Early Fire Detection System by Using Automatic Synthetic Dataset Generation Model Based on Digital Twins

By Hyeon-Cheol Kim, Hoang-Khanh Lam, Suk-Hawn Lee, and Soo-yol Ok

Department of Computer Engineering, Dong-A University, Busan 49315, Republic of Korea

Appl. Sci. **2024**, *14*(5), 1801; https://doi.org/10.3390/app14051801

Submission received: 8 December 2023 / Revised: 15 January 2024 / Accepted: 22 January 2024 / Published: 22 February 2024

Seonmi Choi

Department of Applied Art and Technology
Chung-Ang University

Abstract

- The early detection of fires is a very important task in preventing large-scale accidents; however, there are currently almost no learnable early fire datasets for machine learning

- Generates synthetic fire data according to various fire situations in each specific space
  - ➡ Performs transfer learning using a state-of-the-art detection model with these datasets
  - ➡ Distributes them to AIoT devices in the real space

[Keywords]

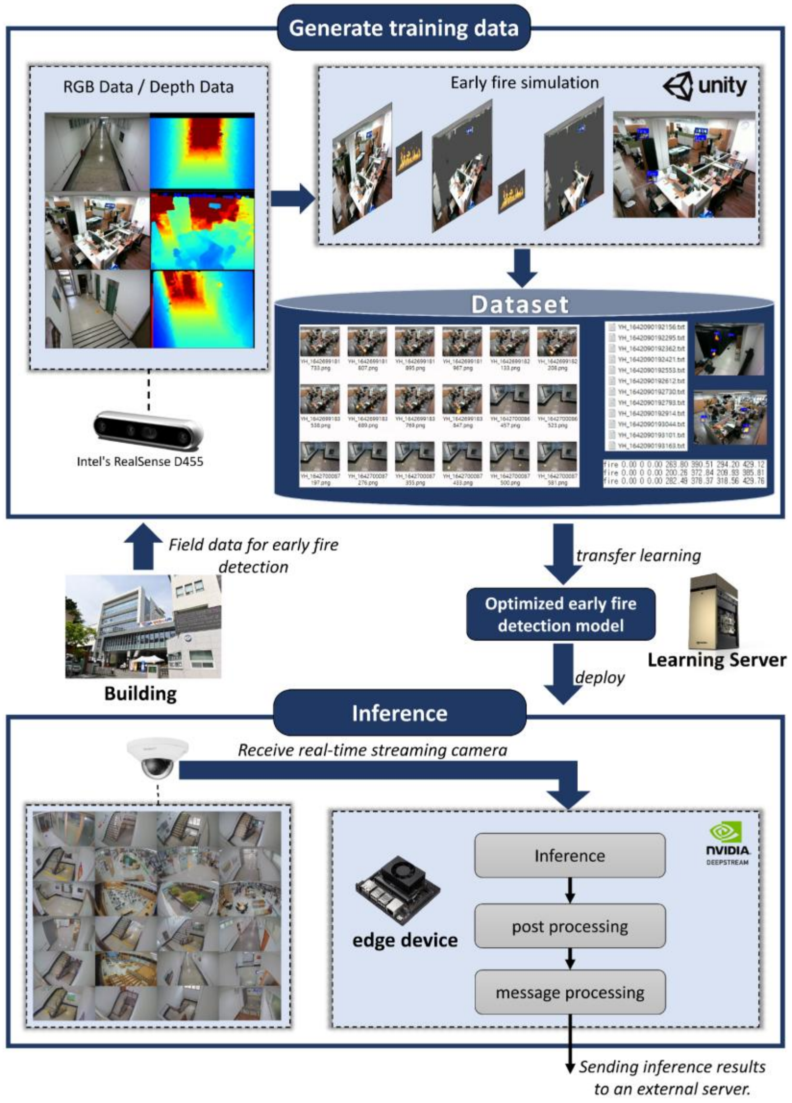
- Digital twin smart city
- Particle system
- Synthetic learning data
- Object detection
- Early fire detection

1. Introduction

- Studies on smart cities: AI & IoT
- Digital Twin: Reflect the real world in a digital virtual space by sending data on the physical space gathered via IoT

⇒ propose an early fire detection system using a digital-twin-based autonomous learning data generation model that can detect fires via image recognition sensors such as CCTVs.

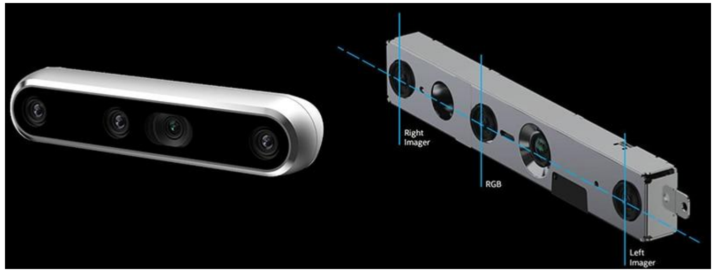
2. Materials and Methods



Real image data collection

- Virtual fire occurrence composite image generation
- Automatic generation of learning datasets
- Fire early model transfer learning
- AI device inference
- Post-processing

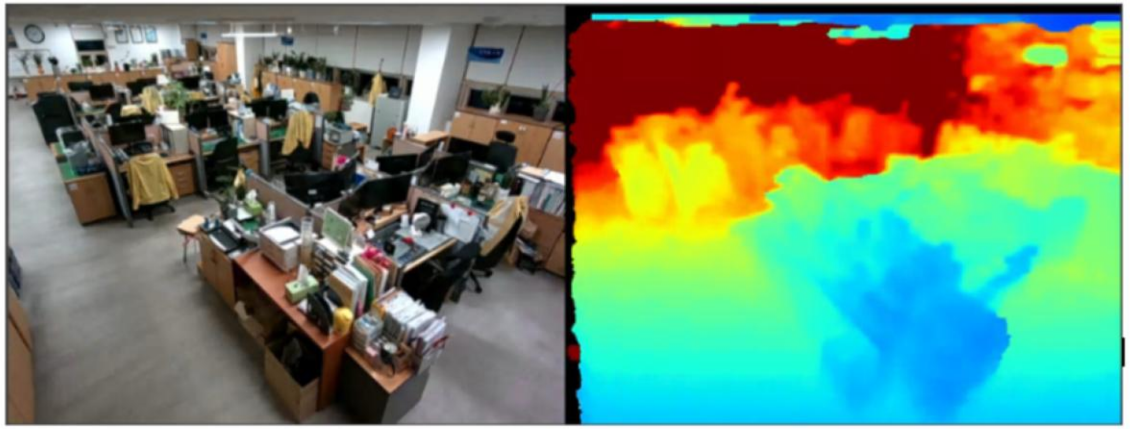
2.1. Real Environment Data



- Intel's RealSense D455

  : Collect the RGB and Depth data required to create virtual data

- RealSense Viewer
- RealSense SDK → desired frame data can be obtained



(a) (b)

