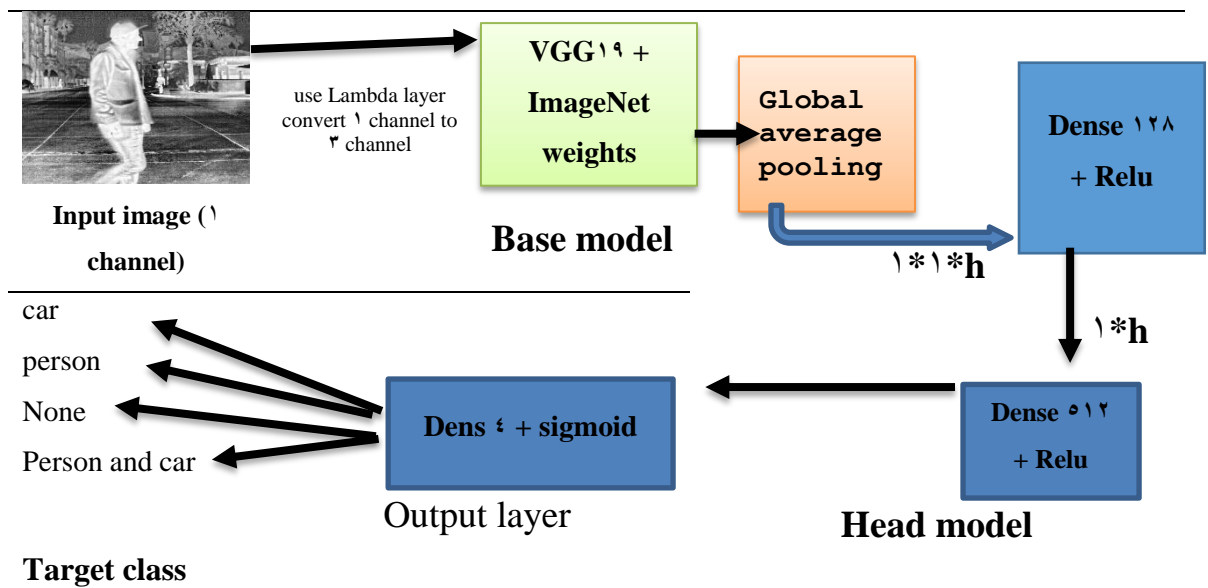
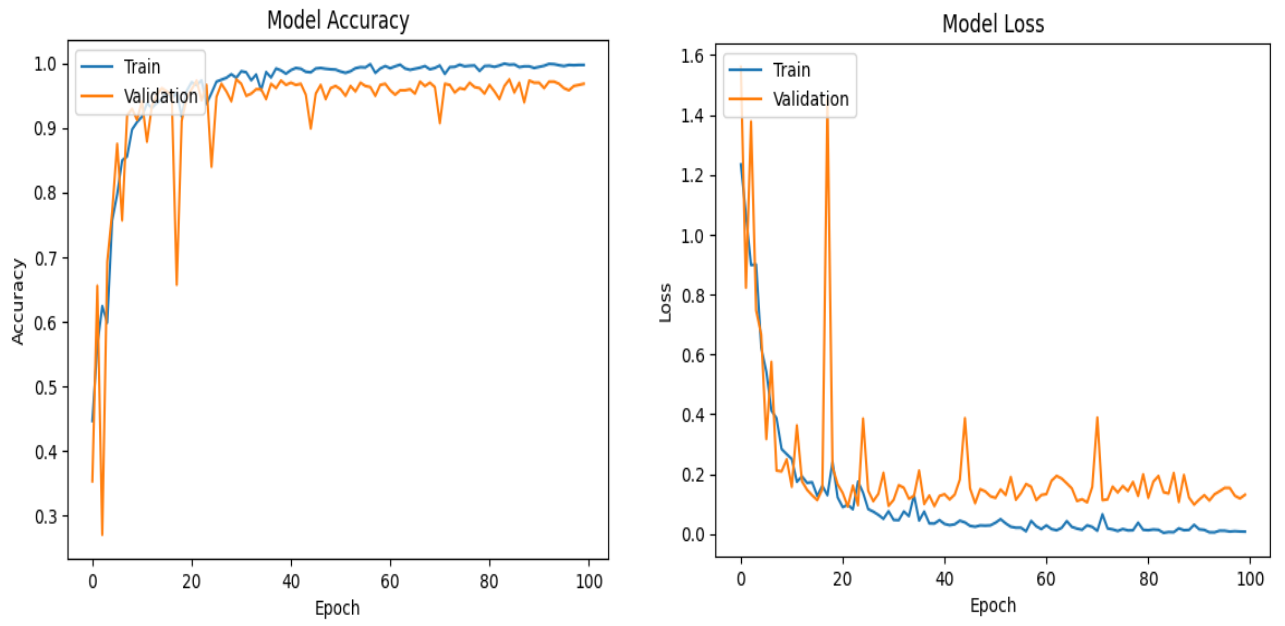


Figure 4.4: Proposed VGG-19 Models.

Table 4.4: Hyperparameter of the VGG-19 Model in Implementation.

Hyper parameter	Value architecture
Optimizer	SGD
Learning rate	0.001
Batch size	32
Early Stopping	20
Epochs	100

Figure 4.5 shows the training and validation accuracy and loss of the neural network for 100 epochs, where Figure 4.5a shows the model accuracy and Figure 4.5b shows the model loss.



(a) Model accuracy

(b) Model loss

Figure 4.6: The proposed architecture's accuracy and loss function.

