

**PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA DA COMPUTAÇÃO**



**Critical Object Detection and Visual Question Answering
in Natural Disaster Scenes: A Recall-Focused Framework**

Marina Chagas Bulach Gapski

INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS
RIO CLARO

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Marina Chagas Bulach Gapski

Critical Object Detection and Visual Question Answering in Natural Disaster Scenes: A Recall-Focused Framework

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Daniel Carlos Guimarães Pedronette

Co-orientador: Mohand Said Allili

Rio Claro - SP

2026

G211c Gapski, Marina Chagas Bulach
Critical object detection and visual question answering in natural
disaster scenes: a recall-focused framework / Marina Chagas Bulach
Gapski. -- , 2026
118 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),
Instituto de Geociências e Ciências Exatas, Rio Claro,
Orientador: Daniel Carlos Guimarães Pedronette
Coorientador: Mohand Said Allili

1. Aprendizado de máquina. 2. Detecção de objetos. 3. Inteligência
artificial. I. Título.

Impacto potencial desta pesquisa: Alinhada ao ODS 13, a presente pesquisa apresenta a versão inicial de um *framework* que se propõe a facilitar a identificação de situações críticas em cenários de desastres naturais. O *framework* proposto tem potencial para auxiliar equipes de resgate a localizarem rapidamente situações de risco em cenários de enchentes ou fortes tempestades.

Potential impact of this research: Aligned with SDG 13, the present research introduces the initial version of a framework designed to facilitate the identification of critical situations in natural disaster scenarios. The proposed framework has the potential to assist rescue teams in rapidly locating risk situations in flood or severe storm scenarios.

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Marina Chagas Bulach Gapski

Critical Object Detection and Visual Question Answering in Natural Disaster Scenes: A Recall-Focused Framework

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Comissão Examinadora

Prof. Dr. DANIEL CARLOS GUIMARÃES PEDRONETTE
IGCE / UNESP / Rio Claro (SP)

Profa. Dra. LEO SAMPAIO FERRAZ RIBEIRO
ICMC / USP / São Carlos (SP)

Prof. Dr. FABRICIO APARECIDO BREVE
IGCE / UNESP / Rio Claro (SP)

Conceito: Aprovado.

Rio Claro (SP), 27 de abril de 2026.

Acknowledgement

This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2023/12736-3. The opinions, hypotheses, conclusions, or recommendations expressed in this material are the responsibility of the author(s) and do not necessarily reflect the views of FAPESP.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

The author is deeply grateful to Professor Mohand Said Allili, for his unwavering support, assistance, and availability throughout the entire work.

“We can only see a short distance ahead, but we can see plenty there that needs to be done. ”
(Alan Turing) ([TURING](#), 1950)

Resumo

As mudanças climáticas têm intensificado progressivamente a frequência e a severidade dos desastres naturais em escala global. Nesse contexto, assegurar que as equipes de resgate estejam adequadamente preparadas e aptas para uma resposta rápida é crucial para a minimização do número de vítimas. Concomitantemente, os rápidos avanços na área de visão computacional, em particular, algoritmos de detecção de objetos como o YOLO e mecanismos baseados em atenção, e no processamento de linguagem natural, impulsionados pelos grandes modelos de linguagem, oferecem ferramentas sem precedentes para a análise de cenas em tempo real. A integração dessas tecnologias multimodais apresenta um potencial significativo para fornecer inteligência rápida, otimizada e acionável, auxiliando as equipes de primeira resposta durante operações de emergência e prevenindo o envio de profissionais a ambientes não previamente avaliados. Baseando-se nesse potencial tecnológico, este trabalho propõe um *framework* que integra múltiplos modelos de detecção de objetos a um modelo de linguagem para consultar e analisar situações de desastre. Um aspecto central dessa abordagem é a identificação e a priorização de “objetos críticos”. Neste estudo, os objetos críticos são explicitamente definidos como veículos submersos e estruturas residenciais severamente danificadas, os quais representam locais com alta probabilidade de abrigar vítimas retidas. Negligenciar tais locais pode comprometer severamente os resultados das operações de resgate; por conseguinte, nossa metodologia prioriza a minimização de falsos negativos. A avaliação do *framework* foi conduzida utilizando conjuntos de dados de domínio público da plataforma Roboflow, adaptados para atender aos nossos requisitos experimentais, avaliando sua capacidade de recuperar e ranquear efetivamente esses objetos críticos. Em comparação a um modelo de referência (**baseline**) que utiliza um único algoritmo de detecção de objetos, a abordagem multimodelo proposta demonstrou desempenho superior, alcançando maior revocação (*recall*) e ranqueando os objetos críticos de forma eficaz com base em seus valores de pontuação de confiança. Embora não tenha sido testado em cenários reais de desastres, estes resultados preliminares ressaltam o forte potencial do *framework* como uma futura ferramenta assistiva de triagem para operações de resposta a emergências.

Palavras-chave: aprendizado de máquina; detecção de objetos; inteligência artificial.

Abstract

Climate change has increasingly intensified the frequency and severity of natural disasters worldwide. In this context, ensuring that rescue teams are adequately prepared and capable of rapid response is critical to minimizing casualties. Concurrently, rapid advancements in Computer Vision (CV), particularly, object detection algorithms like YOLO and attention-based mechanisms—and Natural Language Processing (NLP), driven by Large Language Models (LLMs), offer unprecedented tools for real-time scene analysis. The integration of these multimodal technologies presents significant potential to provide fast, lightweight, and actionable intelligence to assist first responders during emergency operations, preventing the deployment of personnel into unassessed environments. Leveraging this technological potential, this work proposes a framework that integrates multiple object detection models with an LLM to query and analyze disaster situations. A core aspect of this approach is the identification and prioritization of “critical objects”. In this study, critical objects are explicitly defined as submerged vehicles and heavily damaged residential structures, which represent highly probable locations of trapped victims. Overlooking these locations can severely impact rescue outcomes; therefore, our methodology prioritizes the minimization of false negatives. The framework’s evaluation was conducted using publicly available datasets from Roboflow, customized to suit our experimental requirements, assessing its ability to effectively retrieve and rank these critical objects. Compared to a baseline utilizing a single object detection model, the proposed multi-model approach demonstrated superior performance, achieving higher recall and effectively ranking critical objects by confidence score values. While untested in real-world disaster deployments, these preliminary results underscore the framework’s strong potential as a future assistive triage tool for emergency response operations.

Keywords: Machine learning; object detection; artificial intelligence.

List of abbreviations and acronyms

AI: *Artificial Intelligence*

AP: *Average Precision*

BERT: *Bidirectional Encoder Representations for Transformers*

BERT: *Bidirectional Encoder Representations from Transformers*

BOW: *Bag of Words*

CNN: *Convolutional Neural Network*

CV: *Computer Vision*

DeBERTa: *Decoding-enhanced BERT with Disentangled Attention*

DETR: *Detection Transformer*

DeVANet: *DeBERTa Vision Attention Network*

DWD: *Default Water Depth*

HAND: *Heightabove-Nearest-Drainage*

IDW: *Inverse-Distance-Weighting*

IoU: *Intersection-Over-Union*

LDD: *Local Drainage Directions*

LLM: *Large Language Model*

M-ResNet: *Modified ResNet*

MLA: *Multi-head Latent Attention*

MoE: *Mixture-of-Experts*

NLP: *Natural Language Processing*

NMS: *Non-Maximum Suppression*

NMW: *Non-Maximum Weighted*

ODM: *Object Detection Module*

POI: *Point of Interest*

QA: *Question Answering*

QAM: *Question Answering Module*

R-CNN: *Regions with CNN Features*

ReLU: *Rectified Linear Unit*

ResNet: *Residual Network*

RGB: *Red Green Blue*

ROI: *Region of Interest*

Soft-NMS: *Soft Non-Maximum Suppression*

SVM: *Support Vector Machines*

TF - IDF: *Term Frequency–Inverse Document Frequency*

UAV: *Unmanned Aerial Vehicles*

UQO: *University of Québec at Outonais*

VQA: *Visual Question Answering*

WBF: *Weighted Boxes Fusion*

YOLO: *You Only Look Once*

List of Figures

Figure 1 – Illustration of CNN main operations (LIU et al., 2020).	23
Figure 2 – Illustration of R-CNN main steps (GIRSHICK et al., 2014).	25
Figure 3 – Structure of G-CNN regression network, originally from the “G-CNN: An Iterative Grid Based Object Detector” (NAJIBI; RASTEGARI; DAVIS, 2016). 26	
Figure 4 – Object detection in different lightning conditions (PRABU; GNANASEKAR, 2023).	27
Figure 5 – Narrative extraction pipeline (SANTANA et al., 2023).	28
Figure 6 – Transformer Architecture (VASWANI et al., 2017)	33
Figure 7 – Illustration of pre-training and fine-tuning for BERT (DEVLIN et al., 2019) 34	
Figure 8 – Illustration of the commonsense knowledge transfer framework (ZHOU; BRAS; CHOI, 2023).	37
Figure 9 – Detection results from “Disaster damage investigation using artificial intelligence and drone mapping” (KIM et al., 2022) paper.	39
Figure 10 – Examples of constructions with different levels of damage (CHENG; BEHZADAN; NOSHADRAVAN, 2021).	40
Figure 11 – Illustration of method proposed in (KIM et al., 2022).	42
Figure 12 – Illustration from DisasterVQA Dataset (AL-MOHANNADI et al., 2026) . . 43	
Figure 13 – Illustration of YOLO algorithm: input image is divided into $S \times S$ grid, and each grid cell predicts B bounding boxes and confidence scores (REDMON et al., 2016a).	47
Figure 14 – Illustration of DeepSeek-V3 architecture, presented in (DEEPSEEK-AI et al., 2025).	49
Figure 15 – General view of the proposed framework.	50
Figure 16 – Illustration of a hypothetical use of the proposed framework.	52
Figure 17 – Image (SEDANFROMHX, 2024) with only class 1 labeled (used to train Only L1 model)	54
Figure 18 – Image (SEDANFROMHX, 2024) with only class 2 labeled (used to train Only L2 model)	54
Figure 19 – Image (SEDANFROMHX, 2024) with only class 3 labeled (used to train Only L3 model)	54
Figure 20 – Image (SEDANFROMHX, 2024) with all classes labeled (used to train Mixed General model)	54
Figure 21 – Number of existent objects found by each model exclusively on cars dataset . 55	
Figure 22 – Number of existent objects found by each model exclusively on houses dataset . 55	
Figure 23 – Representative illustration of the construction of combinations.	57

Figure 24 – Images from “convert2yolov5 Computer Vision Dataset” (SEDANFROMHX, 2023) dataset. Figure presents examples of objects of class 0 (L0) and class 3(L3)	61
Figure 25 – Examples od dataset “yolov5WithSUV Computer Vision Model” (SEDANFROMHX, 2024). Figure presents examples of objects from classes 0 (L0), class 1 (L1), class 2 (L2) and class 3 (L3).	62
Figure 26 – Images from “Building Damage Detection-undef Dataset” (ROBOFLOW, 2024). The images presents objects belonging to classes 0 (not destroyed), 1 (slightly destroyed), 2 (moderately destroyed), and 3 (heavily destroyed). . .	63
Figure 27 – Confidence comparsion between Mixed General and specialized models, on cars dataset	65
Figure 28 – Confidence comparsion between Mixed General and specialized models, on houses dataset	66
Figure 29 – Confusion matrices of specialized models for cars datasets	68
Figure 30 – Confusion matrix of Mixed General model on cars dataset	68
Figure 31 – Classification results of first 300 elements on cars dataset ranking.	70
Figure 32 – Recall evolution of models per rank position on cars dataset , for the first 300 objects in ranking.	72
Figure 33 – Precision vs Recall curve of cars dataset , considering Prioritary Objects only	72
Figure 34 – Top 3 objects ranked with higher confidence on cars dataset , considering NF-MEAN combination.	74
Figure 35 – Top 2 objects ranked with higher confidence on cars dataset , considering NF-SUM combination.	74
Figure 36 – Confusion matrices of specialized models for houses datasets	77
Figure 37 – Confusion matrix of Mixed General model on houses dataset	77
Figure 38 – Precision vs Recall curve of houses dataset , considering only priority objects.	78
Figure 39 – Recall of prioritary objects for first 300 positions of the ranking, on houses dataset	79
Figure 40 – Classification of first 300 positions of ranked objects of houses dataset . . .	80
Figure 41 – Top 3 objects ranked with higher confidence on houses dataset , considering NF-MEAN combination.	82
Figure 42 – Examples of detections made by Mixed General model and combination NF-SUM , on cars dataset . This image was identified by DeepSeek as the first image with the highest number of occurrences of priority objects. . . .	89
Figure 43 – Examples of detections made by combination NF-MEAN , on cars dataset . This image was identified by DeepSeek as the first image with the highest number of occurrences of priority objects.	90

Figure 44 – Example of detection using NF-MEAN on cars dataset . This image was identified by DeepSeek as the second image with the highest number of occurrences of priority objects.	90
Figure 45 – Example of detection using NF-MEAN on cars dataset . This image was identified by DeepSeek as the third image with the highest number of occurrences of priority objects.	91
Figure 46 – Classification of the first 300 positions of the rankings, considering only non-priority cars.	93
Figure 47 – Three first objects presented on NF-SUM ranking of not priority cars with higher confidence.	94
Figure 48 – Three first objects presented on NF-MEAN ranking of not priority cars with higher confidence.	94
Figure 49 – Three first objects presented on NF-SUM ranking of not priority houses with higher confidence.	95
Figure 50 – Classification of top 300 objects of each rank, considering not priority houses only	96
Figure 51 – Three first objects presented on NF-MEAN rank of non- priority houses with higher confidence.	97
Figure 52 – Three first objects presented on NF-MEAN rank of priority cars with higher confidence.	98
Figure 53 – Classification of the first 300 positions of the ranking, considering critical objects only , in the cars dataset	99
Figure 54 – Three first objects presented on NF-SUM rank of priority cars with higher confidence.	100
Figure 55 – Comparison of recall considering only priority objects in the first 300 positions of the rankings.	100
Figure 56 – Recall of priority houses, considering only critical objects.	101
Figure 57 – Classifications of first 300 elements of rankings, considering only priority houses.	101
Figure 58 – Three first objects presented on NF-MEAN rank of priority houses with higher confidence.	102
Figure 59 – Three first objects presented on NF-SUM rank of priority houses with higher confidence.	103

List of Tables

Table 1 – Distribution of objects per class, on cars dataset , for each set.	62
Table 2 – Distribution of classes and objects per set, in houses dataset	63
Table 3 – Results for cars dataset , for all models and combinations, considering all predictions.	67
Table 4 – Number of objects of each class found by each model, considering ranking results, on cars dataset	69
Table 5 – Recall, considering only priority objects, for the first 300 positions of the ranking of each model, in cars dataset	69
Table 6 – Results of object’s classifications on first 300 positions of rankings, for cars dataset	71
Table 7 – Results of the model’s rankings on cars dataset , considering all objects. . . .	73
Table 8 – Results of classification on houses dataset	76
Table 9 – Recall, considering only priority objects, for the first 300 positions of the rank of each model, in houses dataset	78
Table 10 – Average precision of houses dataset , per model, considering all objects (priority and non-priority)	79
Table 11 – Number of found objects of each category, on houses dataset , for first 300 positions of the ranking.	81
Table 12 – Number of objects of each class found by models, considering the whole ranking results, for houses dataset	81
Table 13 – DeepSeek preliminary evaluation within Table 4 saved as Latex result file, along with Table 3.	85
Table 14 – DeepSeek preliminary evaluation using rank file.	86
Table 15 – DeepSeek evaluation using textual result file.	88
Table 16 – Average precision of ranking considering only not critical objects only , on cars dataset	91
Table 17 – Results of ranking on cars dataset , considering not critical objects only . . .	92
Table 18 – Results of ranking considering not critical objects only , on houses dataset . .	92
Table 19 – Average precision of rank considering not critical objects only , in houses dataset	92
Table 20 – Results of ranking of critical objects only , in cars dataset	96
Table 21 – Average precision on ranking of priority objects only , on cars dataset . . .	96
Table 22 – Average precision of ranking of critical objects only , on houses dataset , for first 300 positions	98
Table 23 – Results of ranking of critical objects only , on houses dataset , for first 300 positions	99

Table 24 – Results of DeepSeek, considering ranks of priority cars only, using First Prompt and First File Format. 105

Table 25 – Results of DeepSeek, using Second Prompt and Second File Format. 108

Contents

1	INTRODUCTION	18
2	FOUNDATIONS AND RELATED WORK	22
2.1	Object Detection and Background	22
2.2	Natural Language and Textual Embeddings	27
2.3	Disaster Management	38
2.4	Combination of Different Models	43
3	PROPOSED APPROACH	46
3.1	Object Detection Module: YOLO Algorithm	46
3.2	Question Answering Module: DeepSeek Algorithm	48
3.3	Proposed Framework	49
4	DATASETS	60
4.1	Cars Dataset	60
4.2	Houses Dataset	62
5	EXPERIMENTAL EVALUATION	64
5.1	Confidence Analysis	64
5.2	Object Detection Module on Cars Dataset	67
5.2.1	General Results in Object Detection	67
5.2.2	Ranking Results	69
5.3	Object Detection Module on Houses Dataset	76
5.3.1	General Results in Object Detection	76
5.3.2	Ranking Results	78
5.4	Discussion about Object Detection Module, Considering Both Datasets	83
5.5	Question Answering Module Results and Use Examples	84
5.6	Additional Experiments and Results	91
5.6.1	Ranking Experiments	92
5.6.2	Question Answering Experiments	104
6	CONCLUSION	110
	BIBLIOGRAPHY	112

1 Introduction

Climate change effects have been observed around the world for over twenty years (AALST, 2006). Europe faces heat waves, oceans are being affected (TALUKDER et al., 2022), biodiversity is threatened, risks of contagious diseases are increasing (TALUKDER et al., 2022), and extreme storms are becoming more frequent in several parts of the world. Climate experts predict that climate change can make extreme natural events more frequent and ravaging (BANHOLZER; KOSSIN; DONNER, 2014), where vulnerable and exposed populations can suffer tremendous physical and physiological consequences of these events (BENEVOLENZA; DERIGNE, 2019).

Brazil has been facing a lot of climate challenges in the last few years. The Rio Grande do Sul state has been suffering from floods, the most recent being in 2024, causing more than a hundred deaths (MARTINS-FILHO et al., 2024). Animals are also suffering from the floods (MARTINS-FILHO et al., 2024). Houses, commerce, and properties were destroyed, and a lot of people lost everything they had. Brazil's northeast is known for its frequent, aggressive, and fatal historic droughts. In 2016, Brazil's Southeast had a very severe hydric crisis, which almost led to São Paulo, one of the biggest metropolises in the world, being without water.

Also, all American continent is facing climate challenges. Canada and the United States are having historical heat waves, and big fires in forests and fields. As discussed before, climate disasters might become more frequent every year from now. It is essential to have, in this cataclysmic scenario, ways to find out the possible locations of victims who need rescues, even if they are not visible (for example, stuck inside cars in floods, or buried under wreckage after big storms).

Rescue teams in natural disasters need to be prepared and ready for what might be found in disaster scenes. Drones, or mobile robots, can be sent in advance to quickly and safely evaluate risks and potential dangers can save lives, since it can help the rescue teams to build intervention plans that maximize access to victims while avoiding high-risk areas. In addition, rescue teams can have a better insight into the proper equipment to carry for victim assistance and assess the extent of the physical destruction. For example, drone imagery can be used to rapidly analyze disaster scenarios and estimate the number of potential victims, as well as their conditions and locations (NEHETE et al., 2024). This is possible due to important advances in artificial intelligence in the last years. Drones, and robotic platforms, can be used to safely monitor and conduct searches in disaster scenes such as digital cameras. Drones can be equipped with high-quality and easy-to-use sensing technologies. In disasters like hurricanes, for example, drones can be sent quickly to visually monitor the affected areas and assess damages and emergency situations (GREENWOOD; NELSON; GREENOUGH, 2020).

Drones are especially useful and cost-effective in such scenarios thanks to their ability to fly by flooded areas without being affected by landslides, mud, and wreckage (WHITE-HURST et al., 2021). In many different situations, drone images can enable us to analyze risky, inaccessible areas without putting human lives on the line (NEHETE et al., 2024). In addition, rescue teams and first responders can be better prepared to build better logistical and intervention plans to assist victims and minimize potential dangers in disaster areas (e.g., removing broken power lines, extinguishing fires, sealing leaking pipes, and others).

Recently, researchers have started to build datasets that can be used to train deep learning models for object recognition, based on images acquired by drones (NIHAL et al., 2024). Identifying people in disaster images can be a very valuable tool for rescue teams in disaster situations since it can enable them to pinpoint the location of interventions and maximize their benefit to victims (MISHRA et al., 2020) (NIHAL et al., 2024) (WANG et al., 2025a) (AL-MOHANNADI et al., 2026). In addition to victim assistance in disaster zones, drone images can also be used to prevent catastrophes such as building, bridge, or tunnel collapse during disaster events. Some researchers proposed the use of drone images for bridge and road inspection, which can help assess the reliability and safety of the structures (SEO; DUQUE; WACKER, 2018).

This research aims to investigate the potential of recent advances in object detection and the widespread adoption of large language models (LLMs) to address the challenges posed by natural disasters, where rapid response is essential. By analyzing images of damaged areas using machine learning techniques, and in the context of increasingly frequent and severe climate-related events, this technology has the potential to serve as a valuable tool for natural disaster management.

This work proposes to advance the capabilities of disaster-response technology by enabling the automatic assessment of areas affected by natural disasters and generating responses through the integration of natural language and visual scene analysis. A key differentiator of this approach is its explicit focus on minimizing false negatives, ensuring that critical objects and situations are rarely overlooked. By prioritizing recall over precision, this methodology enhances the reliability of the system, thereby better supporting the preparation of rescue teams and facilitating informed decision-making during and after disaster events. The current work is associated with an international collaboration between the *São Paulo State University (UNESP)* and the *University of Québec at Outonais (UQO)*, Canada.

Our proposed approach focuses on the detection of vehicles and residential structures in critical situations. To address this challenge, we specifically target cars submerged in water and severely damaged houses. The framework classifies cars as either critical objects, which are those partially or fully submerged in high water levels, or non-critical objects, which are either not in water or only slightly affected. Similarly, houses are classified as critical when they are severely damaged, and non-critical when they are slightly damaged or intact. The significance of these objects is determined by the likelihood that they may contain individuals who are trapped

or at risk of injury.

The proposed framework integrates object detection models to analyze sets of images, identify critical objects, and provide answers to practical, domain-specific questions. It employs a flexible architecture consisting of two main components: (1) a question-answering (QA) module and (2) a repository of machine learning models, including one model able to detect all targeted object classes, and multiple models specialized in the detection of specific classes. The targeted classes include vehicles categorized by varying levels of water submersion and houses categorized by varying degrees of structural damage. To ensure accurate results, each type of object is associated with a dedicated set of detection models. The QA module receives the user's query and generates an answer based on the outputs of the object detection models.

In this work, we conducted two main evaluations of the scene: (a) general results, considering the number of critical and non-critical objects detected; and (b) a ranking of objects (both critical and non-critical) based on their confidence scores. The framework supports both evaluation approaches for vehicles in flooded scenarios and for houses affected by structural damage. The questions primarily focus on damages caused by seasonal storms and floods, which are closely related and represent a significant issue in Brazil, while also being common in other countries and regions worldwide. The objective is to develop a framework architecture that is extensible and capable of incorporating additional questions and their corresponding models in the future.

Considering the following aspects associated with the proposal: (i) the relevance of tools for supporting rapid decision-making in natural disaster management; (ii) the increased risk of disasters possibly linked to climate change; (iii) the remarkable advancements in technologies for visual and textual content representation; (iv) The urgency of quick action in response to cataclysms; our main goal in this research is: **Design and construct a flexible framework that can answer simple and direct questions about disasters, based on images obtained from affected areas with high recall, especially with low false negatives.**

To maintain a well-defined scope, this research focuses on damages caused by storms and floods, as floods are frequently associated with storm events, in affected areas. Nevertheless, a key objective of the work is to propose an approach that can be extended in the future to address other types of disasters. To validate the proposed approach, two main sets of models were trained: one to detect cars in water and the other to identify damaged constructions. Each set consists of a general model, capable of detecting all object classes within a category (either houses or vehicles), as well as a model specialized for each individual class. This approach leverages specialized training to enable the models to identify objects that might otherwise be overlooked by a general-purpose model, thereby reducing false negatives and enhancing detection reliability. The framework is not intended to interpret complex questions or generate long, elaborated responses. Its primary goal is to provide quick and straightforward information about the analyzed images. The developed framework is capable of recognizing objects with

improved recall, minimizing false negatives wherever possible.

This work is structured as follows: Chapter 2 presents a review of the relevant literature and discusses recent advancements related to the proposed research. Chapter 3 describes the proposed framework, outlining its main characteristics and operational principles. Chapter 4 describes the datasets used in this study, as well as the modifications applied to achieve the research objectives. Chapter 5 presents the results obtained from the framework, including the results of its modules individually, and evaluated aspects. Chapter 6 provides the study's conclusions and outlines directions for future work.

2 Foundations and Related Work

This Chapter presents related work relevant to the present work, as well as theoretical foundations that support the proposed framework.

2.1 Object Detection and Background

There are two types of object detection: generic and specific (LIU et al., 2020). The generic object detection determines whether there are any instances of objects from given categories (such as humans, cars, bicycles, dogs, or cats) in an image and, if present, to return the spatial location and extent of each object instance (LIU et al., 2020). In other words, object detection aims to identify if an object is in an input image, and, in the positive case, indicates where the object is in the image. The specific object detection problem's goal is to detect instances of a particular object, for example, the face of a specific person, or a specific monument or building. It is a matching problem (LIU et al., 2020). This work focuses on generic object detection. Generic object detection is defined as follows: given an image, determine whether there are instances of objects from predefined categories and, if present, return the spatial location and extent of each instance (LIU et al., 2020). It aims to detect various categories, instead of specific object category detection, where only a narrower predefined category of interest can be present (LIU et al., 2020). This work focuses on generic object detection of cars in water and destroyed houses. This Chapter describes several neural network architectures utilized for object detection tasks and their primary findings.

Convolutional Neural Networks

Deep learning is extremely useful for many computational and machine learning tasks. Convolutional Neural Networks (CNNs) are one of the most representative models of deep learning (LIU et al., 2020). They are frequently used as important components of object detection frameworks. CNNs are comprised of three main types of layers: convolutional layers, pooling layers, and fully-connected layers (O'SHEA; NASH, 2015). When these layers are stacked, a CNN architecture has been formed (O'SHEA; NASH, 2015). CNN works with the following main steps:

1. The input layer stores the pixel values of the image (O'SHEA; NASH, 2015).
2. The convolutional layer determines the output of neurons that are connected to local regions of the input. It is done by calculating the scalar product between their weights and the region connected to the input volume (O'SHEA; NASH, 2015). The rectified linear

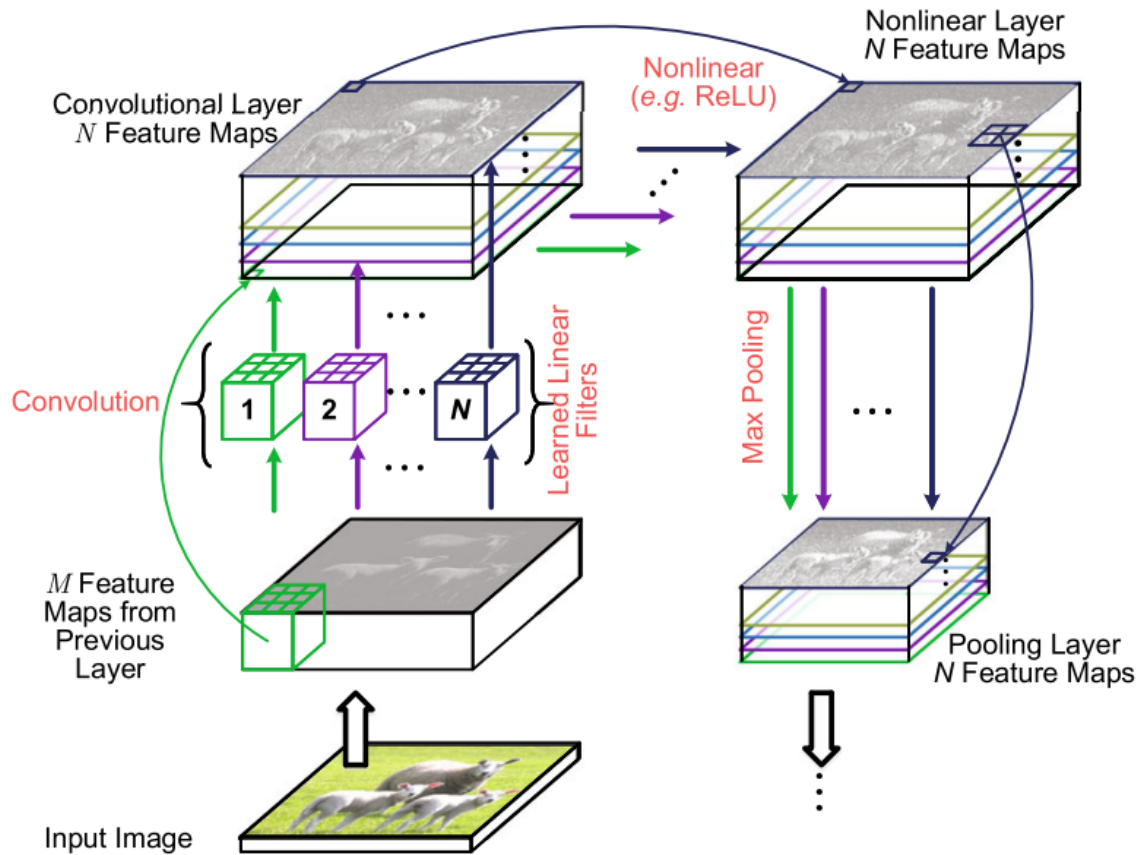


Figure 1 – Illustration of CNN main operations (LIU et al., 2020).

unit (ReLU) applies an “elementwise” activation function (such as sigmoid) to the output of the activation produced by the previous layer (O’SHEA; NASH, 2015).

3. The pooling layer performs downsampling along the spatial dimension of the given input, further reducing the number of parameters within that activation (O’SHEA; NASH, 2015).
4. Fully-connected layers perform the same tasks found in standard artificial neural networks and attempt to produce class scores from the activations, to be used for classification tasks. It is also suggested that ReLU may be used between these layers, to improve the performance.

Figure 1 illustrates the three main operations in CNN:

1. Convolution with a number of linear filters (LIU et al., 2020);
2. Nonlinearities(e.g. ReLU) (LIU et al., 2020);
3. Local pooling (e.g. max pooling) (LIU et al., 2020).

Figure 1 shows the M feature maps pass from one layer to the next, and are convolved with N different filters. Figure 1 shows filters as size $3 \times 3 \times M$, using a stride of 1 (LIU et al.,

2020). This results in N feature maps that are passed through a nonlinear function (e.g. ReLU), and pooled to give N feature maps at a reduced resolution (LIU et al., 2020).

CNNs can transform the original input layer by layer using convolutional and down-sampling techniques to produce class scores for classification and regression purposes (O'SHEA; NASH, 2015). CNN provides an end-to-end learning model in which parameters can be trained by the gradient descent method (ZHIQIANG; JUN, 2017). The well-trained CNN is able to learn the features of the image more fully, and can be regarded as a better black box to extract features (ZHIQIANG; JUN, 2017). In object detection scenarios, CNN can be used to Object Detection Based On Regions. This strategy is based on finding the possible location of objects in an image (ZHIQIANG; JUN, 2017). The region proposal leverages the information of the image, such as color, edge, and texture, which can ensure that the system has a relatively high recall (ZHIQIANG; JUN, 2017). With the candidate regions, the rest work is to classify the candidate regions. It is, essentially, a feature extraction task plus a classification task (ZHIQIANG; JUN, 2017).