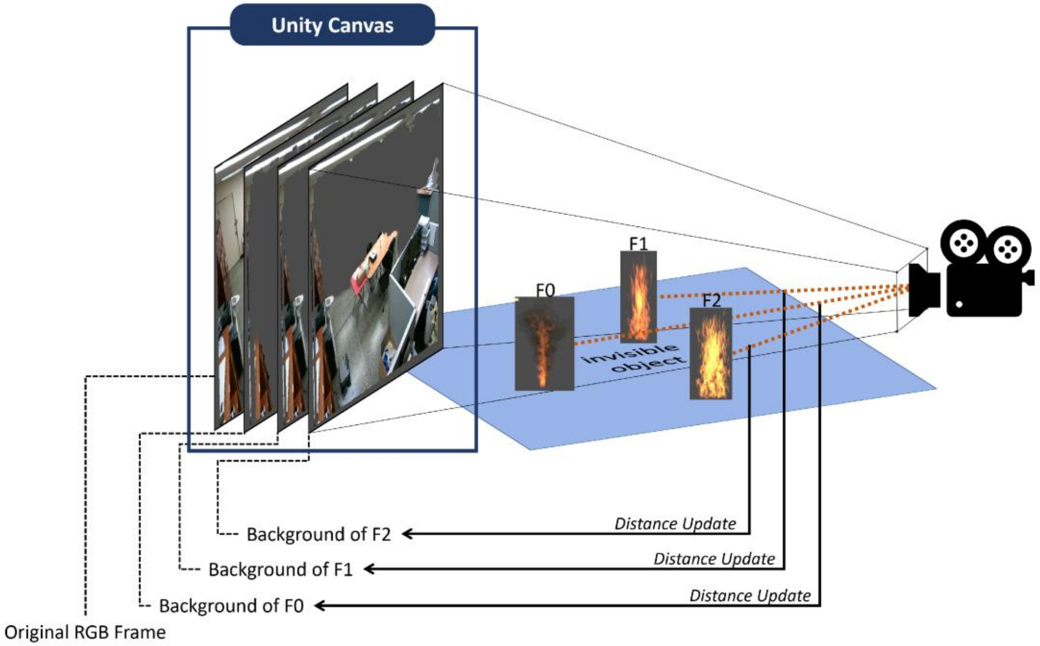
- To secure actual image data at various sites, the recording was conducted considering possible environmental changes at the site, such as when the lights were turned off

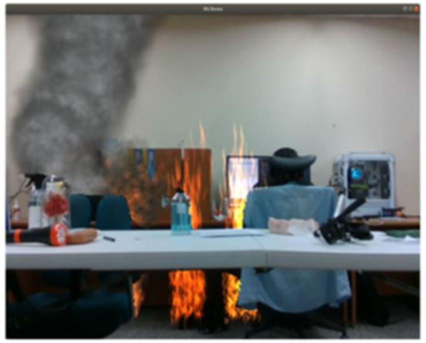
2.2. Virtual Fire Data

- Unity3D's Particle System → Virtual fire

- RealSense SDK

  : Recorded images from the actual environment + Virtual fire
- Background Segmentation Shader

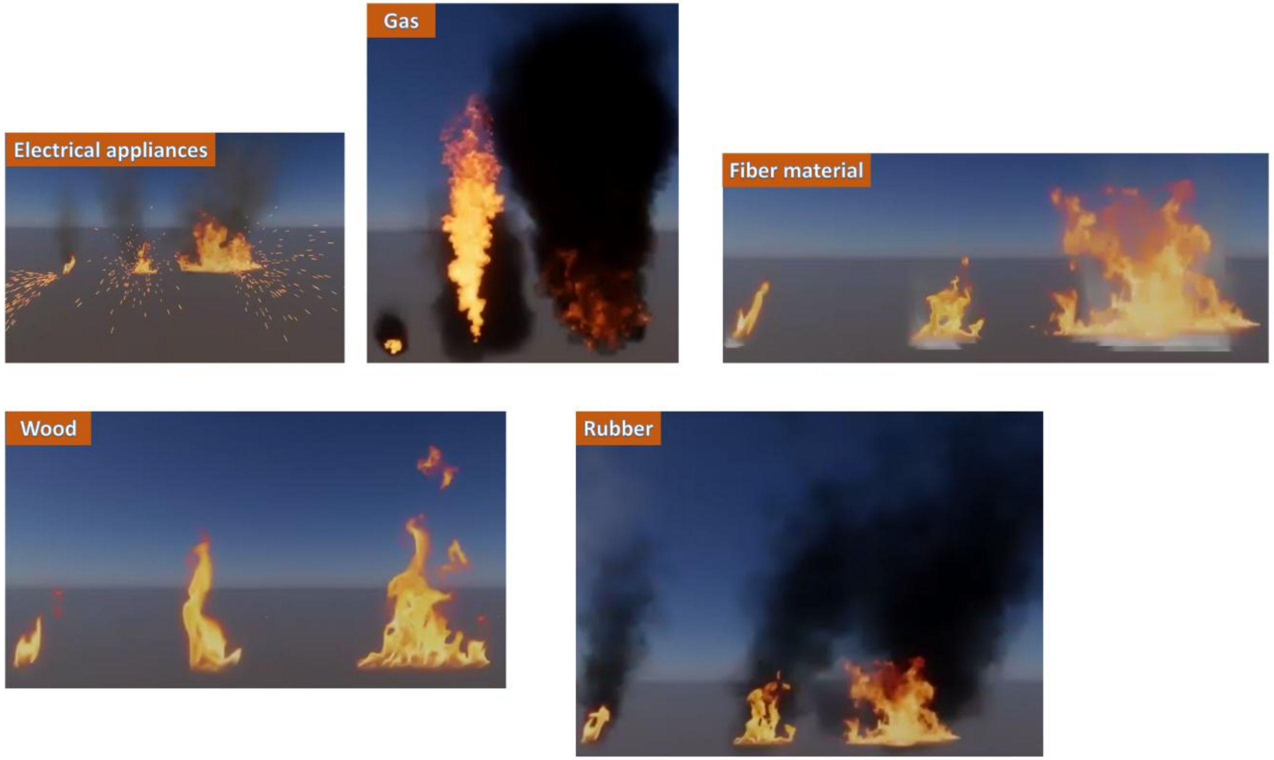




Create unprocessed RGB frames on Canvas in Unity3D

- Begin rendering from the fire that is furthest away from the camera
- Draw each fire, then calculate the distance between it and the virtual camera
- Pixels closest to the fire are covered by rendering the result of transparently executing a Background Segmentation of all pixels having a longer distance value