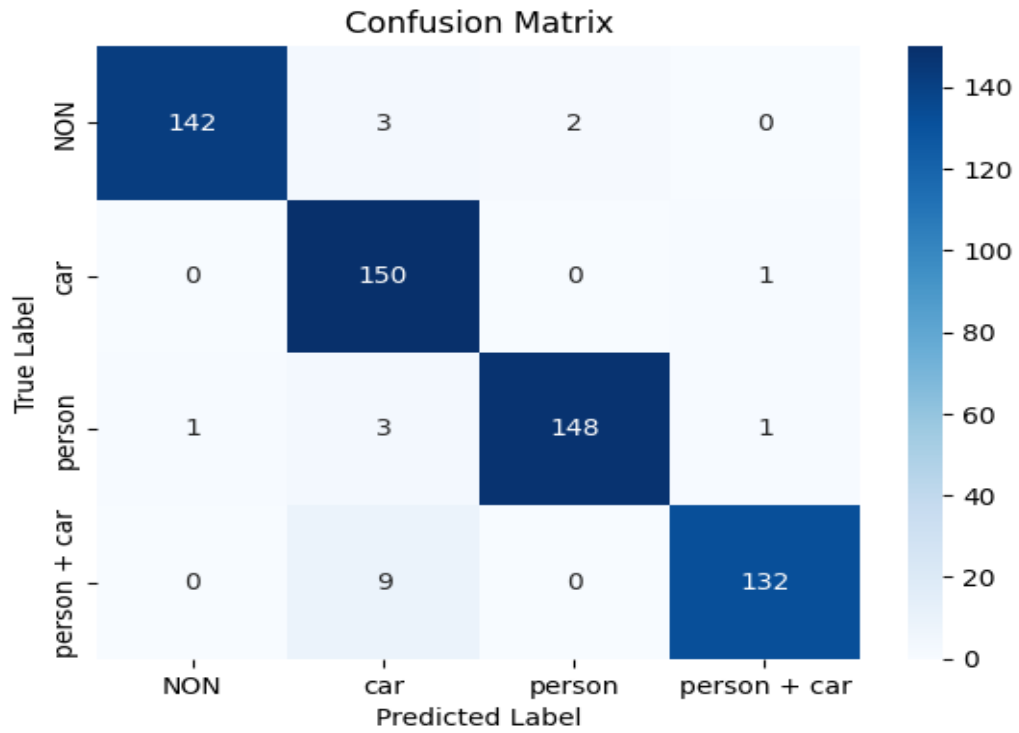


Figure 4.7: The Proposed Architecture's Confusion matrix.

Table 4.6: Specifics of the Outcomes attained While Testing the Model.

Type model	Accuracy	F1-score	Precision	Recall
VGG-19	0.97	0.966	0.968	0.966

Table 4.7: Classification report for each object by using VGG 19

Type class	precision	recall	f1-score
Person	0.99	0.97	0.98
car	0.91	0.99	0.95
None	0.99	0.97	0.98
Person and car	0.99	0.94	0.96

4.5 Test Results of the Segmentation Model

The segmentation model following testing on a dataset made from of 1898 images. Training images has 1443 and validation images 380 images. And test images have 70 image, this dataset has comparable ground truth and was not used in the training procedure. An example of the testing dataset for the detection model with matching ground truth is displayed in Figure (4.1).

Table 4.8: Information on the Outcomes of the YOLO Model Testing.

Algorithm	MAP _{0.5}	MAP _{0.5-95}	P	R	F1 score
Yolov4l-seg	0.82	0.693	0.780	0.762	0.77