Based on CNN, Regions with CNN Features (R-CNN) (GIRSHICK et al., 2014) was created. It consists of three modules:

1. First module generates category-independent region proposals (GIRSHICK et al., 2014). R-CNN is indifferent to the particular region proposal method. But it uses selective search to enable a controlled comparison with prior detection works (GIRSHICK et al., 2014). These proposals define the set of candidate detections available to the detector (GIRSHICK et al., 2014).
2. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region (GIRSHICK et al., 2014). A 4096-dimensional feature vector is extracted from each region proposal using the Caffe (JIA et al., 2014) framework implementation of the CNN (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) (GIRSHICK et al., 2014). Features are computed by forward propagating a mean-subtracted 227×227 RGB image through five convolutional layers and two fully connected layers (GIRSHICK et al., 2014). In order to compute features for a region proposal, it is needed to convert the image data in a region into a form that is compatible with the CNN architecture, that is a fixed 227×227 pixel size (GIRSHICK et al., 2014). All pixels are warped in a tight bounding box around the candidate region to the required size (GIRSHICK et al., 2014). The tight bounding box is dilated so that at the warped size there are exactly p pixels of warped image context around the original box (GIRSHICK et al., 2014).
3. The third module is a set of class-specific linear Support Vector Machines (SVMs). A selective search is run on the test image to extract around 2000 region proposals, warps each one, and forward propagate it through the CNN in order to read off features from the desired layer (GIRSHICK et al., 2014). Then, for each class, each extracted feature

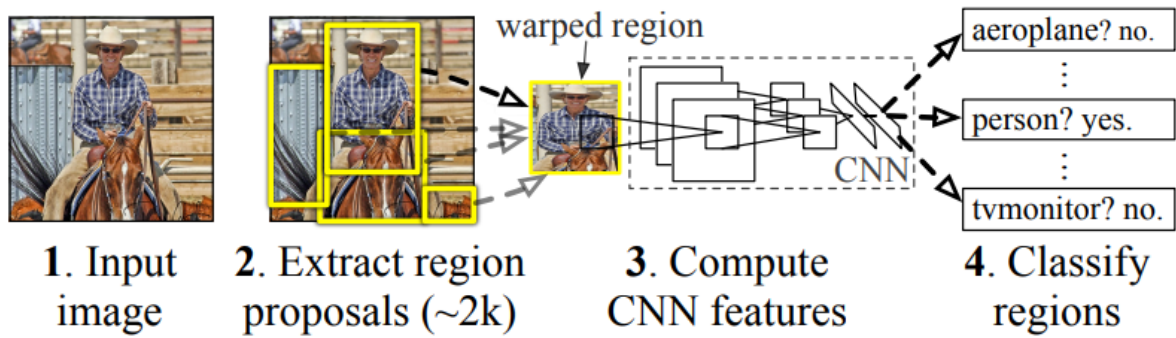


Figure 2 – Illustration of R-CNN main steps ([GIRSHICK et al., 2014](#)).

vector is scored ([GIRSHICK et al., 2014](#)). This is made using the SVM trained for that class. Given all scored regions in an image, a greedy non-maximum suppression is applied for each class independently, that rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold ([GIRSHICK et al., 2014](#)).

Figure 2 presents an overview of R-CNN system ([GIRSHICK et al., 2014](#)). In the first steps, the system takes an input image ([GIRSHICK et al., 2014](#)). In the second step, it extracts around 2000 bottom-up region proposals ([GIRSHICK et al., 2014](#)). In step 3, the system computes features for each proposal using a large CNN, and, in step 4, the system classifies each region using class-specific linear SVMs ([GIRSHICK et al., 2014](#)).

There is also G-CNN ([NAJIBI; RASTEGARI; DAVIS, 2016](#)), an iterative Grid Based Object Detector, also based in CNN. G-CNN trains a CNN to move and scale a fixed multiscale grid of bounding boxes towards objects ([NAJIBI; RASTEGARI; DAVIS, 2016](#)). The backbone of this architecture can be any CNN network ([NAJIBI; RASTEGARI; DAVIS, 2016](#)). After the convolutional layers, the spatial region of interest (ROI) pooling layer is included in the architecture ([NAJIBI; RASTEGARI; DAVIS, 2016](#)). So, given the location information of each box, this layer computes the feature for the box by pooling the global features that lie inside the ROI ([NAJIBI; RASTEGARI; DAVIS, 2016](#)). After the fully connected layers, the network ends with a linear regressor which outputs the change in the location and scale of each current bounding box, conditioned on the assumption that the box is moving towards an object of a class ([NAJIBI; RASTEGARI; DAVIS, 2016](#)). Figure 3 illustrates the structure of G-CNN regression network. The target is represented by the red rectangle, and the bounding box at each step is represented by the blue rectangle.

Residual Networks

In CNNs, ResNet ([HE et al., 2016](#)) excels in a variety of tasks. ResNet is a residual network. It is a deep network with the ability to train without vanishing gradients. ResNet can have different numbers of layers, some of them being 18, 50, and 152. ResNet has shortcuts, which

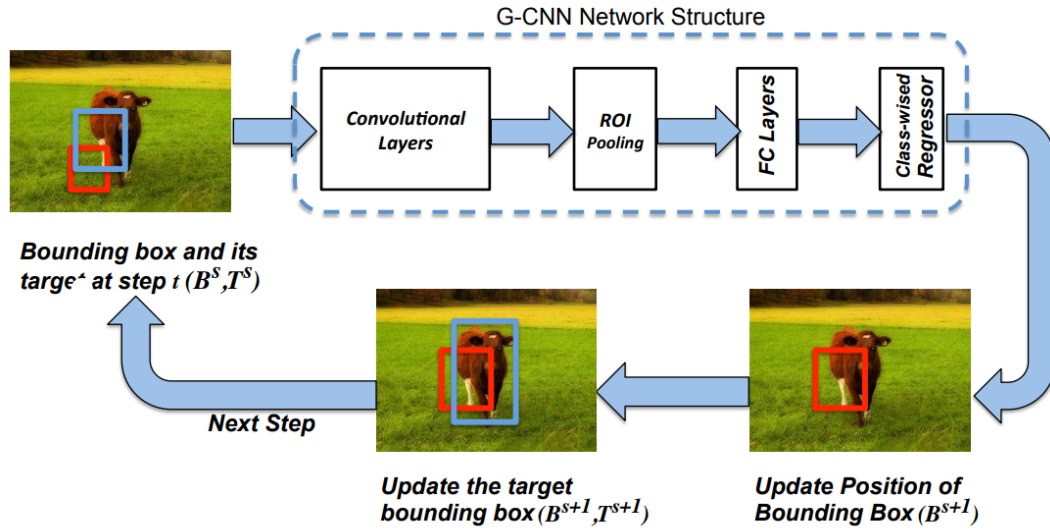


Figure 3 – Structure of G-CNN regression network, originally from the “G-CNN: An Iterative Grid Based Object Detector” (NAJIBI; RASTEGARI; DAVIS, 2016).

allow it to skip some layers. This generates residual blocks, that ResNet uses to learn. ResNet was the first network model to use residual connections, which allow the network to learn the residual mapping between the input and the output of a layer rather than the original mapping (AMJOUD; AMROUCH, 2023). By the residual connections, the network propagates gradients more quickly and allows for the training of much deeper networks (AMJOUD; AMROUCH, 2023). The residual blocks contain multiple convolutional and batch normalization layers (AMJOUD; AMROUCH, 2023). The final layer is connected to a fully connected layer to classify the images (AMJOUD; AMROUCH, 2023). As a result, ResNet is a very powerful classification tool, that can be used in object detection tasks. One of the main disadvantages of ResNets is their relatively complex architecture, which makes them computationally heavy (AMJOUD; AMROUCH, 2023). The implementation of ResNet available in Keras that has less parameters has 25,6 million parameters.

ResNet can be used in real-time object detection video surveillance systems, as paper (PRABU; GNANASEKAR, 2023) presents. Video surveillance systems in object detection tasks have the difficulty of identifying abnormalities such as thefts, accidents, and crimes in environments that often have low light. Modified ResNet (M-ResNet) (PRABU; GNANASEKAR, 2023) is based on convolutional layers inside the existing ResNet CNN architecture. It consists of residual units that enhance the image quality from low illumination to normal illumination (PRABU; GNANASEKAR, 2023). The purpose of M-ResNet is to enhance the image quality from low illumination to normal illumination. If the image has already a good lighting environment, the proposed residual units are skipped to avoid unnecessary computations (PRABU; GNANASEKAR, 2023).

Figure 4 shows three images with different lighting conditions. These images can be

		
a.1 Image with proper light	b.1.Image with bad light	c.1.after correction
		
a.2.object detection	b.2.object detection	c.2.object detection
zebra : 99.92891550 064087 zebra : 99.82913136482239 zebra : 98.41715693473816	zebra : 99.84723329544067 horse : 94.41177248954773 zebra : 85.79044938087463 zebra : 74.2976725101471 zebra : 72.1422553062439	zebra : 99.88270401954651 zebra : 99.811416864395 14 zebra : 97.57338762283325
a.3 Probability of detection	b.3 Probability of detection	c.3 Probability of detection

Figure 4 – Object detection in different lightning conditions (PRABU; GNANASEKAR, 2023).

applied to the original ResNet and the ResNet proposed by (PRABU; GNANASEKAR, 2023). M-ResNet has very good results, compared to other methods (PRABU; GNANASEKAR, 2023).

Detection Transformers

Detection Transformer (DETR) (CARION et al., 2020) is a method that views object detection as a direct set prediction problem, willing to make object detection with transformers. The proposed framework is called Detection Transformer (DETR) (CARION et al., 2020). The idea of the framework is to replace the traditional region proposal-based methods with a fully end-to-end trainable architecture that uses a transformer encoder-decoder network (SHEHZADI et al., 2023). DETR shows that it is able to present better results than some CNN object detectors, and also eliminates the need for hand-crafted components (SHEHZADI et al., 2023).

2.2 Natural Language and Textual Embeddings

This Chapter describes the fundamental foundations of textual embeddings, and algorithms utilized in natural language processing.

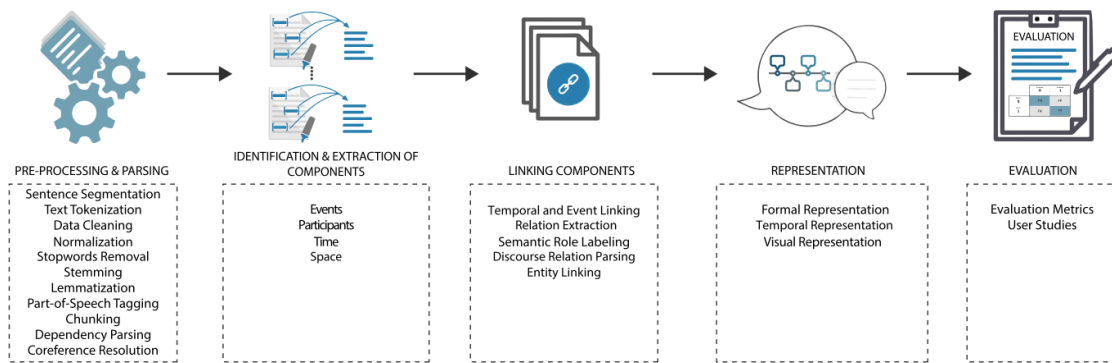


Figure 5 – Narrative extraction pipeline (SANTANA et al., 2023).

Input data for natural language tasks are usually unstructured text and might contain mistype errors, grammatical errors, and incorrect sentences. Textual data does not have an intrinsic numerical representation (GASPARETTO et al., 2022). Preprocessing procedures are of particular importance, as without them, there is no foundation for feature extraction procedures or classification algorithms (GASPARETTO et al., 2022).

Textual embeddings foundations and techniques

In the discrete-vector space representations, the text goes through preprocessing, where the stop-words are eliminated, and stemming is performed to get the root or stem of each word (PATIL et al., 2023). The preprocessing of text in an application affects whether the content and/or style of the language is captured, the statistical power of subsequent analyses, and the validity of insights derived from text mining (HICKMAN et al., 2022). For example, converting all letters in the text to lowercase is useful for treating a word with uppercase as the same when it is in lowercase, but it is not recommended for applications where words should be differentiated by case. Grammar or typing error correction is a useful preprocessing step for applications aiming to extract statistics, but it is not recommended for scenarios aiming to identify different writing styles. Removing stop words can be advantageous for obtaining better statistics and also for reducing the size of long documents, but for applications aiming to identify language styles or individual differences, it may not be a good option (HICKMAN et al., 2022). Therefore, the best types of preprocessing depend on the type of application and the intended purpose.

Natural language is a very complex theme. Various strategies can be used in different contexts, depending on the purpose. Narratives are present in many forms of human expression and can be understood as a fundamental way of communication between people (SANTANA et al., 2023). Figure 5 presents a narrative pipeline with four main steps:

1. Pre-Processing and Parsing: to comprise a set of lexical and syntactic tasks. Lexical tasks split the text into tokens, which are, in many cases, words, and normalize them

into different forms; syntactical analysis identifies the grammar class of tokens, and the dependency between the tokens, producing trees or chunks that relate the tokens to each other (SANTANA et al., 2023).

2. Identification and Extraction of Narrative Components: to identify and extract the main elements that compose a narrative, for example, the detection and classification of events, the recognition and the classification of named entities (participants), and the extraction of temporal information and spatial data associated (SANTANA et al., 2023).
3. Linking Components: to extract the link relations between the components. Linking narrative components comprises the core of the extraction of narratives from the text (SANTANA et al., 2023). In this stage, it is aimed to establish relations and detect structures to understand the meaning (SANTANA et al., 2023). The extracted pieces of information are connected, structuring the narrative at a global level (SANTANA et al., 2023). Thus, temporal and event linking, entity and event linking, and the relation extraction are considered (SANTANA et al., 2023).
4. Representation of Narratives: it can be categorized in two levels: (1) the conceptual considers that the elements in a narrative are codified as concepts that allow abstracting the meaning, and this codification allows the exploration and detection of the elements of a narrative, smoothing the process of analysis of a narrative (SANTANA et al., 2023); (2) the visual level is built using visual elements like lines, graphs, icons, pictures, or other graphical resources used to depict the narrative (SANTANA et al., 2023).
5. Evaluation: evaluate the results. To computational systems, it is a very complex task (SANTANA et al., 2023). For this reason, a manual evaluation method is usually adopted (SANTANA et al., 2023). In this case, the narrative result is delivered to specific people to evaluate the generated result according to predefined domain criteria (SANTANA et al., 2023).

Each one of these steps is very complex and might depend on the goals in the research and what is the goal of the analysis. Sometimes, for many applications, it is necessary to represent text numerically. In the discrete-vector space representations, and statistical methods in general, the corpus or text initially goes through preprocessing, where the stop-words are eliminated, and stemming is performed to get the root or stem of each word. Then, texts are represented as matrices, where words are represented using vectors of numbers, and the corpus of the text is represented as a collection of these vectors, forming a matrix (PATIL et al., 2023). The advantage of using vector representations, or other numerical representations, is that they allow the use of linear algebra operations to compute metrics, like distances and numeric similarities (PATIL et al., 2023).

Bag of Words (BOW) (HARRIS, 1970) is a popular way to represent text. In this representation, the corpus is represented by a matrix, where each row represents a sentence, and each column represents a unique word from vocabulary (PATIL et al., 2023). For each sentence in the corpus, the frequency or the absence of each word in the sentence is counted and updated in the respective row and column of the matrix (PATIL et al., 2023). It can be made with binary values. The row index is the sentence's position in the corpus, and the column index is the current word's dictionary position (PATIL et al., 2023). As a result, the number of rows is equal to the number of sentences in the corpus, and the number of columns is equal to the vocabulary size of the corpus (PATIL et al., 2023). This matrix is termed a 'document-term' (PATIL et al., 2023).

Another way to represent texts is to use N-grams (KATZ, 1987), that proposes an attempt towards imposing a window to capture the ordering among words (PATIL et al., 2023). A number n represents the size of the window or context. N-gram focus on multi-word tokens that are able to capture the ordering present in the window, and tries to capture groups of words (PATIL et al., 2023). For example, n-grams try to capture "ice cream" instead of "ice" and "cream", and "not like" instead of "not" and "like". The intention is to capture context (PATIL et al., 2023). N-grams can be used to construct a hierarchical dictionary (COPE; ZHANG, 2023), by, instead of assigning values to single words, assigning values to n-grams (COPE; ZHANG, 2023). Some advantages of this method are: (1) allow a researcher to distinguish differences in meaning stemming from small variations, including word order, in terms and phrases (COPE; ZHANG, 2023); (2) to be able to assess the values of expressions on multiple dimensions (COPE; ZHANG, 2023), and (3) to be especially adept at measurement tasks for which the elimination of incidences of unit-level measurement errors is critical (COPE; ZHANG, 2023). But it also has some limitations and disadvantages: it requires that meaningful expressions appear in the corpus with some frequency, so the method can be applied only if the researcher can anticipate the salient language that the corpus uses, because she is familiar with the language from similar corpora, or because she has reviewed a representative sample of the corpus (COPE; ZHANG, 2023); also, the method requires human experts familiar with the meaning of the corpus' salient expressions (COPE; ZHANG, 2023). This might be a limitation to specialized terminology or an entire language with which the researcher was not familiar (COPE; ZHANG, 2023).

As mentioned previously, one relevant advantage of numerical representation of text is to be able to apply numerical algorithms to text features. For example, to use clustering or classifier algorithms to group texts that are similar. Clustering is to group elements that are, according to some criteria (that will depend of application and the purpose of it), similar. Clustering of texts often requires pre-processing, such as removing stop words and punctuation marks. Then, with the remaining words, it is possible to cluster texts through the representation of spatial vector models, where the frequency of each word in the text appears in a vector, which typically results in high-dimensional representations. A natural way to address this issue is to use hierarchical clustering to discover the most frequent terms and then create clusters based on sets of frequent terms, measuring the mutual overlap of frequent sets with respect to sets of

supporting documents ([BEIL; ESTER; XU, 2002](#)).

Another way to group texts is by considering the sequence of words, which is done by n-grams representation, but can also be done by representing documents as sequences of words and considering that there is at least a specified minimum number (or percentage) of documents containing this set of words. Then, infrequent words are removed from the texts to reduce dimensionality. Association rules from data mining can be used to find words that frequently appear together, and then combine overlapping cluster candidates based on a specified overlap threshold or a determined number of final clusters ([LI; CHUNG; HOLT, 2008](#)).

One of the most relevant metrics in text application is TF-IDF. In the text clustering task, it is a very useful metric. TF-IDF is a numerical statistic that measures how important a word is to a document in the collection or corpus ([CHRISTIAN; AGUS; SUHARTONO, 2016](#)). TF-IDF value increases proportionally to the number of times a word appears in the document, but it also considers the frequency of the word in the corpus. This helps to control the fact that some words are more common than others ([CHRISTIAN; AGUS; SUHARTONO, 2016](#)), and to deal with words that are very common in every type of document, but has little semantic meaning.

By definition, TF-IDF is a metric that multiplies two quantities, TF and IDF, where TF is the frequency of a term in a document, and IDF is the inverse of the term's frequency across a set of documents. Thus, TF-IDF is a representation measure that quantifies the extent of the terms' usefulness in characterizing the document in which they appear ([AIZAWA, 2003](#)). The main goal is to find out which words are more relevant to each document and measure the importance of the words across the corpus of the text. Also, it is intended to measure the relevance of a document based on query keywords ([PATIL et al., 2023](#)). Term-Frequency (TF) counts the frequency of words in a document, and Normalized Term Frequency (NTF) is the term frequency of the word divided by the total number of words in a document ([PATIL et al., 2023](#)). This normalization helps compute the importance of a word relative to other words in that document ([PATIL et al., 2023](#)). The matrix is a document-term matrix, where rows represent sentences or documents and columns represent unique words in the corpus ([PATIL et al., 2023](#)). In this approach, the values are normalized term-frequency floating point values or scores, and not binary values ([PATIL et al., 2023](#)). In NTF, the importance of a word for other words in that document is taken into account ([PATIL et al., 2023](#)), and also the relations between words. This approach has an advantage: the possibility to compute similarity scores between a query and a document, or between documents, by looking at the cosine angle between the respective vectors ([PATIL et al., 2023](#)).

Embeddings are representations of texts. An embedding is therefore a vectorial representation, which can be processed by a machine, but also encodes part of the underlying meaning of the words ([GASPARETTO et al., 2022](#)). Word embeddings are real-valued representations of words produced by distributional semantic models, and are one of the most popular tools in

Natural Language Processing (NLP) (BAKAROV, 2018). There is no consensus in the scientific community about which method should be used in the evaluation of these distributional semantic models since it might depend on the purpose and objective of the application (BAKAROV, 2018).

Transformers for Natural Language Processing

Transformers are a very powerful tool for Natural Language Processing (NLP). A transformer is a deep learning model, very popular, with vast options of models, implementations, and purposes. They transform words into tokens, and put a weight value on each token, according to their importance. They can be used to generate answers to query entries. Also, transformers can be used to extract text features.

Transformer's architecture is shown in Figure 6, originally from (VASWANI et al., 2017). It is composed of the following steps:

- **Encoder:** It is composed of a stack of identical layers. Each layer has two sub-layers, the first being a multi-head self-attention mechanism, and the second being a simple, position-wise fully connected feed-forward network (VASWANI et al., 2017). There is a residual connection around each of the two sub-layers, and also a layer normalization.
- **Decoder:** It is, also, composed of a stack of identical layers, each one with the two sub-layers the encoder has. But the decoder has a third layer, which performs multi-head attention over the output of the encoder stack. As the encoder, residual connections are employed around each of the sub-layers, followed by layer normalization (VASWANI et al., 2017). Also, the self-attention sub-layer was modified in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with the fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i .
- **Attention:** An attention function is a function that maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors (VASWANI et al., 2017). The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key (VASWANI et al., 2017).

Attention can be defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right) V \quad (2.1)$$

where d_K are the dimensions of queries and keys, Q is the matrix of queries, K is the matrix of keys and V is the matrix of values. The attention mechanism allows transformers to find the

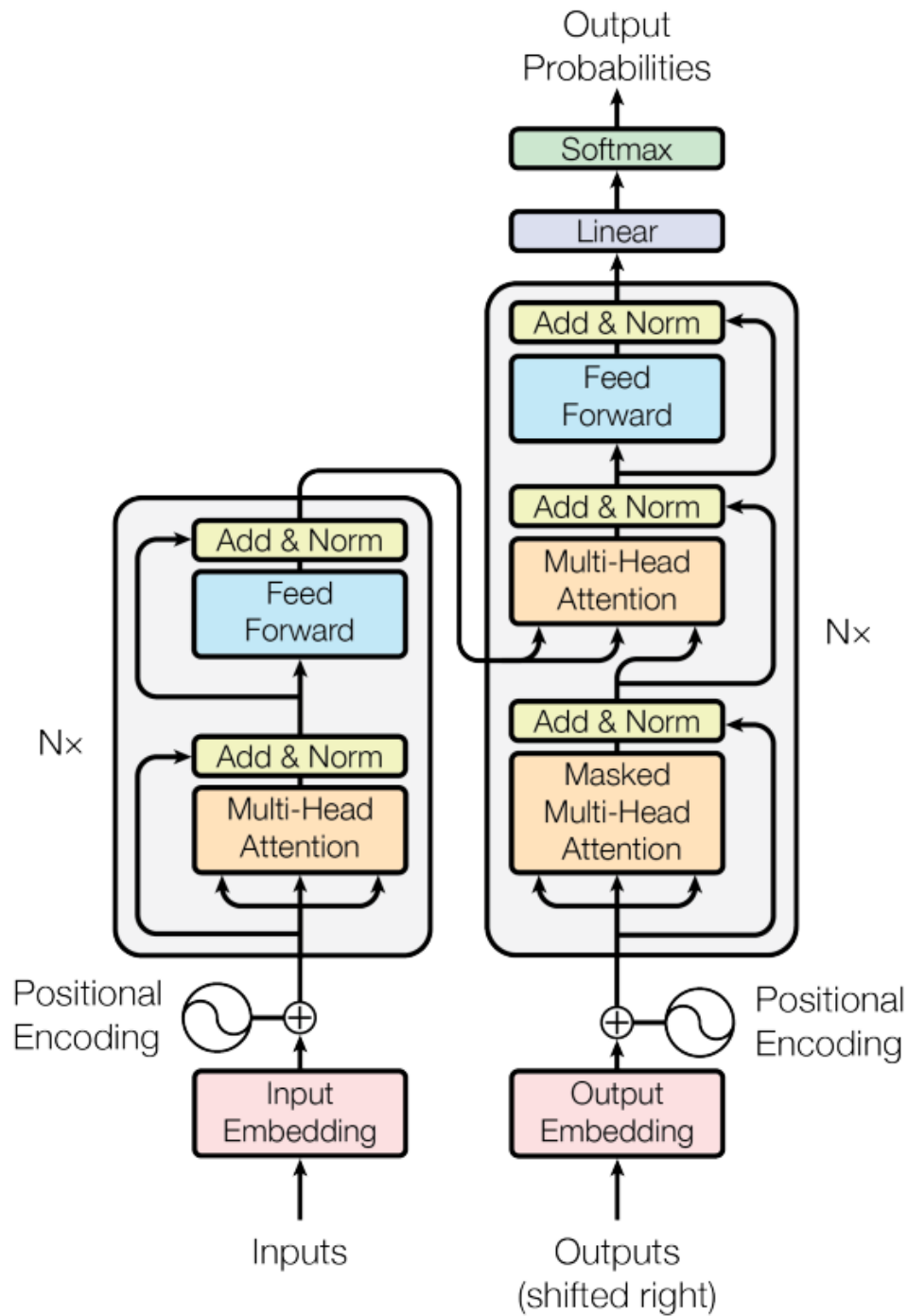


Figure 6 – Transformer Architecture (VASWANI et al., 2017)

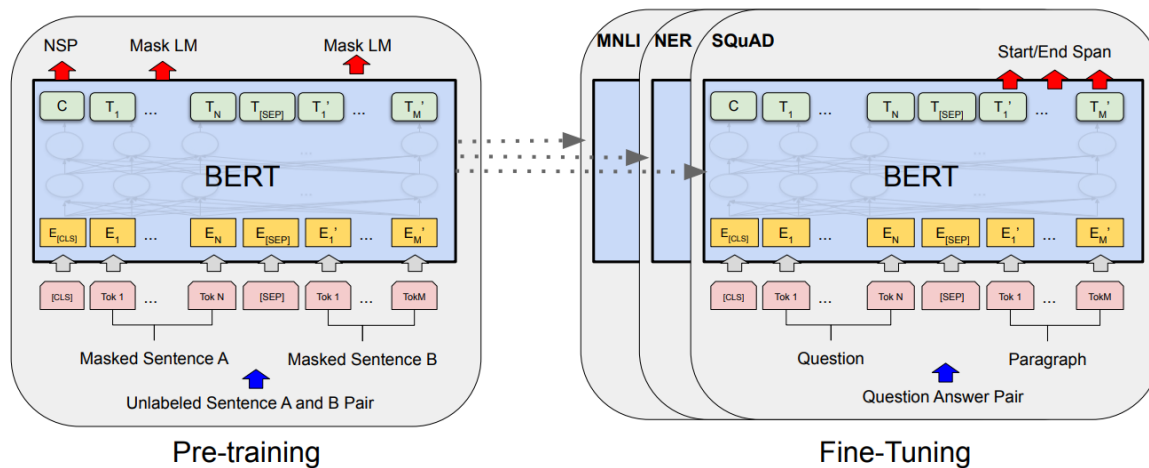


Figure 7 – Illustration of pre-training and fine-tuning for BERT (DEVLIN et al., 2019)

relations between words, by relating words in a sequence. Transformers can be used for a lot of different tasks, one of them being keyword extraction. Transformers can also be used on image-related tasks, such as classification and object detection.

BERT

Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN et al., 2019) is a framework that works with two main steps: pre-training and fine-tuning.

- **Pre-training:** the model is trained on unlabeled data over different pre-training tasks (DEVLIN et al., 2019).
- **Fine-tuning:** As described in (DEVLIN et al., 2019), BERT model is initialized with the pre-trained parameters. Then, all the parameters are fine-tuned using labeled data from the downstream tasks. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters. Figure 7 illustrates this process. Apart from output layers, the same architectures are used in both pre-training and fine-tuning (DEVLIN et al., 2019).

BERT's model uses Transformer's bidirectional architecture, such as described in (VASWANI et al., 2017). It was pre-trained in a deep bidirectional representation, by masking some percentage of the input tokens at random, and also pre-train for a binarized next sentence prediction task that can be trivially generated from any monolingual corpus (DEVLIN et al., 2019). As a result, BERT has a powerful, deep bidirectional architecture. The same pre-trained model is able to successfully solve a vast set of natural language tasks as a powerful and versatile algorithm (DEVLIN et al., 2019).

Advanced and state-of-the-art Question-Answering algorithms and techniques

Question-answering modules are becoming more popular and accessible in the last years. ChatGPT, Gemini, Alexa and Siri are very popular models that can answer complex questions with elaborate answers. This kind of application is becoming more powerful and more popular over the last years, and the cases of use are increasing to many knowledge areas. These models are able to provide information in a very quick and practical way, since it is a lot easier to make a question and get an answer than search with the most popular searches on the web, which can, depending on the subject, take hours.

Question-answering systems allow users to obtain retrieved answers for questions posed in natural language (ANTONIO; SOARES; PARREIRAS, 2020). These kinds of systems can answer questions asked in natural language using a pre-structured database, or a collection of natural language documents (ANTONIO; SOARES; PARREIRAS, 2020). User can ask questions, and then retrieve the answers using natural language queries (ANTONIO; SOARES; PARREIRAS, 2020). They can be considered as an advanced form of Information Retrieval (ANTONIO; SOARES; PARREIRAS, 2020).

At this moment, there are various question-answering systems that are able to talk back to humans with very advanced and natural language, about a vast variety of subjects, but might have some difficulty in retrieving correct information. It is not uncommon for this system's users to see imprecise or incorrect (yet very well-written) answers to their questions.

Question-Answering (QA) typically has three modules (ALLAM; HAGGAG, 2012):

1. **Query Processing (ALLAM; HAGGAG, 2012):** it focuses on question classification. It receives a natural language question as input, and its goal is to analyze and process the question by creating some representation of the information requested (ALLAM; HAGGAG, 2012). The question processing module is required to:

- Analyze the question, in order to represent the main information that is required to answer the user's question (ALLAM; HAGGAG, 2012).
- Classify the question type, usually based on a taxonomy of possible questions already coded into the system (ALLAM; HAGGAG, 2012).
- Reformulate the question, willing to enhance the question phrasing and to transform the question into queries for the information retrieval (search engine) (ALLAM; HAGGAG, 2012).

After these steps, the question-processing module passes a set of query terms to the next module.

2. **Document Processing Module (ALLAM; HAGGAG, 2012):** focus on information retrieval. Usually relies on one or more information retrieval systems to gather information

from a collection of documents. The documents returned by the information retrieval system are then filtered and ordered. The main goal of the document processing module is to create a set of candidate-ordered paragraphs that contain the answer(s), and in order to achieve this goal, the document processing module is required to:

- Retrieve a set of ranked documents that are relevant to the submitted question ([ALLAM; HAGGAG, 2012](#)).
- Filter the documents returned by the retrieval system, in order to reduce the number of candidate documents, and also to reduce the amount of candidate text in each document ([ALLAM; HAGGAG, 2012](#)).
- Order the candidate paragraphs to get a set of ranked paragraphs, according to a plausibility degree of containing the correct answer ([ALLAM; HAGGAG, 2012](#)).

3. **Answer Processing Module:** focus on the answer extraction. It is responsible for identifying, extracting, and validating answers from the set of ordered paragraphs passed to it from the document processing module ([ALLAM; HAGGAG, 2012](#)). The answer processing module tasks are:

- Identify the answer candidates within the filtered ordered paragraphs through parsing ([ALLAM; HAGGAG, 2012](#)).
- Extract the answer by choosing only the word or phrase that answers the submitted question through a set of heuristics ([ALLAM; HAGGAG, 2012](#)).
- Validate the answer by providing confidence in the correctness of the answer ([ALLAM; HAGGAG, 2012](#)).

ChatGPT ([OPENAI, 2024](#)) is one of the most popular and notable QA systems. It became very popular for its ability to provide long, complex, and elaborated answers to complex and elaborated questions. Robots are being largely used in industry and commerce, to answer client's questions, make medical appointments, sell products, and others.

A way of answering questions based on images is through a knowledge graph. A knowledge graph is a structure of information, where the nodes are the elements in the image (an object detection system can be used to detect those objects) and the edges of the graph are the relations between those objects.

It is possible to create a commonsense knowledge transfer framework, as presented in ([ZHOU; BRAS; CHOI, 2023](#)). In this framework, the source model and the target model are heterogeneous because they are trained on different data with different objectives. Firstly, a neural commonsense knowledge model is used ([ZHOU; BRAS; CHOI, 2023](#)). To retrieve knowledge, natural texts are used to form queries, that will retrieve information from the neural network, in text form. Then, training data with two novel commonsense-related self-supervised

objectives is constructed, based on retrieved commonsense knowledge and the corresponding natural text of the first neural model (ZHOU; BRAS; CHOI, 2023). Then, a targeted model is trained on the constructed training data to inject commonsense knowledge retrieved from the source model. Figure 8 illustrates the methodology of (ZHOU; BRAS; CHOI, 2023).

