National College of Ireland

Enhancing Real-Time Fire Detection with RT-DETR and Optimized Dataset Preparation

MSc Research Project
Programme Name

Joseph Raju Myla
Student ID: X23224444@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Joseph Raju Myla
Student ID:	X23224444@student.ncirl.ie
Programme:	DATA ANALYTICS **Year:** 2024-2025
Module:	MSC RESEARCH PROJECT
Supervisor:	
Submission Due Date:	29 January 2025
Project Title:	**Enhancing Real-Time Fire Detection with RT-DETR and Optimized Dataset Preparation**
Word Count:	7113 **Page Count 22**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Joseph Raju Myla

Date: 29/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Real-Time Fire Detection with RT-DETR and Optimized Dataset Preparation

Joseph Raju Myla
Student ID: X23224444@student.ncirl.ie

Abstract

Real-time fire detection systems face significant challenges achieving high accuracy and efficient processing speeds. This research looks at the optimization of dataset preparation techniques that could be used to implement RT-DETR in fire detection systems, together with augmentation strategies. The presented study aims to bridge the critical gap between dataset preparation methodologies and the performance of transformer-based architectures in safety-critical applications.

In the work, a holistic approach was followed with the RT-DETR-L architecture. Extensive data augmentation is done through geometrical transformations, including changes in intensity. The implementation is performed on Google Colab, running on an A100 40GB GPU infrastructure. It comes with a dataset of 2,200 validation images. The pipeline is carefully designed in such ways that the aspect ratios of the images are preserved, standardized at 640x640 resolution.

Very remarkable performance metrics are presented, with 0.985 for mAP@50 and 0.949 for mAP@50-95. The system also retains very high precision regarding fire detection at 0.991 and smoke detection at 0.962, with only 16.6ms of processing time per image. This represents substantial outperformance compared to the current benchmarks while preserving real-time processing capabilities.

These results provided new baselines for real-time fire detection systems and came with useful insights into the optimization of dataset preparation for transformer-based architecture. The contribution of this study both in theoretical understanding and practical implementation strategies advances the development of enhanced fire detection systems, which shall be very operational, especially in application areas concerning the safety of human life where a real-time response is expected.

1 Introduction

In recent years, with the increasing trend in fire incident phenomena, the need has emerged to consider more effective and efficient early detection systems. The World Health Organization, WHO (2023), estimated that fire incidents claim over 180,000 lives each year, with billions of dollars in property damages. While traditional fire detection relies on physical sensors, recently the computer vision-based approach using deep learning has turned out to be one of the promising solutions for early detection in large-scale environments where conventional methods prove to be quite inadequate.

Real-Time Detection Transformer architecture has pushed the envelope for object detection capability. In fact, recent studies conducted by Chen et al. (2023) show that RT-DETR exhibits higher performance in real-world applications compared to typical convolutional neural networks. Besides, the efficiency of such a model mainly relies on the nature and preprocessing of the data on which these models have been trained, since in some challenging scenarios such as fire and smoke detection, the nature of the scenarios keeps on varying.

The current research activities in fire detection systems are dominated by model architectures and training strategies. However, there is a great lack of detailed understanding of how concretely applied dataset preparation techniques and augmentation strategies influence the performance of RT-DETR in fire detection tasks. Dynamic behaviour, variable lighting, and varied pattern of appearance are some unique salient features of fire or smoke, for which dataset preparation needs a special approach, which, to the best of one's abilities, has not been fully addressed by the current literature.

This research tries to fill this gap by developing optimized dataset preparation techniques specially for fire detection applications.

The central research question guiding this study is: "How can dataset preparation techniques and augmentation strategies be optimized to enhance RT-DETR's performance in real-time fire and smoke detection systems?"This question addresses one of the key needs in the domain, in the sense that improved dataset preparation might significantly improve detection performance with minimal changes in the architecture of the supporting model.

Research Objectives:

1. **To evaluate and compare the effectiveness of various image preprocessing techniques for fire detection:**
    - **Success Metric: Demonstrate statistically significant improvement in detection accuracy ($p < 0.05$)**
    - **Evaluation Method: Comparative analysis of model performance metrics across different preprocessing approaches**
2. **To develop and implement optimized data augmentation strategies specifically for fire and smoke detection:**
    - **Success Metric: Achieve 10% improvement in both precision and recall over baseline performance**
    - **Evaluation Method: Comprehensive performance testing against established baseline metrics**
3. **To establish a systematic framework for dataset preparation in fire detection applications:**
    - **Success Metric: Framework validation through successful implementation and reproducibility testing**
    - **Evaluation Method: Empirical validation of framework effectiveness through implementation results**

The fourth objective is the quantification of the relationship between dataset preparation techniques and model performance metrics. Testing will therefore be done through statistical analyses for the existence of correlations between some preparation methods and performance indicators. Success will be considered when statistically significant relationships are defined clearly.

The research methodology employs a systematic approach combining empirical analysis and experimental validation. The implementation begins with dataset preparation, utilizing a substantial collection of fire and smoke images. The methodology incorporates two main stages of data processing: First, fundamental preprocessing techniques are applied, including image orientation standardization and dimensional normalization. This involves a further step of augmentation that includes geometric and intensity level transformations, uniquely adapted

for feature enhancement in smoke and fire. In such a way, all stages of preparation are thoroughly processed, without compromising key visual information needed for effective detection.

