Evaluation of Neural Object Detection Models for Human Detection in Infrared Images

PROJECT REPORT T1000

from the course of studies Computer Science - Artificial Intelligence

at the Cooperative State University Baden-Württemberg
Ravensburg Campus Friedrichshafen

by

Lukas Florian Richter

11.08.2025

Completion Time:	16 Wochen
Student ID, Course:	None, TIK24
Company:	Airbus Defence & Space, Taufkirchen
Supervisor in the Company:	René Loeneke

Declaration of Authorship

In accordance with clause 1.1.13 of Annex 1 to §§ 3, 4 and 5 of the Cooperative State University Baden-Württemberg's Study and Examination Regulations for Bachelor's degree programs in the field of Technology, dated 29.09.2017. I hereby declare that I have written my thesis on the topic:

Evaluation of Neural Object Detection Models for Human Detection in Infrared Images

independently and have used no other sources or aids than those specified. I further declare that all submitted versions are identical.

Taufkirchen 11.08.2025

Lukas Florian Richter

Abstract

This project report evaluates the performance of neural object detection models for detecting humans in infrared images. The study focuses on comparing different variations of the SSD (Single Shot Mutlibox Detector) model architecture, assessing their accuracy and inference speed, and identifying the most suitable model for the given task. Additionally, different preprocessing techniques are evaluated to improve the detection performance.

More specifically, the main contributions of this project are:

- Conceptualization of a simple and cost-efficient hardware setup for the purpose of on-premise human detection in infrared images
- Evaluation of different SSD model architectures
- Comparison between different preprocessing techniques
- Identification of the most suitable model for the given task
- A theoretical pipeline for the secure transmission of the detection results to a remote server

Table of Contents

List of Figures

List of Tables

Code Snippets

List of Acronyms

AI	Artificial Intelligence
AP	Average Precision
API	Application Programming Interface
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DL	Deep Learning
FC	Fully Connected
FN	False Negative
FP	False Positive
FP16	16-bit floating point
FP32	32-bit floating point
FPS	Frames per Second
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HNM	Hard-Negative Mining
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
INT8	8-bit integer
IR	Infrared
IoU	Intersection over Union

ML	Machine Learning
MPS	Metal Performance Shaders
NMS	Non-Maximum Suppression
NN	Neural Network
OSI	Open Systems Interconnection
R-CNN	Region-based Convolutional Neural Network
RAM	Random Access Memory
REST	Representational State Transfer
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
ReLU	Rectified Linear Unit
ResNet	Residual Network
SGD	Stochastic Gradient Descent
SSD	Single Shot MultiBox Detector
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
TPU	Tensor Processing Unit
VGG	Visual Geometry Group
VOC	Visual Object Classes
ViT	Vision Transformer
YOLO	You Only Look Once
mAP	mean Average Precision

Glossary

Batch	A batch is a group of data processed together as a unit.
Batch Gradient Descent	Batch Gradient Descent is an optimization algorithm used to minimize the loss function in machine learning models by iteratively updating the model parameters based on their gradients with respect to the
Exploit	An exploit is a method or piece of code that takes advantage of vulnerabilities in software, applications, networks, operating systems, or hardware, typically for malicious purposes.
Patch	A patch is data that is intended to be used to modify an existing software resource such as a program or a file, often to fix bugs and security vulnerabilities.
Stochastic Gradient Descent	Stochastic Gradient Descent (SGD) is an optimization algorithm used to minimize the loss function in machine learning models by iteratively updating the model parameters based on their partial derivatives with
Vulnerability	A Vulnerability is a flaw in a computer system that weakens the overall security of the system.

1 Introduction

The increasing demand for automated surveillance systems in security-critical applications has driven significant advances in computer vision technologies. While conventional RGB-based surveillance systems remain prevalent, they face inherent limitations in challenging environmental conditions such as low-light scenarios, adverse weather, and nighttime operations. Thermal infrared imaging presents a compelling alternative, offering consistent detection capabilities independent of ambient lighting conditions and providing unique advantages for human detection through body heat signatures.

Key areas to develop:

- Context: Growing need for reliable 24/7 surveillance systems
- Problem: Limitations of RGB cameras in challenging conditions
- Solution: Advantages of thermal imaging for human detection
- Research gap: Need for optimized neural networks for thermal imagery
- Project scope: Evaluation of SSD models for thermal human detection
- Thesis structure: Overview of methodology and contributions
- Industrial relevance: Partnership with Airbus Defence & Space

2 Literature Review and Theoretical Background

The field of object detection has undergone significant evolution from traditional computer vision techniques to sophisticated deep learning architectures. Understanding this progression is essential for contextualizing the current work's contribution to thermal image analysis. This section examines the theoretical foundations of object detection, with particular emphasis on the Single Shot MultiBox Detector (SSD) architecture and its applicability to thermal imagery processing challenges.