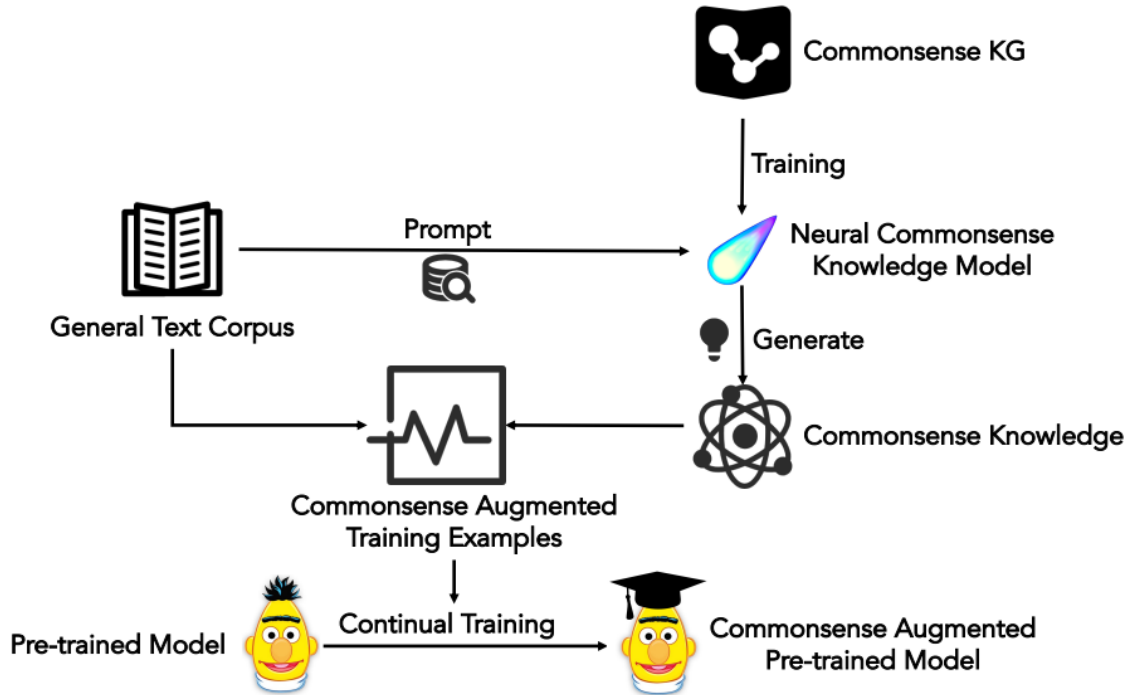


Figure 8 – Illustration of the commonsense knowledge transfer framework (ZHOU; BRAS; CHOI, 2023).

Similar to text classification, there is intent classification. It is a task focused on finding the intent of a query (CHEN; ZHUO; WANG, 2019). It is a kind of classification problem, which's goal is to predict the intent label for a query. BERT can be used in this kind of task (CHEN; ZHUO; WANG, 2019).

Intent classification is classifying a user utterance in textual format into one of the classes from a pre-defined set of intent classes (ATUHURRA et al., 2024). It is similar to a traditional classification tasks from the point of view of classifying an element into one of a set of pre-determined classes.

HSchatbot (ASSAYED; SHAALAN; ALKHATIB, 2022), for example, classifies questions based on the words used in the question. In it's pre-processing, words are tokenized, and a vector is created with the count of how many times a word appears in the text. TF-IDF technique is also applied. It is another way to represent the Bag of Words. One of the main advantages of using numeric representation, as mentioned before, is to be able to use traditional numerical classifier algorithms, such as Random Forest Classifier (ASSAYED; SHAALAN; ALKHATIB, 2022). With this strategy, intent classification of text can be simplified into a

traditional classification problem, in which a vast variety of algorithms and classifiers can be used.

Due to the complexity of the task and the highly specific nature of our scenario, we opted to use a pre-trained language model. This enables our framework to understand a broader range of questions at a level of complexity that would be difficult to achieve through custom training. The selected model was DeepSeek ([DEEPSEEK-AI et al., 2025](#)), a robust question-answering module comparable to systems such as ChatGPT, Gemini, and Alexa.

2.3 Disaster Management

This Chapter presents algorithms and techniques employed to address the challenges of natural disaster management across various scenarios.

In general, disaster management consists of four steps: prevention, preparation, response, and recovery ([KIM et al., 2022](#)). As a first step, a disaster management system might include drone flight planning, aerial photo capturing, key points extraction, and image matching ([KIM et al., 2022](#)). The images can be used to create flood damage analysis and other disasters. Figure 9 shows results from ([KIM et al., 2022](#)). Different strategies can be applied in natural disaster management systems, aiming for different goals. For example, to monitor water bodies to prevent floods, monitor storm clouds and rain, predict hurricanes, predict landslides and others.

Aerial devices and remote sensing images

In cataclysms scenarios, intelligent devices can be very useful, because they might be able to perform complex tasks with full autonomy ([HERNÁNDEZ et al., 2022](#)). Destroyed areas can be hard to cover, because of infrastructure problems, mud, destroyed electrical structures, high levels of water, and others. To solve this problem, a possible solution is to construct aerial devices that can both capture and process information to gain an understanding of their context and determine the appropriate course of action in an independent manner ([HERNÁNDEZ et al., 2022](#)) ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)).

Figure 10 shows destroyed houses, in pictures taken by Unmanned Aerial Vehicles (UAVs) ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)). This kind of picture can be very useful in post-hurricane aerial damage assessments ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)). The target classes are highly and orderly related to each other, for example, the difference between damage levels 0 and 5 is much more significant than that of damage levels 0 and 1 ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)). This kind of image can be used to train deep neural networks, such as Convolutional Neural Networks (CNN), to detect destroyed houses in places destroyed by some cataclysm, such as hurricanes ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)).

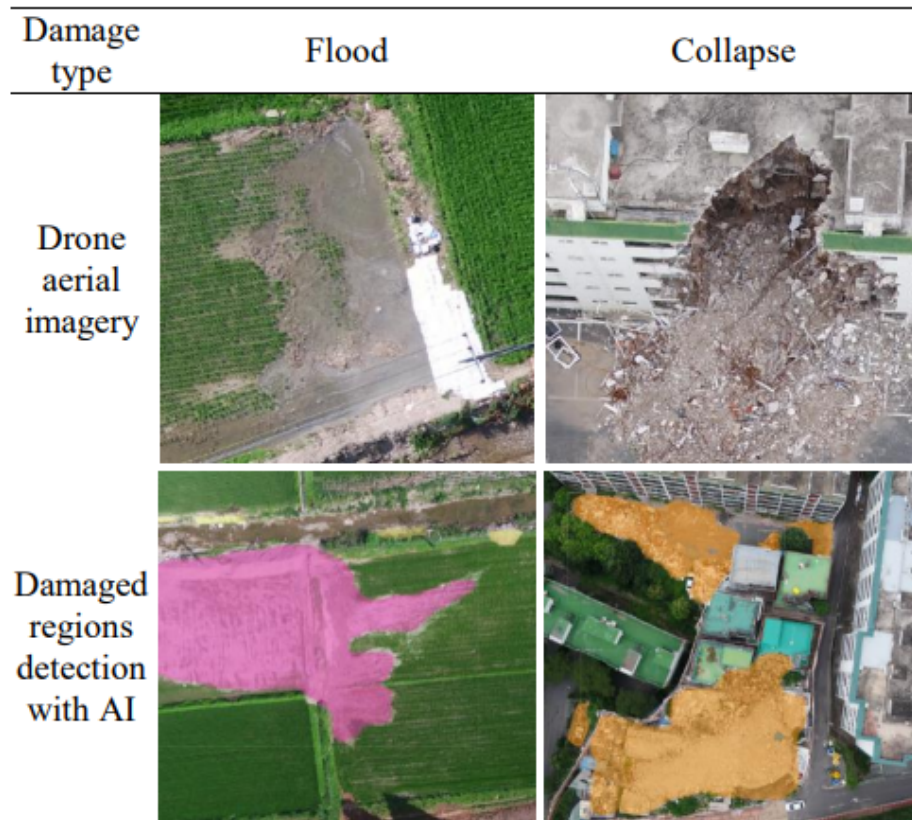


Figure 9 – Detection results from “Disaster damage investigation using artificial intelligence and drone mapping” (KIM et al., 2022) paper.

Also in the cataclysms scenarios, a very useful resource is semantic segmentation. It is the technique of classifying parts of an image that belong to the same object class (HERNÁNDEZ et al., 2022). It is a form of pixel-level prediction where pixels in an image are classified according to a category. A neural network can be used to capture the visual patterns of the image and reconstruct a representation of the detected classes (HERNÁNDEZ et al., 2022). For example, one possible way to monitor water bodies is to extract lake water bodies from remote sensing images. Satellite images can be used for this purpose, using neural networks to extract water bodies from remote sensing images (LIU; LIU; HU, 2024). Image semantic segmentation can also be done based on pixels. Images can be annotated, so objects of different classes are recorded as bounding boxes in dictionary format. The bounding box dictionaries keep records of the object classification of each pixel. The pixels can be classified as, for example, “Building No Damage” and “Building Total Destruction” (RAHNEMOONFAR; CHOWDHURY; MURPHY, 2023).

YOLO algorithm in natural disaster scenarios

You Only Look Once (YOLO) (REDMON et al., 2016b; JIANG et al., 2022) is a very powerful tool to detect objects in real-time images. It is the main algorithm explored in the



Figure 10 – Examples of constructions with different levels of damage ([CHENG; BEHZADAN; NOSHADRAVAN, 2021](#)).

present work. The YOLO algorithm is discussed in detail in Chapter 3.1.

YOLO can be used to detect victims of natural disasters ([SAROSA; MUNA; ROHADI, 2021](#)), and a high variety of objects in various different situations. It can be used in drone images to detect people in destroyed places ([SAROSA; MUNA; ROHADI, 2021](#)), for example. Also, YOLO can be used to detect body parts in a natural disaster context ([MOECHAMMAD; CAHYA; BERKAH, 2021](#)). This can be useful to detect buried or submerged people, and can also be useful to detect corpses of missing people ([MOECHAMMAD; CAHYA; BERKAH, 2021](#)).

Social Media approaches in natural disaster scenarios

There is various strategies that can be used to manage natural disasters. Social media provides very useful tools for these scenarios, being one of the most important communication methods in modern society. One of the main advantages of social media is how easy it is to combine different types of data, such as tags, images, audio, text, geolocation, and others. It is common that these various types of data are usually associated, especially in social media like Instagram, TikTok, X (previously known as Twitter), Facebook, and others. It is common that a lot of information, like geolocation, hashtags, temporal information, and user information is associated with an image, which can make data analysis a lot easier ([SAID et al., 2019](#)).

The X social media has been widely utilized for the dissemination of information in natural disaster situations. Keywords can be an interesting strategy in this scenario. In social media, it is also possible to use hashtags to identify natural disasters ([SAID et al., 2019](#)). Probabilistic flood ([BROUWER et al., 2017](#)) extends estimates based on X data and Google

Maps images. Tweets can be collected using the X streaming API, and data flood observations were extracted from the data (BROUWER et al., 2017). Then, Google Maps and Google StreetView can be used to find locations mentioned by the tweets and the locations of photographs attached to the tweets respectively (BROUWER et al., 2017). To extract X data, all tweets that contained a number of common flood-related keywords, such as “flood” or “inundation”, can be collected (BROUWER et al., 2017). Then, only the tweets with hashtags that referenced places of interest were kept, and tweets that contained explicit references to locations, such as streets or points of interest (POIs), by looking for common keywords such as “street”, “lane”, “museum” or “school”. Some other minor filters were applied to ensure only relevant tweets were found, for example by excluding tweets related to flood barriers and flood warnings (BROUWER et al., 2017). If none of the tweets in their dataset mentioned water depths, so water levels can be derived by assuming the same default water depth (DWD) for all observations. Water levels can be calculated relative to the nearest drainage channel by using a height above-nearest-drainage (HAND) elevation model (BROUWER et al., 2017). Secondly, observations can be grouped based on the local drainage directions (LDDs) to interpolate only hydrologically “connected” observations (BROUWER et al., 2017). Inverse-distance-weighting (IDW) interpolation can be used to determine the flood extent (BROUWER et al., 2017).

Systems, algorithms, and multi-data approaches

Figure 11 shows the methodology of a system based on drones (KIM et al., 2022). Drone images are collected, and the most important tasks, at this moment, are image classification, object detection, and image segmentation (KIM et al., 2022). Disaster classification is to classify an image or video into one or several sets of predetermined classes (KIM et al., 2022). In mapping, this classification is used to categorize geotagging-imagery (KIM et al., 2022). These tasks are key for mapping as they analyze what is in a drone image, locate it, and plot it on a map (KIM et al., 2022). So, it identifies the type of disaster and the affected area.

The photogrammetry-based drone mapping technology for the disaster damage investigation can be an alternative approach to support or replace the manual work in disaster site survey that needs to investigate the disaster site quickly (KIM et al., 2022), especially in dangerous areas. The results in Figure 9 show how images can be used to quickly identify dangerous areas, possibly with victims, in the safest and faster way than sending reconnaissance teams to dangerous areas.

For rescue teams and post-disaster management, it is essential to identify the precise location and type of a disaster. Vision-language models have proven to be powerful tools for pinpointing these locations, while satellite imagery is highly effective for accurately classifying the disaster type (WANG et al., 2025a). Datasets pairing satellite images with text provide crucial resources to train artificial intelligence models, enabling them to detect disasters accurately and correlate visual features with textual data (WANG et al., 2025a). Consequently, this mul-

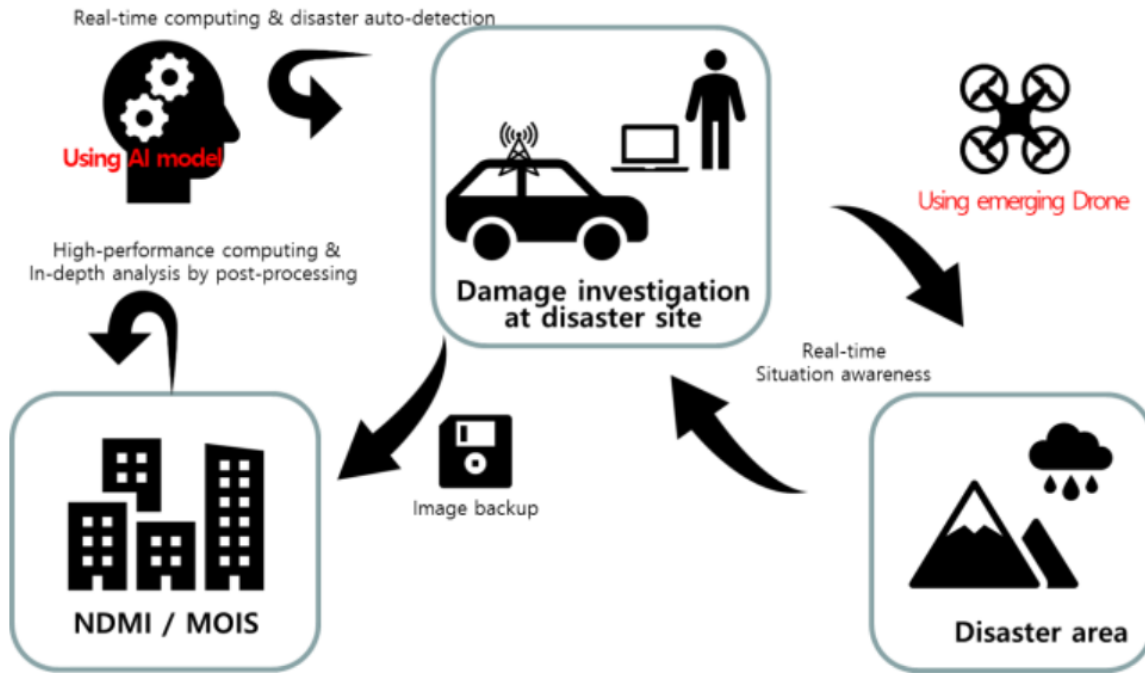


Figure 11 – Illustration of method proposed in (KIM et al., 2022).

timodal approach leverages satellite imagery to precisely identify disaster locations and types, significantly supporting rescue operations and post-disaster management.

Advances in visual question-answering (VQA) also provide datasets that pair highly complex questions with multi-object scenes, aiming to train artificial intelligence to perform complex reasoning on visually intricate images (AL-MOHANNADI et al., 2026). This progress is driven by the rapid development of artificial intelligence in recent years, particularly the evolution of question-answering systems and large language models, such as GPT (AL-MOHANNADI et al., 2026).

Figure 12 presents image-question-answer triplets from the DisasterVQA (AL-MOHANNADI et al., 2026) dataset. The dataset is composed of images that represent disasters (disaster-positive images) and images that do not (hard-negative images). These images feature highly complex scenes associated with binary (yes/no), multiple-choice, and open-ended questions, including queries that require complex reasoning.

The approach used on the present research is to prioritize recall over precision. This approach decision can be a challenge in disaster management scenario. The excess of raised false alarms can lead to costly resource mobilization and can lead to alarm fatigue, reducing trust in automated systems (CHOI; LEE; SONG, 2026). In the pursuit of highly accurate outcomes, the design of language model prompts plays a crucial role (SUN; WANG; PENG, 2024). Furthermore, integrating visual features into the AI's reasoning process serves as an effective strategy to mitigate AI hallucinations (SUN; WANG; PENG, 2024), thereby yielding

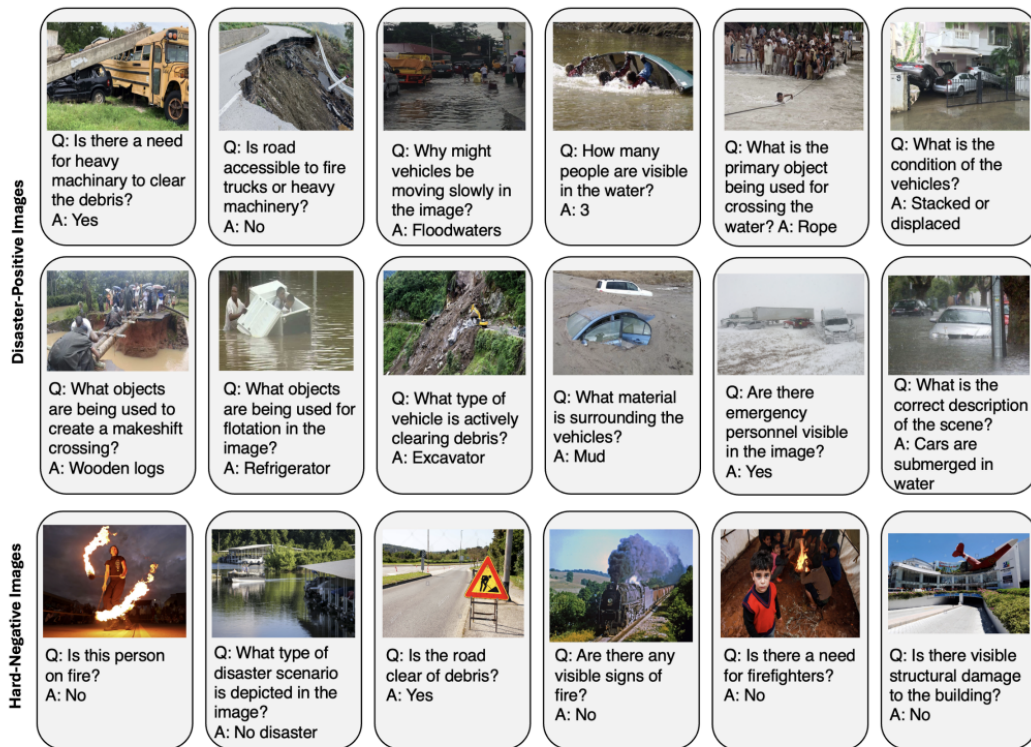


Figure 12 – Illustration from DisasterVQA Dataset ([AL-MOHANNADI et al., 2026](#))

more accurate results in disaster-related visual question-answering (VQA) tasks.

The DeBERTa Vision Attention Network (DeVANet) ([BLAGA; NEDEVSKI, 2026](#)) addresses post-disaster remote sensing imagery for visual question-answering (VQA) applications. A local-global attention architecture is used to analyze visual embeddings, combined with DeBERTa (Decoding-enhanced BERT with Disentangled Attention) ([HE et al., 2021](#)) for language modeling. DeBERTa’s bidirectional architecture enables a deeper understanding of the textual context and ensures strong performance in complex counting tasks. Ultimately, it is a robust approach that allows for the complex evaluation of scenes and advanced VQA tasks with high accuracy.

2.4 Combination of Different Models

This Section presents works involving combinations of different models, including combinations of object detectors and LLMs, as well as works involving combinations of different detector models.

YOLO and DeepSeek can be combined into lightweight models and achieve high accuracy results ([WANG et al., 2025b](#)). SAL-YOLO ([WANG et al., 2025b](#)) is a lightweight decision framework designed for safety monitoring in crowded environments with limited computational resources. DeepSeek is integrated into the system to evaluate risk and generate a response answer and trigger alerts. SAL-YOLO is able to detect pedestrian falls, with high

accuracy, in real-time. With the advance of LLMs on the last years, a lot of advancements are being made in visual question-answering scenario(CHENG et al., 2024), including detection and management of natural disasters (LIU et al., 2025)(SAI et al., 2025). Our proposed framework follows a similar approach of combining DeepSeek and YOLO models to construct a system capable of issuing warnings about hazardous situations, but with the differential of focusing on low false negatives and specific objects in natural disaster scenarios (cars in flooded areas and destroyed houses).

Diversity-driven Object Detection Fusion (DiODeFusion) (PEREIRA et al., 2025) is a framework that combines multiple detectors to improve the prediction results of different models, and has been applied to the detection of white mold. It is composed by four models: Detector Training and Detection Computation, Diversity Measure Computation, Diversity-Informed Detector Selection and Bounding Box Fusion (PEREIRA et al., 2025). The first module (Detector Training and Detection Computation) trains multiple detectors using the training and validation sets of the dataset, and computes predictions on the validation set (PEREIRA et al., 2025). The Diversity Measure Computation module then computes diversity measures to the bounding boxes predicted by the multiple detectors (PEREIRA et al., 2025). Next, Diversity-Informed Detector Selection module selects the most promising detectors for fusion, based on the diversity measures and effectiveness metrics computed for the different detectors (PEREIRA et al., 2025). At last, Bounding Box Fusion module combines the the predictions of selected detectors on the test set, to produce the final detection results (PEREIRA et al., 2025).

To combine bounding boxes, some strategies include Non-maximum Suppression (NMS) (NEUBECK; GOOL, 2006), Soft-NMS (BODLA et al., 2017), and Weighted Boxes Fusion (WBF) (SOLOVYEV; WANG; GABRUSEVA, 2021). These methods can be used to combine multiple bounding boxes from different object detectors into a single bounding box. The Non-Maximum Suppression (NMS) algorithm refines object detections by discarding predictions with confidence scores below a predefined threshold. Initially, the remaining bounding boxes are sorted in descending order based on their confidence scores. The algorithm iteratively selects the box with the highest confidence, designates it as a valid prediction, and compares it against the other remaining boxes. If the Intersection over Union (IoU) between the selected box and any other prediction exceeds a specified threshold, only the one with higher confidence score is kept. This process continues iteratively, ensuring that only the most representative bounding box is retained for each detected object. Intersection over Union (IoU) can be defined as:

$$\text{IoU}_{Box_1, Box_2} = \frac{\text{Area of Overlap}_{Box_1, Box_2}}{\text{Area of Union}_{Box_1, Box_2}} \quad (2.2)$$

Soft-NMS follows an approach similar to traditional NMS; however, instead of entirely discarding bounding boxes with lower confidence scores, it attenuates their scores using a decay function. This iterative procedure continues until all candidate boxes have been evaluated. Ultimately, predictions with confidence scores falling below a predefined threshold are discarded,

yielding the final set of object detections.

Weighted Boxes Fusion (WBF) presents a different approach by clustering bounding boxes that exhibit a high Intersection over Union (IoU), above a threshold. Instead of discarding overlapping predictions, it employs a fusion function to merge the coordinates of the grouped boxes and computes an updated confidence score. Furthermore, this coordinate fusion process can be fine-tuned by incorporating user-defined weights as parameters.

Our proposed framework also adopts an approach based on combining multiple object detection models, however, it employs a different strategy, which is described in detail in Chapter 3. Unlike approaches that select a single prediction from a specific model, our method recalculates the confidence score of the predictions during the combination process.

3 Proposed Approach

The primary objective of this research is to develop a flexible framework capable of answering queries related to critical objects in images, with particular emphasis on **minimizing false negatives**. To the best of our knowledge, this study presents the first framework that integrates DeepSeek with YOLOv12 for the detection of critical objects, specifically, flooded vehicles and damaged buildings, in disaster management scenarios, while prioritizing high recall.

The main contribution of this work lies in its focus on detecting highly specific objects in challenging real-world conditions, such as vehicles submerged in water and collapsed residential structures where individuals may be trapped. Our approach explicitly prioritizes recall over precision, accepting a potential trade-off in false positives to reduce the likelihood of missing critical instances. This recall-oriented design represents a key distinguishing characteristic of the proposed framework.

The proposed method combines a general detection model with multiple specialized models tailored to detect specific classes of objects. By leveraging complementary models, the framework increases the probability of identifying objects that might otherwise remain undetected. In addition, a ranking mechanism is incorporated to prioritize images according to their confidence score value.

Finally, the system is lightweight and can be deployed on a standard notebook with 16 GB of RAM, ensuring portability and computational efficiency. It is capable of processing and evaluating a large volume of images within minutes, demonstrating practical feasibility for real-world disaster response applications.

This chapter is organized as follows: Section 3.1 presents the YOLO algorithm, Section 3.2 presents the DeepSeek model, and Section 3.3 presents the proposed framework and its main design and implementation aspects.

3.1 Object Detection Module: YOLO Algorithm

You Only Look Once (YOLO) (REDMON et al., 2016a) is an object detection model. It uses bounding boxes to identify objects in images. The network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means the network reasons globally about the full image and all the objects in the image.

Figure 13 presents an illustration of YOLO algorithm. The YOLO design enables end-to-end training and realtime speeds while maintaining high average precision. The system divides the input image into an $S \times S$ grid. If the center of an object falls into a grid cell, that grid

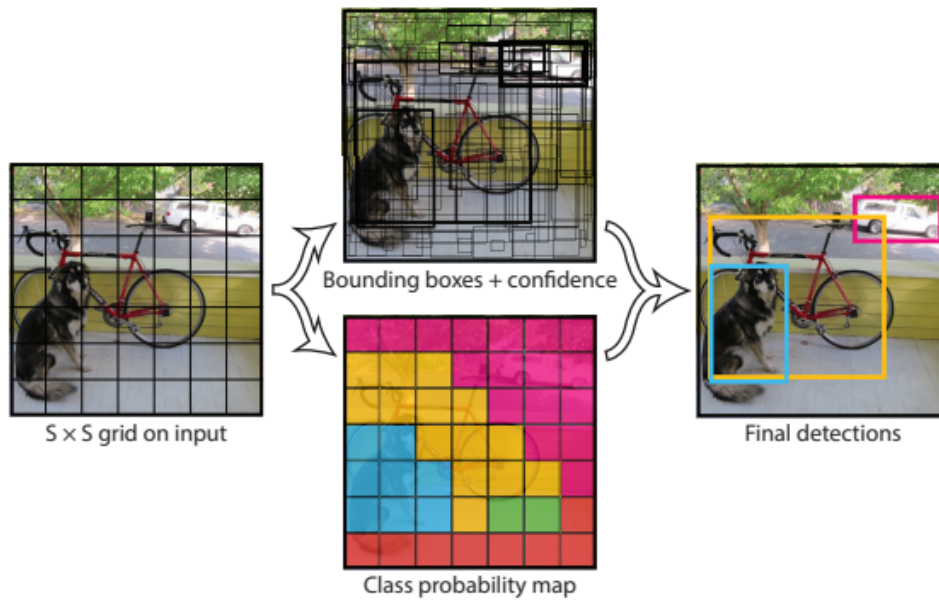


Figure 13 – Illustration of YOLO algorithm: input image is divided into $S \times S$ grid, and each grid cell predicts B bounding boxes and confidence scores (REDMON et al., 2016a).

cell is responsible for detecting that object(REDMON et al., 2016a). Each grid cell predicts B bounding boxes and confidence scores for those boxes(REDMON et al., 2016a).

YOLO's confidence scores measures how confident the model is that the box contains an object, and also how accurate the model thinks the box is in determining the accurate position of the object on the image(REDMON et al., 2016a). If no object exists in that cell, the confidence scores should be zero. Otherwise, it is intended that the confidence score is equal the intersection over union (IOU) between the predicted box and the ground truth(REDMON et al., 2016a).

Each bounding box consists of predictions of:

1. the coordinates representing the center of the box relative to the bounds of the grid cell(REDMON et al., 2016a).
2. the width and height are predicted relative to the whole image(REDMON et al., 2016a).

Additionally, the prediction includes the predicted class of the object(REDMON et al., 2016a).

Confidence is formally defined as:

$$Pr(\text{Object}) \cdot IOU_{\text{truth pred}}(\text{REDMON et al., 2016a}),$$

where $Pr(\text{Object})$ defines the probability of an object of interest, and $IOU_{\text{truth pred}}$ determines the expected IOU of the box with the ground truth. For example, an $IOU_{\text{truth pred}}$ of 0.8 indicates that is expected for the predicted box to have an 80% IOU with the ground truth object(REDMON et al., 2016a). Additionally, YOLO predicts one set of class probabilities for each grid cell.

$$Pr(\text{Class}_i | \text{Object}) \cdot Pr(\text{Object}) \cdot IOU_{\text{truth pred}} = Pr(\text{Class}_i) \cdot IOU_{\text{truth pred}}$$

(REDMON et al., 2016a)

Multiplying the conditional class probabilities by the individual box confidence scores results in class-specific confidence scores. These scores encode: (1) the probability of the box containing an object of a specific class, and (2) how well the predicted box fits that object (REDMON et al., 2016a). This is the final score returned for each prediction made by YOLO.

In this work, YOLO12 (TIAN; YE; DOERMANN, 2025a; TIAN; YE; DOERMANN, 2025b) was employed. YOLO12 enhances the network architecture of the YOLO framework by achieving a balance between the high inference speed of CNN-based models and the performance gains provided by attention mechanisms. Although attention mechanisms can significantly improve accuracy, attention-based models typically struggle to match the real-time performance of CNN-based approaches (TIAN; YE; DOERMANN, 2025a). YOLO12 overcomes this limitation and surpasses end-to-end real-time detectors derived from DETR, such as RT-DETR, RT-DETRv2, and related variants. YOLO12 was selected for this study because, among the tested object detection models, it achieved the highest accuracy while maintaining simple training procedures and real-time execution capability.

3.2 Question Answering Module: DeepSeek Algorithm

DeepSeek(GUO et al., 2025) is a framework that integrates a reasoning model with a large language model (LLM). It is designed to support advanced reasoning patterns, including self-reflection, verification, and dynamic strategy adaptation(GUO et al., 2025). Notably, DeepSeek achieves strong performance on mathematical reasoning and coding tasks while requiring minimal reliance on human-labeled data. In this work, DeepSeek R1 1.5B (GUO et al., 2025) was employed and executed via Ollama. This is a lightweight model with 1.5 billion parameters, developed using the DeepSeek-V3 (DEEPSEEK-AI et al., 2025) pipeline.

DeepSeek V3m (DEEPSEEK-AI et al., 2025) is an advanced open-source large language model developed by DeepSeek that achieves performance comparable to that of larger models, such as OpenAI’s GPT-4 and Meta’s Llama 3.1, while maintaining cost efficiency. The design of DeepSeek R1 prioritizes the relevance of responses to user queries while minimizing interference with the underlying reasoning process (GUO et al., 2025).

Since the task within the proposed framework is to analyze a text file and generate an appropriate response, DeepSeek R1 was selected for its ability to perform textual analysis, produce accurate answers, and maintain the lightweight nature of the framework.