Original image

Ground truth

Predict

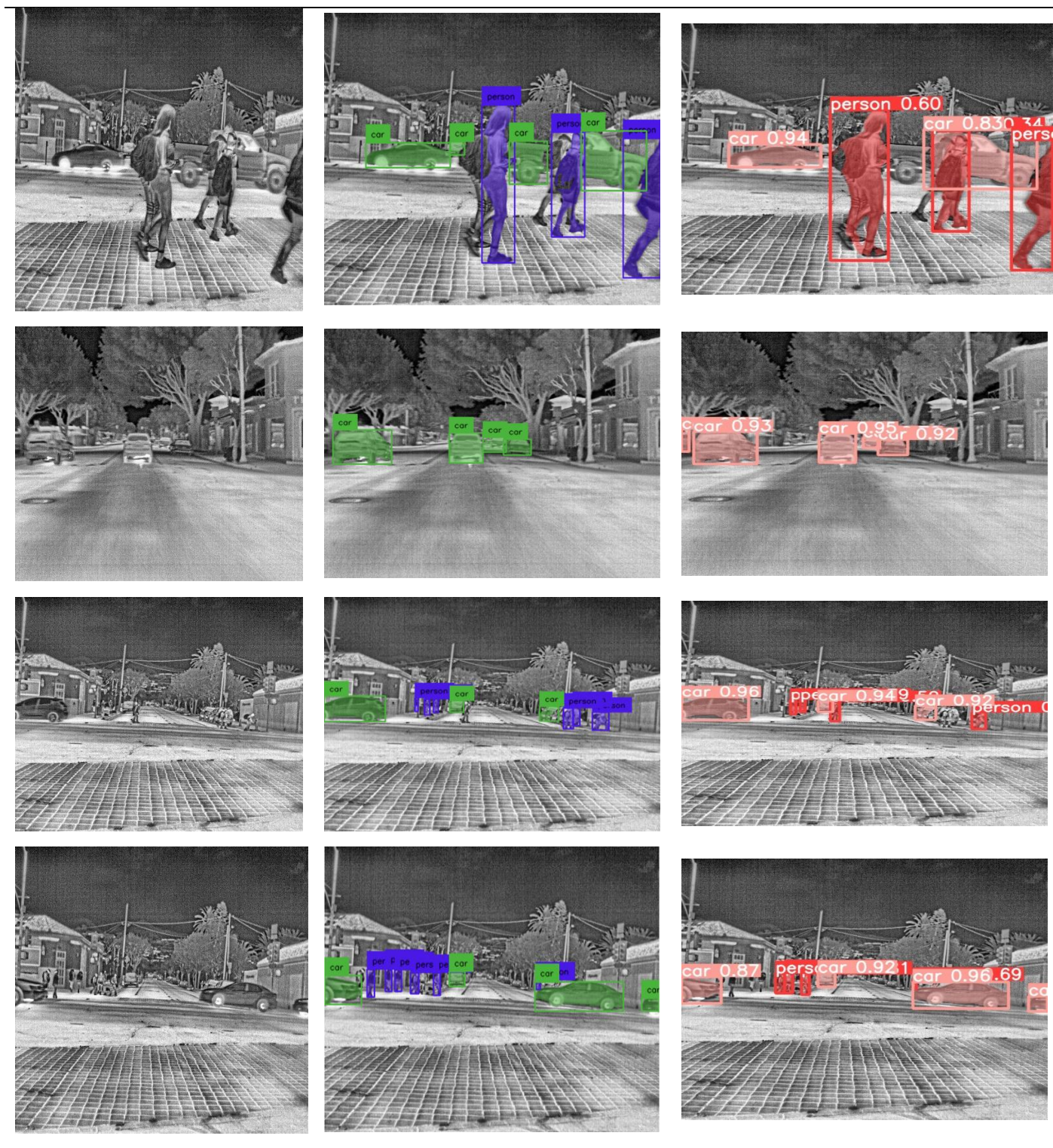


Figure 4.5: Sample of Test Dataset.

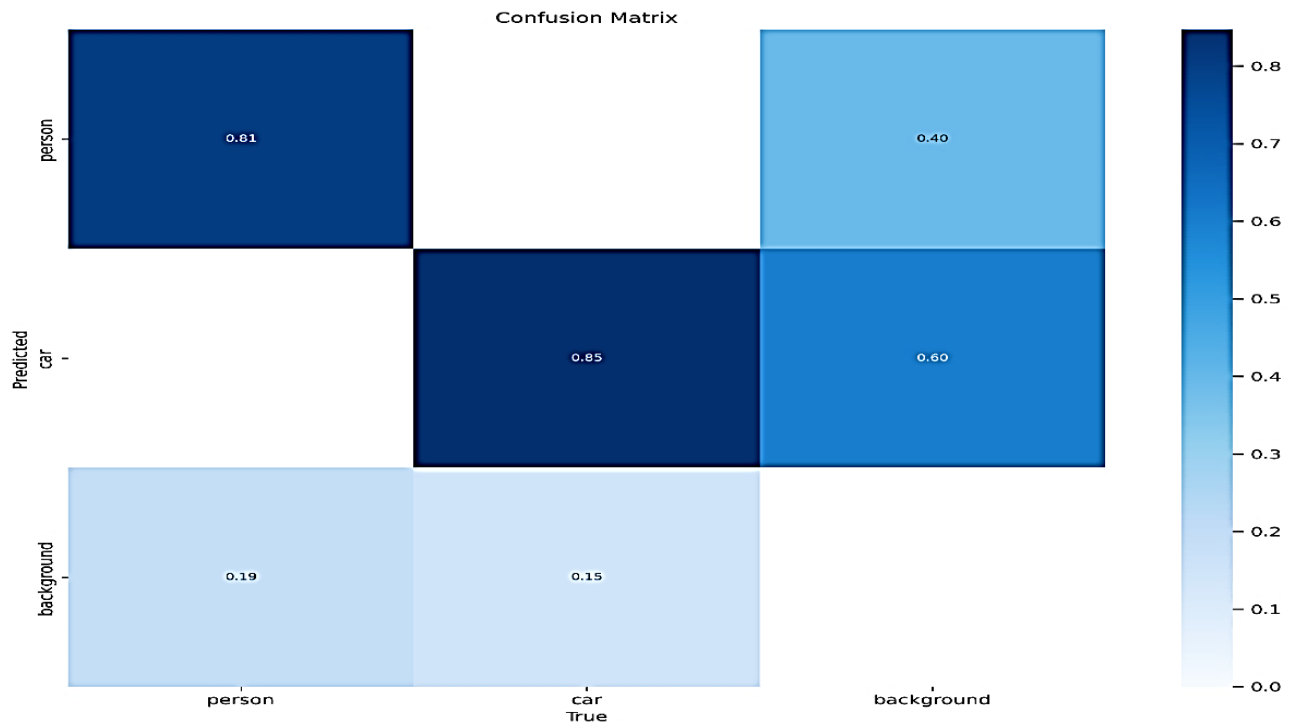


Figure 4.4: Confusion Matrix of The Proposed Architecture

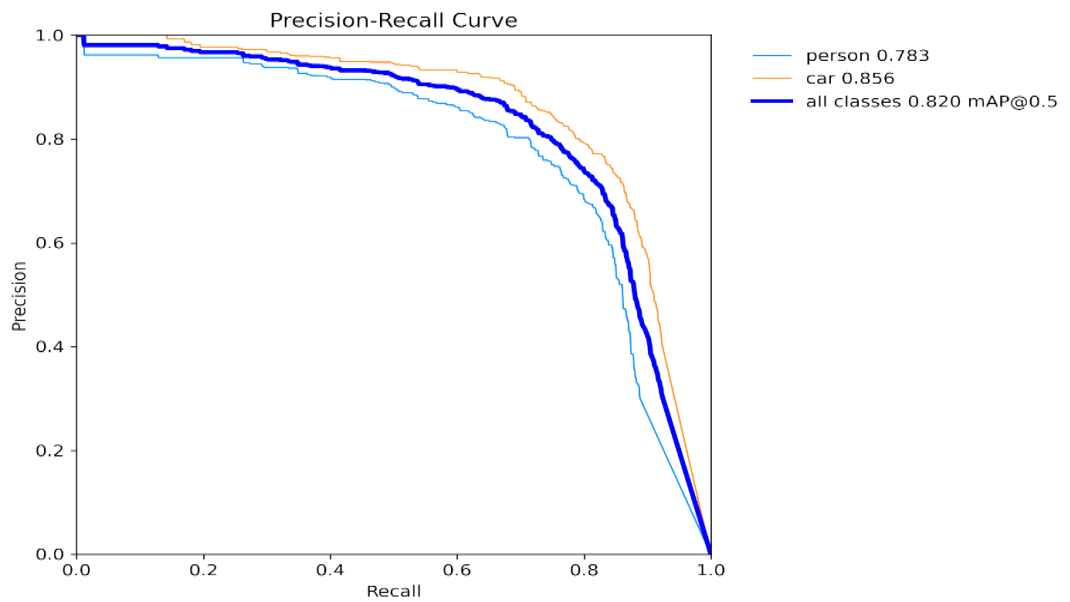


Figure 4.5: Average Precision (AP) of Detection Model.