2.2. Virtual Fire Data



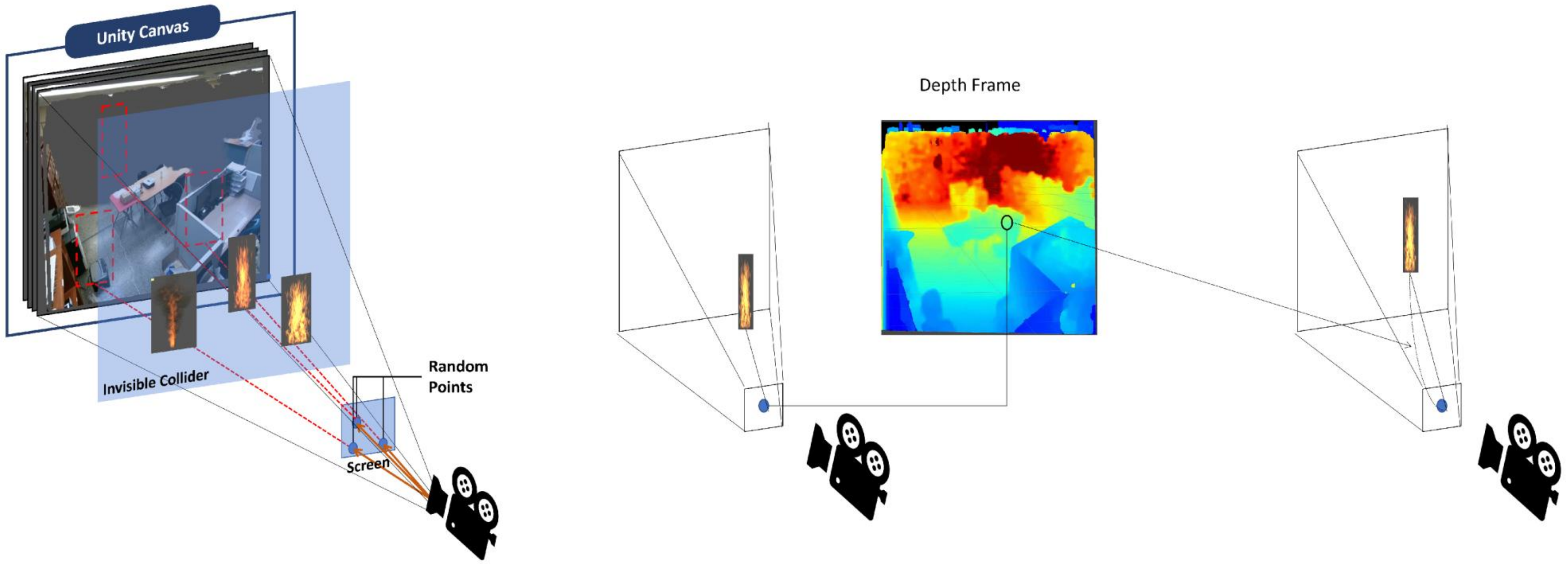
- The scale of the fires: small, medium, large

- Combustible materials(12):

alcohol, animals, electricity, fibers, gasoline, kitchen fires, lamp oil, paint, plastic, rubber, vegetable oil, and wiring

2.3. Automatic Dataset Generation

- A virtual fire must be maintained at an acceptable distance as it approaches the virtual camera viewing angle



(a) (b)

- Created a ray in the virtual space
- Ran Raycast to temporarily place the fire at the point where it hits the virtual collider

- Read from the depth data at the point where each fire was rendered
- distance between the virtual fire and the camera → less than a certain value