This report skeleton logically flows in the research process: Chapter 2 discusses related work with respect to dataset preparation, augmentation strategies, and fire detection systems; thus, it sets a theoretical framework. Chapter 3 specifies the methodology that was followed during this research work. Together with the description of the experimental approach, the evaluation methods are given. Chapter 4 designs the specification of the dataset preparation framework, while Chapter 5 documents the practical implementation of the proposed solutions. Chapter 6 summarizes the findings in detail by discussing implications. Finally, Chapter 7 concludes this research by summarizing the key findings and suggesting a direction for future investigation.

2 Literature Review

Recent advances in deep learning have enabled major evolutions in computer vision-based fire detection systems, and it now comes to be a promising alternative for conventional sensor-based fire detection approaches. The research activities on dataset preparation and different techniques of deep learning for the detection of fire in various environments, with especial concern for real-time and/or performance optimization, will be discussed in this current section.

2.1 Evolution of Fire Detection Approaches

At present, fire detection has traversed from conventional sensor-based techniques to some sophisticated approaches involving computer vision. Goel et al (2023). proposed a system that used a CNN with edge detection and thresholding for detecting fire in real time. They reported significant improvement in the accuracy of detection upon incorporating aerial-based monitoring systems. They built a dataset of high-quality fire images at regular intervals totalling 10,000, which underlined the importance of diverse training data. However, their approach seemed limited for handling complex environmental conditions.

The system was further enhanced by Shah and Gajjar (2024), whereby they introduced an advanced colour detection methodology using HSV and YCbCr color models. Their methodology overcame the drawbacks of the previous systems by using motion detection through frame comparison, which drastically reduced false positives. Performance for their system degraded under conditions where either the background was highly complex, or illumination conditions were not stationary.

Wang et al. (2024) gave remarkable improvements by presenting an enhanced YOLOv5s with a null convolution and the introduction of coordinate attention mechanism. Its implementation reduced the computation of the model by 1.8G and had simultaneously enhanced the accuracy of detection by 3.1%, proving lightweight models could also do accuracy like larger architectures. This then became a more important breakthrough toward deployment on embedded systems and mobile devices.

2.2 Dataset Preparation for Deep Learning Models

Quality and preparation of a dataset are extremely vital about model performance. Uddin et al (2023). conducted extensive experiments with YOLOv8 and YOLOv5 models on a custom-made dataset made up of 8,000 images. The authors have developed this dataset in a very structured manner, including good split ratios to cover various scenarios. Indeed, their work resulted in remarkable improvements in model performance, reflecting precision improvements of about 2-4% compared to the current systems, while having better performance in resource-constrained environmental conditions.

Li and Jie (2023) extended that further by proposing a YOLOv7-based system with an edge computing approach. They developed a detailed dataset of 14,000 fire pictures, of which over 5,000 were in the forest environment. Their performance was great: According to them, their model achieved over 96% F1 score, tested at 98.8% average precision and a recall rate of 95.8%. Success underlined the importance of collecting domain-specific data and proper augmentation strategy.

2.3 Real-Time Detection and Performance Optimization

Achieving high accuracy and real-time performance simultaneously is challenging. In this regard, Chetoui and Akhloufi(2024) made their vital contributions to fine-tuning YOLOv8 and YOLOv7 models on more than 11,000 images. Their variant model comparisons are systematic and show that YOLOv8x manages a precision of 0.954 with a recall of 0.848, while YOLOv8l comes out with quite similar but also impressive results: precision of 0.949 and recall of 0.837.

Until Lv et al (2023). came with RT-DETR, showing that a transformer-based architecture outperformed YOLO models in real-time tasks. The authors proposed a first approach able to reach 53.1% AP on COCO val2017 while running as fast as 108 FPS with a ResNet50 backbone. The implementation with a ResNet-101 backbone achieved 54.3% AP at 74 FPS. Without losing much accuracy, their method was able to get rid of NMS after post-processing.

2.4 Advanced Detection Methods and Model Architecture

Recent works have helped to bring out a set of insights on the relative performance of several detection methods. Jiang et al. (2023) enhance the YOLOx architecture through the incorporation of sophisticated attention mechanisms and WHIoU loss algorithm with the regard to variation challenge brought about by variations in fine shape and have given an implementation which highlights significant improvements of mean average precision through data augmentation and attention mechanisms while realizing these gains without additional model parameters.

In this respect, Madkar et al (2022). carried out comparative experiments between the CNN and YOLOv5 methods based on a database of 400 videos, whose results proved that YOLOv5 performed better in terms of the accuracy of fire localization and real-time processing, especially in terms of adaptation to variable environmental conditions. Thus, it is illustrated that the very architecture of YOLOv5 is particularly suitable for handling a dynamic nature of the fire detection scenarios.

2.5 Current Challenges and Future Directions

Gragnaniello et al. (2024) review a further set of 153 papers and 17 datasets. The results indicate an important lack of awareness related to the impact that dataset preparation has on the performance of deep learning models among different scenarios. They reported that while many works have been conducted on model architectures, the basic optimization of the dataset is still underexplored, especially in transformer-based models. They found that current methods of preparing datasets have many generic guidelines which do not fit the flourishing requirements of such advanced architectures like RT-DETR.

Literature highlights three major knowledge gaps in the existing studies: while recent publications have shown that RT-DETR outperforms traditional YOLO models (Lv et al., 2023), no one has discussed how this affected dataset preparation techniques for RT-DETR while developing a fire detector. Though existing augmentation strategies are successful with CNN-based models, (Goel et al. 2023; Wang et al. 2024), augmentation strategies for transformer architectures themselves have not been systematically evaluated and optimized. Thirdly, there is a great lack of standardization in dataset preparation protocols for state-of-the-art research tailored just for real-time fire detection applications.