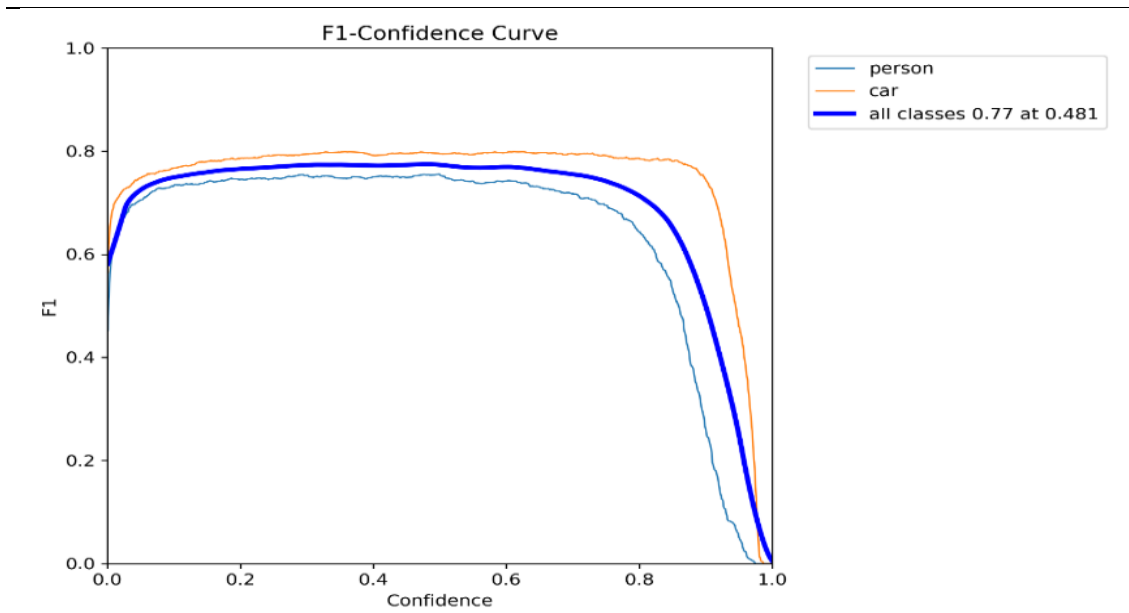
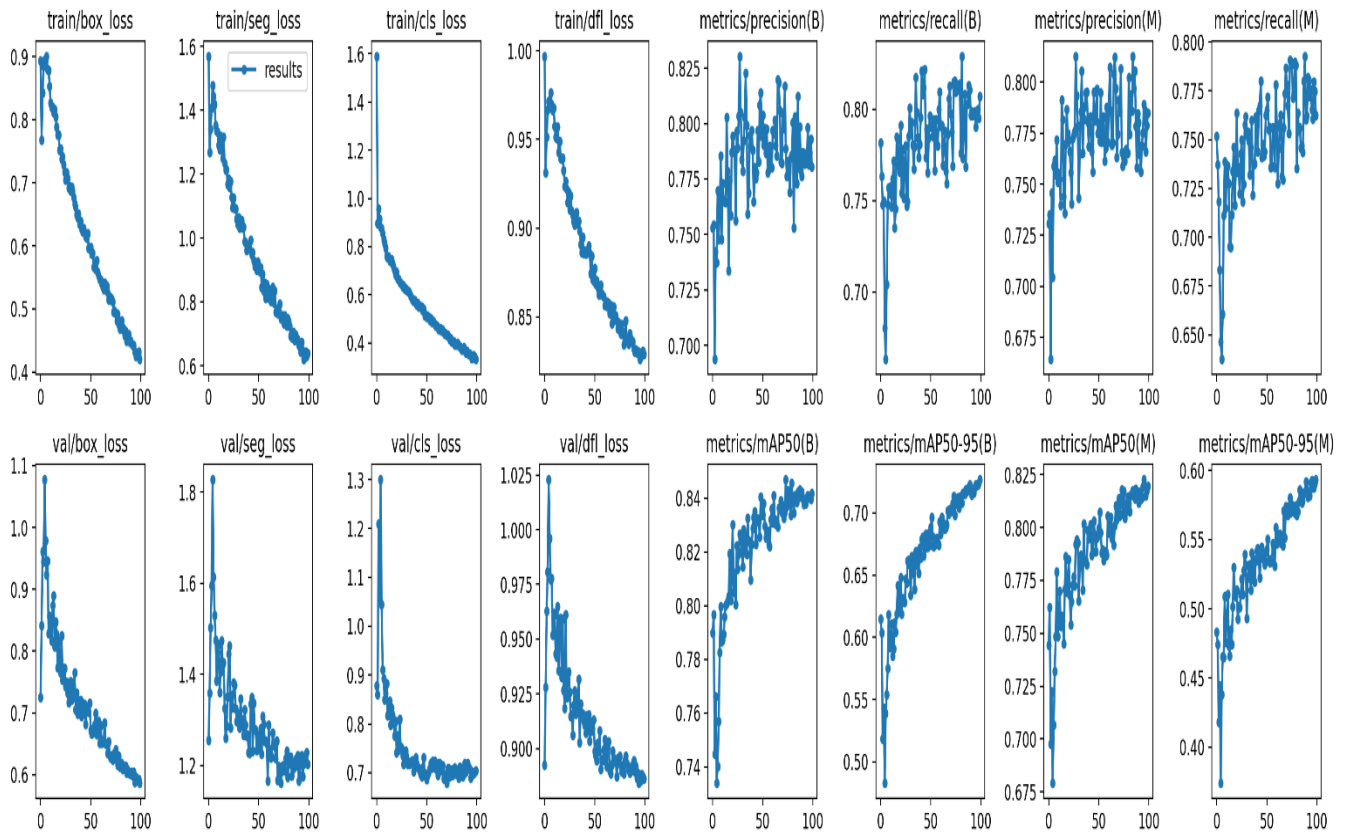