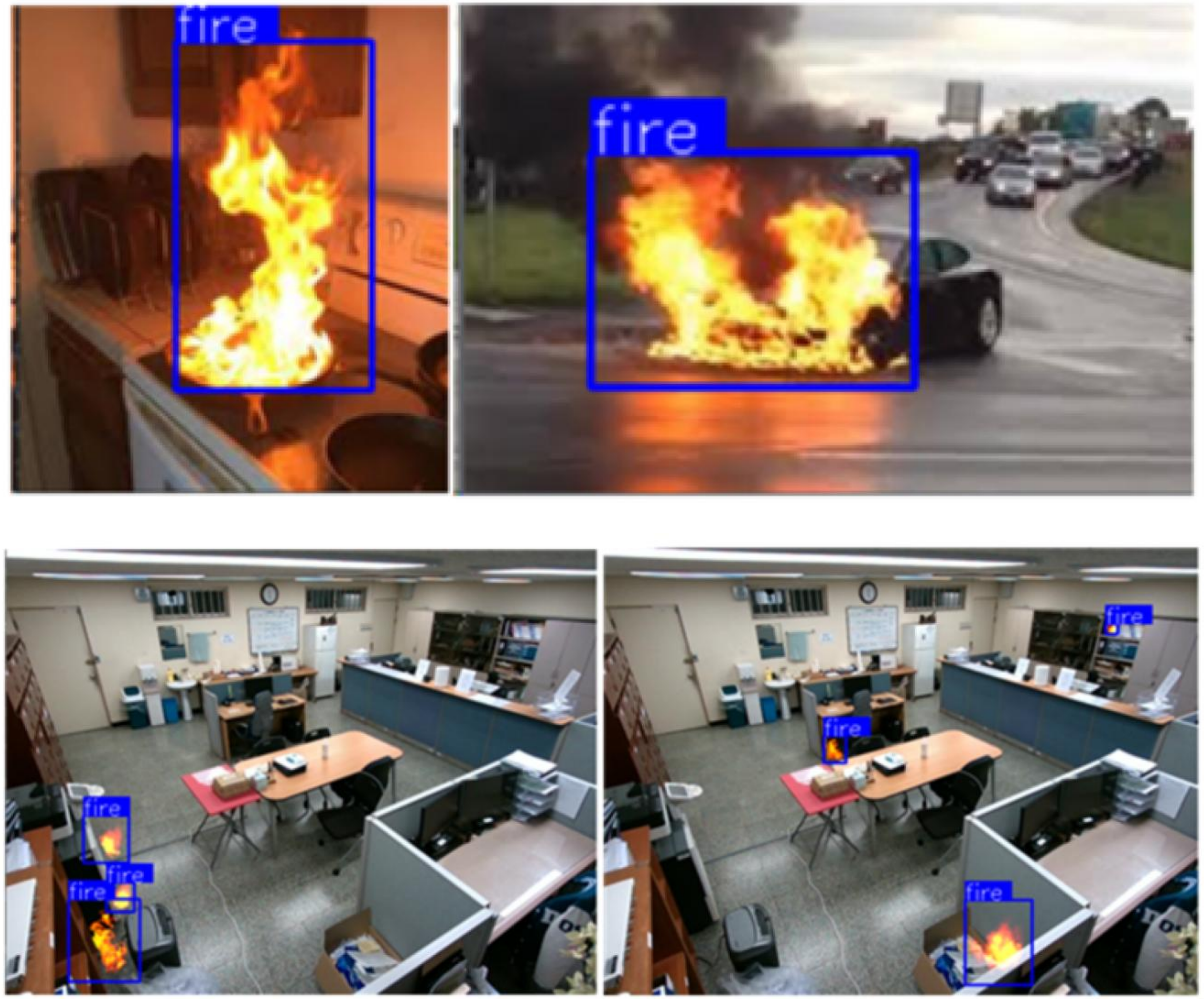
2.3. Automatic Dataset Generation

(a) (b)

- Using BoxCollider for annotation
  $(x_{min}, y_{min}, z_{min}; x_{\max}, y_{\max}, z_{\max})$

- Automatically generated annotation information for fires on images.
  (axis-aligned bounding box, AABB)

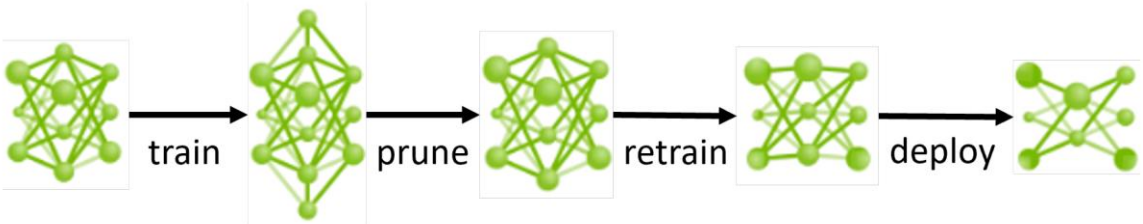
2.3. Automatic Dataset Generation



- Dataset to be utilized in the learning stage:
  automatically generated virtual data
  + actual data(by FireNET)

2.4. Fire Detection Model

- Model learning: NVIDIA TAO Toolkit
  - Transfer Learning Toolkit(TLT)
  - resnet18(Backbone of the YOLOv4 model)



→ Edge device

bounding box display threshold: 0.6

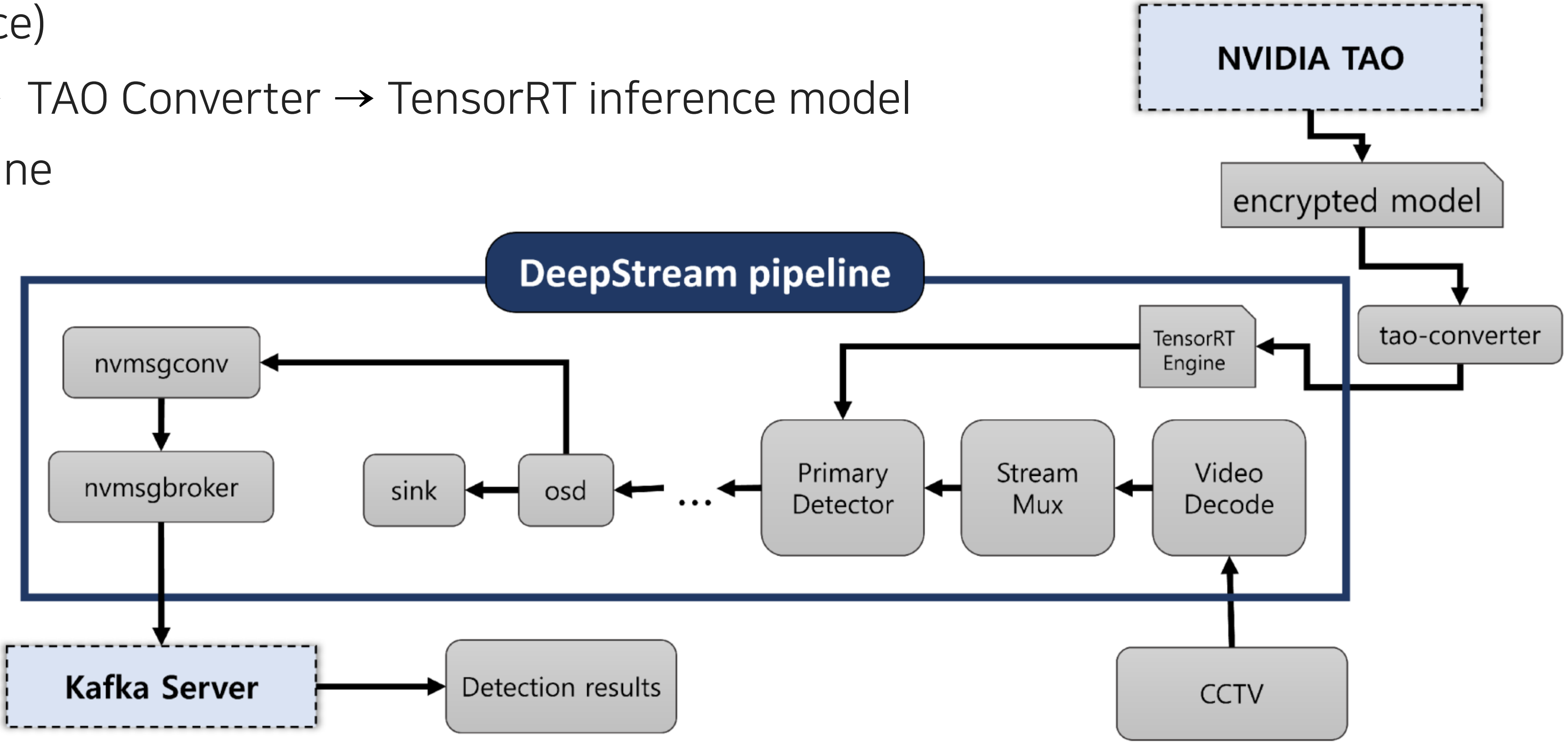
2.4. AI Inference and Post-Processing

- The field-optimal model that learns virtual data is inferred using an AI device at the edge end placed at each site