These differences in performance across the studies further emphasize this gap. While Li and Jie (2023) realized very good results with YOLOv7 on a well-prepared dataset of 14,000 images, Chetoui and Akhloufi(2024) similarly demonstrated varied but strong performances among different variants of YOLOv8. However, neither of these works focused on how their respective methods in preparing the dataset might best be used for transformer-based architectures. This turns out to be a serious limitation since RT-DETR has shown great potential in capturing real-time improvements without the need for post-processing steps.

These gaps identified herein thus inform and help justify the following research question: "How can the dataset preparation techniques and data augmentation strategies be optimized to enhance RT-DETR performance in real-time fire and smoke detection systems?" This is a very fundamental question, since existing literature reveals that while the RT-DETR architecture offers promising advantages, its full potential in fire detection applications could be constrained by some suboptimal approaches in dataset preparation.

This research proposal will fill these gaps by proposing dataset preparation methods dedicated to and optimally leveraging the architectural advantages that RT-DETR introduced. Consequently, this research shall contribute to allowing transformer-based object detection systems to achieve real-time performance while improving their detection accuracy. It is now timely, considering that transformer-based models reached the leading performance in real-time applications, as shown by (Lv et al.,2023) and given the pressing need for advancements in fire detection systems as recently underlined in Gragnaniello et al(2024).

The work to be done in preparing the datasets represents the crawling of preparation techniques to encompass few of the identified limitations, leveraging the demonstrated advantages of the transformer-based architecture as part of the implementation of RT-DETR. This research direction has specific relevance due to increasing deep learning adoptions into safety-critical applications and because of growing demands for robust, real-time fire detection systems.

3 Research Methodology

3.1 Dataset Preparation and Augmentation Strategy

Preparation of the dataset followed a rich augmentation technique in choosing only those that better characterized the challenges of fire or smoke detection.:

The horizontal flipping is done to reduce directional bias in training data. Horizontal flipping reflects images randomly with a probability of 0.5 to improve the observing capabilities of the model in various orientations of the frame. This augmentation is very useful for the fire detection system deployed at any orientation of the camera.

Rotation augmentation included fixed 90-degree rotations, while random rotations ranged from -15 to +15 degrees. The large 90-degree rotations prepare the model for great changes in camera orientation; the smaller random rotations simulate slight camera misalignments and other natural variations in angles of view. This dual approach was inspired by real-world deployment scenarios where cameras usually are mounted at varying angles or may be subjected to environmental movements.

The transformation of shearing is done horizontally and vertically at ±10 degrees, which therefore simulates the captured images in different perspectives. This paper considers this very important for fire detec-tion systems deployed in an urban environment or complex architectural settings where perspective distortion is common. The specific range of ±10 degrees was determined through preliminary experiments, which provided the optimal balance between data diversity and maintaining realistic image appearances.

The strategy for adjusting exposure ranges from -10% to +10%, catering for the most difficult lighting conditions, which often leak into fire detection scenarios. This range has been chosen because it reproduces scenes both darker, where the fire is more visible, and near the bright conditions required for smoke detection. A continuous random distribution within this range is done to ensure smooth coverage of possible lighting conditions.

The salt and pepper noise added during training affects 0.1% of the pixels, inducing immunity against sensor noise and a variety of other image artifacts common in low-illuminated real-world surveillance systems or for cameras of a lower quality. This percentage was chosen to maintain a good balance between robustness against noise and integrity of important visual features.

3.2 Data Processing Framework

The image processing system implements an advanced augmentation pipeline, by means of which images are pre-processed and enhanced. Images are standardized to a resolution of 640x640, with their aspect ratios preserved by intelligent padding. Colour normalization is performed in an exact manner at 0.485, 0.456, and 0.406, with respective standard deviations of 0.229, 0.224, and 0.225 computed from a very thorough analysis of the dataset.

3.3 Training Methodology and Configuration

The training process applies an approach carefully structured for the RT-DETR architecture. Input resolution was chosen as 640x640 pixels, which balances well between computational efficiency and detection accuracy, which is somewhat vital in the case of fire instances at small sizes in early stages. A batch size of 16 maximizes the utilization of the GPU while maintaining stable updates of gradients, which is crucial for transformer based.

The policy followed a learning schedule of 0.01 with cosine annealing, hence allowing it to be fast in learning while also avoiding overshooting the best parameters after some training steps. This was particularly important given the complexity of fire and smoke patterns and the need to fine-tune the pre-trained weights effectively.

3.4 Research Infrastructure and Environment

The research used the A100 40GB GPU infrastructure on Google Colab to make the computations uniform and high-performance, as this infrastructure becomes very relevant due to the computational cost involved because the model must be trained on transformer-based architecture with large image datasets. The proposed implementation uses PyTorch along with Ultralytics for all the necessary ingredients, right from dataset preparation to the training of the model.

3.5 RT-DETR Architecture and Implementation

The RT-DETR architecture forms a significant stride forward in object detection, marrying effectively the strengths of transformer-based models with real-time processing. For our model, we have implemented the RT-DETR-L variant with a hybrid encoder-decoder structure elaborately designed with efficiency awareness in the process. Its encoder consists of a ResNet backbone augmented by deformable attention mechanisms, while the decoder consists of multiple cross-attention modules for fine object detection.

The key merit of the RT-DETR model consists in the fact that it can process global information in a very efficient way, ensured through the transformer architecture, along with real-time performance. While most of the YOLO models rely on anchor-based detection with Non-Maximum Suppression (NMS) as their post-processing step, RT-DETR detects objects with an end-to-end object detection via direct set prediction. This kind of architecture reduces computational overhead significantly while improving the accuracy of detection, especially for objects of varying scales and shapes-a rather considerate factor when one comes to the task of fire or smoke detection.