Figure 4.10: depicts the F1 Confidence of the YOLOv4 algorithm



4.9 Summary

This chapter assesses the output of classification and segmentation. The following four performance metrics were determined: recall, accuracy, precision, and f₁-score. Future works and the conclusion will be explained in the upcoming chapter.

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

๑.๑ Overview

This section concludes the ideas discussed in the thesis and makes reference to upcoming projects.

๑.๒ Conclusion

Numerous observations may be made regarding the study's duration, and the following are the conclusions drawn from the data:

- ❖ Deep neural network-based classification systems are can to be the most accurate method available, outperforming other conventional methods in terms of accuracy and loss functions.
- ❖ Transfer learning leverages pre-trained models, reducing the computational burden and allowing for faster training and deployment of models on new tasks.
- ❖ Using CLAHE increased image resolution.
- ❖ Using GlobalAveragePooling^{๒D} in this classification model is an accomplishing spatial summarization, lowering dimensionality, boosting the model's capacity for generalization, and increasing computing efficiency.
- ❖ In classification model, Use SGD instead of the Adam optimizer. Help increase precision and eliminate overfitting.

5.3 Future work

While many issues have been resolved by effectively implementing our suggested techniques for classification and segmentation (person, automobile), our study has encountered other challenges. When we improve the model in the near future, we wish to eliminate all constraints:

- Using Slicing Aided Hyper Inference (SAHI) with YOLO to detect and segment small-sized objects.
- Using CNN for feature extraction and Single Shot MultiBox Detector (SSD) for classification.
- Using mobilenetv3 to extract features and using U-NET for segmentation.

REFERENCES

- [1] R. Ippalapally, S. H. Mudumba, M. Adkay, and N. V. HR, "Object detection using thermal imaging," in 2020 *IEEE 14th India Council International Conference (INDICON)*, 2020: IEEE, pp. 1-6.
- [2] M. Krišto and M. Ivašić-Kos, "Thermal imaging dataset for person detection," in 2019 *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019: IEEE, pp. 1126-1131.
- [3] B. Unhelkar, H. M. Pandey, and G. Raj, *Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2021*. Springer, 2022.
- [4] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE access*, vol. 7, pp. 128837-128868, 2019.
- [5] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem, "Investigations of object detection in images/videos using various deep learning techniques and embedded platforms—A comprehensive review," *Applied sciences*, vol. 10, no. 9, p. 3280, 2020.
- [6] U. Mittal, S. Srivastava, and P. Chawla, "Object detection and classification from thermal images using region based convolutional neural network," *Journal of Computer Science*, vol. 10, no. 7, pp. 961-971, 2019.
- [7] H. Bergenroth, "Use of Thermal Imagery for Robust Moving Object Detection," ed, 2021.
- [8] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [9] E. Valldor, "Person detection in thermal images using deep learning," ed, 2018.
- [10] M. Buric, M. Pobar, and M. Ivasic-Kos, "Ball detection using YOLO and Mask R-CNN," in 2018 *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018: IEEE, pp. 319-323.
- [11] P. Bhavsar, "Object Detection with Convolutional Neural Networks," *Data Driven Investor website*, [Accessed], vol. 18, 2019.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 61, no. 6, pp. 843-910, 2014.

- [13] Y. Cho, N. Bianchi-Berthouze, N. Marquardt, and S. J. Julier, "Deep thermal imaging: Proximate material type recognition in the wild through deep learning of spatial surface temperature patterns," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1-13.
- [14] R. Dorent *et al.*, "CrossMoDA 2021 challenge: Benchmark of cross-modality domain adaptation techniques for vestibular schwannoma and cochlea segmentation," *Medical Image Analysis*, vol. 83, p. 102628, 2022.
- [15] B. Khalid, M. U. Akram, and A. M. Khan, "Multistage deep neural network framework for people detection and localization using fusion of visible and thermal images," in *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4-6, 2020, Proceedings 9*, 2020: Springer, pp. 138-147.
- [16] Z. Kütük and G. Algan, "Semantic segmentation for thermal images: A comparative survey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 286-290.
- [17] A. A. Hania, "Mengenai Artificial Intelligence, Machine Learning, & Deep Learning," *J. Teknol. Indones*, vol. 1, pp. 1-6, 2017.
- [18] N. Awalia, "Identifikasi Penyakit Leaf Mold Pada Daun Tomat Menggunakan Model Densenet Berbasis Transfer Learning," *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, vol. 8, no. 1, pp. 49-52, 2022.
- [19] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12-20, 2020.
- [20] T. F. Kusumaningrum, "Implementasi convolution neural network (CNN) untuk klasifikasi jamur konsumsi di Indonesia menggunakan Keras," 2018.
- [21] R. Firmansyah, "Implementasi deep learning menggunakan convolutional neural network untuk klasifikasi bunga," Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta, 2020.
- [22] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised machine learning: a brief primer," *Behavior Therapy*, vol. 51, no. 5, pp. 670-687, 2020.
- [23] D. Bzdok, M. Krzywinski, and N. Altman, "Machine learning: supervised methods," *Nature methods*, vol. 15, no. 1, p. 5, 2018.
- [24] S. M. Abas, A. M. Abdulazeez, and D. Q. Zeebaree, "A YOLO and convolutional neural network for the detection and classification of

leukocytes in leukemia," *Indones. J. Electr. Eng. Comput. Sci*, vol. 20, no. 1, pp. 200-213, 2022.