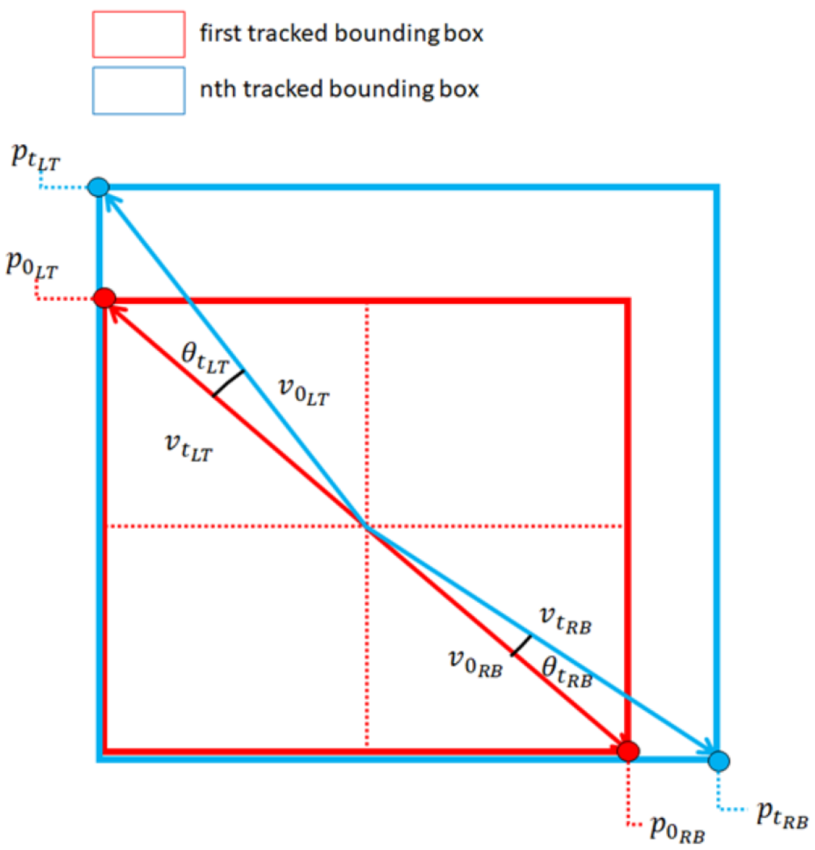
- NVIDIA Jetson Xavier NX(Device)
- Encoded model file(by TAO) → TAO Converter → TensorRT inference model
- DeepStream: Gstreamer pipeline


```
{
  "messageid" : "6360e7bb-8d57-4fc9-beb0-fcf3350809c1",
  "@timestamp" : "2023-08-23T07:36:29.034Z",
  "source_width" : 854,
  "source_height" : 480,
  "fireDetectionModule" : {
    "id" : "DONGA_TEST_A6000_DS6.2_CAM0",
    "detectionInfo" : [
      {
        "classID" : 0,
        "objectID" : 0,
        "left" : 377.0,
        "top" : 277.0,
        "width" : 61.0,
        "height" : 73.0,
        "confidence" : 0.961508810520172l2
      }
    ]
  }
}
```


A message about the detection results sent by deepstream.



- Post-processing for both future research and the purpose of counting the number of fires, fire tracking is crucial