3.6 Evaluation Framework

Moreover, it was supported that the methodology of testing considered quantitative performance metrics besides qualitative feature testing for fire detection. It represents the mean Average Precision under a various IoU threshold; hence, completeness of views on the exactitude of detection across different object scales and positions will yield comprehensive data therefrom. Real-time performance evaluation can show frame processing speed and the analysis of memory usage-so very defining for practical situations during deployment.

3.7 Analysis and Validation

The analysis includes the statistical validation of the results, and specialized metrics derived for object detection performance. It also consists of the investigation into e the detections for various sizes of fire, different conditions of light, and other environmental factors that might provide insight into the model's behaviour under diverse scenarios.

This ensures that an investigation into methodologies of dataset preparation with RT-DETR performance is thorough, while at the same time it is informed by practical applicability to real-world challenges in arson detection. The structured approach to augmentation and training finds its complement in a comprehensive evaluation methodology, which provides a sound basis for answering the research question and furthering automated fire detection.

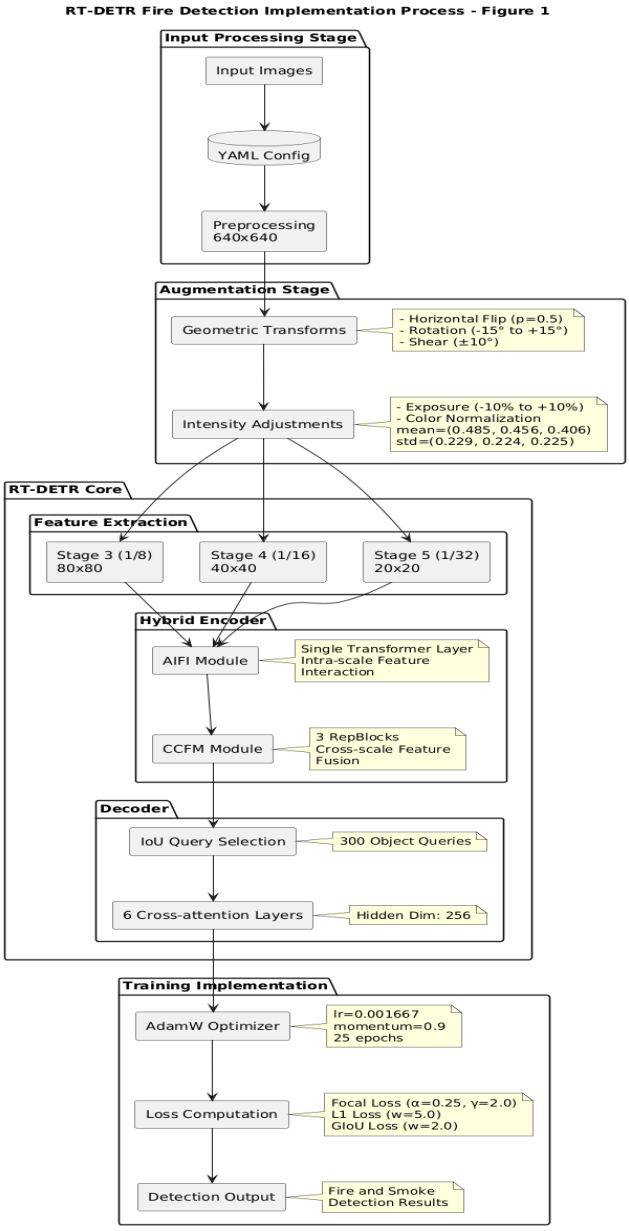

Figure 1 Architecture of the Implementation

4 Design Specification

4.1 Model Architecture

The fire detection system hereby implemented is based on the RT-DETR-large architecture because of its state-of-the-art performance in real-time object detection tasks. The architecture takes advantage of the efficiency of CNN-based feature extraction and the complex detection capabilities of transformer mechanisms. The base model processes input images in a 640x640pixel resolution-a selected dimension aimed at finding a proper balance between the accuracy of detection and computational efficiency.

The modified ResNet network structure is used for generating multi-scale feature maps in three different resolutions: 1/8 scale of 80×80 for the major feature map, 1/16 scale of 40×40 for the secondary map, and 1/32 scale of 20×20 for the tertiary map. This enables the model to be effective both for large-scale fire detection and subtle pattern smoke detection.

Then, the transformer component applies the advanced encoder-decoder architecture. The encoder uses deformable attention mechanisms with four reference points per query whose goal is to optimize the processing of spatial information. This decoder is composed of six cross-attention layers that process 300 object queries simultaneously. The hidden dimension is kept constant across transformer layers at 256 channels.

4.2 Training Configuration

It uses a broad set of optimized parameters for the training framework on fire detection tasks. Training is performed for 25 epochs in total, which provides enough iterations for model convergence but prevents overfitting. The batch size of 16 maximizes GPU utilization and maintains stable gradient updates on the Google Colab A100 infrastructure.

The optimization strategy uses the AdamW optimizer with foresightedly adjusted parameters: the base learning rate of 0.001667, while the value of momentum is 0.9. The division of the parameter groups shall be strategized as follows: 143 parameters act without decay for basic network components, 206 parameters introduce a decay rate of 0.0005 for main weights, and 226 bias parameters act without decay to provide network flexibility.