- [20] A. R. H. Khayyat, A. A. Al-Moadhen, and M. R. H. Khayyat, "Splicing detection in color image based on deep learning of wavelet decomposed image," in *AIP Conference Proceedings*, 2020, vol. 2290, no. 1: AIP Publishing.
- [21] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in 2018 *Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2018: IEEE, pp. 122-129.
- [22] C. C. Johnson, T. Quackenbush, T. Sorensen, D. Wingate, and M. D. Killpack, "Using first principles for deep learning and model-based control of soft robots," *Frontiers in Robotics and AI*, vol. 8, p. 604398, 2021.
- [23] M. I. Arifin, "Klasifikasi Penyakit pada Orchidaceae Menggunakan Pengolahan Citra dengan Metode Convolution Neural Network (CNN)," Politeknik Perkapalan Negeri Surabaya, 2019.
- [24] M. Taylor, C. Bennett, M. Schoen, and S. Hoque, "Advances in artificial neural networks as a disease prediction tool," *J. Cancer Res. Ther*, vol. 9, pp. 1-11, 2021.
- [25] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning (2016)," *arXiv preprint arXiv:1603.04730*, 2016.
- [26] A. Bazaga, M. Roldan, C. Badosa, C. Jimenez-Mallebrera, and J. M. Porta, "A Convolutional Neural Network for the automatic diagnosis of collagen VI-related muscular dystrophies," *Applied Soft Computing*, vol. 80, p. 105772, 2019.
- [27] S. Shafique and S. Tehsin, "Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks," *Technology in cancer research & treatment*, vol. 17, p. 1033033818802789, 2018.
- [28] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.
- [29] R. Chauhan, K. K. Ghanshala, and R. Joshi, "Convolutional neural network (CNN) for image detection and recognition," in 2018 *first international conference on secure cyber computing and communication (ICSCCC)*, 2018: IEEE, pp. 278-282.

- [35] J. Teuwen and N. Moriakov, "Convolutional neural networks," in *Handbook of medical image computing and computer assisted intervention*: Elsevier, 2020, pp. 481-501.
- [36] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network-a deep learning approach," *Procedia computer science*, vol. 132, pp. 779-788, 2018.
- [37] D. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary intelligence*, vol. 10, no. 1, pp. 1-22, 2022.
- [38] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 101, p. 107398, 2021.
- [39] D.-X. Zhou, "Universality of deep convolutional neural networks," *Applied and computational harmonic analysis*, vol. 48, no. 2, pp. 787-794, 2020.
- [40] J.-Á. Román-Gallego, M.-L. Pérez-Delgado, and S. V. San Gregorio, "Convolutional neural networks used to date photographs," *Electronics*, vol. 11, no. 2, p. 227, 2022.
- [41] P. Dell'Aversana, "Artificial Neural Networks and Deep Learning: A Simple Overview," *A Global Approach to Data Value Maximization. Integration, Machine Learning and Multimodal Analysis*, 2019.
- [42] S.-H. Han, K. W. Kim, S. Kim, and Y. C. Youn, "Artificial neural network: understanding the basic concepts without mathematics," *Dementia and Neurocognitive Disorders*, vol. 17, no. 2, pp. 83-89, 2018.
- [43] M. Susila, B. Irawan, and C. Setianingsih, "Deteksi Penyakit Pada Daun Pakcoy Dengan Pengolahan Citra Menggunakan Metode Convolutional Neural Network," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 9347-9354, 2020.
- [44] Y. Yohannes, S. Devella, and K. Arianto, "Deteksi Penyakit Malaria Menggunakan Convolutional Neural Network Berbasis Saliency," *JUITA: Jurnal Informatika*, vol. 8, no. 1, pp. 37-44, 2020.
- [45] H. A. Pratiwi, M. Cahyanti, and M. Lamsani, "Implementasi Deep Learning Flower Scanner Menggunakan Metode Convolutional Neural Network," *Sebatik*, vol. 20, no. 1, pp. 124-130, 2021.
- [46] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611-629, 2018.

- [47] A. T. R. Dzaky and W. F. Al Maki, "Deteksi Penyakit Tanaman Cabai Menggunakan Metode Convolutional Neural Network," *eProceedings of Engineering*, vol. 8, no. 2, 2021.
- [48] M. Yani, S. Budhi Irawan, Si, MT, and S. Casi Setiningsih, MT, "Application of transfer learning using convolutional neural network method for early detection of terry's nail," in *Journal of Physics: Conference Series*, 2019, vol. 1201, no. 1: IOP Publishing, p. 012052.
- [49] Z. Li, S. H. Wang, R. R. Fan, G. Cao, Y. D. Zhang, and T. Guo, "Teeth category classification via seven- layer deep convolutional neural network with max pooling and global average pooling," *International Journal of Imaging Systems and Technology*, vol. 29, no. 4, pp. 077-083, 2019.
- [50] T.-Y. Hsiao, Y.-C. Chang, H.-H. Chou, and C.-T. Chiu, "Filter-based deep-compression with global average pooling for convolutional networks," *Journal of Systems Architecture*, vol. 90, pp. 9-18, 2019.
- [51] W. Ketwongsa, S. Boonlue, and U. Kokaew, "A new deep learning model for the classification of poisonous and edible mushrooms based on improved alexnet convolutional neural network," *Applied Sciences*, vol. 12, no. 9, p. 3409, 2022.
- [52] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112-119, 2020.
- [53] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 *international conference on engineering and technology (ICET)*, 2017: Ieee, pp. 1-6.
- [54] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," in 2017 *international conference on communication and signal processing (ICCSP)*, 2017: IEEE, pp. 0888-0892.
- [55] S. Tiwari, "Activation functions in neural networks," *geeksforgeeks.org*, 2020.
- [56] S. Jadon, "Introduction to different activation functions for deep learning," *Medium, Augmenting Humanity*, vol. 16, p. 140, 2018.
- [57] X. Jiang, Y. Pang, X. Li, J. Pan, and Y. Xie, "Deep neural networks with elastic rectified linear units for object recognition," *Neurocomputing*, vol. 270, pp. 1132-1139, 2018.
- [58] A. F. Agarap, "Deep learning using rectified linear units (relu). arXiv 2018," *arXiv preprint arXiv:1803.08375*, 1803.