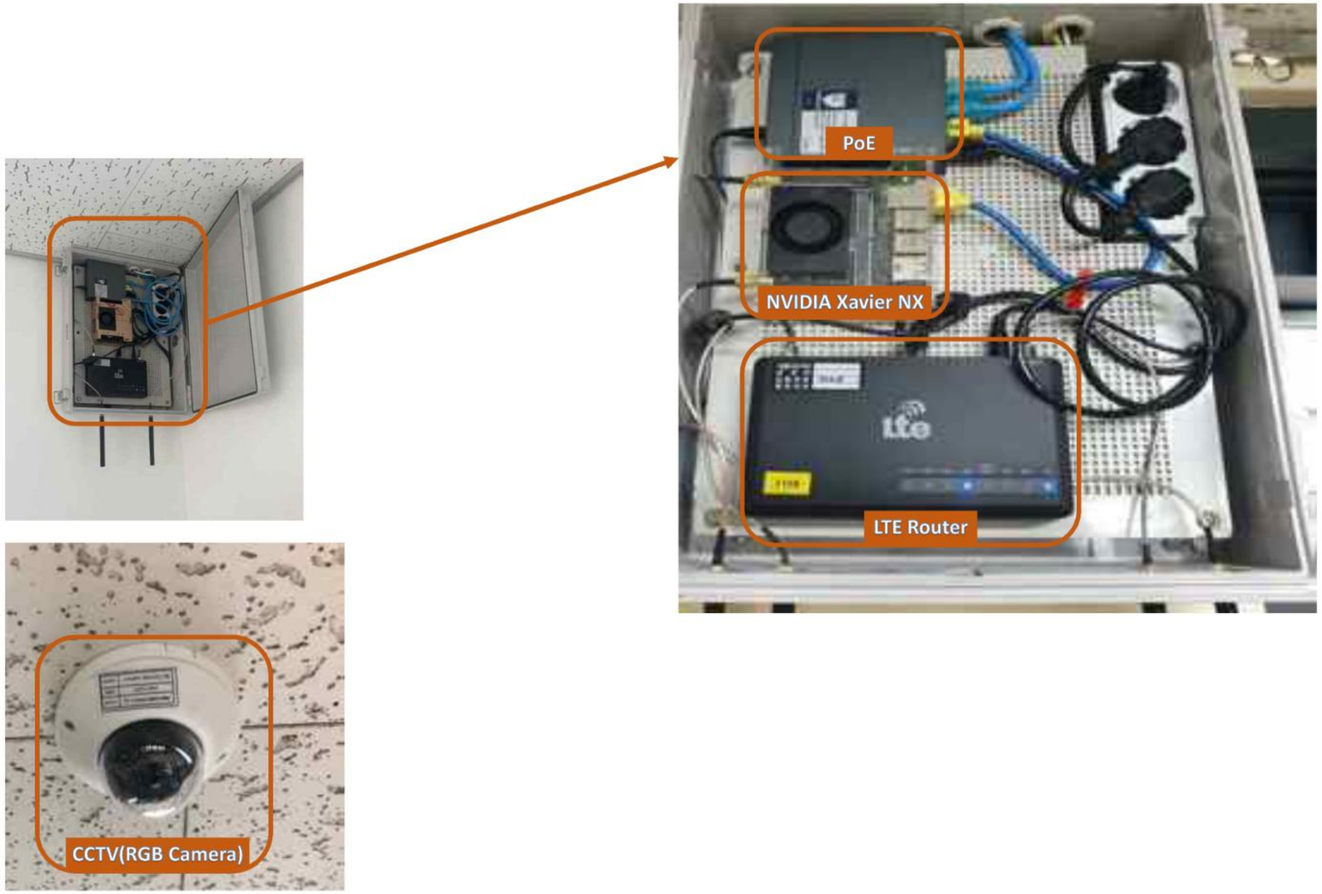


$$d_t = |p_{0_{RB}} - p_{0_{LT}}| - |p_{t_{RB}} - p_{t_{LT}}|; d_{cummulative} = \sqrt{\sum_{t=1}^{n} (d_0 - d_t)^2}$$

$$cos\theta_{t_{LT}} = \frac{\vec{v_{0_{LT}}} \cdot \vec{v_{t_{LT}}}}{|\vec{v_{0_{LT}}}||\vec{v_{t_{LT}}}|}; cos\theta_{t_{RB}} = \frac{\vec{v_{0_{RB}}} \cdot \vec{v_{t_{RB}}}}{|\vec{v_{0_{RB}}}||\vec{v_{t_{RB}}}|}$$

0.9-0.98 was the best threshold to verify whether it was a fire

3.1. IoT Installation



IoT:

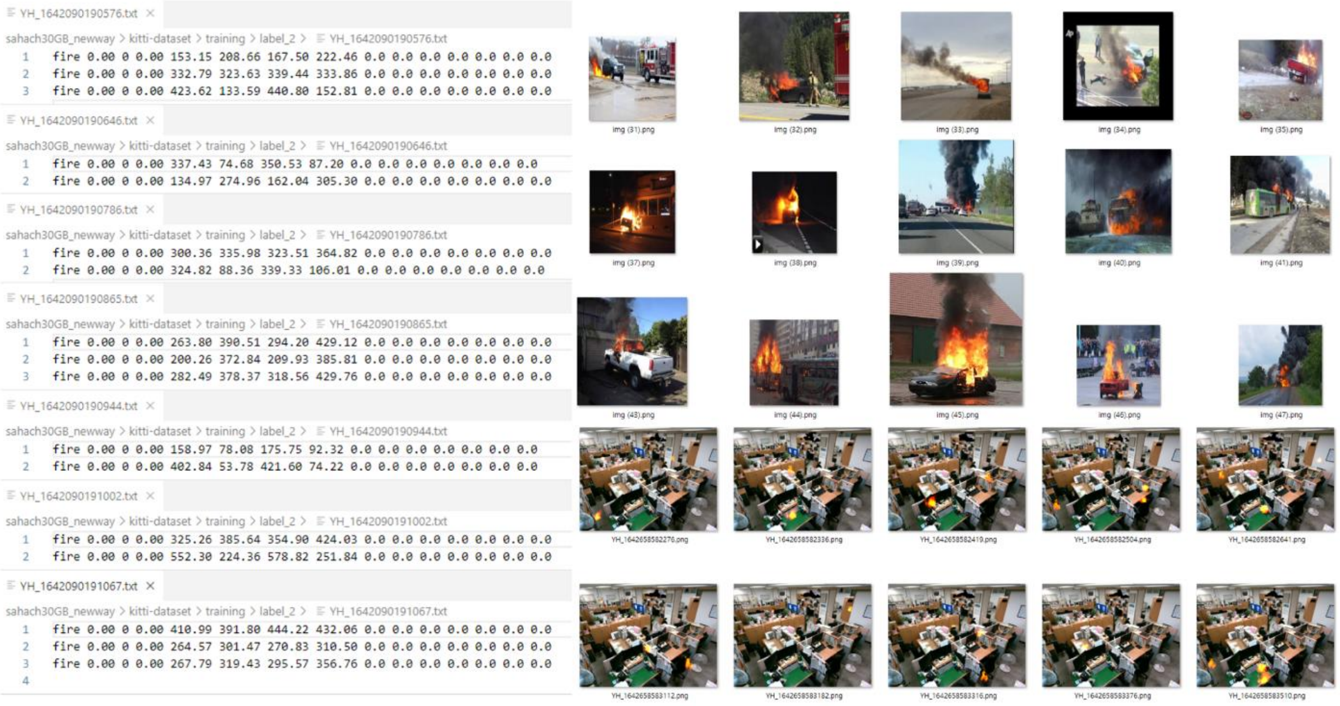
CCTV

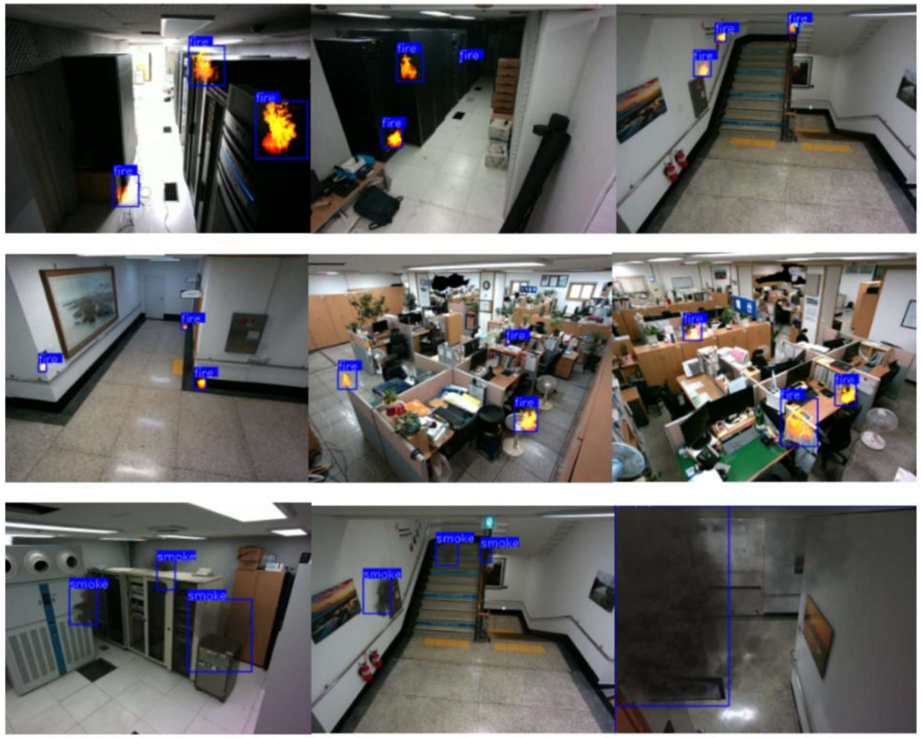
+ NVIDIA Xavier NX

+ LTE Router

+ PoE

3.2. Virtual Fire Data





- Recorded RGB and Depth data for 43 indoor locations
- Generated 7000 virtual flame particle data points and 7000 virtual smoke particle data points for each location
- Total of approximately 600,000 virtual fire data points

Number of training images for a single specific environment.

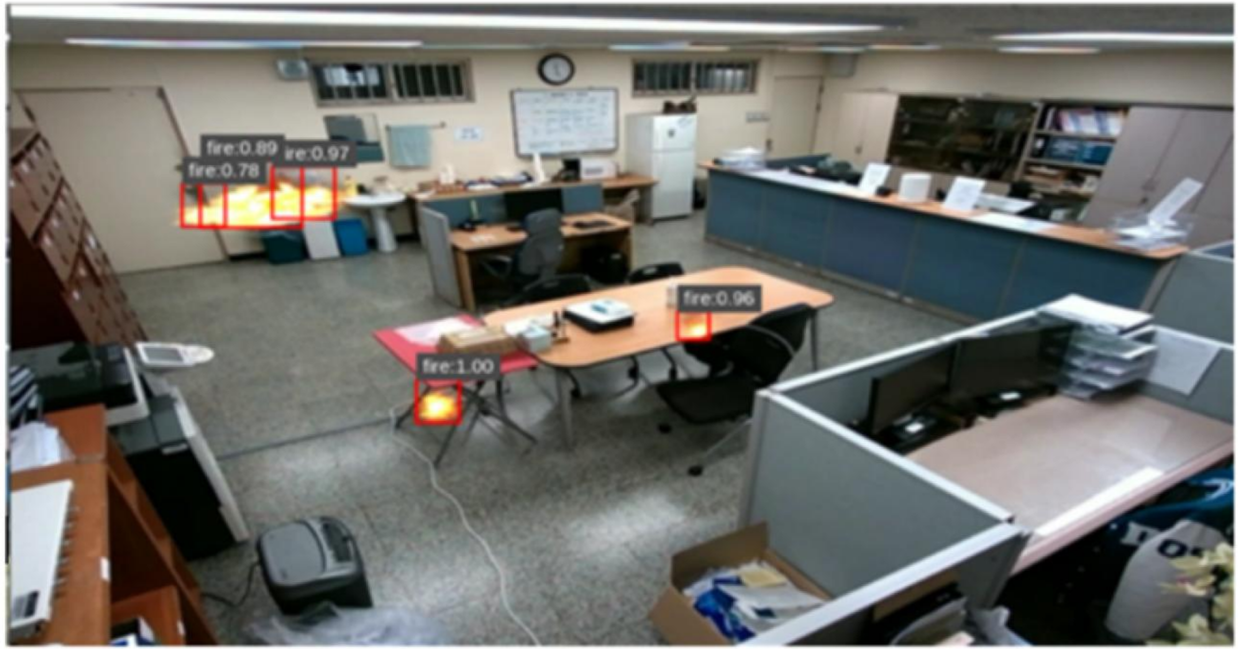
Data	Virtual Data	Real-World Data	Total
Training data	4375	412	4787
Testing data	625	90	715
Total	5000	502	5502

- NVIDIA DGX A100

Model	Unpruned Model Parameters	AP	Pruned Model Parameters	AP	Retrain/Model
DetectNetV2	11,200,458	0.93515	9,561,530	0.96316	0.85367
FasterRCNN	12,751,352	0.9528	10,434,616	0.9506	0.81831
YOLOv4	34,829,183	0.90909	3,659,191	0.9091	0.10506
EfficientDet	3,876,308	0.426	2,130,676	0.426	0.54966
DINO	-	0.83	-	-	-
D-DERT	-	0.71433	-	-	-

- Object Detection model:
  DetectNet_v2, FastRCNN, YOLOv4, EfficientDet, DINO

- Used fire particles to generate virtual data to infer new data not used for learning



Inferring new fire footage from fires used to generate data for training

- confidence: 0.95, no false detection

Detection of fire shapes that have never been used in training

- a few misdetection results

3.3. Results

- Differences in inference performance between the models.

(a)YOLOv4 (b)DetectNetV2

(a) (b)