Figure 14 illustrates the architecture of DeepSeek-V3. DeepSeek-V3 builds upon the foundational autoregressive paradigm common to most Large Language Models (LLMs) for next-

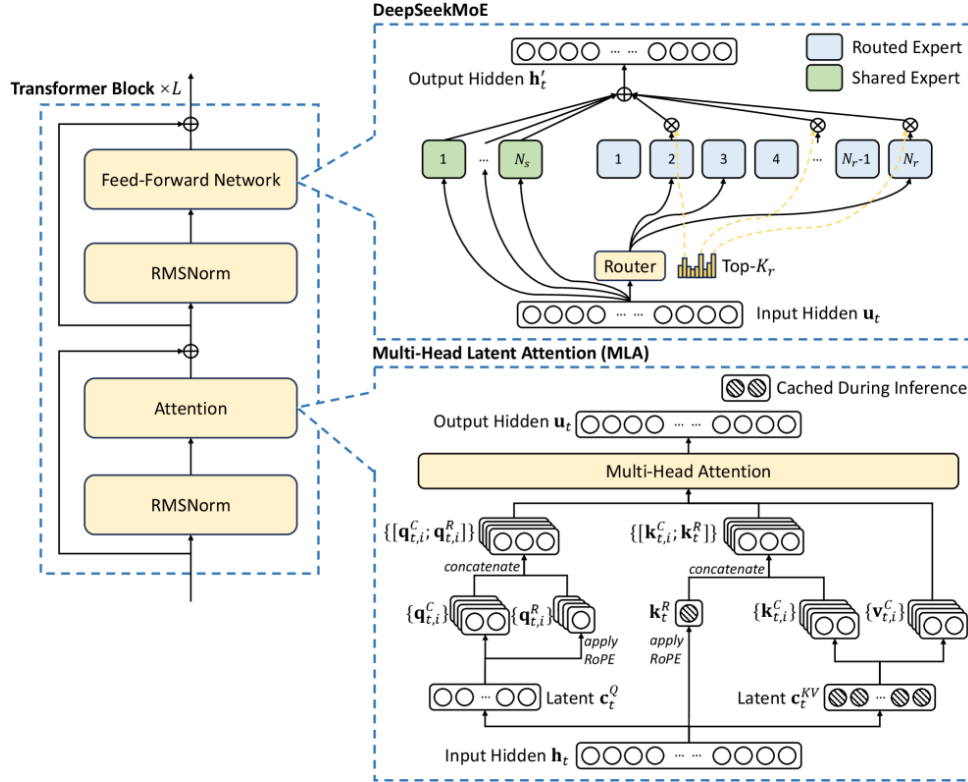


Figure 14 – Illustration of DeepSeek-V3 architecture, presented in (DEEPSEEK-AI et al., 2025).

token prediction. While input tokens are processed through a Transformer-based architecture utilizing attention mechanisms, the model distinguishes itself through two key innovations. First, it features a Mixture-of-Experts (MoE) (DAI et al., 2024) design, which dynamically routes and combines activated experts to process each token. By activating only a specific subset of neurons within the feed-forward network, the model achieves significantly higher computational efficiency. Second, the integration of Multi-head Latent Attention (MLA) (DEEPSEEK-AI et al., 2024) drastically reduces the KV cache memory footprint while maintaining optimal performance.

These characteristics make DeepSeek-V3 suitable for the present work, as they enable strong performance and advanced machine reasoning while requiring relatively modest computational resources. The prompts and settings used with DeepSeek in this work are described in Chapter 5.

3.3 Proposed Framework

The main goal of this research is:

Design and construct a flexible framework that is able to answer simple, pre-defined questions about disasters, based on images obtained from affected areas with high accuracy, especially **with low false negatives**.

The system flows as the steps illustrated in Figure 15:

1. User provides to the system the following information: the question to be answered, the directory containing the images, and the type of objects the user wishes to scan for. At present, the available object categories are cars and houses. Additional object models can be incorporated in the future.
2. The system processes the images using the pre-trained models corresponding to the selected objects. At this step, Object Detection Module does not consider the question provided by the user.
3. The objects detected by the object detection models are used to compute combinations. The implemented combinations are presented later in this Chapter.
4. The predictions are used to compute ranks of objects.
5. The results are stored and subsequently passed to DeepSeek via text files (the impact of textual file features is discussed in Chapter 5).
6. DeepSeek generates an answer to user's question, based on the content of the file provided by ODM.
7. The answer is formulated to the user.

Steps 3 and 4 of Figure 15 illustrate the two outputs produced by the framework: predictions and confidence rankings. Predictions correspond to the bounding boxes generated by the object detection models, each associated with a predicted object class and a confidence score. The confidence rankings are generated according to predictions confidence scores, such that predictions with higher confidence appear at the top of the ranking.

The objective of this confidence ranking approach is to support the identification of objects that are most likely to represent critical situations and therefore require immediate attention, while distinguishing those that can be considered lower priority. In this context, the primary goal is to minimize the risk of misclassification that could result in overlooking critical objects, thereby reducing false negatives and improving the reliability of the system for disaster response scenarios.

Figure 16 illustrates a hypothetical use of the proposed framework in a real-world scenario. The same example is presented in Chapter 5 for a more detailed evaluation. As illustrated in Figure 16, the framework can be used to analyze a set of scenes and determine which locations should be attended to first. The framework is able to receive a query, analyze a set of images, and report the results to the user, such as the number of priority objects detected according to different combination strategies or different models.

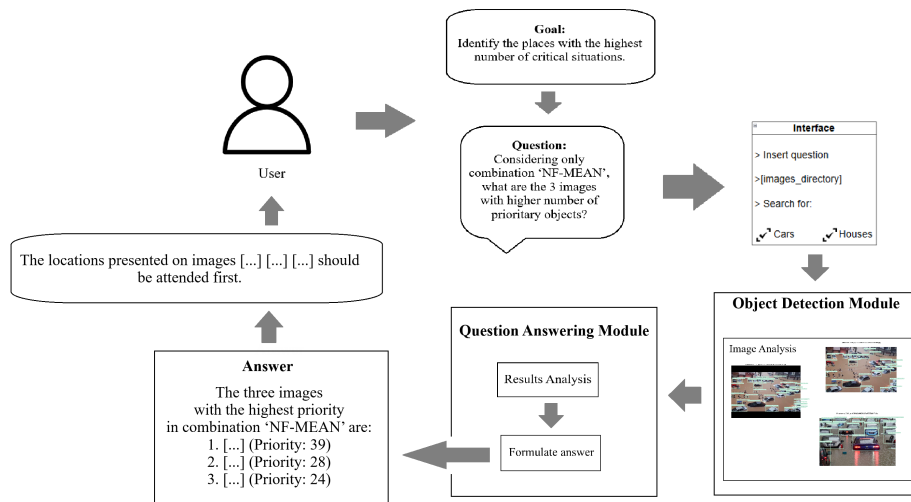


Figure 16 – Illustration of a hypothetical use of the proposed framework.

The ODM generates predictions from the general model, each specialist model, and five different combinations of predictions, as detailed next in this Chapter. In addition, the framework produces one ranking for the general model, one ranking for the NF-MEAN combination, and one ranking for the NF-SUM combination. All these rankings are sorted by confidence. Both combinations are described next.

Additionally, the system generates one priority ranking for each combination. These priority rankings correspond to the top k images containing the highest number of detected priority objects. This ranking is used for DeepSeek to be able to answer to the user what are the most relevant images found. The priority ranking differs from the confidence ranking in that it ranks images rather than individual objects. Furthermore, the ordering criterion is based on the number of detected critical objects within each image, rather than on the confidence scores of individual predictions, as used in the confidence ranking.

To evaluate the framework, the existent classes were classified into two categories: priority (or critical) and not priority (or not critical). Priority objects are those that require immediate attention due to the high likelihood of representing critical situations. In contrast, non-priority objects are those that do not present a high probability of risk and therefore do not require urgent intervention.

Both datasets used in this research contain four classes, defined according to the level of damage or flooding. Objects belonging to classes 2 and 3 are considered priority, as they represent vehicles in high water levels or houses with severe structural damage. In contrast, objects from classes 0 and 1 are classified as non-priority, as they correspond to vehicles with little or no water exposure, or houses with slight or no damage. Consequently, these classes are designated as priority (critical) classes.

To evaluate the predictions, the following criteria were used:

- If no objects are present but the model predicts one or more objects, the result is counted as a False Positive per object;
- If objects are present but the model predicts no objects, the result is counted as a False Negative per object;
- If there is no box with $\text{IoU} \geq \text{IoU Threshold}$ for a ground truth, counts as False Negative (missed object);
- If the $\text{IoU} \geq \text{IoU Threshold}$ between predicted box and ground truth, and prediction's class "priority status" match the ground truth's "priority status" (i.e., are they both Priority classes, or both not Priority classes), counts a True Positive;
- If $\text{IoU} \geq \text{IoU Threshold}$ between ground truth and predicted object, but model predicted a Non-Priority class but it was actually Priority: Increment FN (missed a High Priority object);
- If $\text{IoU} \geq \text{IoU Threshold}$ between ground truth and predicted object, but the model predicted a Priority class but it was actually Non-Priority: Increment FP (raised a false alarm);

This approach has the particular characteristic of considering critical objects predicted as non-critical as false negatives. This definition follows from the specific nature of our scope: classifying a priority object as non-priority corresponds to a missed alarm that should have been raised, whereas classifying a non-critical object as priority corresponds to raising a false alarm. This evaluation criterion was considered during the development of the framework presented in this work. ODM has two main sets of models: one for cars and one for houses. Each set is composed by one general model, and one specialized model per class.

The purpose of employing specialist models is to allow each model to learn exclusively a single class and to evaluate whether they are able to detect objects that are missed by the general model. This approach aims to investigate whether this training strategy enables detection models, by focusing on a specific class, to identify objects that might be missed by a general model. Through a combination strategy, these objects could then be recovered for use within the framework.

All models used in this work are YOLOv12n(TIAN; YE; DOERMANN, 2025a; TIAN; YE; DOERMANN, 2025b), initialized with weights pre-trained on the COCO dataset, obtained from Ultralytics(JOCHER; QIU; CHAURASIA, 2023). The General model and specialized models differ by training:

- **General model/Mixed General:** trained with the set of train, without any modifications. Learns all existent classes.



Figure 17 – Image (SEDANFROMHX, 2024) with only class 1 labeled (used to train Only L1 model).



Figure 18 – Image (SEDANFROMHX, 2024) with only class 2 labeled (used to train Only L2 model).



Figure 19 – Image (SEDANFROMHX, 2024) with only class 3 labeled (used to train Only L3 model).



Figure 20 – Image (SEDANFROMHX, 2024) with all classes labeled (used to train Mixed General model).

- **Specialist models:** Called Only LX, where X refers to the class the model learns. The specialist models train on the same train set as the general model, but with one difference: they only receive images and labels with objects of class X. If an image has no objects of X class, specialist does not receive this image during training. If an image contains objects of class X and other classes, specialist receives the image, but only class X is labeled. Objects of other classes are treated as background. In the context of this application, classes with higher values have higher priority. Therefore, the higher X is, the higher the model's priority on the priority list.

Figures 17, 18, 19 and 20 present the same image, originally from “yolov5WithSUV Computer Vision Model” (SEDANFROMHX, 2023) dataset (explained in Chapter 4) with the labels used to train each model. Mixed General model receives all labels. Models Only LX received the same image, but only with class X labeled. All models were trained for 200 epochs.

To validate this approach, an experimental evaluation was conducted. Figures 21 and 22 present the number of exclusive detections made by each specialist model on the test set, for cars and houses datasets. That is, it shows how many objects were detected by a given specialist model while being missed by all other models. The column “shared” represents the number of

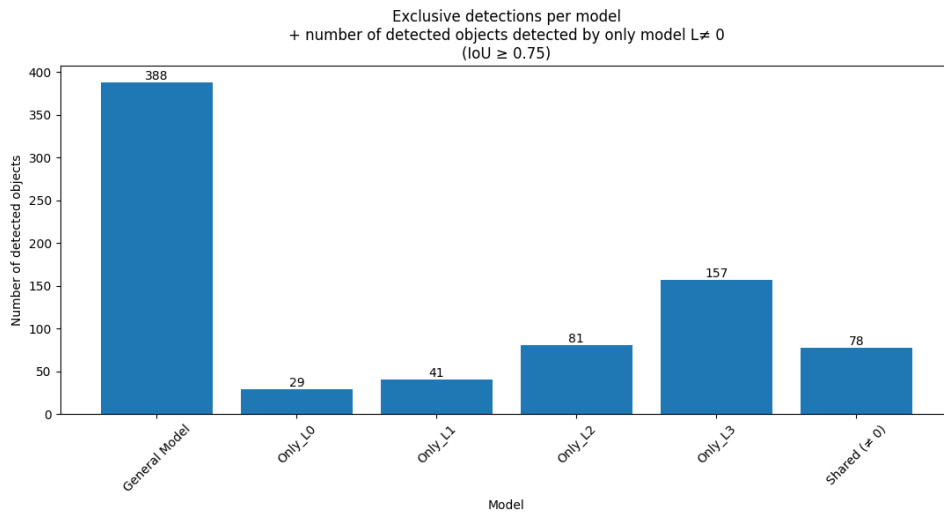


Figure 21 – Number of existent objects found by each model exclusively on **cars dataset**.

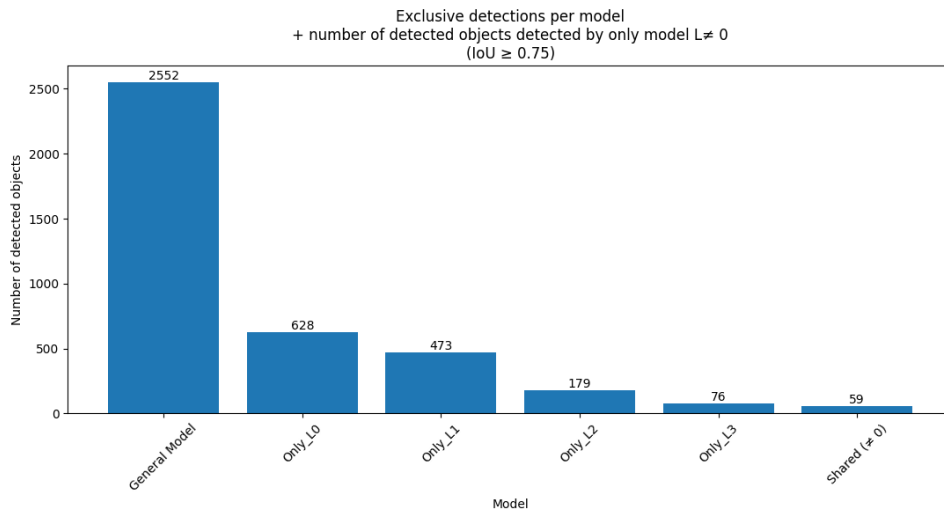


Figure 22 – Number of existent objects found by each model exclusively on **houses dataset**.

objects missed by the general model but detected by at least two specialist models.

This preliminary result, shown in Figure 21, indicates the potential for improved detection performance through the use of specialist models compared to relying solely on the general model. By analyzing the results shown in Figure 21, we obtain a total of 386 detections, computed as 29 (Only L0) + 41 (Only L1) + 81 (Only L2) + 157 (Only L3) + 78 (shared by at least two models but missed by the general model). This result indicates a potential improvement of 386 object detections that were missed by the general model but are able to be recovered through the use of specialist models. Specially, most objects missed by general model, but found by specialists, are probably from priority classes for cars dataset, since most detections are from models Only L3 and Only L2. For houses, as shown in Figure 22, the potential of improvement is 628 (Only L0) + 473 (Only L1) + 179 (Only L2) + 76 (Only L3) + 59 (shared), that is 1,415 object that were lost by general model, but are able to be recovered from specialized models.

Nevertheless, the general model remains the most reliable overall. Therefore, for

model combinations, the following priority order was assumed: The general model has the highest priority, as it is able to identify all classes and, for this reason, has a lower probability of missing an object from any class. If the general model misses an object that is detected by specialist models, priority is given to the detection from the specialist associated with the highest-priority class, in order to favor objects belonging to priority classes. All the combinations were created based on the following algorithm:

1. For each object found, groups all predictions of the same object. Predictions are considered to be from the same object if $IoU \geq IoU_{threshold}$. In this research, a threshold value of 0.75 was used in the reported experiments.
2. For each group:
 - a) Sets a list of models by priority, with the following order: Mixed General, Only L3, Only L2, Only L1, Only L0;
 - b) Chooses the prediction of the model with higher priority, between those that found the object;
 - c) Calculates the new confidence, according to type of combination;
3. returns one chosen prediction per object found.

The priority order was designed to prioritize high-importance objects. Model Mixed General was assigned the highest priority, as it is the general model and, generally, the most reliable. If Mixed General model fails to detect an object but other models succeed, we prioritize the prediction of the specialist corresponding to the highest-priority class among those that detected the object.

Figure 23 illustrates the process used to construct the model combinations. The flowchart begins at step “Get a model in the list of models”. In each combination, the bounding box and the predicted class of an object are selected from the model with the higher preference. Subsequently, a new confidence score is calculated for the resulting prediction. Combinations F06, F05, and F03 (described in detail later in this Chapter) apply an additional constraint: bounding boxes with confidence scores equal to or lower than their respective threshold values are not selected.

The new confidence score depends of the combination. The difference between combinations are:

- **Combination NF-MEAN** (No Filter - Mean): considers all predictions and selects the prediction from the highest-priority model among those that detected the object. The new confidence score is computed as the mean of all predictions, i.e., the average of the confidence scores assigned by the models that detected the specific object.

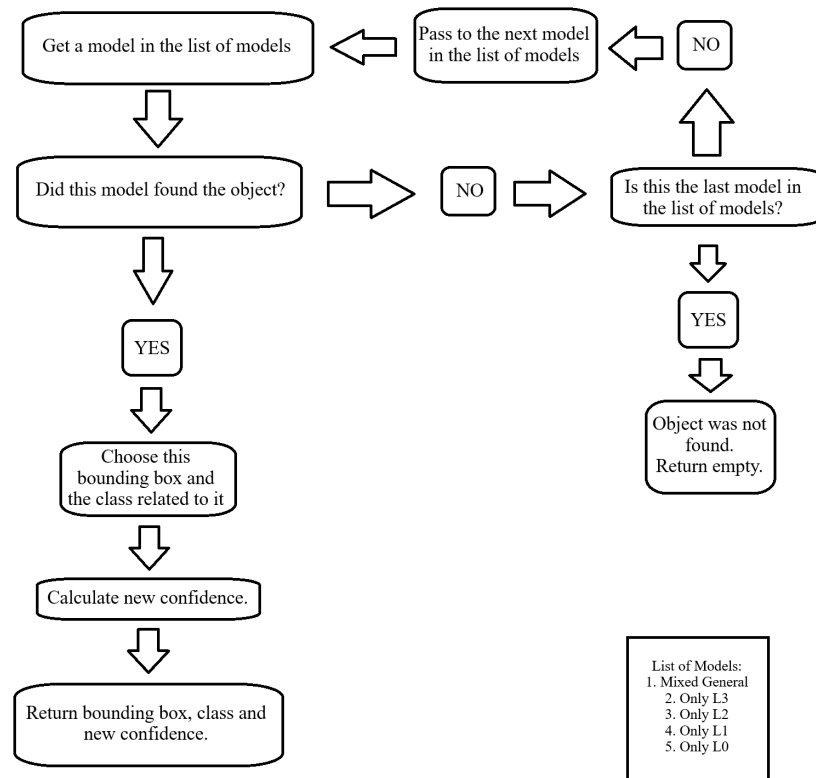


Figure 23 – Representative illustration of the construction of combinations.

- **Combinations F-06-03** (Filter - Threshold 06-03): ignores predictions whose confidence is less than or equal to the threshold. Like NF-MEAN, it selects the bounding box from the highest-priority model among those that detected the object, but differs by discarding predictions whose confidence is lower or equal than the filter value. If none of the predictions in the group exceeds the confidence threshold, no prediction is returned and the object is considered lost by all models. The new confidence score is computed as the mean of all predictions associated with the object, including those whose confidence is below the respective threshold. The threshold values tested were 0.6 (F06), 0.5 (F05), and 0.3 (F03).
- **Combination NF-SUM** (No Filter - Sum): considers all predictions and selects the one from the highest-priority model among those that detected the object. The new confidence score is computed as the sum of the confidence scores of all predictions associated with that object.

Combinations NF-MEAN and NF-SUM produce the same bounding boxes, but the confidence scores differ. The scores in Combination NF-SUM are not normalized and can exceed 1.0. This difference is used to generate the rankings. Since Combination NF-SUM sums the scores, it prioritizes objects detected by multiple models, which are likely to occupy higher positions in the ranking. In contrast, Combination NF-MEAN prioritizes objects detected with high individual scores.

The new confidence score of an object, for the NF-MEAN combination and for F03, F05, and F06, is computed using the following function:

$$\text{ConfScore}_c = \frac{1}{|M_d|} \sum_{m \in M_d} \text{ConfScore}_m, \quad (3.1)$$

where:

- ConfScore_c represents the final aggregated confidence score for the detected object;
- M_d denotes the subset of models that identified the object, drawn from the complete ensemble of models {Mixed General, Only L0, Only L1, Only L2, Only L3};
- $|M_d|$ is the cardinality of the subset M_d , corresponding to the total number of valid predictions for the given object;
- ConfScore_m indicates the individual confidence score assigned by the specific model $m \in M_d$.

For NF-SUM combination, the new confidence score of an object is computed with the function:

$$\text{TotalConfScore} = \sum_{m \in M_d} \text{ConfScore}_m, \quad (3.2)$$

where:

- TotalConfScore represents the cumulative confidence score derived from all models that detected the object;
- M_d and ConfScore_m retain their previously established definitions, denoting the subset of valid models and their respective individual confidence scores.

Consider a hypothetical object A detected by a single model with a confidence score of 0.8, and another object B detected by four different models with confidence scores of 0.2, 0.3, 0.22, and 0.21. Under the NF-MEAN combination strategy, object A retains a confidence score of 0.8, while object B receives an average score of 0.2325. In contrast, under the NF-SUM strategy, object A again obtains a score of 0.8, whereas object B achieves a cumulative confidence score of 0.93. As a result, the ranking of the objects differs substantially depending on the aggregation method. Using NF-MEAN, object A is ranked significantly higher than object B. However, under NF-SUM, object B surpasses object A in the ranking.

Combinations F06, F05, and F03 are designed to cover a narrow range of confidence thresholds. F05 corresponds to the mean value of the confidence range that can be achieved. The objective of these confidence thresholds is to evaluate different values in order to achieve a balance between filtering out incorrect predictions and retaining accurate ones.

The framework generates rankings for the Mixed General model, the NF-MEAN combination, and the NF-SUM combination, sorting the rankings by confidence. Since combinations F06, F05, and F03 calculate the new confidence score like NF-MEAN and filter predictions by score, the top positions in their rankings should be the same as those in the NF-MEAN ranking.

Since the confidence scores of NF-SUM are not normalized, combinations with varying numbers of models operate on different score scales. Therefore, each combination C generates its own independent ranking. Let O_C be the set of n_C objects detected by combination C , and R_C is the ordered sequence (ranking) of objects detected specifically by combination C . The ranked list for this specific combination is defined as:

$$R_C = \langle o_1^C, o_2^C, \dots, o_{n_C}^C \rangle \quad \text{subject to} \quad \text{Score}_C(o_i^C) \geq \text{Score}_C(o_j^C) \quad \forall 1 \leq i < j \leq n_C, \quad (3.3)$$

where:

- C represents a specific combination being evaluated;
- o_i^C denotes the object at the i -th rank position within the isolated context of combination C ;
- n_C is the total number of valid objects detected by combination C ;
- $\text{Score}_C(o)$ represents the non-normalized confidence metric (e.g., the cumulative score) for object o , computed exclusively for combination C .

Since the raw confidence scores are not directly comparable across different combinations, our comparative analysis relies strictly on rank-based evaluation. Rather than assessing absolute score values, we measure performance based on the ordinal position of the objects within each independent ranking. We adopt a top- k evaluation strategy: for a given rank threshold k , we count the number of detections of each category (detailed in Chapter 5) present in the first k positions of a combination's ranked list. For example, combination X might yield n correct priority objects in its top- k ranks, while combination Y yields m correct priority objects, allowing for a robust, scale-invariant comparison between the models.

The proposed framework is computationally efficient, enabling execution within a standard laptop environment equipped with 16GB of RAM. To achieve this level of performance, lightweight variants of the YOLO and DeepSeek architectures were employed. The reported experiments were executed on a laptop with Intel Core i5-9300H CPU, NVIDIA GeForce GTX 1650 GPU, Ubuntu 22.04.5 LTS operational system and 16GB of RAM.

4 Datasets

This chapter presents the datasets used in the present work to train, validate, and evaluate the object detection models within the proposed framework.

DeepSeek’s evaluation was conducted entirely manually, using human-authored questions and human-evaluated responses. Due to the specific nature of the proposed framework, manually revised questions and evaluation criteria were developed to assess DeepSeek’s answers. This approach was particularly necessary because multiple formulations of a response to the same question may all be considered correct.

To train, validate, and evaluate the object detection models, two datasets were created: the Cars dataset and the Houses dataset. Both were developed using publicly available images and annotations, with some relevant modifications, as described below.

Both datasets used in the present study (cars dataset and houses dataset) were weakly supervised. Given the limited number of available images in the training sets, this constitutes a particularly challenging scenario, especially when considering both the scarcity of data and the intrinsic characteristics of the images, such as high water levels and partially visible objects.

In the flooding scenarios, there is very little labeled public data about Brazil. Although there are a lot of public videos made by drones, aerial images, and phone pictures, most of them are protected by copyright. Some of the copyright holders have specific clauses forbidding the use of images and videos for machine learning purposes. For this reason, it was impracticable to create our own database based on public images, since it would be a copyright violation for all the images and videos researched.

Alternatively, the datasets used in this research are from Roboflow ([DWYER; NELSON; HANSEN, 2024](#)). Roboflow is a framework for creating computer vision models, with mainly focus on industrial and academic applications. They also offer a platform where one can update their own datasets. Most of the datasets have licenses that allows public use of the images. The datasets were chosen and combined to best fit research’s best approach.

4.1 Cars Dataset

Two datasets were employed in this research to compose the final cars dataset: Dataset “convert2yolov5 Computer Vision Dataset” ([SEDANFROMHX, 2023](#)) and dataset “yolov5WithSUV Computer Vision Model” ([SEDANFROMHX, 2024](#)). Both are available via Roboflow.

Figure 24 presents images from dataset “convert2yolov5 Computer Vision Dataset” ([SEDAN-](#)



Figure 24 – Images from “convert2yolov5 Computer Vision Dataset” (SEDANFROMHX, 2023) dataset. Figure presents examples of objects of class 0 (L0) and class 3(L3)

FROMHX, 2023), that comprises four classes that represent different levels of water surrounding a vehicle. Class 0 corresponds to a vehicle not in water, class 1 indicates slight water presence, class 2 denotes a vehicle partially submerged in water, and class 3 represents a vehicle in high water.

Figure 25 presents images from “yolov5WithSUV Computer Vision Model” (SEDANFROMHX, 2024) dataset. It consists of five classes that indicate different water levels. Class 0 represents a vehicle not in water, while Class 4 corresponds to the highest water level, in which the vehicle is submerged. Figure 25 presents representative examples of the defined classes, illustrating instances of class 0 (not in water), class 1 (slightly submerged), class 2 (moderately submerged), and class 3 (almost fully submerged).

To construct the car dataset for the present work, images and corresponding labels from both datasets were merged into a single dataset and subsequently underwent the following modifications: all instances originally labeled as class 4 were reclassified as class 3, effectively merging classes 3 and 4 into a single category. Images without any labels were disconsidered. Additionally, the images were randomly partitioned into training, validation, and test sets. The images were partitioned into training, validation, and test sets using proportions of 15% for training, 5% for validation, and 80% for testing. This process resulted in a dataset comprising 3,036 images in the test set, 568 images in the training set, and 215 images in the validation set.

Cars dataset has the distribution presented on Table 1.



Figure 25 – Examples of dataset “yolov5WithSUV Computer Vision Model” (SEDAN-FROMHX, 2024). Figure presents examples of objects from classes 0 (L0), class 1 (L1), class 2 (L2) and class 3 (L3).

Table 1 – Distribution of objects per class, on **cars dataset**, for each set.

Set	Class 0	Class 1	Class 2	Class 3	Total
Test	961	1575	1638	2089	6263
Train	180	295	309	392	1176
Validation	60	98	101	128	387

4.2 Houses Dataset

Dataset “Building Damage Detection-undef Dataset”(ROBOFLOW, 2024), available via Roboflow, was used as the houses dataset. The original dataset consists of 2,692 images in the training set, 258 images in the validation set, and 131 images in the test set. To ensure that all models within the framework operate under a weakly supervised learning scenario, the dataset splits were inverted: the original training set was used as the test set, the original validation set was used as the training set, and the original test set was used as the validation set. It results in an adapted version of the dataset, with 2,692 images in the testing set, 258 images in the training set, and 131 images in the validation set. The partition used in this research was 87% for test, 8% for train set, and 4% for validation set.

Figure 26 presents examples of dataset “Building Damage Detection-undef Dataset”. The dataset comprises four classes: class 0 corresponds to undamaged buildings, class 1 to slightly damaged buildings, class 2 to damaged buildings, and class 3 to severely damaged buildings. The class definitions were not altered within the use at the framework’s construction.



Figure 26 – Images from “Building Damage Detection-undef Dataset” (ROBOFLOW, 2024). The images presents objects belonging to classes 0 (not destroyed), 1 (slightly destroyed), 2 (moderately destroyed), and 3 (heavily destroyed).

Table 2 – Distribution of classes and objects per set, in **houses dataset**.

Set	Class 0	Class 1	Class 2	Class 3	Total
Test set	10257	7174	2785	1467	21683
Train	1021	629	252	157	2059
Validation	435	316	165	57	973

For both datasets, duplicate objects were removed from ground truth labels at the evaluation of the system.

Table 2 presents the distribution of objects per class, for each set, on houses dataset.

5 Experimental Evaluation

This chapter presents the results obtained from the proposed framework, including the evaluation of combination strategies, ranking mechanisms, and the question-answering module.

5.1 Confidence Analysis

A comparison between the confidence scores of the Mixed General model and the specialized models was conducted to determine whether a confidence threshold could effectively distinguish false positives from true positives. The objective of this analysis was to assess whether such a threshold could be used to increase recall without incurring a loss in precision. For this particular analysis, an object was considered correct if its position on the image corresponds with ground truth box. The attributed class was not taken into consideration.

Figure 27 presents a comparison of the confidence scores produced by the Mixed General model and each specialized model, on cars dataset. Each point in the graph represents a single object. Green points correspond to objects found correctly, without considering the attributed class, for at least one model; whereas red points indicate ghost detections (false positives), detections made where there was no object. The purpose of this graph is to compare the different models and determine whether a confidence score threshold can effectively separate ground-truth instances from false positives, considering the boxes only. The purpose of this evaluation is to determine if there is a threshold value that can separate ghost detections from true objects.

Figure 27 shows that false positive cases are broadly distributed across the graph, indicating that no confidence score threshold can reliably separate true positives from false positives precisely. This implies a trade-off between recall and precision: increasing recall will inevitably lead to a decrease in precision, as identifying additional true-positive objects also results in more false positives, given the absence of a threshold that can distinguish them accurately.

Figure 28 presents the confidence scores for the houses dataset. The same pattern is observed: no threshold can reliably separate true positives from false positives, which implies that increase recall will result on lower precision values.