- [59] S. Frei, G. Vardi, P. L. Bartlett, N. Srebro, and W. Hu, "Implicit bias in leaky relu networks trained on high-dimensional data," *arXiv preprint arXiv:2210.07082*, 2022.
- [60] Z. Qiumei, T. Dan, and W. Fenghua, "Improved convolutional neural network based on fast exponentially linear unit activation function," *Ieee Access*, vol. 7, pp. 101309-101317, 2019.
- [61] D. S. Putra, M. Azmi, and W. Purwanto, "ANN Activation Function Comparative Study for Sinusoidal Data," in *Journal of Physics: Conference Series*, 2022, vol. 2406, no. 1: IOP Publishing, p. 012029.
- [62] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.
- [63] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310-316, 2017.
- [64] R. Zaheer and H. Shaziya, "A study of the optimization algorithms in deep learning," in 2019 *third international conference on inventive systems and control (ICISC)*, 2019: IEEE, pp. 036-039.
- [65] L. Zhang, H. J. Gao, J. Zhang, and B. Badami, "Optimization of the convolutional neural networks for automatic detection of skin cancer," *Open Medicine*, vol. 10, no. 1, pp. 27-37, 2020.
- [66] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia computer science*, vol. 131, pp. 977-984, 2018.
- [67] Y.-D. Zhang, C. Pan, X. Chen, and F. Wang, "Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling," *Journal of computational science*, vol. 27, pp. 07-18, 2018.
- [68] X. Chen, S. Liu, R. Sun, and M. Hong, "On the convergence of a class of adam-type algorithms for non-convex optimization," *arXiv preprint arXiv:1808.02941*, 2018.
- [69] A. H. Ali, M. G. Yaseen, M. Aljanabi, and S. A. Abed, "Transfer Learning: A New Promising Techniques," *Mesopotamian Journal of Big Data*, vol. 2022, pp. 31-32, 2022.

- [٧٠] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV٢ network," *Procedia computer science*, vol. ١٩٧, pp. ١٩٨-٢٠٧, ٢٠٢٢.
- [٧١] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (٢٠١٠-٢٠٢٠)," *IEEE Transactions on Artificial Intelligence*, vol. ١, no. ٢, pp. ١٥١-١٦٦, ٢٠٢٠.
- [٧٢] W. Hastomo, "Convolution Neural Network Arsitektur Mobilenet-V٢ Untuk Mendeteksi Tumor Otak," in *Prosiding Seminar SeNTIK*, ٢٠٢١, vol. ٥, no. ١, pp. ١٧-٢١.
- [٧٣] R. O. Ekoputris, "MobileNet: Deteksi Objek pada Platform Mobile," *Medium, May*, vol. ٩, ٢٠١٨.
- [٧٤] E. I. Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*, vol. ٥, no. ٢, pp. ٨١-٩١, ٢٠٢١.
- [٧٥] H. Hendriyana and Y. H. Maulana, "Identification of Types of Wood using Convolutional Neural Network with Mobilenet Architecture," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. ٤, no. ١, pp. ٧٠-٧٦, ٢٠٢٠.
- [٧٦] H. F. Jessar, A. T. Wibowo, and E. Rachmawati, "Klasifikasi Genus Tanaman Sukulen Menggunakan Convolutional Neural Network," *eProceedings of Engineering*, vol. ٨, no. ٢, ٢٠٢١.
- [٧٧] F. E. Ramadhan, "Penerapan image classification dengan pre-trained model mobilenet dalam client-side machine learning," *Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah ...*, ٢٠٢٠.
- [٧٨] R. Tadjudin and D. Rosmala, "IMPLEMENTASI MOBILENETV٢ DAN FRAME DIFFERENCE UNTUK PENENTUAN KECEPATAN KENDARAAN," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. ٧, no. ٢, pp. ١٩٣-٢٠٤, ٢٠٢١.
- [٧٩] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv٢: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, ٢٠١٨, pp. ٤٥١٠-٤٥٢٠.
- [٨٠] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, "Fundus image classification using VGG-١٩ architecture with PCA and SVD," *Symmetry*, vol. ١١, no. ١, p. ١, ٢٠١٨.

- [81] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease," *AgriEngineering*, vol. 4, no. 4, pp. 871-887, 2022.
- [82] A. I. Károly, P. Galambos, J. Kuti, and I. J. Rudas, "Deep learning in robotics: Survey on model structures and training strategies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 266-279, 2021.
- [83] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261-318, 2020.
- [84] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.
- [85] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agronomy* 2022, 12, 319," ed: s Note: MDPI stays neutral with regard to jurisdictional claims in published ..., 2022.
- [86] H. Le, M. Nguyen, W. Q. Yan, and H. Nguyen, "Augmented reality and machine learning incorporation using YOLOv3 and ARKit," *Applied Sciences*, vol. 11, no. 13, p. 6006, 2021.
- [87] D. Dima *et al.*, "Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100," *International Journal of Computer Vision*, vol. 130, no. 1, pp. 33-55, 2022.
- [88] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv 2023," *arXiv preprint arXiv:2304.00001*.
- [89] U. YOLOv8, "Available online: <https://docs.ultralytics.com/> (accessed on 21 June 2023).".
- [90] Y. Li, Y. Ma, and Y. Long, "Protocol for assessing neighborhood physical disorder using the YOLOv8 deep learning model," *STAR protocols*, vol. 5, no. 1, p. 102778, 2024.
- [91] P. Musa, F. Al Rafi, and M. Lamsani, "A Review: Contrast-Limited Adaptive Histogram Equalization (CLAHE) methods to help the application of face recognition," in 2018 *third international conference on informatics and computing (ICIC)*, 2018: IEEE, pp. 1-6.
- [92] K. A. Dar and S. Mittal, "An enhanced adaptive histogram equalization based local contrast preserving technique for HDR images," in *IOP Conference*

Series: Materials Science and Engineering, 2021, vol. 1022, no. 1: IOP Publishing, p. 012119.