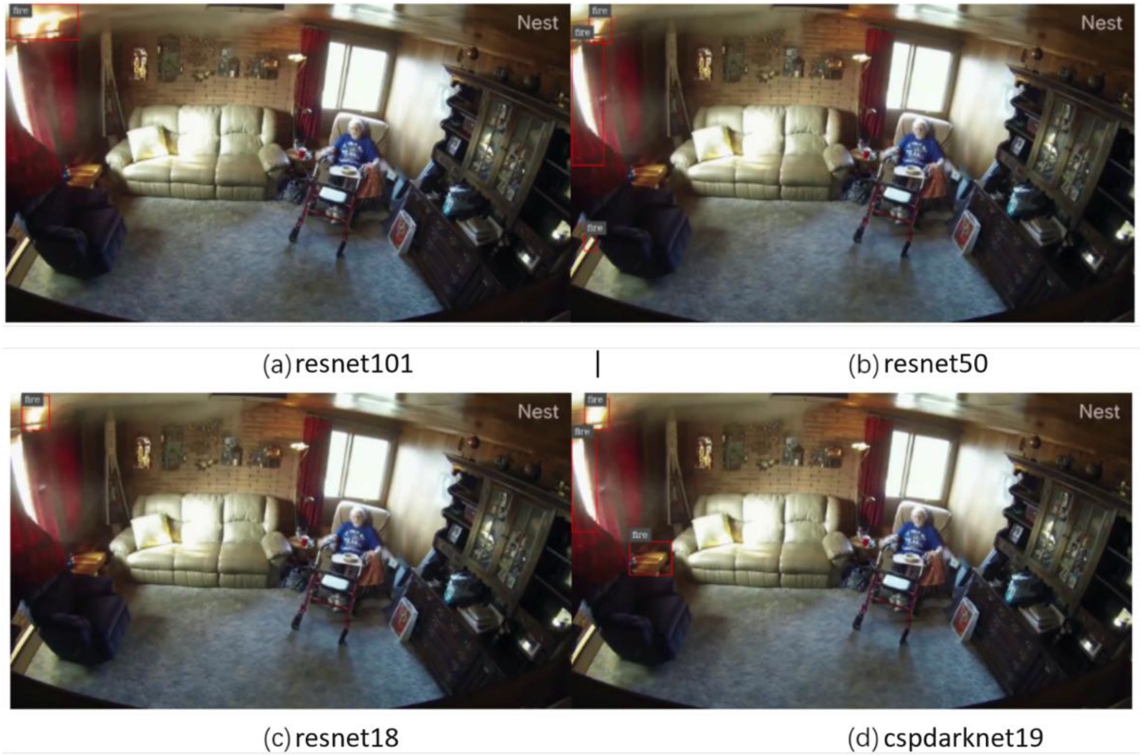
3.3. Results

Backbone	Model Parameters	mAP	Retrain Model Parameters	mAP	Retrain/Model
resnet18	11,200,458	0.90798	3,659,191	0.9088	0.10506
resnet50	85,346,559	0.90798	22,902,807	0.90792	0.26835
resnet101	122,286,335	0.90673	3,000,711	0.90365	0.02453
cspdarknet19	53,444,895	0.9062	38,253,879	0.90847	0.71576

- Evaluation of the detection model according to the backbone of the YOLOv4 model.

- After pruning before relearning, Resnet101, which had the largest number of layers, showed fewer parameters than the other backbones

3.3. Results



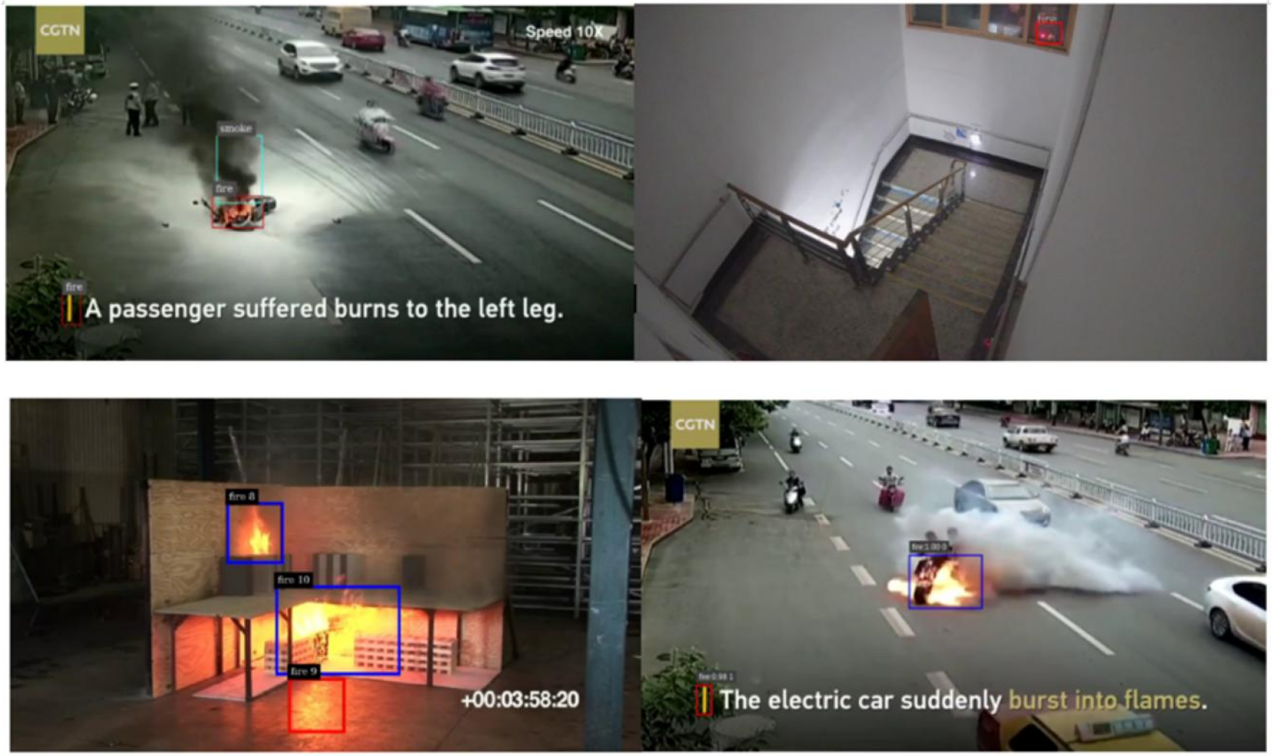
- In the case of resnet18, it was confirmed that the initial state of fire was detected well, and the error detection was very low.

- Inference for very small fires with the YOLOv4+resnet18 model.

3.4. Post-Processing

- False detection by simple colors sometimes occurred
→ post-processing process was added to inspect the physical properties of the fire's time series

- nvTracker: tracks each fire object for the inference of the model, calculates the variation in the size or shape of the detection bounding boxes of previous frames for tracked fires



4. Conclusions

- In this study, we propose a model that automatically generates digital-twin-based field-optimal learning data for the early detection of fires.

- Limitations

1. Detailed descriptions such as reflections of fire or background elements burning and turning into ash were not addressed
2. Since the background of the training images is constructed from recorded video data
3. Moving camera is also in our plan for future work
4. This proposed method is trained and tested on NvidiaTAO, which has a lack of supporting models

   → YOLOv8

- Ultimately, this undertaking not only adds value to the realm of research, but also plays a pivotal role in advancing the creation of a sophisticated digital twin dedicated to fire evacuation.