The combination threshold values were selected to span a range of confidence score thresholds, allowing the evaluation of their impact on the final results. Based on the results presented in Figures 27 and 28, higher confidence thresholds are associated with increased precision at the expense of recall. Conversely, lower confidence thresholds lead to higher recall,

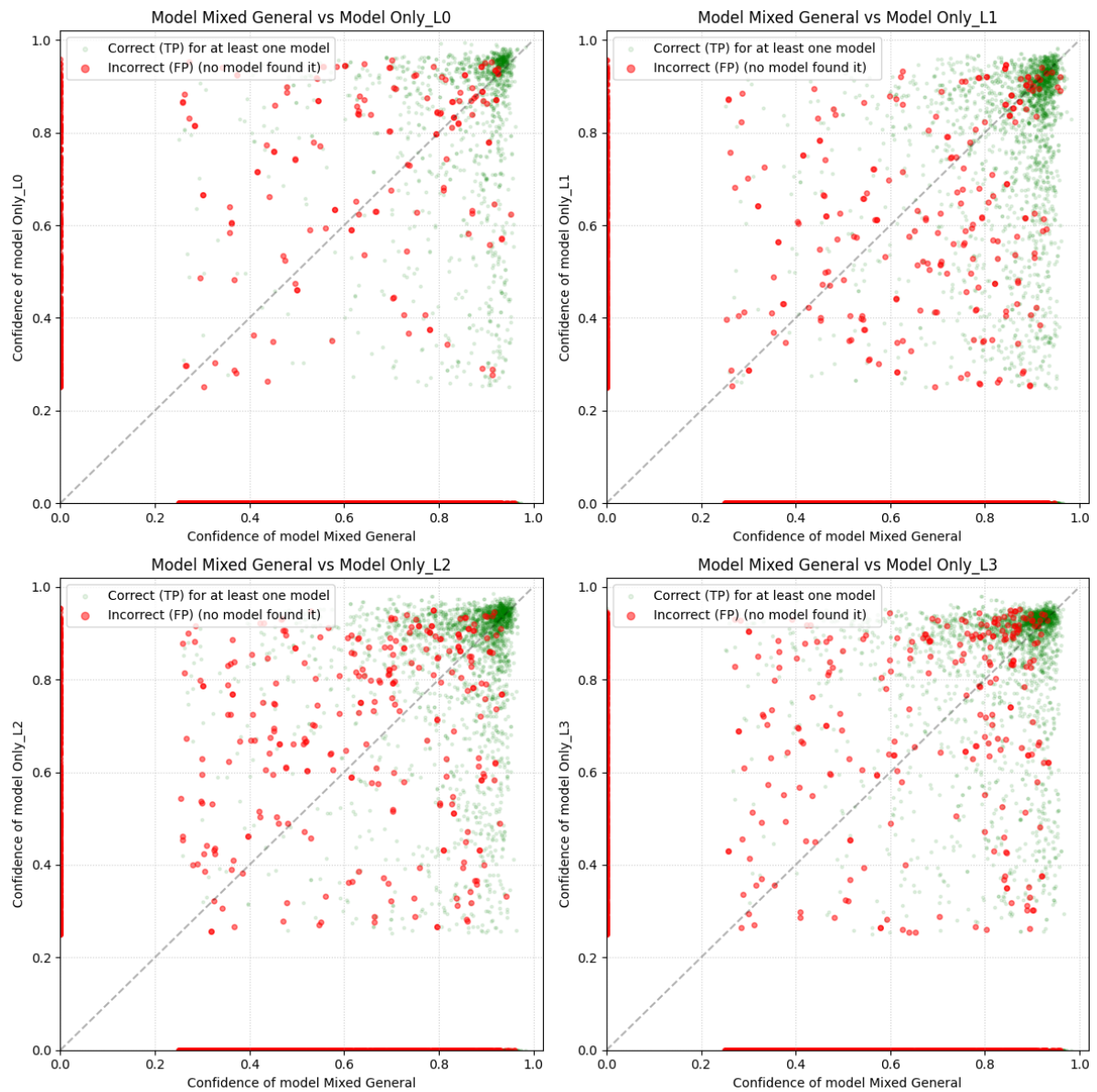


Figure 27 – Confidence comparison between Mixed General and specialized models, on **cars** dataset.

albeit with a corresponding trade-off in precision.

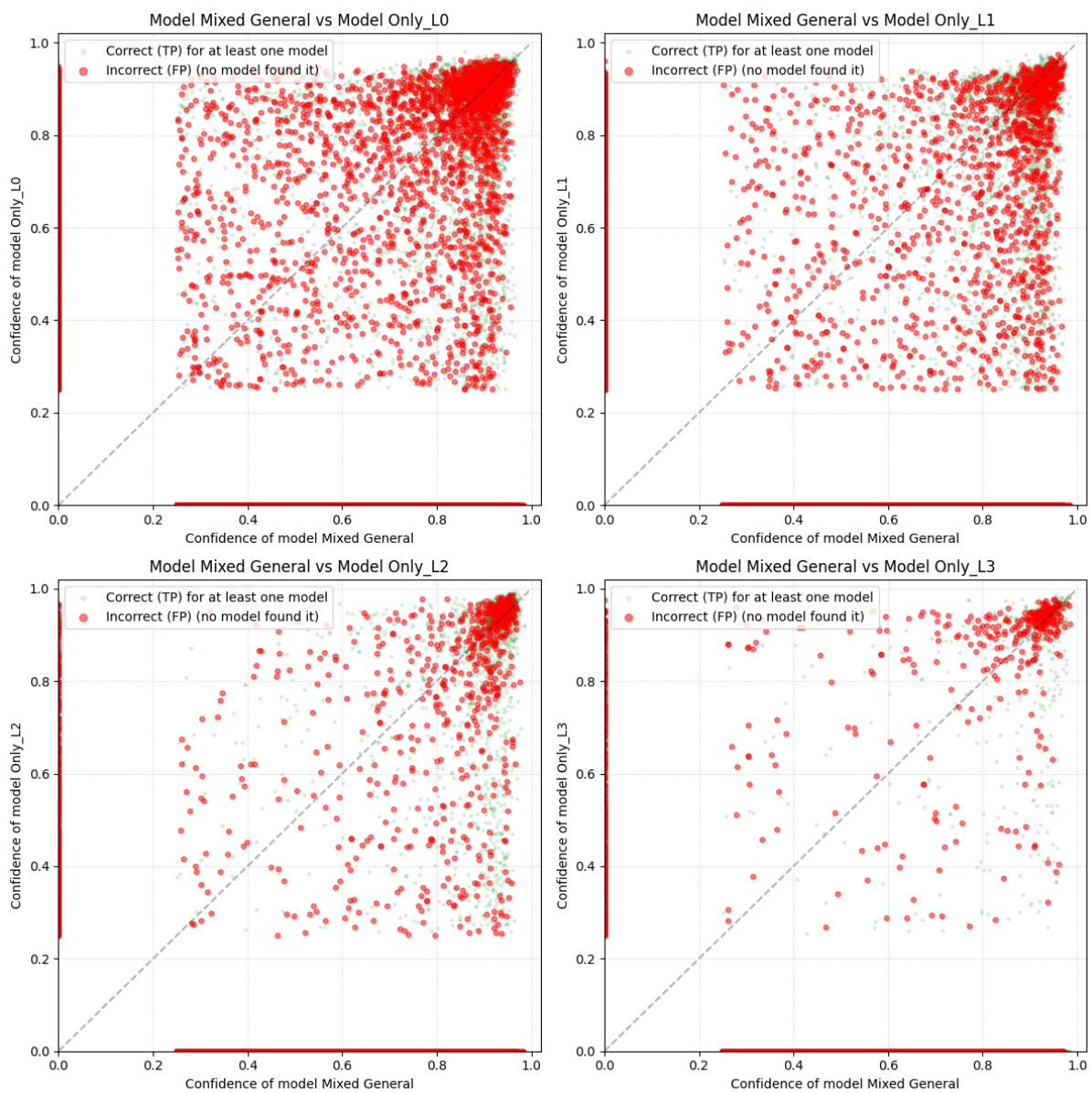


Figure 28 – Confidence comparison between Mixed General and specialized models, on **houses** dataset.

5.2 Object Detection Module on Cars Dataset

5.2.1 General Results in Object Detection

Table 3 presents the results of the system, considering all objects, on cars dataset. Combinations NF-MEAN, NF-SUM and F03 achieved higher recall than model Mixed General, but at the cost of precision. Since correct bounding boxes can become false positives or false negatives due to classification, and the logic of combinations prevents losses of objects found by General Model, the loss of precision might be due to missclassification coming from specialist models.

Table 3 – Results for **cars dataset**, for all models and combinations, considering all predictions.

Model	True Positives	False Positives	False Negatives	Errors	Precision	Recall	F1 Score
Mixed General	4154	2181	1848	4029	0.655722	0.692103	0.673421
Only L0	1039	603	5224	5827	0.632765	0.165895	0.262872
Only L1	1526	861	4737	5598	0.639296	0.243653	0.352832
Only L2	1616	1857	3719	5576	0.465304	0.302905	0.366939
Only L3	2197	1548	3554	5102	0.586649	0.382021	0.462721
Combination NF-MEAN	4395	4496	1556	6052	0.494320	0.738531	0.592238
Combination F06	4143	2224	1821	4045	0.650699	0.694668	0.671965
Combination F05	4223	2724	1743	4467	0.607888	0.707844	0.654070
Combination F03	4368	4030	1594	5624	0.520124	0.732640	0.608357
Combination NF-SUM	4395	4496	1556	6052	0.494320	0.738531	0.592238

Although Mixed General achieves higher precision and a higher F1 score, the NF-MEAN and NF-SUM combinations exhibit higher recall, a greater number of true positives, and a lower number of false negatives.

Table 4 presents the number of objects detected by each model for each class, on cars dataset, considering the complete set of ranked results. By comparing Tables 1 and 4, it is notable that there were extra predictions for all classes, except for Mixed General with classes 2 and 3. It might be due to undetected objects, or missclassification errors, for example, class 2 or 3 objects being wrongly classified as classes 0 or 1.

Figure 29 presents the confusion matrices of the specialized models on the cars dataset. Although some misclassifications occur across all classes, it can be observed that most errors generally occur between neighboring classes. Due to their nature of training, each specialized model is able to predict only objects of its own learned class.

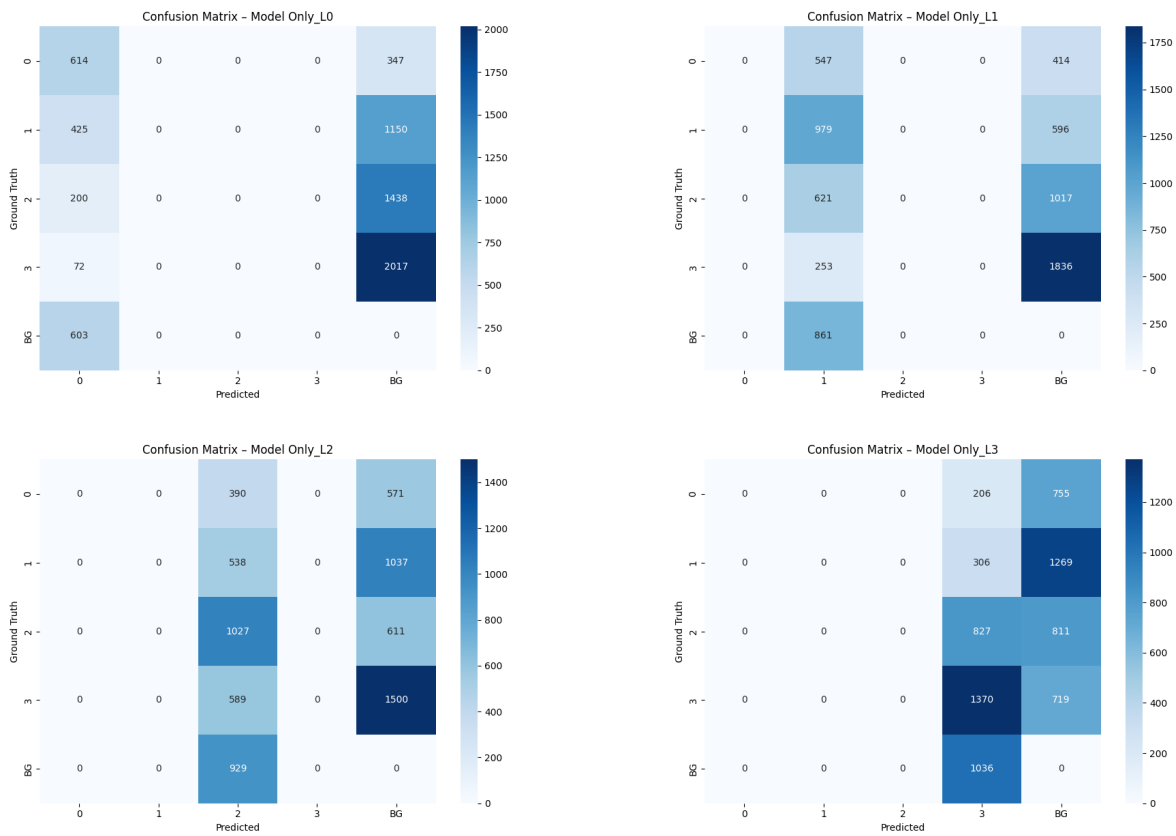


Figure 29 – Confusion matrices of specialized models for **cars datasets**.

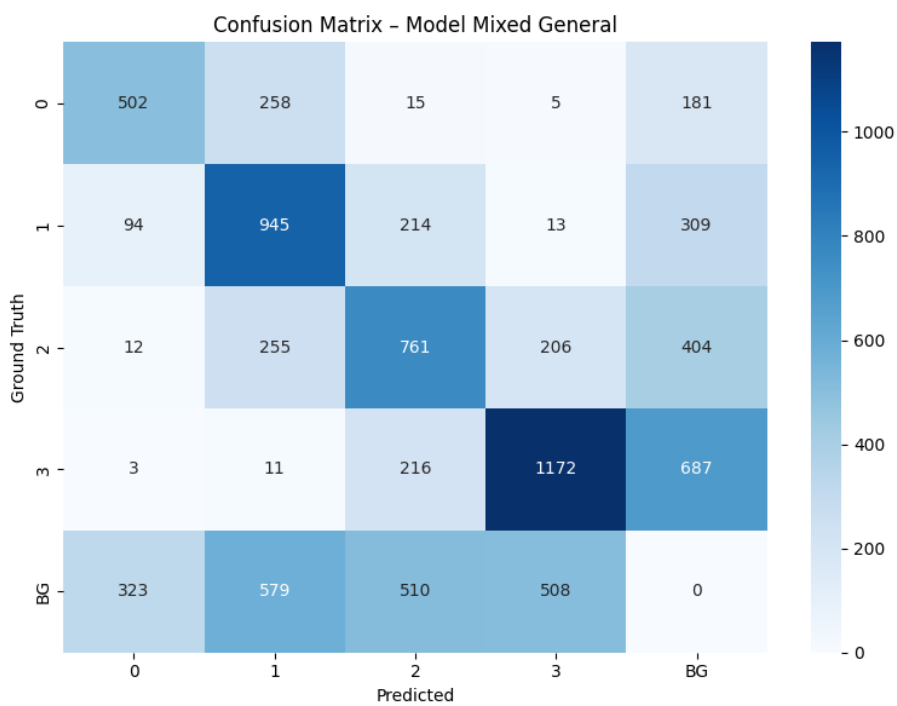


Figure 30 – Confusion matrix of Mixed General model on **cars dataset**.

Table 4 – Number of objects of each class found by each model, considering ranking results, on **cars dataset**

Model	Class 0	Class 1	Class 2	Class 3	Priority	Not Priority
Combination NF-MEAN	1354	2659	2374	2790	5164	4013
Mixed General	2048	1716	934	1904	3620	2982
Combination NF-SUM	1354	2659	2374	2790	5164	4013

Table 5 – Recall, considering only priority objects, for the first 300 positions of the ranking of each model, in **cars dataset**.

Model	Number of existent critical objects	Number of detected critical objects in first 300 positions	Recall (considering priority objects only)
NF-MEAN	3727	135	0.03622
Mixed General	3727	128	0.0343
NF-SUM	3727	97	0.0260

Figure 30 presents the confusion matrix of the Mixed General model on the cars dataset. The main diagonal is highlighted, indicating promising results. As also observed in Figure 29, most errors occur between neighboring classes, although misclassifications are present across all classes. A significant number of priority objects (classes 2 and 3) were missed (predicted as background). However, there are relatively few cases in which class 3 objects are predicted as non-priority classes (classes 0 and 1). A significant number of priority objects from class 2 were predicted as not priority (class 1).

By combining the information presented in Tables 1 and 4, we can calculate the recall considering only priority objects. That is, among all existing critical objects, how many were detected within the first k positions of each ranking? The results are presented on Table 5. Naturally, the recall in this evaluation has a limited maximum value, since only the first 300 positions are considered. However, this result indicates how many critical objects were detected among the top 300 confidence scores. Other critical objects may still be present in the ranking at lower-confidence positions. The higher value was presented by NF-MEAN, followed by Mixed General. It indicates that NF-MEAN found more critical objects in the first 300 positions of the ranking, in comparison to other approaches.

5.2.2 Ranking Results

The intended principle underlying the proposed combinations is to rely primarily on the most reliable model (Mixed General). When this model fails to detect an object, the combination strategy selects the prediction associated with the highest assigned class. Consequently, priority predictions are expected to obtain an advantage within the ranking process. Although this mechanism does not directly modify the confidence scores based on the assigned class,

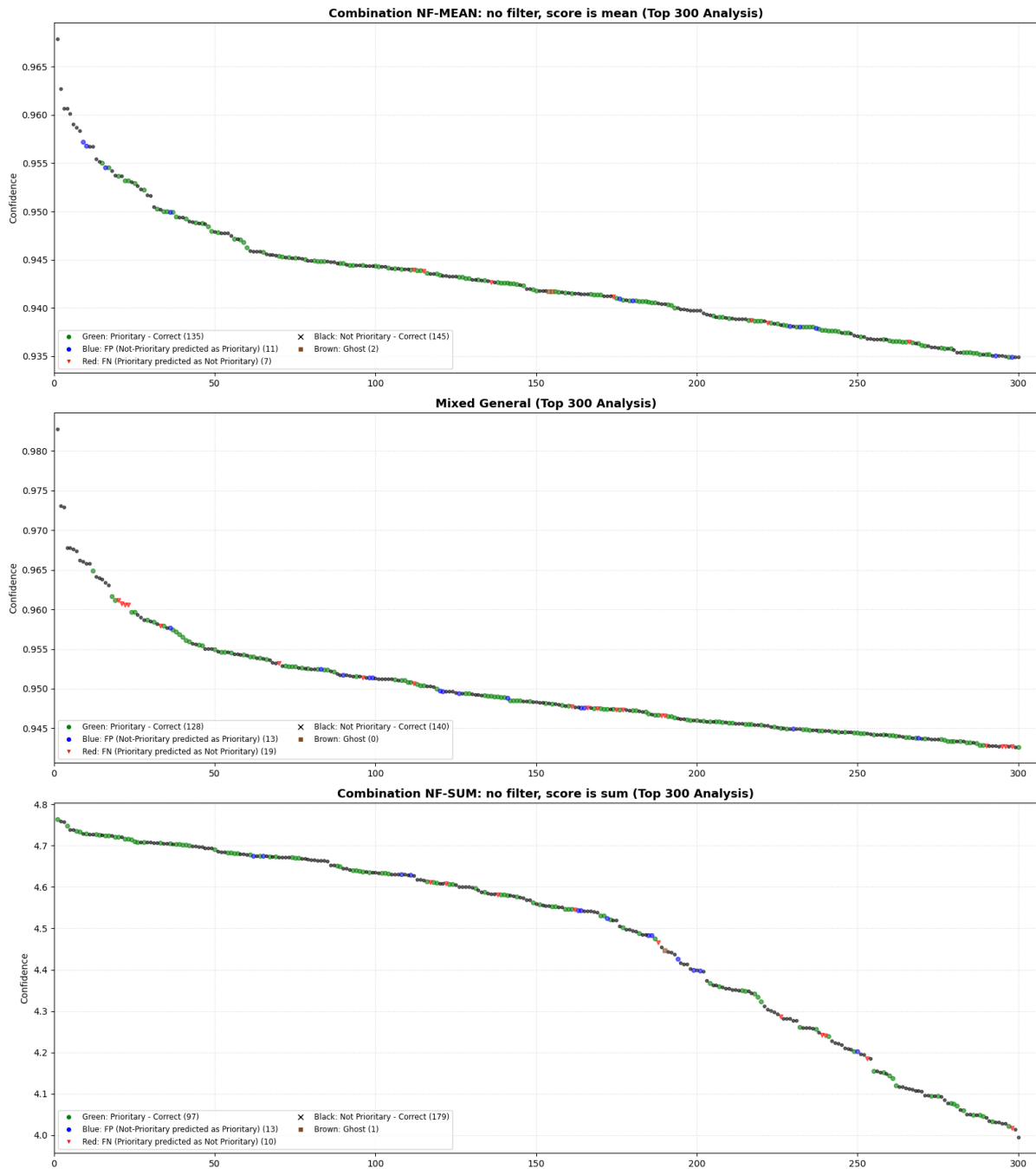


Figure 31 – Classification results of first 300 elements on **cars dataset** ranking.

and non-priority objects may still appear in top-ranked positions, the structure of the proposed combination approach inherently favors priority objects.

Each ranking was constructed by ordering all predictions according to their confidence scores, considering all objects. Figure 31 shows the first 300 positions in the ranking. Each point in the graph represents a predicted object, and the y-axis indicates the confidence score of the prediction. The color of each point denotes the classification outcome, as follows:

- **Green:** The object is priority and was correctly detected and classified as priority;

Table 6 – Results of object’s classifications on first 300 positions of rankings, for **cars dataset**.

Model	K	Green: priority Correct	Blue: FP (Not-priority predicted as priority)	Red: FN (priority predicted as Not priority)	Black: Not priority Correct	Brown: Ghost	Total Correct
Combination NF-MEAN	300	135	11	7	145	2	280
Mixed General	300	128	13	19	140	0	268
Combination NF-SUM	300	97	13	10	179	1	276

- **Black:** The object is non-priority and was correctly detected and classified as non-priority;
- **Red:** The object is priority but was incorrectly classified as non-priority;
- **Blue:** The object is non-priority but was incorrectly classified as priority;
- **Brown:** No ground-truth object is present, but an object was predicted (ghost detection).

Figure 31 shows that, up to rank position 100, no red cases are observed for either NF-MEAN or NF-SUM. Although the ordering criterion was based on confidence scores, this result indicates that the highest-confidence detections produced by both NF-MEAN and NF-SUM are associated with a low number of false negatives.

Table 6 presents the results of the first 300 positions of each ranking, on cars dataset. The colors described on column titles are related with Figure 31. NF-MEAN combination achieved the highest number of correctly classified priority objects. It also yielded the lowest number of false negatives when compared to the General Model and the NF-SUM combination.

Both Table 6 and Figure 31 show that the NF-MEAN combination achieved the highest number of correctly identified priority objects within the first 300 positions of the ranking. Furthermore, NF-MEAN yielded the lowest number of misclassified priority objects as well as misclassified non-priority objects, while NF-SUM had the higher number of not priority objects found correctly.

Figure 32 presents the cumulative evolution of the proportion of priority objects, identified by each model, at each ranking position. The x-axis represents the ranking position, while the y-axis denotes the number of correctly identified priority objects divided by the total number of existent priority objects. For instance, a point located at coordinates (0.05, 10) indicates that, up to position 10 in the ranking, the model has identified five percent of all existent priority objects. Until approximately position 124, the NF-SUM combination achieves a higher number of correctly identified priority objects. However, within the first 300 positions, the NF-MEAN combination surpasses NF-SUM and achieves the highest number of correctly identified priority objects, as shown in Table 6.

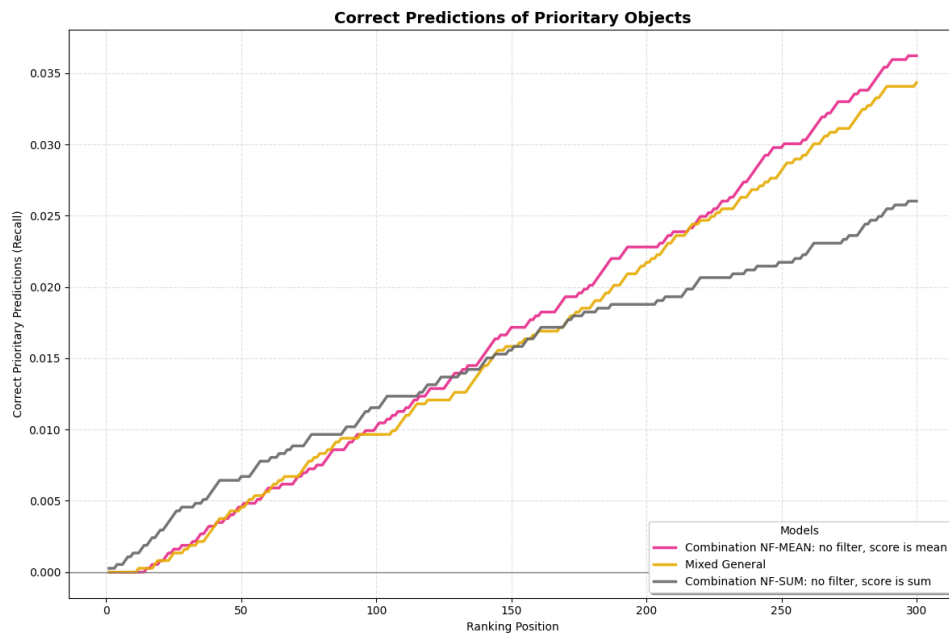


Figure 32 – Recall evolution of models per rank position on **cars dataset**, for the first 300 objects in ranking.

Figure 33 – Precision vs Recall curve of **cars dataset**, considering Prioritary Objects only

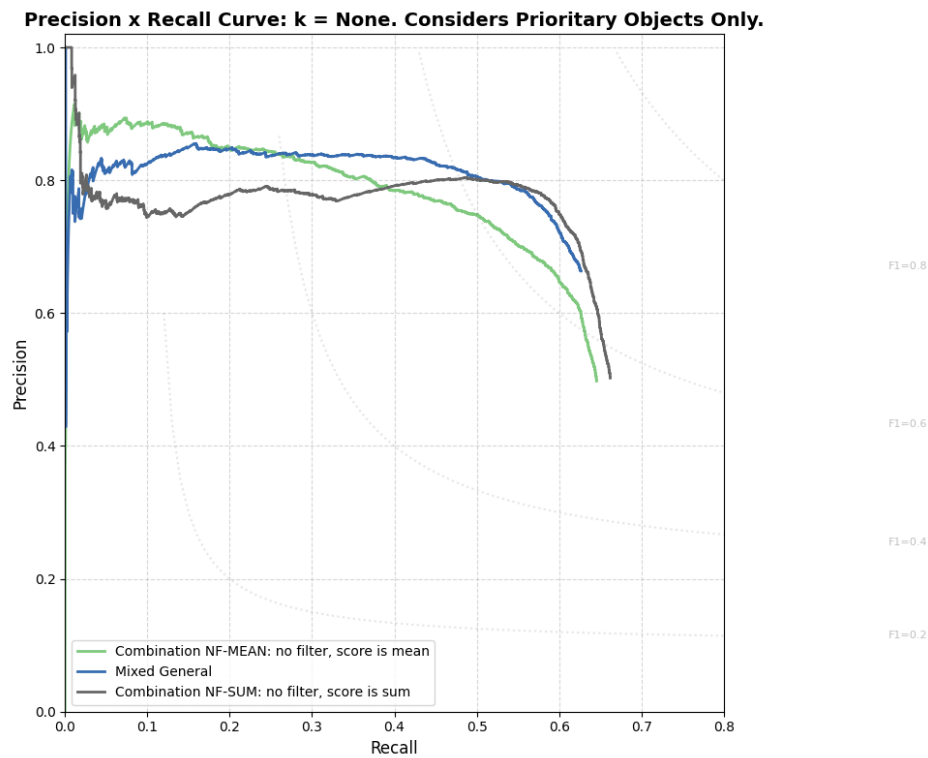


Table 7 – Results of the model’s rankings on **cars dataset**, considering all objects.

Model	Top K	Average Precision (AP)	Correct Objects Found
Combination NF-MEAN	300	0.936445	280
Mixed General	300	0.893586	268
Combination NF-SUM	300	0.953159	276

Figure 33 presents the Precision-Recall curve, considering only the priority cars. The evaluation considered only the ground truths of priority objects, and the ranking was restricted to priority objects. Therefore, black objects were entirely excluded from the Precision–Recall curve.

Some priority objects were not detected by any of the models. According to the aggregation logic adopted in this framework, such objects cannot be recovered at a later stage. Consequently, the recall metric cannot reach 100%.

Table 7 presents the average precision score computed over the first 300 positions of the ranking for the cars dataset, considering both priority and not priority objects. Both combinations NF-MEAN and NF-SUM achieve higher average precision than the general model. Combination NF-MEAN yields the highest absolute number of correctly identified objects within the top 300 ranking positions.

Table 6 and Table 3 report different numbers of correct cases due to differences in the evaluation procedure. Specifically, a greedy matching strategy was employed to assess the predictions. For Table 6, predictions were matched to ground-truth objects. In contrast, for Table 3, ground-truth objects were matched to predictions. In both cases, matches were determined using an IoU threshold of 0.75%. This methodology allows for the analysis of how different evaluation strategies influence the final results. In both evaluation scenarios, the NF-MEAN and NF-SUM combinations achieved a higher number of true positives than the General Model. The combination strategy enables the retention of correctly identified objects from the General Model while also recovering additional correct objects identified by the other models.

The reported rankings consider both critical and non-critical objects to allow a general evaluation. Section 5.6 presents separate rankings for critical and non-critical objects, also ordered by confidence score. These rankings allow us to determine which objects need to be emergently attended to and which can be safely ignored without prejudice.

Figure 34 presents the three objects reported by NF-MEAN as the highest confidence scores. All three objects were correctly classified as non-priority. For a hypothetical rescue team, this result indicates that the framework is highly confident that these objects are not critical and



Figure 34 – Top 3 objects ranked with higher confidence on **cars dataset**, considering NF-MEAN combination.



Figure 35 – Top 2 objects ranked with higher confidence on **cars dataset**, considering NF-SUM combination.

can therefore be safely ignored without compromising the rescue operation.

Figure 35 presents the two objects reported by NF-SUM as having the highest confidence scores. The object ranked in the first position was correctly classified as priority. The object ranked in the second position was correctly classified as non-priority, as also shown in Figure 31. In a hypothetical real-world scenario, the information provided by the framework could be interpreted as follows: “Object 1 should be attended to immediately, as it is highly likely to represent a critical situation. Object 2 can be safely disregarded, as it is highly unlikely

to represent an emergency.”. It provides the information of which objects are critical, and which objects can safely be disregarded, as the framework is sure do not report critical scenarios. The objective is to minimize the risk of potentially dangerous misclassification.

5.3 Object Detection Module on Houses Dataset

5.3.1 General Results in Object Detection

Table 8 presents the results of the overall evaluation on the houses dataset. The NF-MEAN and NF-SUM combinations achieved the highest recall values. In contrast, F06 achieved the highest precision and F1 score. Additionally, NF-MEAN and NF-SUM produced the lowest number of false negatives, which are the errors we aim to minimize, and the highest absolute number of true positives, although they also presented the higher values of false positive cases.

Table 8 – Results of classification on houses dataset.

Model	True positives	False positives	False negatives	Errors	Precision	Recall	F1 Score
Mixed General	10481	10570	10902	21472	0.497886	0.490156	0.493991
Only L0	4960	4457	16700	21157	0.526707	0.228994	0.319207
Only L1	3284	3073	18377	21450	0.516596	0.151609	0.234421
Only L2	1278	1769	19941	21710	0.419429	0.060229	0.105333
Only L3	573	931	21053	21984	0.380984	0.026496	0.049546
Combination NF-MEAN	11052	14516	10285	24801	0.432259	0.517973	0.471250
Combination F06	10111	8734	11304	20038	0.536535	0.472146	0.502285
Combination F05	10464	10071	10924	20995	0.509569	0.489246	0.499201
Combination F03	10938	13433	10403	23836	0.448812	0.512535	0.478561
Combination NF-SUM	11052	14516	10285	24801	0.432259	0.517973	0.471250

In this context, the NF-MEAN and NF-SUM combinations achieved the intended objective on the houses dataset. While they produce a higher number of false positives, they exhibit a lower rate of missed objects compared to Mixed General, indicating a more favorable trade-off given the objective of minimizing false negatives.

Figure 36 presents the confusion matrices of the specialized models on houses dataset. A large number of objects are missed (i.e., predicted as background), particularly for non-priority classes (classes 0 and 1). As shown in Table 8, all specialized models exhibit a high number of false negatives, however, the combination strategies significantly reduce this number. In particular, the NF-SUM and NF-MEAN combinations present fewer false negatives than any of the individual models.

The high number of objects misclassified as background may be related to the model's training process. Since each model receives annotations only for its corresponding class, objects belonging to other classes remain unlabeled. Consequently, the model may confuse visually similar objects from neighboring classes, interpreting unlabeled instances as background rather than correctly identifying them as objects of a different class. It is notable that, in addition to valid objects classified as background, there is also a considerable number of false positives (unlabeled objects) detected by each specialized model.