- [93] Ž. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 099-106, 2021.
- [94] C. Zhou *et al.*, "Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision," *Aquaculture*, vol. 507, pp. 457-460, 2019.
- [95] H. R. Aradhya, "Object detection and tracking using deep learning and artificial intelligence for video surveillance applications," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.
- [96] G. Li, Z. Song, and Q. Fu, "A new method of image detection for small datasets under the framework of YOLO network," in 2018 *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2018: IEEE, pp. 1031-1035.
- [97] M. Lokanath, K. S. Kumar, and E. S. Keerthi, "Accurate object classification and detection by faster-RCNN," in *IOP conference series: materials science and engineering*, 2017, vol. 263, no. 0: IOP Publishing, p. 052028.
- [98] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, p. 101, 2023.
- [99] C. D. Rodin, L. N. de Lima, F. A. de Alcantara Andrade, D. B. Haddad, T. A. Johansen, and R. Storvold, "Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems," in 2018 *International Joint Conference on Neural Networks (IJCNN)*, 2018: IEEE, pp. 1-8.
- [100] Y. Nam and Y.-C. Nam, "Vehicle classification based on images from visible light and thermal cameras," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1-9, 2018.
- [101] M. Bhattarai and M. Martinez-Ramon, "A deep learning framework for detection of targets in thermal images to improve firefighting," *IEEE Access*, vol. 8, pp. 88308-88321, 2020.
- [102] M. Ivašić-Kos, M. Krišto, and M. Pobar, "Human detection in thermal imaging using YOLO," in *Proceedings of the 2019 0th International Conference on Computer and Technology Applications*, 2019, pp. 20-24.

- [103] H. Zhang, X.-g. Hong, and L. Zhu, "Detecting small objects in thermal images using single-shot detector," *Automatic Control and Computer Sciences*, vol. 55, no. 2, pp. 202-211, 2021.

الخلاصة

تلعب الصور الحرارية دورًا حاسمًا في اكتشاف الأشياء لمختلف التطبيقات، بما في ذلك المراقبة والأمن والأتمتة الصناعية وملاحة المركبات، حيث يمكنها التقاط التوقعات الحرارية. ومع ذلك، فإن التحديات مثل التباين المنخفض، والأنماط الحرارية المتقلبة، وغياب مجموعات البيانات الحرارية المحددة تتطلب خوارزميات متخصصة لتحديد الكائن وموقعه بدقة. يستكشف هذا البحث مجال اكتشاف الأشياء في الصور الحرارية، وهو جانب حيوي لتطبيقات مثل المراقبة والأنظمة المستقلة.

استخدمت هذه الأطروحة منهج معادلة الرسم البياني التكيفي المحدود للتباين (CLAHE) في معالجة الصور لتحسين التباين. تحاول مرحلة ما قبل المعالجة هذه تحسين أداء خوارزميات التعرف على الكائنات اللاحقة من خلال معالجة المشكلات المتعلقة بالتباين في الصور الحرارية

القسم الأول من هذه الدراسة هو تحديد نموذج الشبكة العصبية التلافيفية الذي يمكن استخدامه في حالات التصنيف. باستخدام ٢,٩٥٦ صورة حرارية، قامت الدراسة بتقييم الدقة والدقة ودرجة F1 واستدعاء بنيت V2 MobileNet و V19 VGG.

أظهرت النتائج أن دقة V2 MobileNet بلغت ٩٧% ودرجة F1 ٩٦.٨%، في حين بلغت دقة V19 VGG ٩٧% ودرجة F1 ٩٦.٦%. تتوفر هذه الدراسة رؤى قيمة حول تطبيق طرق تعلم النقل لتصنيف الصور الحرارية وتوفير نظرة شاملة عن فعالية الخوارزميات المختلفة في مواجهة التحديات التي تطرحها ظروف التصوير الحراري.

في القسم الثاني من هذه الدراسة، نستفيد من نهج YOLOv8، الذي تم الاعتراف به على نطاق واسع للكشف الدقيق عن الكائنات في الوقت الفعلي، مما يعزز أداء الكشف في منطقة الصور الحرارية الصعبة. تم تدريب هذه التقنية باستخدام مجموعة بيانات مختارة تحتوي على ١,٨٩٨ صورة، تم استخدام ١٤٤٣ منها للتدريب، و ٣٨٠ للتحقق من الصحة، و ٧٥ للاختبار. استخدمنا هذا النهج لتقسيم الكائنات في الصور الحرارية. يُظهر متوسط دقة الخوارزمية (mAP٥٠) البالغة ٨٢% مدى فعاليتها. تؤكد هذه النتائج أن YOLOv8 مناسب

لتطبيقات التصوير الحراري، وهي مناطق جديدة تتطلب اكتشافاً دقيقاً للأشياء في ظل ظروف شديدة الحرارة.



جامعة كربلاء
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

تجزئة الصور الحرارية والتعرف عليها باستخدام تقنيات التعلم العميق

رسالة ماجستير
مقدمة إلى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء وهي جزء من
متطلبات نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة

حيدر علي مفتن

بإشراف

الاستاذ المساعد الدكتور علي رضا حسون