Key areas to develop:

- Evolution from traditional methods (HOG, SIFT) to deep learning
- Comparison of one-stage vs. two-stage detection models
- SSD architecture fundamentals and anchor box mechanisms
- Backbone network analysis (VGG vs. ResNet trade-offs)
- Thermal imaging characteristics and preprocessing challenges
- Existing work on infrared human detection
- Gap analysis: Limited research on SSD for thermal surveillance

2.1 Object Detection Fundamentals

Most object detection methods can be broadly categorized into two main approaches: traditional methods and deep learning-based methods. Traditional methods mainly rely on handcrafted features and sliding window techniques, while deep learning-based methods leverage convolutional neural networks (CNNs) or vision transformer (ViT) architectures to automatically learn features from data.

2.1.1 Traditional Object Detection Methods

Examines feature-based and sliding window techniques, such as Haar-like features and HOG descriptors.

2.1.2 Deep Learning-Based Object Detection

Discusses the evolution of deep learning models, including R-CNN, Fast R-CNN, Faster R-CNN, and YOLO, highlighting their strengths and limitations.

2.2 Single Shot MultiBox Detector (SSD) Architecture

Detailed explanation of SSD model architecture, including backbone networks (VGG, ResNet) and detection mechanisms.

2.2.1 Backbone Networks for Feature Extraction

Explores the role of backbone networks (VGG, ResNet) in feature extraction and their impact on SSD performance.

2.2.2 Feature Maps and Anchor Boxes

Describes the multi-scale feature maps and anchor boxes used in SSD for object detection.

2.2.3 MultiBox Loss Function

Explains the MultiBox loss function that combines localization loss and confidence loss for training SSD models.

Non-Maximum Suppression (NMS):

Examines the NMS technique used to filter duplicate detections and improve detection accuracy.

2.3 Thermal Image Processing

Discusses characteristics of thermal images, preprocessing techniques (inversion, edge enhancement), and challenges specific to infrared imagery.

3 Methodology

This study employs a systematic experimental approach to evaluate the effectiveness of SSD-based neural networks for human detection in thermal imagery. The methodology encompasses dataset selection and preparation, implementation of multiple model variants with different backbone architectures, application of thermal-specific preprocessing techniques, and comprehensive evaluation metrics. The experimental design ensures reproducible results while addressing the unique challenges posed by infrared image characteristics.

Key areas to develop:

- Dataset description: FLIR ADAS v2, AAU-PD-T, OSU-T, M3FD, KAIST-CVPR15
- Model configurations: SSD300-VGG16 vs. SSD300-ResNet152
- Training setup: Pretrained vs. scratch initialization strategies
- Preprocessing techniques: Image inversion and edge enhancement
- Data augmentation and split strategies (train/validation/test)
- Evaluation metrics: mAP, precision, recall, inference speed
- Hardware setup and computational requirements
- Statistical significance testing approach

3.1 Dataset Description

Details the thermal image datasets (FLIR ADAS v2, AAU-PD-T, OSU-T, M3FD, KAIST-CVPR15) and their characteristics.

3.2 Model Implementation

Explains the implementation of SSD models with different backbones and preprocessing configurations.

3.3 Experimental Design

Outlines the systematic approach to comparing model variants and the evaluation framework.

4 Results and Analysis

The experimental evaluation reveals significant performance variations across different model configurations and preprocessing approaches when applied to thermal human detection tasks. This section presents comprehensive results from training 16 distinct model variants, combining backbone architectures (VGG16 vs. ResNet152), initialization strategies (pretrained vs. scratch), and preprocessing techniques (none, inversion, edge enhancement, combined). The analysis demonstrates clear patterns in model behavior and identifies optimal configurations for thermal surveillance applications.

Key areas to develop:

- Training convergence analysis: Loss curves and stability patterns
- Detection accuracy results: mAP scores across all model variants
- Preprocessing impact: Quantitative comparison of enhancement techniques
- Backbone architecture comparison: VGG16 vs. ResNet152 performance
- Initialization strategy effects: Pretrained vs. scratch training outcomes
- Computational efficiency: Inference speed and memory requirements
- Dataset-specific performance: Results breakdown by thermal dataset
- Error analysis: Common failure cases and detection limitations

4.1 Training Performance

Reports training loss curves, converegence behavior, and computational requirements for different mdoel variants.