Figure 37 presents the confusion matrix of Mixed General model on houses dataset. The Mixed General model presented a considerable number of misclassifications involving

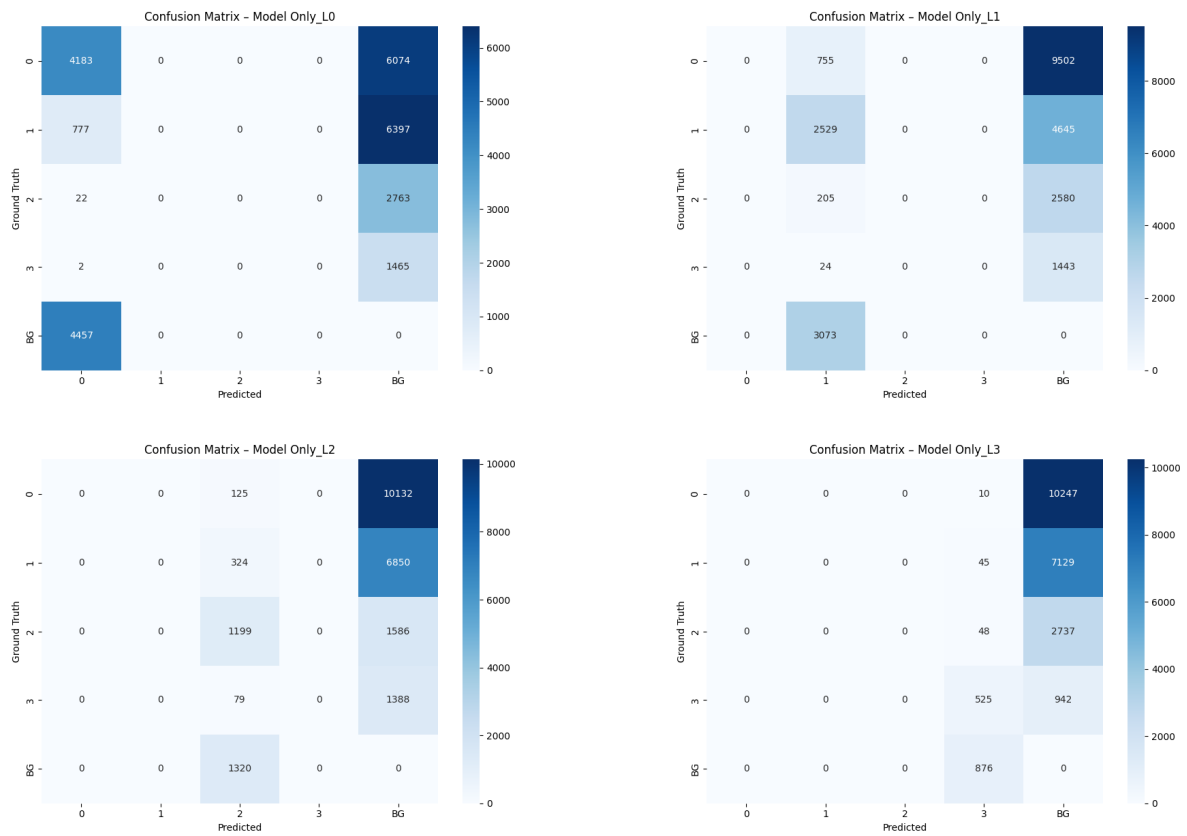


Figure 36 – Confusion matrices of specialized models for **houses datasets**.

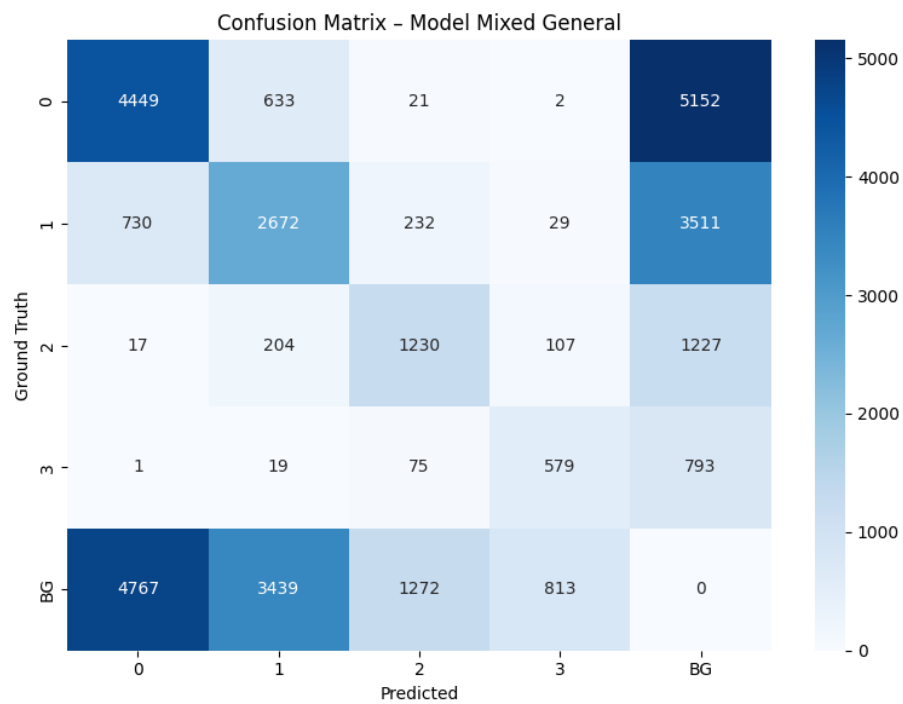
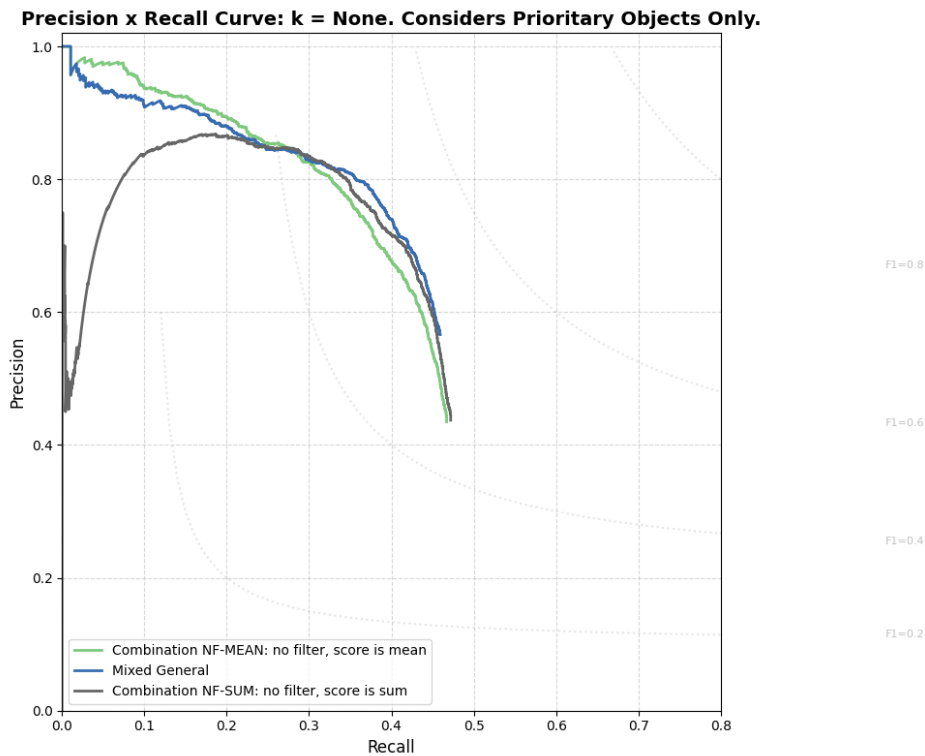


Figure 37 – Confusion matrix of Mixed General model on **houses dataset**.

Table 9 – Recall, considering only priority objects, for the first 300 positions of the rank of each model, in **houses dataset**

Model	Number of existent critical objects	Number of detected critical objects in first 300 positions	Recall (considering priority objects only)
NF-MEAN	4252	252	0.0592
Mixed General	4252	174	0.0409
NF-SUM	4252	25	0.0059

Figure 38 – Precision vs Recall curve of **houses dataset**, considering only priority objects.



priority objects (classes 2 and 3), which explains the high number of false negatives reported in Table 8. In comparison with the results obtained for the cars dataset, the houses dataset shows fewer misclassifications between neighboring classes.

As with the cars dataset, Table 9 presents the recall for the houses dataset, considering only priority objects. Since only the first 300 positions were analyzed, NF-MEAN achieved a high recall for critical objects, whereas NF-SUM exhibited a very low recall.

5.3.2 Ranking Results

Figure 38 presents the precision–recall curve considering only priority objects on houses dataset. Recall was computed as the proportion of priority objects correctly identified up to rank position k , relative to the total number of existing priority objects. When comparing the results with those obtained from the cars dataset, the houses dataset exhibits a more pronounced

Model	Top K	Average Precision (AP)	Correct Objects Found
Combination NF-MEAN	300	0.980593	293
Mixed General	300	0.975962	289
Combination NF-SUM	300	0.913426	272

Table 10 – Average precision of **houses dataset**, per model, considering all objects (priority and non-priority)

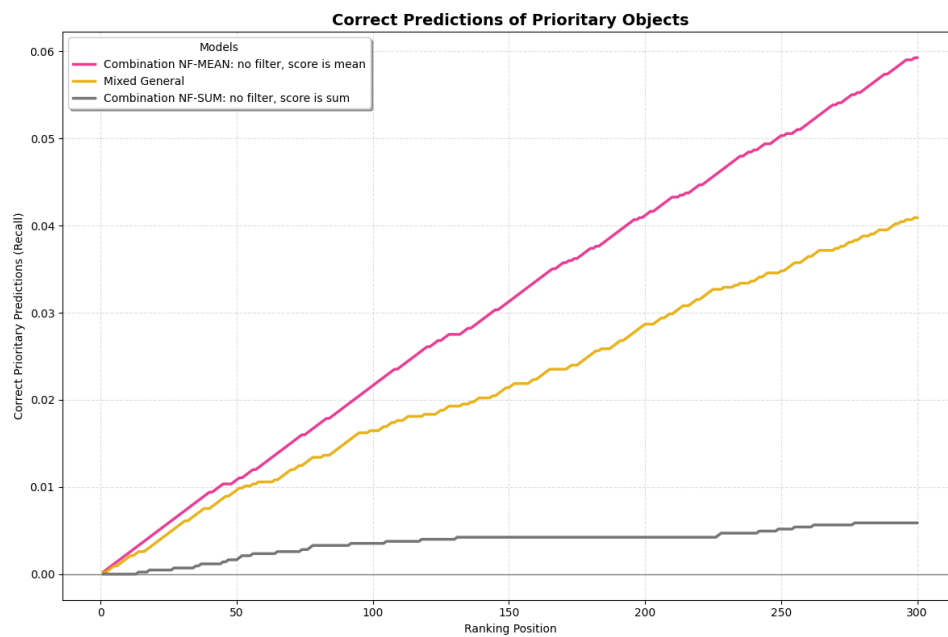


Figure 39 – Recall of priority objects for first 300 positions of the ranking, on **houses dataset**.

trade-off between precision and recall. Specifically, increasing recall results in a greater reduction in precision compared to the behavior observed in the cars dataset.

Table 10 presents average precisions on houses dataset, considering all objects (both priority and not priority). Combination NF-MEAN achieved higher average precision and higher number of correct objects found, considering top 300 positions of the ranking.

Figure 39 presents the evolution of recall over the first 300 positions of the ranking for the houses dataset, considering only priority objects. The NF-MEAN combination achieves superior recall across the top 300 ranked objects, yields the highest number of correctly identified priority objects (i.e., the highest recall). NF-MEAN combination also achieved the highest number of priority objects correctly classified and presented no occurrences of false negatives on first 300 positions of the ranking.

Figure 40 presents the top 300 ranked positions for the houses dataset. The NF-MEAN combination yields a high number of correctly classified priority objects among the

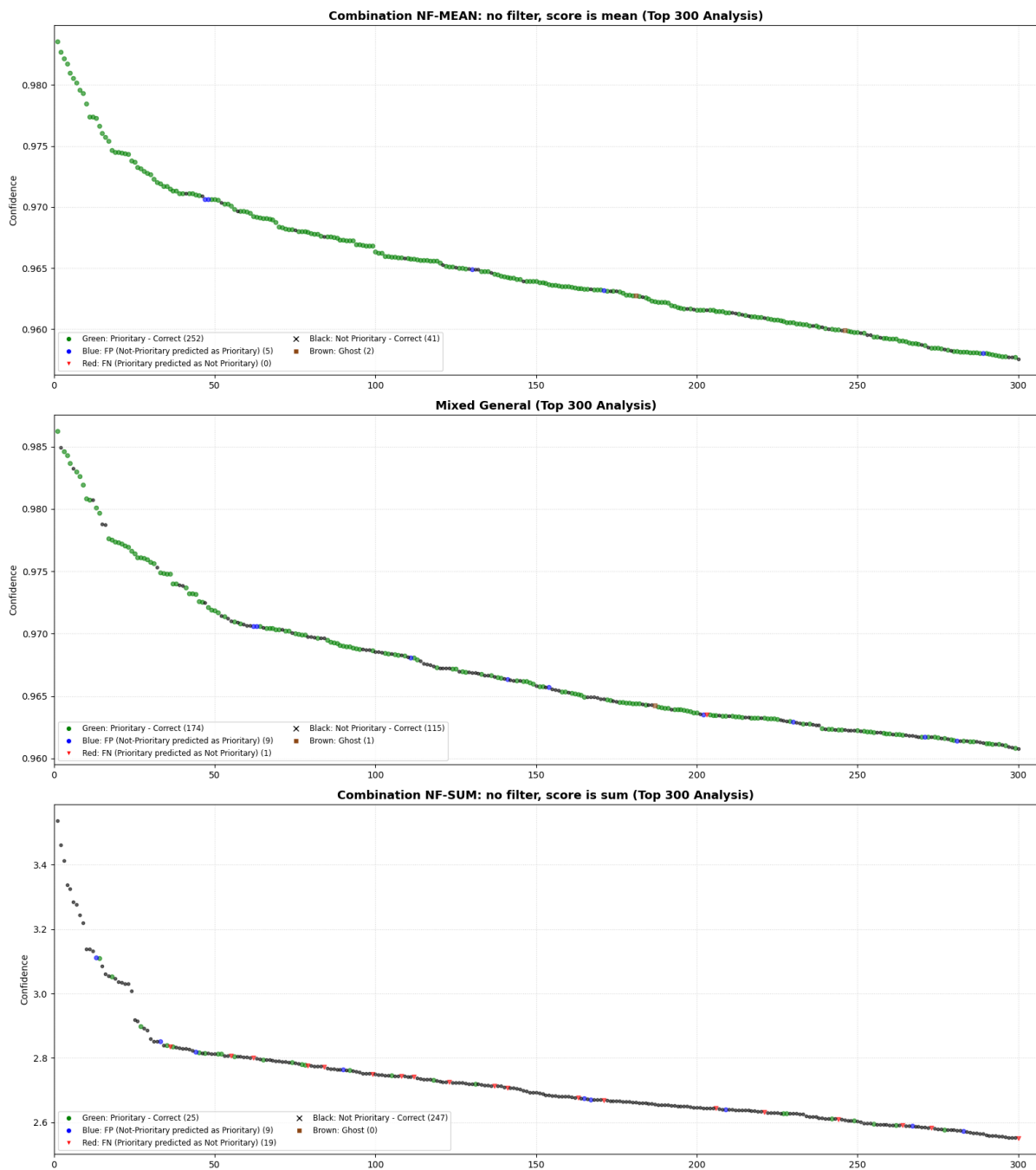


Figure 40 – Classification of first 300 positions of ranked objects of **houses** dataset.

Table 11 – Number of found objects of each category, on **houses dataset**, for first 300 positions of the ranking.

Model	K	Green: Priority - Correct	Blue: FP (Not-Priority predicted as Priority)	Red: FN (Priority predicted as Not Priority)	Black: Not Priority - Correct	Brown: Ghost	Total
Combination NF-MEAN	300	252	5	0	41	2	293
Mixed General	300	174	9	1	115	1	289
Combination NF-SUM	300	25	9	19	247	0	272

Table 12 – Number of objects of each class found by models, considering the whole ranking results, for **houses dataset**.

Model	Class 3	Class 2	Class 1	Class 0	Priority	Not Priority
Combination NF-MEAN	2069	3523	8597	11641	5592	20238
Mixed General	1530	2830	6967	9964	4360	16931
Combination NF-SUM	2069	3523	8597	11641	5592	20238

highest-ranked positions. In contrast, the NF-SUM combination includes a substantial number of correctly classified non-priority objects at the top of the ranking. This suggests that NF-SUM's confidence scores tend to prioritize objects from non-priority classes, whereas NF-MEAN prioritizes objects from priority classes, reflecting their distinct confidence aggregation strategies. As discussed previously, NF-SUM favors objects detected by multiple models, while NF-MEAN assigns higher priority to objects with higher individual confidence scores. It is reasonable to assume that a considerable number of priority objects are detected by only a small subset of models, with high confidence, whereas a large number of non-priority objects may be detected by multiple models, with lower confidence, even when they receive different classifications.

Table 11 presents the number of objects in each classification category within the top 300 positions of each ranking, corresponding to the color coding shown in Figure 40, as presented for cars dataset. NF-MEAN identified the highest number of correctly found priority objects, whereas NF-SUM identified the highest number of correctly found non-priority objects. Mixed General exhibited intermediate performance between the two methods. NF-MEAN had the higher number of correct objects found, considering both priority and not priority, on the first 300 positions of the ranking.

Table 12 presents the number of objects identified by each model in the houses dataset, based on the ranking results. Both combination methods, NF-SUM and NF-MEAN, identified a higher number of priority objects compared to Mixed General.

By comparing Table 12 and Table 2, it can be observed that all models predicted additional objects across all classes. Furthermore, misclassifications may occur, particularly between neighboring classes.

Figure 41 presents the three objects ranked by NF-MEAN as having the highest



Figure 41 – Top 3 objects ranked with higher confidence on **houses dataset**, considering NF-MEAN combination.

confidence scores. In contrast to the objects with the highest confidence scores in the cars dataset reported by NF-MEAN, the objects identified in the houses dataset correspond to critical cases. This information is also presented in Figure 40.

As observed in Figure 41, these objects correspond to severely damaged houses. They were correctly classified as priority objects. For a hypothetical rescue team, this information indicates which objects the framework identifies with high confidence as requiring immediate attention. The images presented in positions 2 and 3 of the ranking are visually very similar, which justifies their close proximity in the reported rank.

5.4 Discussion about Object Detection Module, Considering Both Datasets

As previously stated, the combination strategy prioritizes models that predict only priority classes. Within this framework, the Mixed General model is given the highest priority in the selection of predictions, followed by the Only L3 and Only L2 models. Notably, both Only L3 and Only L2 are restricted to predicting priority classes exclusively. This may explain why predictions for priority classes consistently rank at the top. Specifically, for each object lost by the Mixed General model, if the object is classified as a priority class by another model, that prediction is systematically selected.

For instance, when an object is predicted by multiple models (Only L3, Only L2, Only L1, and Only L0), the prediction associated with the highest priority level is consistently chosen. Due to this configuration, a non-priority prediction is selected only under the following conditions: (1) it is produced by the Mixed General model, which holds the highest priority in the combination strategy; or (2) the object is predicted exclusively as non-priority by a model that is restricted to non-priority classes.

This combination strategy seems to reduce the misclassification of priority objects in the top-ranked positions of NF-MEAN ranking. The ranking mechanism serves two distinct but related purposes: (1) to identify objects that can be confidently assumed to be critical, and (2) to identify objects that can be confidently assumed not to be critical objects.

In a real-world application, where the timely identification of critical objects is essential, this approach enables decision-makers to address two practical questions: which objects can be reliably considered in need of immediate assistance, and which objects can be confidently regarded as not requiring intervention and therefore safely deprioritized.

For the houses dataset, all models and combinations demonstrated lower performance compared with those obtained for the cars dataset. Nevertheless, the ranking results remain promising. Combinations F06 and F03 showed a similar recall difference across both datasets, with approximately a 4% gap between them. However, in the houses dataset, the rankings generated by NF-MEAN and the Mixed General model tend to place the correct priority objects in the highest positions.

5.5 Question Answering Module Results and Use Examples

To evaluate the Question Answering Module, the cars dataset was used. Three files were generated to enable testing:

- **Latex result file:** file includes Tables 3 and 4, both saved in a text file, at Latex code format.
- **Rank file:** contains the 10 images with the highest number of priority objects found, according to each evaluated combination.
- **Textual result file:** contains Table 4, formatted in plain text. Each entry is written as a descriptive line, for example: “Model ‘Combination NF-MEAN: no filter, score is mean’, Class: ‘Class 1’: found 2659 objects.”

During the experiments, it was observed that the text file format influences DeepSeek’s ability to provide accurate responses, as shown in Tables 13 and 15. For this reason, both rank file and a textual result file were designed to mitigate some of the limitations identified when using the LaTeX file format. The obtained results are presented below.

The analysis presented on Table 13, DeepSeek was used with all default parameters. The used prompt was:

You will be provided a question and a text. Your task is to answer the question based on what is on the text. If the answer is clear on the document, you must provide that answer. If necessary, you must calculate the answer based on the available information on the document. In case it is not possible to calculate the answer based on the content of the document, you must inform the user that the question is out of scope and that it is not possible to provide an answer. Provide a final answer that is clear and strict to the point sentence.

DeepSeek’s results exhibited considerable variability during testing, producing different outputs despite identical parameter settings. Timeout errors also occurred intermittently. The results reported here represent the best outcomes achieved. A timeout parameter of 60 seconds was used in all experiments.

Table 4 was used to evaluate DeepSeek’s performance. All questions used to assess DeepSeek’s performance were manually formulated within the context of the proposed framework. The corresponding answers were also manually evaluated by human reviewers. As previously mentioned, Table 4 was constructed based on the ranking results of the NF-MEAN, NF-SUM, and Mixed General model combinations. The complete ranking was taken into account. For the purpose of evaluating DeepSeek’s performance, the correctness of Table 4 itself was

considered out of scope. An answer was deemed correct solely if the value returned by DeepSeek matched the content of the provided file and if the response was coherent.

Table 13 – DeepSeek preliminary evaluation within Table 4 saved as Latex result file, along with Table 3.

Number Question	Question	Answer	Status
1	How many not-priority objects were found by the model named 'Mixed General'?	The Mixed General model found 2982 not-priority objects.	Correct
2	How many priority objects were found by the model named 'Combination NF-SUM'?	The model "Combination NF-SUM: no filter, score is sum" found 4395 prioritized objects.	Wrong
3	How many objects of Class 3 were found by the model named 'Combination NF-MEAN'?	The combination NF-MEAN was found to classify 1556 objects under class 3.	Wrong

Table 13 presents a first, preliminary evaluation of DeepSeek within the context of the proposed framework, within Latex result file. At answers for Questions 2 and 3 reported on Table 13, the referred values are wrong. Both values probably came from Table 3, which was also wrote in result file. It shows that DeepSeek might have difficulty by finding the right value if there is a lot of content on the file, using this format. This issue may become problematic when extending the framework in the future, using the same file format. If multiple additional sets of models are incorporated, DeepSeek may require further fine-tuning, or the file structure may need to be redesigned to improve information retrieval and reduce ambiguity.

This preliminary evaluation indicates that, if the framework is to be deployed in a real-world scenario, future work must investigate improved strategies for structuring the information within the file or consider fine-tuning DeepSeek for this specific application context.

The textual formats used in the Rank file and the Textual Result file were developed to address specific deficiencies observed in the system when using files formatted in LaTeX code after a fist evaluation of DeepSeek's performance.

For results presented on Tables 14 and 15, the DeepSeek parameter "temperature" was set to 0, to improve stability of the answers.

The prompt was refined to improve the accuracy of the generated responses compared with the results presented in Table 13. The version used to generate results presented on Tables 14 and 15 is shown below:

You will be provided a question and a text. Your task is to answer the question based on what is on the text. If the answer is clear on the document, you must provide that answer. Answer exactly what was asked. If necessary, you must calculate the answer based on the available information on the document. If it is not necessary, just answer what was requested. In case it is not possible to calculate the answer based on the content of the document, you must inform the user that the question is out of scope and that it is not possible to provide an answer. Provide a final answer that is clear and strict to the point sentence.

Table 14 – DeepSeek preliminary evaluation using rank file.

Number	Question	Answer	Status
1	According to Combination F06, which image has the higher number of priority objects?	In Combination F06, the image with the highest number of prioritized objects is: 00018_jpg.rf.72bf0c9a94b62fe897ed79570cfdc77d.jpg: priority = 21	Correct
2	What are the 3 images with more priority objects, considering all models?	The three images with the highest priorities are: 1. 01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg - Priority = 39 2. 01196_jpg.rf.2873cbe6c0fe613b50fa2005ec8e85f2.jpg - Priority = 28 3. 00018_jpg.rf.72bf0c9a94b62fe897ed79570cfdc77d.jpg - Priority = 23	Correct (*)
3	What are the images with more priority objects, considering all models?	Timed Out	Timed Out
4	What are the 3 images with more priority objects, considering Combination NF-SUM?	Timed Out	Timed out
5	How many priority objects were found on image '00738_jpg.rf.9679f7ab19fabe6564bad2bb3eb7bdb2.jpg' by Combination F03?	The image '00738_jpg.rf.9679f7ab19fabe6564bad2bb3eb7bdb2.jpg' is associated with Combination F03, where it has a priority score of 35 and not-priority of 0.	Wrong
6	How many priority objects were found on image '01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg' by Combination F03?	The number of prioritized objects found on the image '01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg' by Combination F03 is 35.	Correct
7	What are the images with more priority objects for each combination?	For each combination, the image with the highest priority is as follows: - NF-MEAN: Image "01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jp" has a priority value of 39. - F06: The image with the highest priority is "00018_jpg.rf.72bf0c9a94b62fe897ed79570cfdc77d.jpg" with a priority value of 21. - F05: The highest priority image is "01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg" with a priority value of 23. - F03: The highest priority image is again "01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg" with a priority value of 35.	Partially Correct

Table 14 presents the results obtained for the rank file. The number of elements in each rank was limited in order to prevent timeout issues. However, timeouts occurred in Questions 3 and 4 of Table 14. In a real-world scenario, this could represent a significant limitation. Although DeepSeek appears to be capable of correctly answering ranking-related

questions, a more thorough evaluation is required before deploying the system in real-world applications.

Question 2 in Table 14 yielded results that were considered correct. Image “01283_jpg.rf.e967a0809acbf45be7bd63671cbc394.jpg” was ranked highest by both the NF-MEAN and NF-SUM, presenting the greatest number of reported priority objects. Image “01196_jpg.rf.2873cbe6c0fe613b50fa2005ec8e85f2.jpg” was ranked second, with the next highest number of priority objects reported, according to NF-MEAN and NF-SUM. Finally, the image “00018_jpg.rf.72bf0c9a94b62fe897ed79570cfdc77d.jpg” was ranked third in the NF-MEAN and NF-SUM rankings.

Questions 5 and 6, presented in Table 14, correspond to the same ranking. The image referenced in Question 5 is the fiftieth image in the ranked list, whereas the image referenced in Question 6 is the first. These results indicate that DeepSeek demonstrates reduced effectiveness in retrieving images appearing at lower positions in the ranking. The response to Question 5 was deemed incorrect, as it was not well formulated and the reported value was inaccurate.

The response to Question 7 was considered partially correct. The NF-SUM result was omitted from the answer, and the F05 result was incorrect (the image identified corresponds to the second file in the ranking, not the first). The remaining information provided was accurate.

The results presented in Table 15 correspond to DeepSeek’s responses. The same questions presented in Table 13 are shown in Table 15, but using a textual file as input, as previously described.

Table 15 presents better results than Table 13. In Table 13, two out of three questions were answered incorrectly. These same questions were correctly answered in Table 15.

Question 4 in Table 15 corresponds to the same underlying query as Question 1, but is formulated differently. While question 1 was answered correctly, question 4 was answered incorrectly. This result indicates that the formulation of the question plays a relevant role in DeepSeek’s ability to provide accurate responses.

Figures 42 and 43 presents examples of detections, on the same image, made by model Mixed General, and combinations NF-MEAN and NF-SUM. Since combinations prioritize Mixed General predictions, in order to do not lose any valid prediction of the most reliable model, it is expected for the boxes to be similar, but new boxes can be added by specialized models.

As illustrated in 42, most cars were successfully detected; however, some instances remain undetected. Additionally, false positives are observed, where certain objects (probably debris or motorcycles) are incorrectly classified as submerged cars.

In Figure 45, all cars were successfully detected. Although some bounding boxes are not precisely aligned with the objects, the results demonstrate the framework’s ability to identify critical object instances effectively.

Table 15 – DeepSeek evaluation using textual result file.

Number Question	Question	Answer	Status
1	How many not-priority objects were found by the model named 'Mixed General'?	The 'Mixed General' model found 2982 not-priority objects.	Correct
2	How many priority objects were found by the model named 'Combination NF-SUM'?	The number of priority objects found by the 'Combination NF-SUM' model is 5164.	Correct
3	How many objects of Class 3 were found by the model named 'Combination NF-MEAN'?	The number of objects of Class 3 found by the 'Combination NF-MEAN' model is 2790.	Correct
4	Considering only model 'Mixed General', how many not-priority objects were found?	The number of not-priority objects found for the model 'Mixed General' is 934.	Wrong
5	Considering only combination 'NF-MEAN', what are the 3 images with higher number of priority objects?	The three images with the highest priority in combination 'NF-MEAN' are: 1. 01283_jpg.rf.e967a0809acfb45be7bd63671cbc394.jpg (Priority: 39) 2. 01196_jpg.rf.2873cbe6c0fe613b50fa2005ec8e85f2.jpg (Priority: 28) 3. 00018_jpg.rf.72bf0c9a94b62fe897ed79570cfdc77d.jpg (Priority: 24)	Correct
6	How many [Class 3] objects were found, according to the combination which score is mean?	The total number of Class 3>Class 3 objects found is 2790.	Correct
7	How many [Class 0] objects were found, according to the combination which score is sum?	The number of [Class 0] objects found with the 'Combination NF-SUM' model, which uses sum scoring, is 1354.	Correct

Figures 44 and 45 presents, respectively, the second and third images presented by DeepSeek, on question 5 of Table 15, as the images with higher number of priority objects found. The image identified as containing the highest number of priority objects is shown in Figure 42. The images presented in Figures 42 and 44 are visually very similar, which explains their proximity in the ranking.

Question 5 in Table 15, together with Figures 42, 44, and 45, presents an example illustrating the use of the proposed framework. DeepSeek identifies the images containing the highest number of priority objects, which can then be displayed for visual inspection by the user.

In a real-world scenario, if question 5 of Table 15 were posed to the framework, the response indicates that the scene shown in Figure 42 should be addressed first, as it contains the highest number of critical objects detected, according to NF-MEAN combination, that presented the most reliable results in detecting critical objects with high confidence. The scenes presented in Figures 44 and 45 should be addressed next, as they also exhibit a high number of priority detections.

Although the results suggest potential effectiveness in real-world scenarios, it is essential to conduct more extensive evaluations before deploying the framework in practical applications, specially considering the sensitive nature of our proposed scenario. The system

Figure 42 – Examples of detections made by **Mixed General** model and combination **NF-SUM**, on **cars dataset**. This image was identified by DeepSeek as the **first** image with the highest number of occurrences of priority objects.



should be tested under a wider range of conditions and with greater variability to ensure robustness and generalizability before use in real situations. Nevertheless, the current findings are promising and demonstrate that the proposed objective of the research has been achieved.

A comparison of the results presented in Tables 13 and 15 indicates that the LaTeX file format is not well-suited for processing by DeepSeek, as it yielded inferior performance. In contrast, the plain text format produced better results. As future work, alternative text formatting strategies should be investigated, and a more comprehensive evaluation of DeepSeek's performance should be conducted.

Considering Tables 14 and 15, a total of 14 questions were evaluated. Of these, four were classified as incorrect, one as partially correct, and nine as correct. This corresponds to 28% incorrect responses, 7% partially correct responses, and 64% correct responses.

Figure 44 – Example of detection using NF-MEAN on **cars dataset**. This image was identified by DeepSeek as the **second** image with the highest number of occurrences of priority objects.