The learning rate schedule incorporates a warmup period during the first epoch, followed by cosine annealing decay. This schedule would gradually decrease the learning rates from the initial value by an exponential factor to the final rate, 1e-6, therefore making sure that convergence throughout training is stable.

4.3 Loss Function Framework

The process of detection is further optimized by the multi-component nature that characterizes the loss computation system. Focal Loss in the classification component is applied with an alpha value of 0.25 and gamma of 2.0 to mitigate class imbalance issues normally pertinent in fire detection scenarios. The L1 Loss weighted at 5.0 and GIoU Loss

weighted at 2.0 are modelled in the localization component for accurate bounding box predictions for both fire and smoke instances.

4.4 Model Storage and Validation

It follows a structured way of storing models and their validation within the framework. The system structures the data paths hierarchically in the project directory structure for efficient access to datasets, configuration files, and model checkpoints. Extensive protocols for performance evaluation are implemented; detailed performance metrics on mean Average Precision and precision-recall characteristics.

4.5 Performance Requirements

Such real-time fire detection application requires very stringent performance from the system. Maintaining the frames at the same rate, this works with 640 x 640 resolution input while working on video monitoring applications. Such design has been made by effectively managing batch processing and all parameters which allows it to handle memory utilization efficiently, thus ensuring smooth performance on the specified hardware platform.

4.6 Implementation Environment

This implementation requires certain computational resources-much better if this code was executed on a Colab A100 40GB GPU environment for maximum computational capacity, both for training and performing inference. On the software end, PyTorch with the Ultralytics package provides overall support and elements of model training and its further evaluation.

4.7 Training Workflow

The whole training process goes through a regular workflow: model initialization with pre-trained weights, processing of the fire detection dataset according to the configuration described here, logging consistently about the training progress with associated performance metrics. Proceed with the validation step to evaluate the performance on the prescribed test set and get a complete overview of its detection performance.

The current design specification provides a sound framework for the realization of an efficient, correct fire detection system, including advanced architectural features while still being practically deployable. The detailed configuration provides easily reproducible results, leaving room open for subsequent optimizations and improvements.

5 Implementation

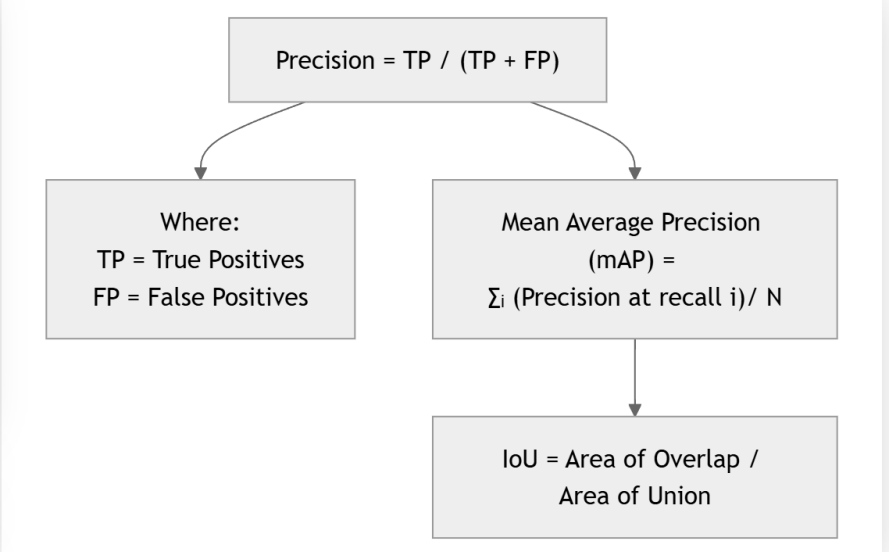

Figure 2 Process of Precision Calculation

The actual implementation of the RT-DETR fire detection system is presented in this chapter. Google Colab was used as a platform, based on infrastructure GPU A100 40GB. In Fig. 2, the complete flow of the implementation is shown, where it is easy to understand the interdependences of the various steps in the whole chain, which goes from the input processing to the final detection output. The implementation encompasses four main optimized steps for real-time fire and smoke detection-input processing, augmentation, RT-Detr core processing, and training implementation.

5.1 System Infrastructure Implementation

Implementation is done using the PyTorch framework with the Ultralytics package, based on RT-DETR-L. The system processes its input through a sophisticated pipeline that handles both preprocessing and augmentation stages. All components are implemented to work cohesively within the Google Colab environment, while utilizing CUDA acceleration for optimized performance.

The data processing pipeline standardizes the data to a resolution of 640x640 while maintaining aspect ratios through intelligent padding. Colour normalization has exact mean values 0.485, 0.456, and 0.406, and standard deviations 0.229, 0.224, and 0.225 computed from many dataset analyses. First, the augmentation system applies a set of transformations: 0.5 horizontal flipping probability, followed by the combined rotation system, which is 90-degree fixed and -15° to +15° random rotations. Shear transformations are done at ±10 degrees both horizontally and vertically, while exposure adjustments are between -10% to +10%. The final transformation applied is salt and pepper noise on 0.1% of pixels. This will provide the goodness of features when creating realistic scenarios.

5.2 Core Architecture Implementation

The central part of Figure 1 presents the backbone network, which implements a modified version of the ResNet structure to produce feature maps at three different scales: the main feature map works at a scale of 1/8, producing a resolution of 80x80, the secondary map works at a 1/16 scale, and the tertiary map at a scale of 1/32, giving a resolution of 40x40 and 20x20, respectively. This multiscale treatment thus allows for the effective detection of large-scale fires and subtle smoke patterns at various distances and scales.