4.2 Detection Accuracy Analysis

Provides detailed mAP scores and detection performance metrics for each model configuraiton and preprocessing technique.

4.3 Preprocessing Impact Evaluation

Analzyes the effects of image inversion and edge enhancement on detection performance.

5 Discussion

The experimental results provide valuable insights into the practical applicability of SSD architectures for thermal human detection systems. While certain configurations demonstrate superior performance, the choice of optimal model depends on specific deployment requirements, including accuracy thresholds, computational constraints, and operational environments. This section interprets the findings within the context of real-world surveillance applications and addresses the broader implications for thermal imaging-based security systems.

Key areas to develop:

- Performance trade-offs: Accuracy vs. computational efficiency analysis
- Preprocessing effectiveness: When and why certain techniques work better
- Backbone selection criteria: Situational advantages of VGG16 vs. ResNet152
- Real-world deployment implications: Edge computing considerations
- Limitations and constraints: Environmental factors affecting performance
- Comparison with existing thermal detection systems
- Cost-benefit analysis for industrial implementation
- Future optimization potential and research directions

5.1 Model Performance Comparison

Compares SSD-VGG and SSD-ResNer performance and discusses trade-offs between accuracy and computational efficiency.

5.2 Practical Deployment Considerations

Discusses real-world application scenarios and system requirements for thermal surveillance.

6 Conclusion and Future Work

This thesis has systematically evaluated the application of Single Shot MultiBox Detector architectures for human detection in thermal imagery, providing empirical evidence for optimal model configurations in surveillance applications. The comprehensive analysis of 16 model variants across multiple thermal datasets has yielded practical insights for deploying neural networks in infrared-based security systems. The findings contribute to both academic understanding and industrial implementation of thermal computer vision technologies.

Key areas to develop:

- Key findings summary: Best-performing model configurations identified
- Methodological contributions: Systematic evaluation framework for thermal detection
- Practical implications: Guidelines for industrial thermal surveillance deployment
- Technical achievements: Successful adaptation of RGB models to thermal domain
- Research limitations: Dataset constraints and environmental factors
- Future research directions: Advanced architectures and multi-modal approaches
- Industry impact: Potential applications beyond security surveillance
- Recommendations: Implementation guidelines for practitioners

7 Examples

Just a couple of examples to demonstrate proper use of the typst template and its functions.

7.1 Acronyms

Use the `acr` function to insert acronyms, which looks like this Hypertext Transfer Protocol (HTTP).

Application Programming Interfaces are used to define the interaction between different software systems.

REST is an architectural style for networked applications.

7.2 Glossary

Use the `gls` function to insert glossary terms, which looks like this:

A Vulnerability is a weakness in a system that can be exploited.

7.3 Lists

Create bullet lists or numbered lists.

- This
- is a
- bullet list

1. It also
2. works with
3. numbered lists!

7.4 Figures and Tables

Create figures or tables like this:

7.4.1 Figures

Figure 1 — Image Example

7.4.2 Tables

	Area	**Parameters**
cylinder.svg	$$\pi h \frac{D^2 - d^2}{4} \quad (1)$$	h: height D: outer radius d: inner radius
tetrahedron.svg	$$\frac{\sqrt{2}}{12} a^3 \quad (2)$$	a: edge length

Table 1 — Table Example

7.5 Code Snippets

Insert code snippets like this:

```
const ReactComponent = () => {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
};

export default ReactComponent;
```

Listing 1 — Codeblock Example

7.6 References

Cite like this K. R. Akshatha, A. K. Karunakar, S. B. Shenoy, A. K. Pai, N. H. Nagaraj, and S. S. Rohatgi [1]. Or like this [2].

You can also reference by adding `<ref>` with the desired name after figures or headings.

For example this Table 1 references the table on the previous page.

8 Conclusion

References

[1] K. R. Akshatha, A. K. Karunakar, S. B. Shenoy, A. K. Pai, N. H. Nagaraj, and S. S. Rohatgi, "Human Detection in Aerial Thermal Images Using Faster R-CNN and SSD Algorithms," *Electronics*, vol. 11, no. 7, p. 1151, Jan. 2022, doi: 10.3390/electronics11071151.

[2] M. A. Farooq, P. Corcoran, C. Rotariu, and W. Shariff, "Object Detection in Thermal Spectrum for Advanced Driver-Assistance Systems (ADAS)," no. arXiv:2109.09854. arXiv, Oct. 2021. doi: 10.48550/arXiv.2109.09854.