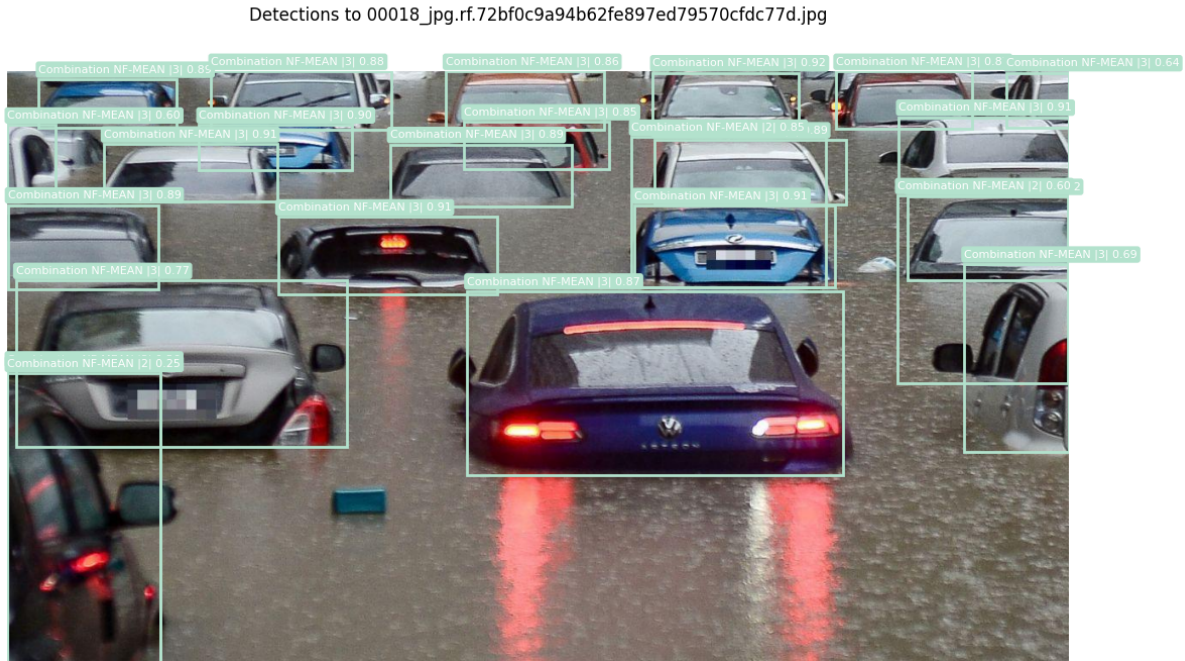


Figure 45 – Example of detection using NF-MEAN on **cars dataset**. This image was identified by DeepSeek as the **third** image with the highest number of occurrences of priority objects.

Table 16 – Average precision of ranking considering only **not critical objects only**, on **cars dataset**.

Model	Top K	Average Precision (AP)	Correct Objects Found
NF-MEAN	300	0.965961	283
Mixed General	300	0.894659	272
NF-SUM	300	0.962051	279

5.6 Additional Experiments and Results

This Section presents additional experiments with rankings and the Question Answering Module. Two separate additional rankings were created: one considers only objects classified as critical, and the other considers only objects classified as non-critical. The goal of these separate evaluations is to determine, by ordering objects according to confidence score, which objects require immediate attention and which can be ignored with a high degree of confidence without being negligent regarding risky situations. Section 5.6.1 presents the rankings experiments, and 5.6.2 presents the experiments with Question Answering Module.

Table 17 – Results of ranking on **cars dataset**, considering **not critical objects only**.

Model	K	Green: Priority - Correct	Blue: FP (Not-Priority predicted as Priority)	Red: FN (Priority predicted as Not Priority)	Black: Not Priority - Correct	Brown: Ghost	Total
NF-MEAN	300	0	0	17	283	0	300
Mixed General	300	0	0	28	272	0	300
NF-SUM	300	0	0	21	279	0	300

Table 18 – Results of ranking considering **not critical objects only**, on **houses dataset**.

Model	K	Green: Priority - Correct	Blue: FP (Not-Priority predicted as Priority)	Red: FN (Priority predicted as Not Priority)	Black: Not Priority - Correct	Brown: Ghost	Total
NF-MEAN	300	0	0	2	298	0	300
Mixed General	300	0	0	6	294	0	300
NF-SUM	300	0	0	23	277	0	300

Table 19 – Average precision of rank considering **not critical objects only**, in **houses dataset**.

Model	Top K	Average Precision (AP)	Correct Objects Found
NF-MEAN	300	0.997631	298
Mixed General	300	0.987350	294
NF-SUM	300	0.935719	277

5.6.1 Ranking Experiments

Table 16 presents the average precision of the rankings considering only non-priority cars. Both NF-MEAN and NF-SUM achieved higher values than the Mixed General method.

Table 17 and Figure 46 present the classification results for the first 300 positions of the ranking, considering only not priority cars. NF-MEAN achieved the highest number of correctly classified objects and the lowest number of false negatives caused by misclassification. Both NF-MEAN and NF-SUM produced no false negatives before position 50.

Figure 47 presents the three objects ranked with the highest confidence among non-priority cars according to NF-SUM, while Figure 48 shows the same rank, according to NF-MEAN. Although all objects were correctly classified as non-priority, NF-SUM identified objects that are not even part of scenes involving water.

Table 18 presents the results for the top 300 objects ranked with the highest confidence, considering only not priority houses. The colors in the Table correspond to those shown in Figure 50. NF-MEAN achieved the highest number of correctly classified non-critical objects and the lowest number of false negatives (i.e., critical objects incorrectly detected as non-critical).

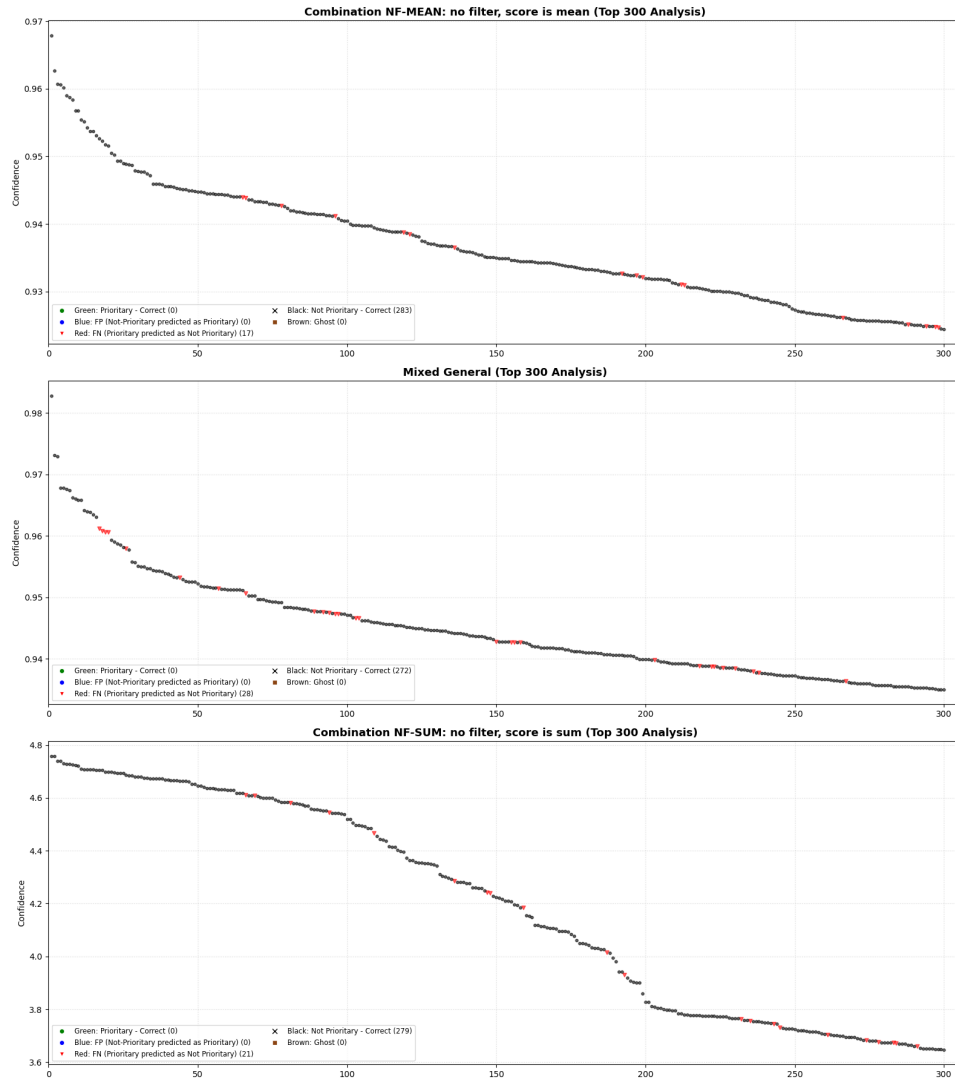


Figure 46 – Classification of the first 300 positions of the rankings, considering only non-priority cars.

Table 19 presents the average precision of first 300 positions of each ranking, considering not priority houses only. NF-MEAN achieved the higher value, with almost 100% average precision. By comparing the results with Table 18, it is notable that NF-MEAN presents very few mistakes on first 300 rank positions.

Figure 49 presents the three objects classified with the highest confidence by the NF-SUM method, considering only non-priority houses. As observed in Figure 50, the first objects in all rankings are correctly classified as non-priority. The houses shown in Figure 49 represent cases that could be safely ignored in a natural disaster scenario, allowing response efforts to prioritize damaged houses without significant losses.

Figure 51 presents the top three non-priority houses according to the NF-MEAN method. The object shown in position 2 appears to be close to potentially critical objects; however, it does not seem to be critically damaged. The objects presented in positions 1 and 3 clearly do not represent emergency situations and could be safely ignored in a natural disaster scenario.

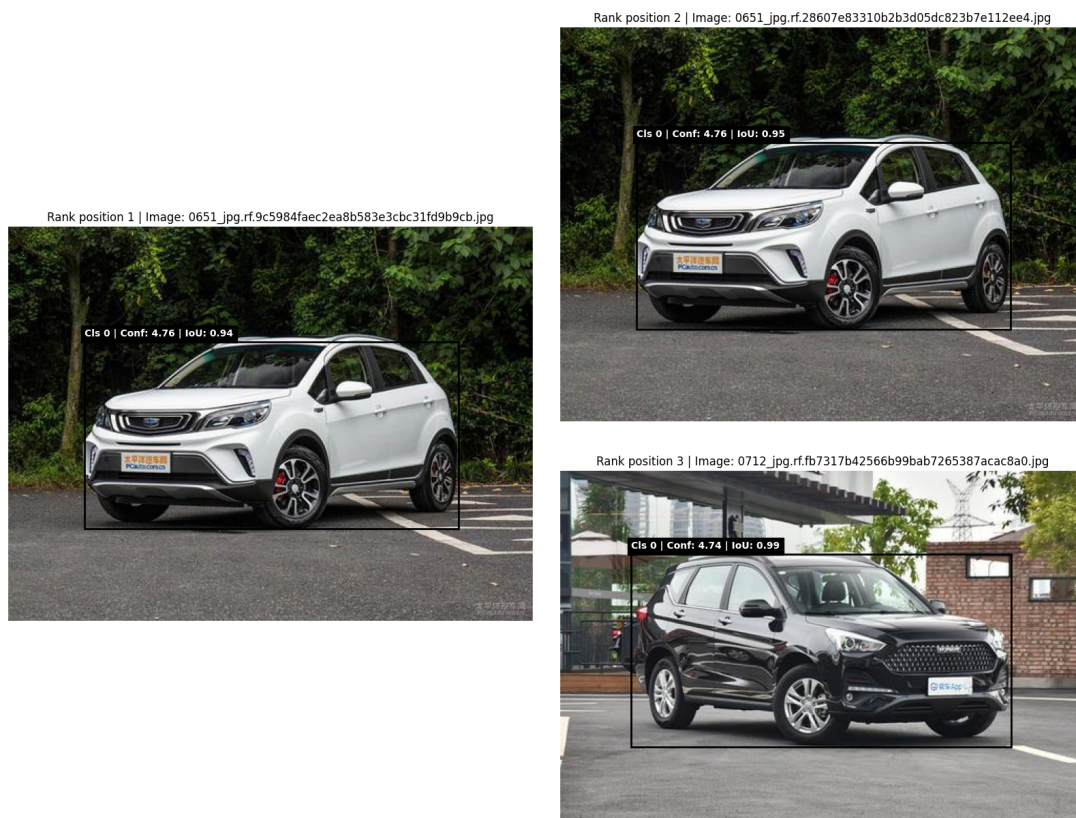


Figure 47 – Three first objects presented on **NF-SUM** ranking of **not priority** cars with higher confidence.



Figure 48 – Three first objects presented on **NF-MEAN** ranking of **not priority** cars with higher confidence.



Figure 49 – Three first objects presented on NF-SUM ranking of **not priority** houses with higher confidence.

Figure 52 presents the objects listed by NF-MEAN with the highest confidence scores, considering only priority cars. As shown in Figure 20, the two objects that appear in positions 1 and 2 of the ranking are not priority, representing false positive cases (non-critical objects incorrectly classified as critical). By comparing these results with the general car ranking shown in Figure 31, it can be observed that the first objects in the NF-MEAN ranking are not priority. The first priority objects that appear are false positives (i.e., they were classified as critical but are labeled as non-critical in the ground truth).

Table 20 and Figure 53 present the first 300 positions of the rankings and their corresponding classifications. Although NF-MEAN presents two false positives in the very first positions of the ranking, it achieves the highest number of correctly predicted priority cars within the first 300 positions, as well as the lowest number of false positives within these positions.

Table 21 presents the average precision of the rankings. Mixed General achieved

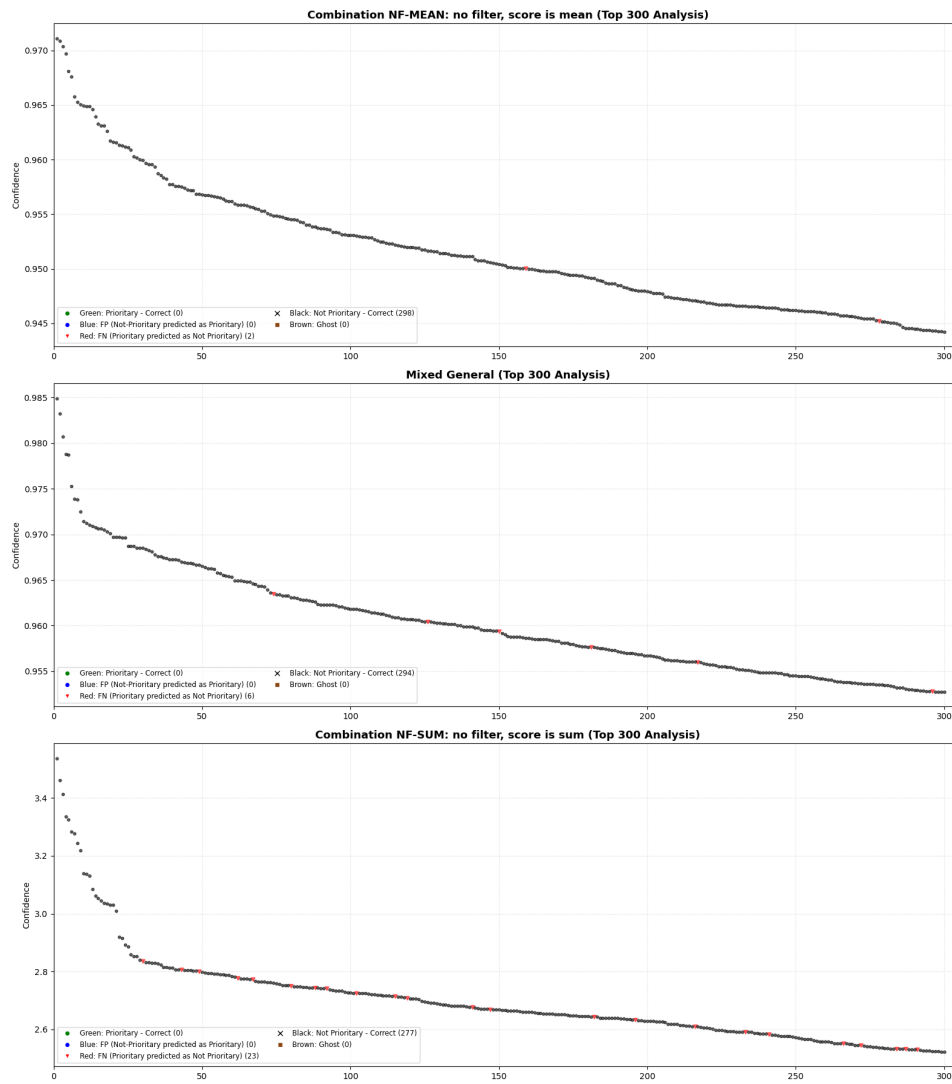


Figure 50 – Classification of top 300 objects of each rank, considering **not** priority houses only.

Table 20 – Results of ranking of **critical** objects only, in **cars** dataset.

Model	K	Green: Priority - Correct	Blue: FP (Not-Priority predicted as Priority)	Red: FN (Priority predicted as Not Priority)	Black: Not Priority - Correct	Brown: Ghost	Total
NF-MEAN	300	279	17	0	0	4	300
Mixed General	300	276	24	0	0	0	300
NF-SUM	300	256	41	0	0	3	300

Table 21 – Average precision on ranking of **priority** objects only, on **cars** dataset

Model	Top K	Average Precision (AP)	Correct Objects Found
NF-MEAN	300	0.902256	279
Mixed General	300	0.912752	276
NF-SUM	300	0.881186	256



Figure 51 – Three first objects presented on NF-MEAN rank of **non- priority** houses with higher confidence.

the highest average precision score, although NF-MEAN found the highest number of correctly identified objects within the first 300 positions.

Figure 54 presents the first three images ranked by NF-SUM with the highest confidence, considering priority cars only. As shown in Figure 53, all objects were correctly classified as priority. The visual similarity between the images evidences a coherence in NF-SUM results.

Figure 55 presents a comparison of recall across the rankings, considering priority cars only. NF-SUM has a slight advantage in the first positions; however, after position 50, NF-MEAN and Mixed General outperform it.

Table 22 presents the average score of the ranking considering only priority houses. In comparison with the priority cars rank, presented in Table 21, Mixed General and NF-MEAN achieved higher average precision values for the priority cars rank. In contrast, NF-SUM

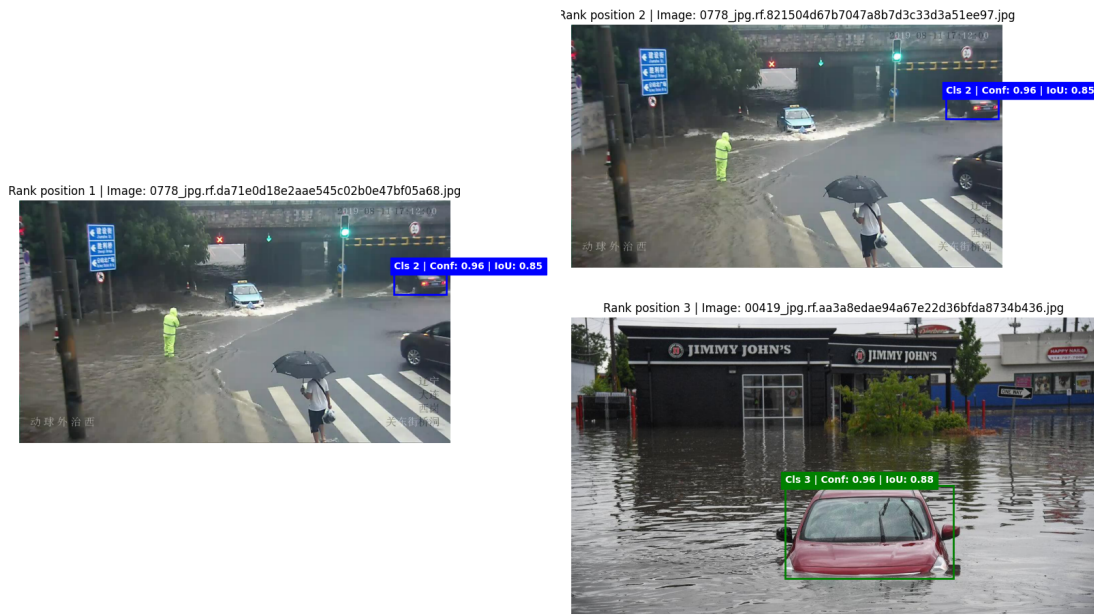


Figure 52 – Three first objects presented on NF-MEAN rank of **priority cars** with higher confidence.

Table 22 – Average precision of ranking of **critical objects only**, on **houses dataset**, for first 300 positions

Model	Top K	Average Precision (AP)	Correct Objects Found
NF-MEAN	300	0.980451	293
Mixed General	300	0.958556	283
NF-SUM	300	0.784025	262

presented a higher value for the priority houses rank.

Figure 56 presents the comparison of recall for the first 300 positions of each rank, considering only priority houses. NF-MEAN and Mixed General show strong recall in the top-ranked positions. Moreover, NF-MEAN appears to have a slight advantage between positions 50 and 300 when compared to the rankings produced by the other methods.

Figure 57 and Table 23 present the classification results for the first 300 elements of the rankings considering only priority houses. NF-MEAN achieved the highest number of correctly classified objects and the lowest number of ghost detections. Additionally, NF-MEAN presented the lowest number of false positives (i.e., non-critical objects incorrectly classified as critical).

Figure 58 presents the three objects reported by the NF-MEAN ranking with the highest confidence, considering only priority houses. Figure 59 also presents the objects in the top three positions of the NF-SUM ranking. As observed in Figure 57, the first image in the NF-SUM ranking corresponds to a false positive. In Figure 59, it can be observed that the object shown does not appear to be severely damaged. The remaining objects in the top three positions of both rankings are correctly classified as priority.

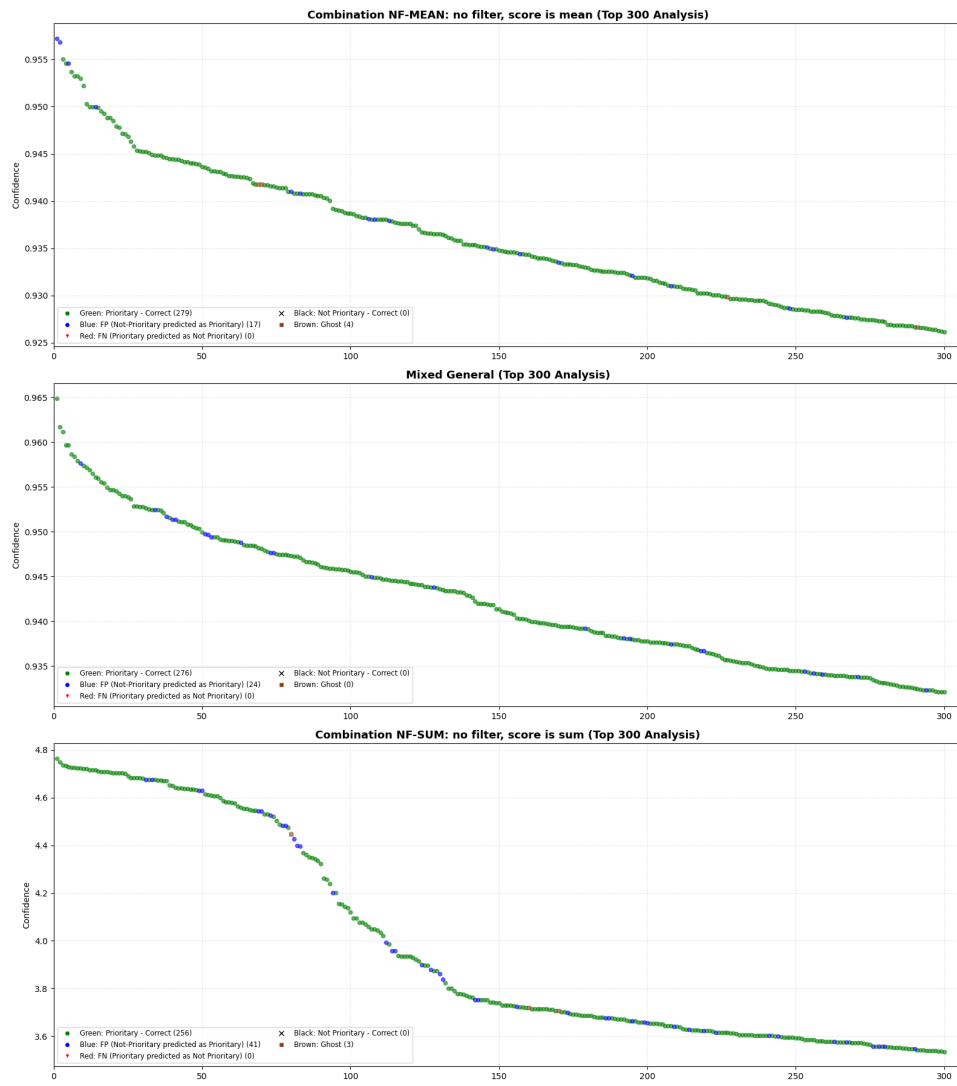


Figure 53 – Classification of the first 300 positions of the ranking, considering **critical objects only**, in the **cars dataset**.

Table 23 – Results of ranking of **critical objects only**, on **houses dataset**, for first 300 positions

Model	K	Green: Priority - Correct	Blue: FP (Not-Priority predicted as Priority)	Red: FN (Priority predicted as Not Priority)	Black: Not Priority - Correct	Brown: Ghost	Total
NF-MEAN	300	293	6	0	0	1	300
Mixed General	300	283	12	0	0	5	300
NF-SUM	300	262	34	0	0	4	300



Figure 54 – Three first objects presented on NF-SUM rank of **priority cars** with higher confidence.

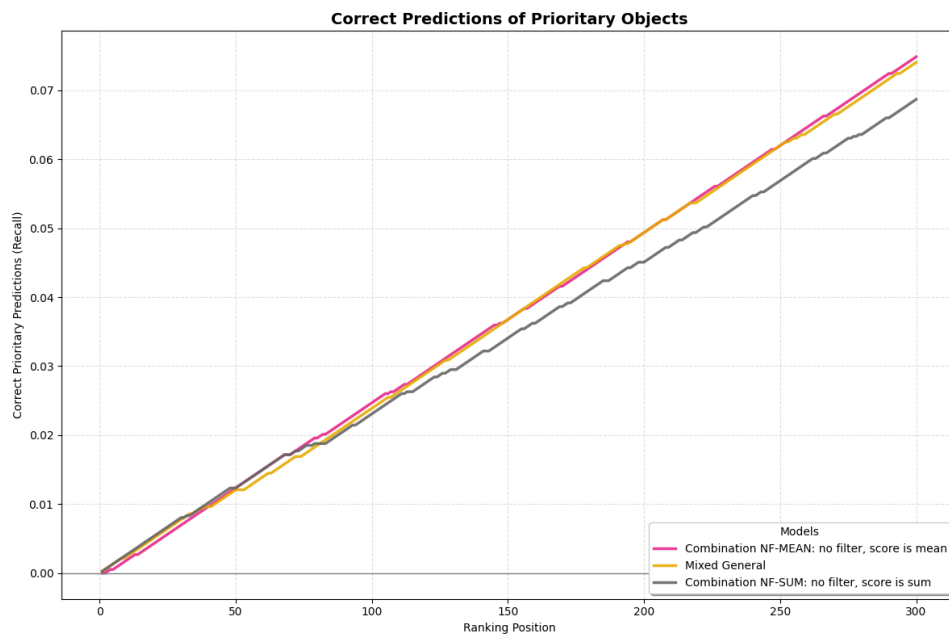


Figure 55 – Comparison of recall considering only priority objects in the first 300 positions of the rankings.

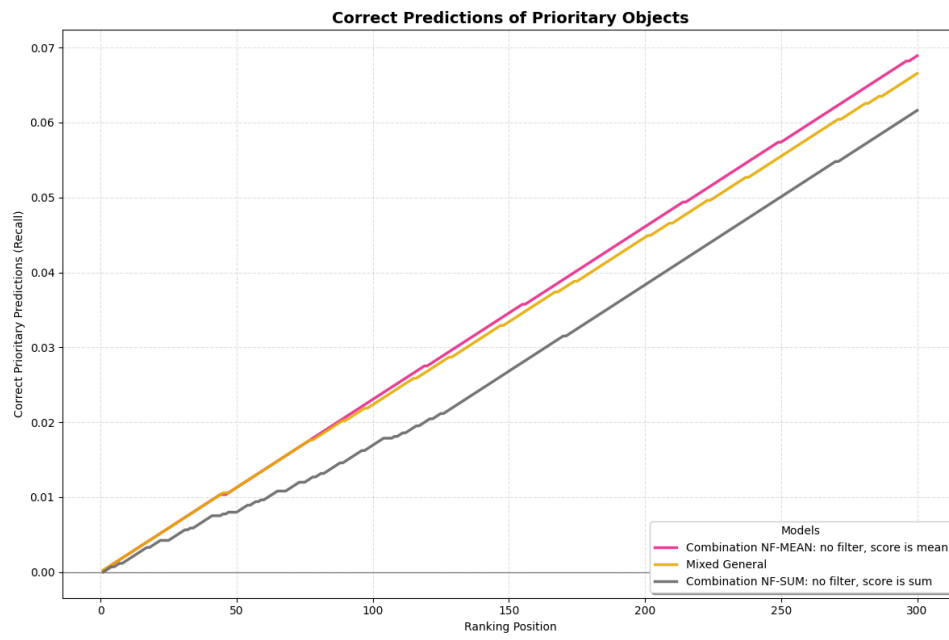


Figure 56 – Recall of priority houses, considering only critical objects.

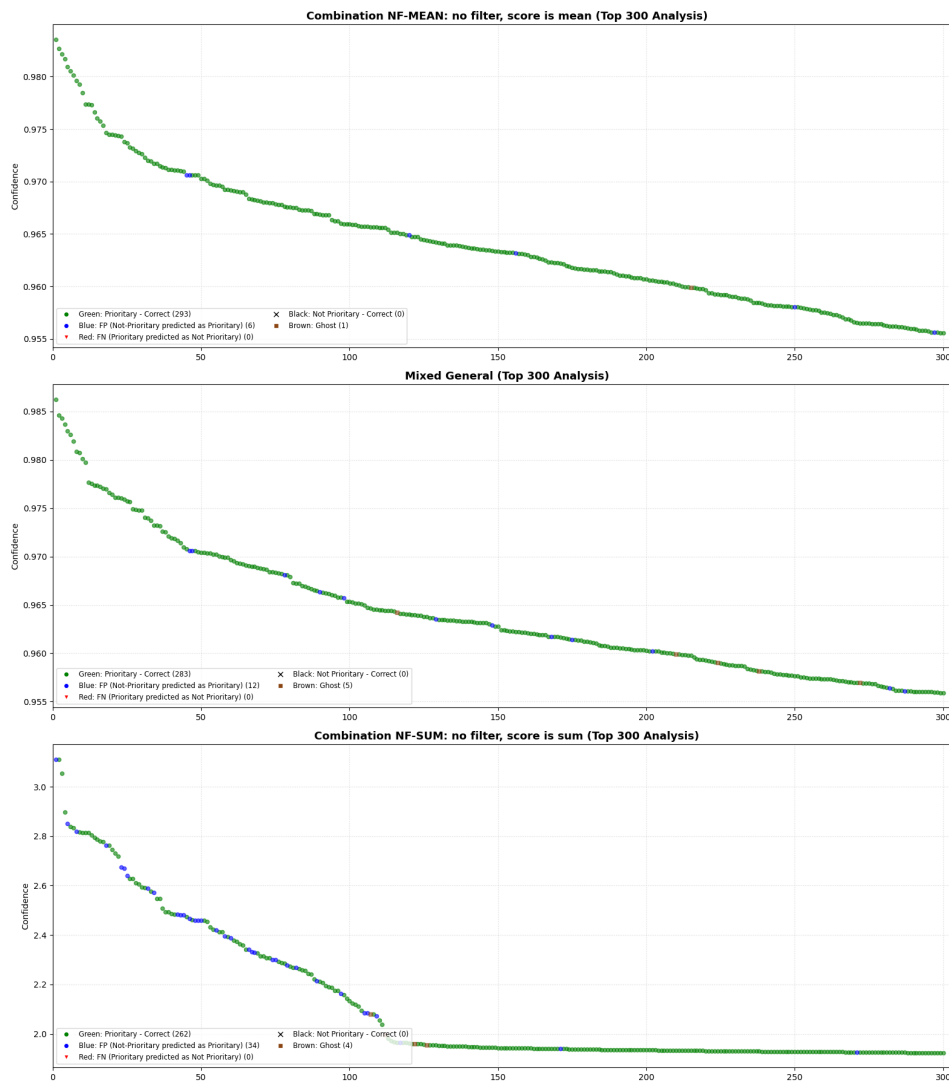


Figure 57 – Classifications of first 300 elements of rankings, considering only priority houses.



Figure 58 – Three first objects presented on NF-MEAN rank of **prioritary houses** with higher confidence.