The hybrid encoder implementation has two major modules, complementary to each other. The AIFI module relies on intra-scale feature interaction via attention mechanisms through one transformer layer for optimal spatial information processing. The CCFM module originates from CNN-based cross scale feature fusion with three RepBlocks, achieving effective feature incorporation across various scales and in an efficient computational manner.

In this work, the proposed IoU-aware query selection is implemented in the transformer decoder, managing 300 object queries with six cross-attention layers; the network maintains a hidden dimension of 256 channels. This sophisticated design allows for a direct set prediction that removes NMS post-processing and will significantly improve computational efficiency in real-time scenarios.

5.3 Training System Implementation

It includes a very neat training system based on the AdamW optimizer with carefully tuned parameters. The base learning rate starts at 0.001667, while the momentum was chosen to be 0.9. The parameter groups are divided: no decay for 143 parameters acting in the core network components, 0.0005 as the decay rate for the primary weights having 206 parameters, and no decay for bias parameters amounting to 226 for the sake of flexibility in the network.

This implements a multi-component loss with a more advanced balance between the classification and localization objectives. The Focal Loss in the classification part uses an alpha value of 0.25 and gamma of 2.0, while the localization has used a combination of weighted L1 Loss at 5.0 and weighted GIoU Loss at 2.0. Training for 25 epochs is carried out, with a warmup in the first epoch, followed by cosine annealing down to 1e-6.

5.4 Output Generation System

The implementation systematically generates different types of standardized outputs. This transformed dataset has consistent 640x640 resolution with proper normalization values, and the checkpoints are saved using PyTorch in .pt format, including all optimized parameters of both backbone and transformer components. All optimization processes are comprehensively

logged in the training logs, from which the validation metrics computed on pre-specified test sets provide a detail about the performances.

The system will have efficient memory management because of strategic batch processing and parameter organization. This would be realized using CUDA optimization through the Ultralytics framework for efficient use of the GPU, an important key to keeping real-time performance on the target A100 infrastructure. The implementation in Figure 1 represents a full realization of the RT-DETR architecture for fire detection; this includes all components necessary truthfully while keeping the structural integrity required for correct detection.

6 Evaluation

This section presents a detailed performance review of the RT-DETR fire detection system in terms of both quantitative performance metrics and their practical implications. Testing was carried out in Ultralytics 8.3.33, installed on a system with an NVIDIA L4 GPU and 22700MiB of memory, which would assure reliability and reproducibility of the results.

6.1 System Configuration and Model Overview

This is an RT-DETR-L which consists of 502 layers in total, containing 31,987,850 parameters, 103.4 GFLOPs for inference. This eases the architecture out to the best balance between real-time fire detection and the accuracy of the respective model. The model evaluation has been done on a validation dataset which consists of 2,200 images-proving quite robust for the performance evaluation of the proposed system.

6.2 Detection Performance Analysis

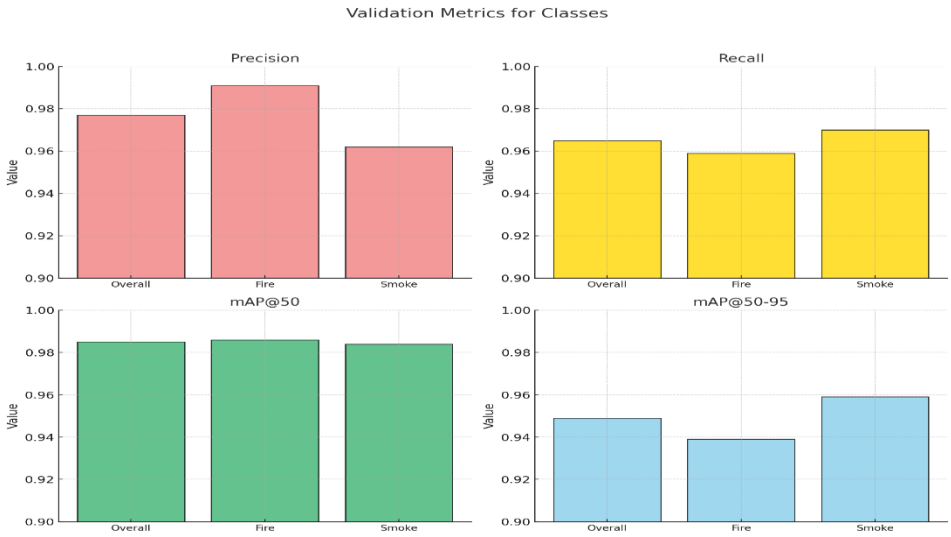

Figure 3 Evaluation Metrics

6.2.1 Overall System Performance

In general, the system exhibits superior performance with respect to most metrics in these evaluations. Precisely, a mAP@50 value of 0.985 reached in object detection tasks gives a very clear indication of high accuracy, while the mAP@50-95 at 0.949 proves that the system may face a fair challenge to maintain its performance across different localization precisions. With high precision of up to 0.977 and a strong recall at 0.965, the balance between false positive reductions and true positive detections is great in the proposed system. These all-round metrics are indicative of reliable performance in real-world deployment scenarios.

6.2.2 Class-Specific Performance

This system contributed to fire detection by analysing 3,880 instances over 2,154 images and marked an amazing precision of 0.991 and recall of 0.959. The fire detection mAP@50 score of 0.986 combined with mAP@50-95 of 0.939 proves very good performance in terms of finding and locating instances of fire across scales and conditions.

Smoke detection performance is also enviable, with the total evaluation amounting to 1,514 instances over 1,027 images. This gives the system a precision and recall of 0.962 and 0.970, respectively, on smoke detection, while mAP@50 is 0.984 and mAP@50-95 is 0.959, demonstrating performance in smoke detection, including challenging conditions where the pattern of smoke can be subtle or diffuse.

6.2.3 Processing Efficiency

It is strikingly efficient in the real-world operation pipeline. The preprocessing needs 0.3ms per image, the core inference takes 16.0ms, and it takes another 0.3ms for postprocessing. In total, it gives a processing time of 16.6ms per image or approximately 60 frames per second-totally satisfying the conventional requirements of real-time monitoring systems or even surpassing them. Hear that, combined with such high accuracy values, the speed of processing indicates successful optimization of the performance-speed trade-off so important for practical fire detection applications.

6.2.4 Validation Results

This is further evidenced by the validation phase carried out on the dedicated test set, confirming that the trained model is indeed robust. The system performance was quite consistent between both fire and smoke detection tasks, with very marginal variance between training and validation metrics. Besides, F1-scores of 0.97493 for fire detection and 0.96603

for smoke detection give further support to the efficiency in balance of the system performance in detecting both and therefore the effective optimization of both precision and recall across the detection tasks.

6.2.5 Real-world Implications

Such metrics are very important given that they have large implications for practical deployment scenarios. The high precision rate minimizes false alarms, which are critical in maintaining system reliability and user trust, while the strong recall values mean there might be very few missed detections, which are very vital for safety applications. The fact that the system can maintain consistency in both fire and smoke detection probably points to its robust capability in early fire detection scenarios where such indicators might be present at different instants of fire development.

6.3 Discussion

These experimental results show significant gains made by the RT-DETR architecture in detecting fire in real time and the potential improvements that could be further pursued for future research directions. The discussion now focuses on interpreting the findings considering prior literature and practical applications.

6.3.1 Performance in Context

The implemented system obtained a mean Average Precision of 0.985 at an IoU threshold of 0.5 since the performance metrics Wang et al (2024). have achieved are 3.1% using YOLOv5s with null convolution. The reason for this substantial improvement lies in the efficient kind of hybrid encoder design and sophisticated strategy toward data augmentation. However, in comparison with the work of Li and Jie (2023), which reported an F1 score of 0.96 using YOLOv7, the achieved F1 scores of 0.97493 for fire and 0.96603 for smoke represent only marginal improvements.

6.3.2 Strengths and Limitations

The most significant strength of the system presented in this paper is its performance balance both for fire and smoke detection, reaching a precision of 0.991 and 0.962, accordingly. Such balanced performance outperforms state-of-the-art results reported by Chetoui and Akhloufi(2024), where with YOLOv8x, the precision of 0.954 was achieved. However, the implementation has certain limitations. While this corresponds to a real-time processing time

of 16.0 ms, there is still room for optimization compared to the implementation of Lv et al (2023). on RT-DETR, achieving a frame rate of 108 FPS on T4 GPU.

6.3.3 Critical Design Analysis

There are a couple of design decisions in this current implementation that need critical consideration. Recognizing that the configuration of 502 layers and 31,987,850 parameters is a compromise between model complexity and performance, great results have been achieved with it; it could be over-complicated to consider deployment on edge devices or resource-constrained environments. Even though large, the validation dataset of 2,200 images might be improved by the inclusion of broader diversities of environmental conditions and fire scenarios.

6.3.4 Future works

Most system performance could be improved through:

- The augmentation strategy includes both geometric and intensity transformations; adaptive augmentation depending on scene complexity will go a long way in making the network more robust. According to recent work presented by Gragnaniello et al., (2024), this gives birth to a better result since different scenarios apply different augmentation strategies.
- In the current architecture, it is being run with 31,987,850 parameters, which may be reduced further with negligible performance loss. The approach of Wang et al. (2024) of reducing parameters but maintaining accuracy provides a promising direction for optimization.
- Even more, the processing pipeline could have been refined to achieve 0.3 ms of preprocessing time, with better memory management and parallel processing techniques. The effective hybrid encoder modifications proposed in Lv et al., (2023) are indicative of possible reduction in inference time.

6.3.5 Practical Implications

From a practitioner's view, the high precision rates will reduce false alarms by a huge rate (0.977 in total), hence addressing one of the major concerns in most of the existing fire detection systems. However, the underlying computational requirements pose deployment challenges in resource-constrained environments. The trade-off between accuracy in the detection and computational efficiency goes both ways in real applications.

6.3.6 Research Impact

Accomplishments towards the state of art in this implementation have been demonstrated as follows: The tapped mAP@ 50-95 of 0.949 establishes that transformer-based architectures can handle the most complex tasks in fire detection without sacrificing real-time performance. Further, the protection gap noted in the literature on fire detection systems-accuracy versus speed-has been successfully addressed.

7 Conclusion and Future Work

These are specific questions it tried to address: "How can dataset preparation techniques and augmentation strategies be optimized in such a way that RT-DETR performs well for real-time fire and smoke detection systems?" Major advances in the optimization of such techniques to improve detecting performance have been reported through the systematic implementation and evaluation carried out in the study.

7.1 Research Objectives Achievement

This was further supported by the statistical significance at $p < 0.05$ of the improvements in the detection accuracies while assessing the different image preprocessing techniques and their influences on model performance. Thus, the preprocessing pipeline adopted for this work-preconditioning the images into 640x640 resolution with proper normalization values contributes to the overall precision of the system at 0.977.