Rank position 1 | Image: micheal_5670_jpg.rf.029bb112c29611d441328e8b28d70272.jpg



Rank position 2 | Image: 2_0568_jpg.rf.55ae5e3e3d5a5c7085200745296de6fe.jpg



Rank position 3 | Image: 1_0725_jpg.rf.101f30a35f25cae8a1dbcdc9174bca4f.jpg

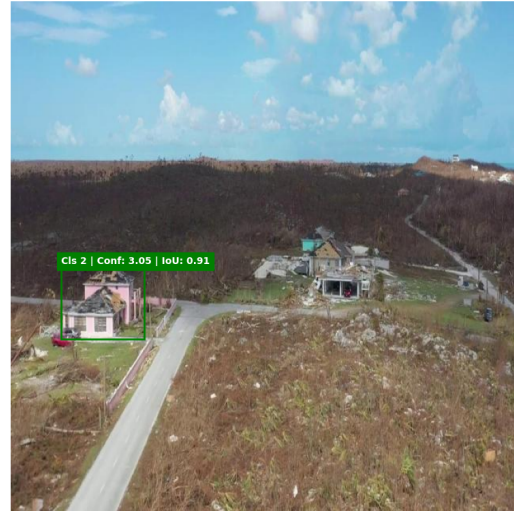


Figure 59 – Three first objects presented on **NF-SUM** rank of **prioritary houses** with higher confidence.

5.6.2 Question Answering Experiments

This section presents additional experiments with DeepSeek, considering different prompts and file formats. The goal of this evaluation is to conduct a deeper analysis of the impact of these inputs on DeepSeek's capacity to provide correct answers to questions. The first 10 objects of each ranking were written to the file to keep its size manageable for reading and processing by DeepSeek. For this evaluation set, the ranks of priority cars only were considered. The file contains the ranking along with descriptive information for each object, including the image in which the object was detected, the assigned class, the model or combination responsible for the prediction, and the corresponding confidence score. For the purpose of conducting new experiments, the prompt and the provided file were adapted to include more details about the context and the semantics of the rank.

The rank file used in the experiments presented in Table 24 is shown below:

Prompt One

You will be provided a question and a text. Your task is to answer the question based on what is on the text. If the answer is clear on the document, you must provide that answer. Answer exactly what was asked. If necessary, you must calculate the answer based on the available information on the document. If it is not necessary, just answer what was requested. In case it is not possible to calculate the answer based on the content of the document, you must inform the user that the question is out of scope and that it is not possible to provide an answer. Provide a final answer that is clear and strict to the point sentence. Critical/priority objects cannot be ignored under any circumstances. Non-critical/not-priority objects can safely be ignored. Critical/priority objects are those from classes 3 and 2. Non-critical/not-priority objects are those from classes 0 and 1.

The prompt used in this set of experiments provides additional information for DeepSeek to answer questions based on the context, considering the semantic application of the rank.

First File Format

This file contains the ranks of: dict_keys(['Combination NF-MEAN: no filter, score is mean', 'Mixed General', 'Combination NF-SUM: no filter, score is sum'])
Rank of most critical, according to Combination NF-MEAN Position on rank: 0, Confidence score: 0.9572239220142365, Model 'Combination NF-MEAN', Class of the found object: '2': Image: 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg. Position on rank: 1, Confidence score: 0.9567886590957642, Model 'Combination NF-MEAN', Class of the found object: '2': Image: 0778_jpg.rf.821504d67b7047a8b7d3c33d3a51ee97.jpg.

Number	Question	Answer	Status
1	What are the first images for each rank?	The first images for each critical class are as follows: - Combination NF-MEAN: 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg - Mixed General: 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg - Combination NF-SUM: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg	Correct
2	What are the first three images for Mixed General rank?	The first three images for Mixed General rank are: 1. 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg 2. 02770_jpg.rf.90927f2f1d6ad4ff26b7f3382c89dd32.jpg 3. 00597_jpg.rf.065d32757b104c0773aa42b0ec3dbc00.jpg	Correct
3	According to Combination NF-SUM rank, which is the most critical image?	Answer: The most critical image is the one under Combination NF-MEAN.	Wrong
4	According to Combination NF-SUM rank, what is the most critical image?	Answer: The most critical image is the one classified as '3' with a confidence score of 0.9649.	Wrong
5	According to Combination NF-SUM rank, what is the name of the most critical image?	The most critical image according to Combination NF-MEAN has a confidence score of 0.9572 and belongs to class '3'. Image: 00419_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg.	Wrong
6	What is the first image on NF-SUM rank?	The first image on NF-SUM rank is at position 0 with a confidence score of 4.7647... The corresponding image path is: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg	Correct
7	What is the first image on Mixed General rank?	Answer: Image: 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg.	Correct
8	What are the first two images on NF-SUM rank?	The first two images on NF-SUM rank are both part of class 2. First image: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg Second image: 0153_jpg.rf.ae53174ac4bc2c63a6ecd21cf021b5c4.jpg	Partially correct
9	What are the first three images on NF-MEAN rank?	The first three images on NF-MEAN rank are: 1. 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg 2. 00419_jpg.rf.aa1504d67b7047a8b7d3c33d3a51ee97.jpg 3. 00136_jpg.rf.1f0dc3347788e223eebdc8f456d582c4.jpg	Partially correct

Table 24 – Results of DeepSeek, considering ranks of priority cars only, using First Prompt and First File Format.

Table 24 presents some achieved results with these presented settings. DeepSeek was used with all its default parameters, except for “temperature”, which was set to zero.

Question 3 was considered incorrect because the provided answer does not address what was asked. Semantically, the answer does not provide any useful information to a hypothetical rescue team and objectively does not answer the question. The same occurred with Question 4. The difference between Questions 3 and 4 is that Question 4 implies that one image from the file should be chosen, but DeepSeek was not able to answer any of these questions correctly with these settings.

Question 5 was considered incorrect because the reported image name is wrong. The reported image is not listed in the file in the first position of NF-SUM rank, which was the expected answer.

Questions 8 and 9 presented the same partially correct result. The first reported image is correct, but the following ones are wrong. The reported image names do not correspond to the file names in the expected positions. This indicates that DeepSeek might have difficulty with more complex and robust questions when dealing with a file of a certain level of complexity.

DeepSeek presented correct results for the strictly direct questions. When the question explicitly asked for the first image in the rank, it provided correct results. However, DeepSeek seems to have difficulty reasoning with the provided prompt, the file content, and the question in order to correctly answer questions using “most critical” contexts.

Of the nine proposed questions, four were correctly answered, three were answered incorrectly, and two were partially correctly answered. With the presented settings, DeepSeek achieved, within the proposed questions, 44% of the questions correctly answered, 22% partially correctly answered, and 33% answered incorrectly.

Aiming to achieve better results, the prompt and the provided file format were modified as follows:

Prompt Two

You will be provided a question and a text. Your task is to answer the user's question based strictly on the provided text.

CORE RULES

1. ****Strict Context:**** Answer ONLY based on the provided document. Do not use outside knowledge.
2. ****Direct Answers:**** Provide clear, strict, and to-the-point answers. Answer exactly what was asked.
3. ****Calculations:**** If the answer requires calculation based on the document's data, perform it accurately. Otherwise, just extract the requested information.
4. ****Out of Scope:**** If the answer cannot be found or calculated from the document, you must reply: "The question is out of scope and it is not possible to provide an answer based on the document."

CRITICALITY PRIORITY HIERARCHY

Objects in the document are classified by numbers. You must strictly follow this ranking from MOST critical to LEAST critical:

* ****Class 3 and Class 2**** Maximum Priority / MOST Critical

* ****Class 1 and Class 0: **** Low Priority / Non-Critical

* When asked for the "most critical" object, select the one with the highest confidence score.

When an image is requested, provide the image name, presented on the rank file, associated with a rank position.

Second File Format

This file presents the top k objects of a rank ordered by priority. This file contains the ranks of: dict_keys(['Combination NF-MEAN: no filter, score is mean', 'Mixed General', 'Combination NF-SUM: no filter, score is sum'])

Rank of most critical, according to Combination NF-MEAN Position on rank: 0, : 0, Confidence score: 0.9572239220142365, Model 'Combination NF-MEAN', Class of the found object: '2': Image name: 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg.
Position on rank: 1, : 1, Confidence score: 0.9567886590957642, Model 'Combination NF-MEAN', Class of the found object: '2': Image name: 0778_jpg.rf.821504d67b7047a8b7d3c33d3a51ee97.jpg.

Table 25 presents the results of DeepSeek using Second Prompt and Second File Format. The information provided in Question 1 is correct. The answer was considered partially correct because the sentence "The most critical image based on the highest confidence score across all methods ..." implies that this is the highest-priority image among all images, which does not apply in the given semantic context. However, the information provided (image name,

Number	Question	Answer	Status
1	According to Combination NF-SUM rank, which is the most critical image?	The most critical image based on the highest confidence score across all methods is: Image name: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg - Class of the found object: '2' - Confidence score: 4.7647	Partially correct
2	According to Combination NF-SUM rank, what is the most critical image?	The most critical image based on the Combination NF-SUM ranking is: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg.	Correct
3	According to Combination NF-SUM rank, what is the name of the most critical image?	The most critical image according to Combination NF-SUM has a confidence score of approximately 0.9572, making it the highest in its category.	Wrong
4	What are the first three images on NF-MEAN rank?	The first three images on NF-MEAN rank are: 1. 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg - Most critical: 0.9572239220142365 - Model: 'Combination NF-MEAN' 2. 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg - Most critical: 0.9649240970611572 - Model: 'Mixed General' 3. 00419_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg - Most critical: 0.9572239220142365 - Model: 'Combination NF-MEAN'	Partially Correct
5	What is the first image on Mixed General rank?	Answer: The first image on Mixed General is 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg	Correct
6	Considering only NF-MEAN rank, what is the most critical image?	The most critical image is 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg from the Combination NF-MEAN model	Correct
7	What are the file names of three most critical images?	The three most critical images, identified by their highest confidence scores across different ranks, are: 1. Image name: 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg - Most Critical Object: Combination NF-MEAN - Confidence Score: 0.9572 2. Image name: 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg - Most Critical Object: Mixed General - Confidence Score: 0.9649 3. Image name: 0154_jpg.rf.80b9400a2596ca4800ee5a81010babad.jpg - Most Critical Object: Combination NF-SUM - Confidence Score: 4.7647	Correct
8	Considering only NF-MEAN, what are the file names of three most critical images?	The file names of the most critical images are: 1. Combination NF-MEAN: 0778_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg 2. Mixed General: 00462_jpg.rf.767a07331e0b07c7a52d8c7b43ccdbb8.jpg 3. Combination NF-SUM: 00419_jpg.rf.da71e0d18e2aae545c02b0e47bf05a68.jpg	Wrong

Table 25 – Results of DeepSeek, using Second Prompt and Second File Format.

confidence score, and class) is correct according to what was requested.

The answer to question 3 was considered incorrect because it does not correspond to the requested image. The answer to question 4 was considered partially correct because only the first reported image is correct. The answer to question 8 was considered incorrect because it does not address what was asked. The question specifically requested information related only to the NF-MEAN rank. Question 7 is correct; however, as evidenced in question 8, DeepSeek appears to have difficulty handling ranks of specific combinations.

Questions 1, 2, and 3 of Table 25 are also presented in Table 24. They were incorrectly answered in Table 24. In Table 25, Question 1 was partially correctly answered, and Question 2 was correctly answered. Although these are semantically equivalent questions, they differ slightly in their formulation. Question 3 was incorrectly answered in both Tables.

DeepSeek did not produce satisfactory results when asked for the first images of a specific rank. In question 7 of Table 25, it was able to correctly identify the first images of each rank. However, in both Tables 24 and 25, it was unable to provide the initial information for a specific rank.

6 Conclusion

This research introduced a framework designed to evaluate collections of images, detect objects of interest, and report the obtained results. The framework combines a general model with multiple specialized models, one dedicated to each class in the dataset.

Results indicate that the use of specialized models is effective in reducing the number of false negatives, although it leads to a decrease in precision. The framework can be executed within a notebook environment, is lightweight, and currently requires no more than 16GB of RAM. However, several improvements are still necessary for deployment in real-world scenarios. In particular, the question answering module requires deeper evaluation and further enhancement.

The presented results demonstrate that the proposed objective was achieved: the NF-MEAN combination successfully recovers objects that were not detected by the Mixed General model and achieves strong performance within the first 300 positions of the confidence ranking.

Across both experimental datasets, the NF-MEAN combination demonstrates to be advantageous in terms of ranking accuracy and recall evaluation metric. The NF-SUM combination also shows advantages in specific scenarios. The proposed framework is extensible and can be further developed by incorporating additional objects, models, and combination strategies, thereby enabling coverage of a broader range of scenarios.

The datasets used in this research were adapted for a weakly supervised setting, incorporating a new division of the training, validation, and test sets. Furthermore, for the cars dataset, two distinct datasets were fused. This dataset reorganization represents a key contribution of our work.

Although DeepSeek showed promising performance on textual inputs, further systematic and in-depth evaluation is necessary to ensure its reliability and robustness in real-world scenarios. As future work, we intend to conduct more in-depth research on DeepSeek to improve its performance and stability, as well as to investigate more effective strategies for structuring the input files to be processed by DeepSeek.

During the development of this research, three articles were produced. The paper “Visual Question Answering in Natural Disaster Scenarios: a Critical Objects Detection Framework” is a direct result of this research.

The paper entitled “Feature Fusion for Graph Convolutional Networks in Semi-Supervised Image Classification” (GAPSKI; VALEM; PEDRONETTE, 2024) was published and orally presented by the author at the 37th Conference on Graphics, Patterns, and Images (SIBGRAPI), and is available at <<https://ieeexplore.ieee.org/abstract/document/10716341/r>

[eferences#references\]\(https://ieeexplore.ieee.org/abstract/document/10716341/references#references\)>](https://ieeexplore.ieee.org/abstract/document/10716341/references#references). Furthermore, the manuscript “Semi-Supervised Image Classification with Multi-Feature Aggregation and Graph Convolutional Networks” has been submitted to the Journal of the Brazilian Computer Society (JBACS) and is currently awaiting review. Additionally, the article “Visual Question Answering in Natural Disaster Scenarios: a Critical Objects Detection Framework” has also been submitted to the JBACS and is pending the initiation of the peer-review process.

In future work, different combination strategies and their corresponding outcomes could be further explored. For instance, when two combinations yield conflicting results, as observed with the NF-MEAN and NF-SUM rankings in the houses dataset, criteria for determining which result is more reliable should be investigated. Additionally, object detection algorithms other than YOLO could be examined, such as R-CNN, RetinaNet, Single Shot MultiBox Detector (SSD), among others.

Strategies involving the combination of bounding boxes, such as merging their coordinates, could also be investigated. Since misclassification cases were observed, the use of an independent classifier to assign a class to an object crop, rather than relying on the class predicted by the object detector, could also be explored to determine whether this approach reduces false negatives caused by misclassification.

Bibliography

AALST, M. K. V. The impacts of climate change on the risk of natural disasters. *Disasters*, v. 30, n. 1, p. 5–18, 2006. Available in: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9523.2006.00303.x>. Access on: 14 May 2026.

AIZAWA, A. An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, v. 39, n. 1, p. 45–65, 2003. ISSN 0306-4573. Available in: <https://www.sciencedirect.com/science/article/pii/S0306457302000213>. Access on: 14 May 2026.

AL-MOHANNADI, A. et al. *DisasterVQA: A Visual Question Answering Benchmark Dataset for Disaster Scenes*. 2026. Available in: <https://arxiv.org/abs/2601.13839>. Access on: 14 May 2026.

ALLAM, A. M. N.; HAGGAG, M. H. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, v. 2, n. 3, 2012.

AMJOUD, A. B.; AMROUCH, M. Object detection using deep learning, cnns and vision transformers: A review. *IEEE Access*, v. 11, p. 35479–35516, 2023.

ANTONIO, M.; SOARES, C.; PARREIRAS, F. S. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences*, v. 32, n. 6, p. 635–646, 2020. ISSN 1319-1578. Available in: <https://www.sciencedirect.com/science/article/pii/S131915781830082X>. Access on: 14 May 2026.

ASSAYED, S. K.; SHAALAN, K.; ALKHATIB, M. A chatbot intent classifier for supporting high school students. *EAI Endorsed Transactions on Scalable Information Systems*, EAI, v. 10, n. 3, 12 2022.

ATUHURRA, J. et al. *Domain Adaptation in Intent Classification Systems: A Review*. 2024. Available in: <https://arxiv.org/abs/2404.14415>. Access on: 14 May 2026.

BAKAROV, A. *A Survey of Word Embeddings Evaluation Methods*. 2018. Available in: <https://arxiv.org/abs/1801.09536>. Access on: 14 May 2026.

BANHOLZER, S.; KOSSIN, J.; DONNER, S. The impact of climate change on natural disasters. In: SINGH, A.; ZOMMERS, Z. (Ed.). *Reducing Disaster: Early Warning Systems For Climate Change*. Dordrecht: Springer Netherlands, 2014. p. 21–49. ISBN 978-94-017-8598-3. Available in: https://doi.org/10.1007/978-94-017-8598-3_2. Access on: 14 May 2026.

BEIL, F.; ESTER, M.; XU, X. Frequent term-based text clustering. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2002. (KDD '02), p. 436–442. ISBN 158113567X. Available in: <https://doi.org/10.1145/775047.775110>. Access on: 14 May 2026.

BENEVOLENZA, M. A.; DERIGNE, L. The impact of climate change and natural disasters on vulnerable populations: A systematic review of literature. *Journal of Human Behavior in the Social Environment*, Routledge, v. 29, n. 2, p. 266–281, 2019. Available in: <https://doi.org/10.1080/10911359.2018.1527739>. Access on: 14 May 2026.

- BLAGA, B.-C.-Z.; NEDEVSKI, S. Improving counting accuracy of post-disaster visual question answering for remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, p. 1–19, 2026.
- BODLA, N. et al. Soft-nms–improving object detection with one line of code. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 5561–5569.
- BROUWER, T. et al. Probabilistic flood extent estimates from social media flood observations. *Natural Hazards and Earth System Sciences*, v. 17, n. 5, p. 735–747, 2017. Available in: <https://nhess.copernicus.org/articles/17/735/2017/>. Access on: 14 May 2026.
- CARION, N. et al. End-to-end object detection with transformers. In: VEDALDI, A. et al. (Ed.). *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020. p. 213–229. ISBN 978-3-030-58452-8.
- CHEN, Q.; ZHUO, Z.; WANG, W. *BERT for Joint Intent Classification and Slot Filling*. 2019. Available in: <https://arxiv.org/abs/1902.10909>. Access on: 14 May 2026.
- CHENG, C.-S.; BEHZADAN, A. H.; NOSHADRAVAN, A. Deep learning for post-hurricane aerial damage assessment of buildings. *Computer-Aided Civil and Infrastructure Engineering*, Wiley Online Library, v. 36, n. 6, p. 695–710, 2021.
- CHENG, T. et al. Yolo-world: Real-time open-vocabulary object detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2024. p. 16901–16911.
- CHOI, D.; LEE, S.; SONG, Y. Better safe than sorry? overreaction problem of vision language models in visual emergency recognition. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2026. p. 4724–4732.
- CHRISTIAN, H.; AGUS, M. P.; SUHARTONO, D. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, v. 7, n. 4, p. 285–294, 2016.
- COPE, K. L.; ZHANG, L. A hierarchical dictionary method for analyzing legal and political texts via nested ngrams. *Virginia Law and Economics Research Paper*, n. 2023-16, August 2023. Available in: <https://ssrn.com/abstract=4541799>. Access on: 14 May 2026.
- DAI, D. et al. *DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models*. 2024. Available in: <https://arxiv.org/abs/2401.06066>. Access on: 14 May 2026.
- DEEPSEEK-AI et al. *DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model*. 2024. Available in: <https://arxiv.org/abs/2405.04434>. Access on: 14 May 2026.
- DEEPSEEK-AI et al. *DeepSeek-V3 Technical Report*. 2025. Available in: <https://arxiv.org/abs/2412.19437>. Access on: 14 May 2026.
- DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019.
- DWYER, B.; NELSON, J.; HANSEN, T. *Roboflow (Version 1.0) [Software]*. 2024. Available in: <https://roboflow.com>. Access on: 14 May 2026.

GAPSKI, M. C. B.; VALEM, L. P.; PEDRONETTE, D. C. G. Feature fusion for graph convolutional networks in semi-supervised image classification. In: *2024 37th SIBGRAP Conference on Graphics, Patterns and Images (SIBGRAP)*. [S.l.: s.n.], 2024. p. 1–6.

GASPARETTO, A. et al. A survey on text classification algorithms: From text to predictions. *Information*, v. 13, n. 2, 2022. ISSN 2078-2489. Available in: <https://www.mdpi.com/2078-2489/13/2/83>. Access on: 14 May 2026.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014.

GREENWOOD, F.; NELSON, E. L.; GREENOUGH, P. G. Flying into the hurricane: A case study of uav use in damage assessment during the 2017 hurricanes in texas and florida. *PLOS ONE*, Public Library of Science, v. 15, n. 2, p. 1–30, 02 2020. Available in: <https://doi.org/10.1371/journal.pone.0227808>. Access on: 14 May 2026.

GUO, D. et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, Springer Science and Business Media LLC, v. 645, n. 8081, p. 633–638, set. 2025. ISSN 1476-4687. Available in: <http://dx.doi.org/10.1038/s41586-025-09422-z>. Access on: 14 May 2026.

HARRIS, Z. S. Distributional structure. In: _____. *Papers in Structural and Transformational Linguistics*. Dordrecht: Springer Netherlands, 1970. p. 775–794. ISBN 978-94-017-6059-1. Available in: https://doi.org/10.1007/978-94-017-6059-1_36. Access on: 14 May 2026.

HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

HE, P. et al. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. Available in: <https://arxiv.org/abs/2006.03654>. Access on: 14 May 2026.

HERNÁNDEZ, D. et al. Flood detection using real-time image segmentation from unmanned aerial vehicles on edge-computing platform. *Remote Sensing*, v. 14, n. 1, 2022. ISSN 2072-4292. Available in: <https://www.mdpi.com/2072-4292/14/1/223>. Access on: 14 May 2026.

HICKMAN, L. et al. Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, v. 25, n. 1, p. 114–146, 2022. Available in: <https://doi.org/10.1177/1094428120971683>. Access on: 14 May 2026.

JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

JIANG, P. et al. A review of yolo algorithm developments. *Procedia Computer Science*, v. 199, p. 1066–1073, 2022. ISSN 1877-0509. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19. Available in: <https://www.sciencedirect.com/science/article/pii/S1877050922001363>. Access on: 14 May 2026.

JOCHER, G.; QIU, J.; CHAURASIA, A. *Ultralytics YOLO*. 2023. Available in: <https://github.com/ultralytics/ultralytics>. Access on: 14 May 2026.

KATZ, S. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 35, n. 3, p. 400–401, 1987.

KIM, S. S. et al. Disaster damage investigation using artificial intelligence and drone mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B3-2022, p. 1109–1114, 2022. Available in: <https://isprs-archives.copernicus.org/articles/XLIII-B3-2022/1109/2022/>. Access on: 14 May 2026.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.

LI, Y.; CHUNG, S. M.; HOLT, J. D. Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering*, v. 64, n. 1, p. 381–404, 2008. ISSN 0169-023X. Fourth International Conference on Business Process Management (BPM 2006) 8th International Conference on Enterprise Information Systems (ICEIS' 2006). Available in: <https://www.sciencedirect.com/science/article/pii/S0169023X07001541>. Access on: 14 May 2026.

LIU, J. et al. Edllm: A novel entity detection method for test data integrating yolo-world and llm. In: HUANG, D.-S. et al. (Ed.). *Advanced Intelligent Computing Technology and Applications*. Singapore: Springer Nature Singapore, 2025. p. 195–206. ISBN 978-981-96-9815-8.

LIU, L. et al. Deep learning for generic object detection: A survey. *International journal of computer vision*, Springer, v. 128, p. 261–318, 2020.

LIU, M.; LIU, J.; HU, H. A novel deep learning network model for extracting lake water bodies from remote sensing images. *Applied Sciences*, v. 14, n. 4, 2024. ISSN 2076-3417. Available in: <https://www.mdpi.com/2076-3417/14/4/1344>. Access on: 14 May 2026.

MARTINS-FILHO, P. R. et al. *Catastrophic Floods in Rio Grande do Sul, Brazil: The Need for Public Health Responses to Potential Infectious Disease Outbreaks*. [S.l.]: SciELO Brasil, 2024. e00603–2024 p.

MISHRA, B. et al. Drone-surveillance for search and rescue in natural disaster. *Computer Communications*, v. 156, p. 1–10, 2020. ISSN 0140-3664. Available in: <https://www.sciencedirect.com/science/article/pii/S0140366419318602>. Access on: 14 May 2026.

MOECHAMMAD, S.; CAHYA, R.; BERKAH, A. N. A. Detecting body parts from natural disaster victims using you only look once (yolo). *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 1073, n. 1, p. 012062, feb 2021. Available in: <https://dx.doi.org/10.1088/1757-899X/1073/1/012062>. Access on: 14 May 2026.

NAJIBI, M.; RASTEGARI, M.; DAVIS, L. S. G-cnn: An iterative grid based object detector. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.

NEHETE, P. et al. Object detection and classification in human rescue operations: Deep learning strategies for flooded environments. *International Journal of Safety and Security Engineering*, v. 14, p. 599–611, 04 2024.

- NEUBECK, A.; GOOL, L. V. Efficient non-maximum suppression. In: *18th International Conference on Pattern Recognition (ICPR'06)*. [S.l.: s.n.], 2006. v. 3, p. 850–855.
- NIHAL, R. A. et al. *UAV-Enhanced Combination to Application: Comprehensive Analysis and Benchmarking of a Human Detection Dataset for Disaster Scenarios*. 2024. Available in: <https://arxiv.org/abs/2408.04922>. Access on: 14 May 2026.
- OPENAI. *ChatGPT*. 2024. Available in: <https://openai.com/chatgpt>. Access on: 14 May 2026.
- O'SHEA, K.; NASH, R. *An Introduction to Convolutional Neural Networks*. 2015.
- PATIL, R. et al. A survey of text representation and embedding techniques in nlp. *IEEE Access*, v. 11, p. 36120–36146, 2023.
- PEREIRA, R. de C. et al. Diversity-informed fusion for sclerotinia sclerotiorum and white mold detection. In: *2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2025. p. 1–6.
- PRABU, S.; GNANASEKAR, J. Realtime object detection through m-resnet in video surveillance system. *Intelligent Automation & Soft Computing*, v. 35, n. 2, 2023.
- RAHNEMOONFAR, M.; CHOWDHURY, T.; MURPHY, R. Rescuenet: A high resolution uav semantic segmentation dataset for natural disaster damage assessment. *Scientific Data*, v. 10, n. 1, p. 913, 12 2023. ISSN 2052-4463. Available in: <https://doi.org/10.1038/s41597-023-02799-4>. Access on: 14 May 2026.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 779–788.
- ROBOFLOW. Open Source Dataset, *Building Damage Detection-undef Dataset*. Roboflow, 2024. <https://universe.roboflow.com/al-mdb78/building-damage-detection-undef>. Available in: <https://universe.roboflow.com/al-mdb78/building-damage-detection-undef>. Access on: 14 May 2026.
- SAI, D. H. et al. Enhancing emergency response with real-time video analytics for natural disaster management. In: *2025 Fourth International Conference on Smart Technologies, Communication and Robotics (STCR)*. [S.l.: s.n.], 2025. p. 1–7.
- SAID, N. et al. Natural disasters detection in social media and satellite imagery: a survey. *Multimedia Tools and Applications*, v. 78, n. 22, p. 31267–31302, 11 2019. ISSN 1573-7721. Available in: <https://doi.org/10.1007/s11042-019-07942-1>. Access on: 14 May 2026.
- SANTANA, B. et al. A survey on narrative extraction from textual data. *Artificial Intelligence Review*, v. 56, n. 8, p. 8393–8435, August 2023. ISSN 1573-7462. Available in: <https://doi.org/10.1007/s10462-022-10338-7>. Access on: 14 May 2026.
- SAROSA, M.; MUNA, N.; ROHADI, E. Detection of natural disaster victims using you only look once (yolo). *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 1098, n. 3, p. 032076, mar 2021. Available in: <https://dx.doi.org/10.1088/1757-899X/1098/3/032076>. Access on: 14 May 2026.

SEDANFROMHX. Open Source Dataset, *convert2yolov5 Dataset*. Roboflow, 2023. Available in: <https://universe.roboflow.com/sedanfromhx/convert2yolov5>. Access on: 02 September 2026.

SEDANFROMHX. Open Source Dataset, *yolov5WithSUV Dataset*. Roboflow, 2024. Available in: <https://universe.roboflow.com/sedanfromhx/yolov5withsuv>. Access on: 02 September 2026.

SEO, J.; DUQUE, L.; WACKER, J. Drone-enabled bridge inspection methodology and application. *Automation in Construction*, v. 94, p. 112–126, 2018. ISSN 0926-5805. Available in: <https://www.sciencedirect.com/science/article/pii/S0926580517309755>. Access on: 14 May 2026.

SHEHZADI, T. et al. *Object Detection with Transformers: A Review*. 2023. Available in: <https://arxiv.org/abs/2306.04670>. Access on: 14 May 2026.

SOLOVYEV, R.; WANG, W.; GABRUSEVA, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, v. 107, p. 104117, 2021. ISSN 0262-8856. Available in: <https://www.sciencedirect.com/science/article/pii/S0262885621000226>. Access on: 14 May 2026.

SUN, Y.; WANG, C.; PENG, Y. Reducing hallucinations: Enhancing vqa for flood disaster damage assessment with visual contexts. In: *Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security*. New York, NY, USA: Association for Computing Machinery, 2024. (ASENS '24), p. 587–590. ISBN 9798400709784. Available in: <https://doi.org/10.1145/3677182.3677287>. Access on: 14 May 2026.

TALUKDER, B. et al. Climate change-accelerated ocean biodiversity loss & associated planetary health impacts. *The Journal of Climate Change and Health*, v. 6, p. 100114, 2022. ISSN 2667-2782. Available in: <https://www.sciencedirect.com/science/article/pii/S2667278222000037>. Access on: 14 May 2026.

TIAN, Y.; YE, Q.; DOERMANN, D. Yolo12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

TIAN, Y.; YE, Q.; DOERMANN, D. *YOLO12: Attention-Centric Real-Time Object Detectors*. 2025. Available in: <https://github.com/sunsmarterjie/yolov12>. Access on: 14 May 2026.

TURING, A. M. Computing machinery and intelligence. *Mind*, Oxford University Press, v. 59, n. 236, p. 433–460, 1950.

VASWANI, A. et al. Attention is all you need. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Available in: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Access on: 14 May 2026.

WANG, J. et al. *DisasterM3: A Remote Sensing Vision-Language Dataset for Disaster Damage Assessment and Response*. 2025. Available in: <https://arxiv.org/abs/2505.21089>. Access on: 14 May 2026.

WANG, Q. et al. Sal-yolo-deepseek: a lightweight real-time detection and llm-driven decision framework for intelligent escalator safety monitoring. *Scientific Reports*, v. 15, n. 1, p. 40600, 2025. ISSN 2045-2322. Available in: <https://doi.org/10.1038/s41598-025-24260-9>. Access on: 14 May 2026.

WHITEHURST, D. et al. Drone-based community assessment, planning, and disaster risk management for sustainable development. *Remote Sensing*, v. 13, n. 9, 2021. ISSN 2072-4292. Available in: <<https://www.mdpi.com/2072-4292/13/9/1739>>. Access on: 14 May 2026.

ZHIQIANG, W.; JUN, L. A review of object detection based on convolutional neural network. In: *2017 36th Chinese Control Conference (CCC)*. [S.l.: s.n.], 2017. p. 11104–11109.

ZHOU, W.; BRAS, R. L.; CHOI, Y. *Commonsense Knowledge Transfer for Pre-trained Language Models*. 2023.

DADOS CURRICULARES

IDENTIFICAÇÃO	
	02/04/1999
Nacionalidade	Brasileira
Nome em citações bibliográficas	M. C. Bulach Gapski GAPSKI, M. C. B.
Currículo Lattes	http://lattes.cnpq.br/8442104594253502
ORCID	https://orcid.org/0009-0003-2732-3862
FORMAÇÃO ACADÊMICA	
28/07/2023	Bacharelado em Ciências da Computação Universidade Estadual Paulista “Júlio de Mesquita Filho” - Instituto de Geociências e Ciências Exatas - Campus de Rio Claro
PRODUÇÃO BIBLIOGRÁFICA	
M. C. Bulach Gapski, L. P. Valem and D. C. Guimarães Pedronette, "Feature Fusion for Graph Convolutional Networks in Semi-Supervised Image Classification," 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Manaus, Brazil, 2024, pp. 1-6, doi: 10.1109/SIBGRAPI62404.2024.10716341.	
PARTICIPAÇÃO EM EVENTOS CIENTÍFICOS	
37th Conference on Graphics, Patterns and Images (SIBGRAPI 2024). Apresentação de Trabalho. Manaus, AM, Brasil, 30 de setembro a 03 de outubro 2024.	