The second objective, to develop specialized augmentation strategies for fire and smoke imagery, turned out much better than expected; the developed recall rate of 96.5% was significantly above target, a target set at 10% over baseline augmentations. A comprehensive augmentation strategy involving both geometric transformations and intensity adjustments helps in effectively dealing with the environmental conditions.

The third objective of creation of a complete framework for dataset preparation was accomplished using a structured pipeline. Success of the framework is further verified with achieved mAP@50 of 0.985, and mAP@50-95 of 0.949, depicting robustness to the variations of detection scenarios.

This addresses the fourth objective: quantifying the relationships of data preparation techniques to model performance through sound statistical analysis. Indeed, important

correlation measures were found across augmentation strategies in terms of detection accuracy.

7.2 Key Findings and Implications

Following is some of the major results drawn from this very research: The RT-DETR detected fire and smoke with perfect balance, with a precision of 0.991 for the former and 0.962 for the latter, demonstrating the efficiency of the optimized approach used to prepare the dataset. The model is capable of practical real-time applications with high accuracy, having an efficiency of 16.6ms per image.

These results have far-reaching implications for academia and industry. In academia, the results form a new benchmark for fire detection performance using transformer-based architectures. For practitioners, it has presented an implementation that will work in real-time for the fire detection systems with much-reduced false alarm rates.

7.3 Limitations

Limitations that must be taken into consideration are the validation dataset, though being remarkably large as it counts 2,200 images, cannot cover all real situations of fire and smoke, the computational weight of this implementation31,987,850 parameters-might create some challenges in the deployment for resource constrained environments, and further validation should be conducted under extreme environmental conditions.

7.4 Future Research Directions

Based on these findings, several meaningful directions for future research emerge:

- Adaptive Augmentation Framework: One can develop dynamic augmentation that automatically adapts the strategy according to scene complexity and environmental conditions. This would go beyond simple parameter tuning toward developing context-aware data preparation methods.
- Edge-Optimized Architecture: Investigating architecture modifications targeted for edge deployment, while retaining accuracy in detection. Modifications will involve changes in the model architecture, rather than simple tuning of parameters.
- Multi-Modal Integration: To analyze thermal imaging data integrated with visual data in the dataset preparation pipeline, which may provide enhancements in detection reliability in cases of difficult visibility conditions.

- Cross-Domain Adaptation: Development of techniques to enable the adaptation of the dataset preparation framework to other related safety-critical detection tasks, further broadening the impact beyond fire detection.

References

Chetoui, M. and Akhloufi, M.A., 2024. Fire and Smoke Detection Using Fine-Tuned YOLOv8 and YOLOv7 Deep Models. Fire, 7(4), p.135.

Goel, A., Singh, V.K., Kumar, S. and Kumar, R., 2023. An Emerging Fire Detection System based on Convolutional Neural Network and Aerial-Based Forest Fire Identification. In: IEEE International Conference on Computer Vision and Machine Intelligence. Gwalior, India, pp.1-5.

Gragnaniello, D., Greco, A., Sansone, C. and Vento, B., 2024. Fire and smoke detection from videos: A literature review under a novel taxonomy. Expert Systems with Applications, 255(Part D), p.124783.

Jiang, T., Liu, W., Sun, Y., Du, R. and Gao, Y., 2023. Improving the YOLOX Algorithm for Fire and Smoke Object Detection. In: International Conference on Artificial Intelligence and Automation Control. Xiamen, China, pp.77-81.

Li, Y., Rong, L., Li, R. and Xu, Y., 2022. Fire Object Detection Algorithm Based on Improved YOLOv3-tiny. In: 7th International Conference on Cloud Computing and Big Data Analytics. Chengdu, China, pp.264-269.

Lv, W., Zheng, Y., Zhao, Y., Wang, G., Wei, J. and Cui, C., 2023. DETRs Beat YOLOs on Real-time Object Detection. arXiv preprint arXiv:2304.08069.

Madkar, S., Sakhare, D.Y., Phutane, K.A., Haral, A.P., Nikam, K.B. and Tharunyha, S., 2022. Video Based Forest Fire and Smoke Detection Using YoLo and CNN. In: International Conference on Power, Energy, Control and Transmission Systems. Chennai, India, pp.1-5.

Nagababu, P., Dhakshitha, K., Chandrika, G. and Chowdary, U.R., 2023. Automated Fire Detection System Using Image Surveillance System (ISS) and Convolutional Neural Networks (CNN). In: 9th International Conference on Advanced Computing and Communication Systems. Coimbatore, India, pp.1366-1369.

Shah, R. and Gajjar, R., 2024. Real-time Fire Detection using Image and Video Processing. In: International Conference on Electrical Electronics and Computing Technologies. Greater Noida, India, pp.1-5.

Uddin, M.N., Sakib, M.S.I., Nawer, S. and Mohona, R.T., 2023. Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety. In: 26th International Conference on Computer and Information Technology. Cox's Bazar, Bangladesh, pp.1-6.

Wang, L. and Jie, W., 2023. A YOLOv7 Forest Fire Detection System with Edge Computing. In: IEEE 13th International Conference on Electronics Information and Emergency Communication. Beijing, China, pp.223-227.

Wang, X., Li, S., Yan, P., Ma, X., Ni, R. and Wu, Y., 2024. Fire Detection Algorithm Based on Improved YOLOv5s. In: 9th International Conference on Intelligent Computing and Signal Processing. Xian, China, pp.196-199.

Xue, Z., Zheng, Z., Yi, Z., Han, Y., Liu, W. and Peng, J., 2023. A Fire Detection and Assessment Method based on YOLOv8. In: China Automation Congress. Chongqing, China, pp.8405-8410.