Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 04, December 2026.

LEAKAGE-FREE EVALUATION OF MASK R-CNN FOR FIRE REGION SEGMENTATION IN STILL IMAGES: ACCURACY AND FAILURE MODES

Abdullah Shekha Hamad

(Received: 7-Sep.-2026, Revised: 4-Nov.-2026, Accepted: 5-Nov.-2026)

ABSTRACT

Public fire datasets are built partly from video, so the same scene repeats many times. Splitting such images at random leaves near-copies in both the training and test sets, which makes accuracy look better than it is. We hashed 2,164 labeled images, grouped visually similar frames into 1,319 scene groups and split by group rather than by image. Under a random split, 49.1% of test images had a near-copy in training and over five training seeds mask AP was 7.1% higher than on the leakage-free split (0.561 ± 0.005 against 0.524 ± 0.005). Within the random split, the same models scored 0.632 on test images with a training near-copy and 0.497 on the others. On the leakage-free test set of 323 images, Mask R-CNN with a ResNet-50 FPN backbone reached a mask AP@[0.50:0.95] of 0.524 ± 0.005 and AP@0.50 of 0.808, with pixel IoU 0.739 and Dice 0.850, at about 27 FPS on a laptop GPU. The main weakness is small fire: the model does much worse on small fire regions than on large ones (mask AP 0.301 against 0.670). Over-segmentation is rare (1.5% to 4.0% of test images per run) and most of it covers real flames that the ground-truth labels do not include.

KEYWORDS

Fire detection, Instance segmentation, Mask R-CNN, Data leakage, Near-duplicate detection.

1. INTRODUCTION

Visual fire detection complements point sensors. Smoke and heat detectors work less well in large, open or well-ventilated spaces, such as warehouses and tunnels, because combustion products must reach the sensor before an alarm is raised, while a camera sees a fire from a distance and raises it within seconds. Most work puts a bounding box around the flame. A box is enough for an alarm, but it cannot measure burning area, flame perimeter or rate of spread: an irregular flame leaves a variable amount of background inside it and the overestimation cannot be calibrated away. Pixel-level segmentation isolates the flame and gives the geometry needed to judge severity.

Reported accuracies for learned fire segmentation are hard to compare and architecture does not explain the spread. Fire images are expensive and risky to collect, so most public datasets are built from video frames and overlapping sources. Consecutive frames of one fire are nearly identical, so a random split puts the same fire on both sides of the train/test boundary.

This is measurable, not hypothetical. In the 2,163 labeled still images used here, many of them video frames, a random split placed near-duplicates of 49.1% of the test images into the training set. Training the same configuration on both partitions with five random seeds gave a test mask AP of 0.561 ± 0.005 on the random split against 0.524 ± 0.005 once cross-subset near-duplication was removed.

This paper therefore reports a leakage-free evaluation. Near-duplicates are found by perceptual hashing, merged into scene groups and assigned to sub-sets as whole groups and the partition is verified to contain none at the 6-bit threshold. On it, we characterize Mask R-CNN with a ResNet-50 FPN backbone under COCO average precision, pixel-level agreement and instance-level matching. The detection method itself is not new and similar tools already exist (Sub-section 2.3). Our contribution is a measured case study for fire instance segmentation. The error analysis then examines the cases that the aggregate metrics hide.

The contributions of this work are as follows:

- We measure how much train/test leakage a public fire segmentation dataset holds and what it costs. Under a random split, near-duplicates of 49.1% of the test images sit in the training set. Rather than

A. Hamad is with IT department, Choman Technical Institute, Erbil Polytechnic University, Erbil, Iraq. Email: abdullah.hamad@epu.edu.iq

infer the effect from duplicate counts, we train the identical setup on both partitions with five seeds and compare.

- We build a leakage-free partition and release the procedure behind it. Images are grouped by perceptual hash and groups rather than images are assigned to sub-sets. We verify that no near-duplicate within the 6-bit threshold survives the split and we publish the partition manifest and the detection script. We also report how the results change with the hash threshold.
- We provide a reproducible baseline on the clean partition, characterized under three metric families, so the numbers can be read against both the detection and the semantic-segmentation literature, reported as mean ± standard deviation over five seeds (Sub-sections 3.6 and 0). The training log, environment record and per-image results will also be released upon publication.
- We analyze the remaining errors. Small fire regions are the main weakness. Over-segmentation, with high recall and low precision, is rare and most of it covers real flames that the labels miss (Sub-section 5.4).

2. Related Work

2.1 Traditional Fire Detection

Rule-based flame recognition begins with color. Phillips et al. [1] built a color lookup table from hand-labeled flame pixels and combined it with temporal variation and Chen et al. [2] formalized the decision as RGB and HSI inequalities. Çelik and Demirel [3] proposed a generic YCbCr model. Because static color rules cannot separate flame from similarly colored objects, later work added temporal cues: Töreyin et al. [4] analyzed flicker in the wavelet domain, Habiboğlu et al. [5] built covariance descriptors over spatio-temporal blocks and Ko et al. [6] classified candidate regions with a support vector machine. These methods are fast and need no training data, but they encode how fire looks under the illumination their thresholds were tuned for and they fail in still images, where no temporal signal exists. Chino et al. [7] addressed that case with BoWFire, combining pixel color with superpixel texture and reported that texture cut false positives at a cost in recall of up to 15% for all methods, but one. Their diagnosis still applies: color-driven methods fire on regions of high illumination and on reddish-yellowish objects. Sub-section 5.4 shows that a learned model still makes this error, but rarely.

2.2 Learned Fire Segmentation

Convolutional networks have largely displaced hand-designed rules and deep fire segmentation splits into two branches. The semantic branch uses encoder–decoder architectures, with U-Net [8] and DeepLabv3+ [9] as common baselines on the Corsican Fire Database [10] and FLAME [11]. Ghali et al. [12] compared vision-transformer segmenters against a U-Net baseline, Perrolas et al. [13] embedded a U-Net in a quad-tree search for high-resolution aerial imagery and Harkat et al. [14] studied loss function and encoder choice for DeepLabv3+ across three pooled datasets. The instance branch is smaller. Guan et al. [15] is closest to this work: an improved Mask Scoring R-CNN with a DSA-ResNet-50 backbone on FLAME, reporting precision 91.85%, recall 88.81% and mean IoU 82.31%. The underlying architecture is Mask R-CNN [16], a two-stage detector with a parallel mask branch, usually built on ResNet-50 [17] with a Feature Pyramid Network [18]. It remains the reference architecture for instance segmentation and COCO pre-training [19] makes it viable on far smaller datasets. Abuowaida et al. [20] proposed a three-stage instance-segmentation algorithm for multi-object images built on the same region-proposal design: an improved ResNet backbone with a squeeze-and-excitation block after each ResNet block, a region proposal network and an average-position RoI layer that avoids the quantization of RoI pooling. On COCO, it reached a mask AP of 0.458, against 0.357 for Mask R-CNN with a ResNet-101 backbone.

Two features of this literature matter here and both concern reporting rather than correctness. First, metric conventions are not shared: the semantic branch reports IoU, F1 and pixel accuracy, while the instance branch reports IoU and F1, but, in the case of [15], no COCO average precision. Instance results are therefore often not comparable with the detection literature they come from, which is why we report all three families. Second, the train/test split is usually not described. Among the studies surveyed, [13] says its partition is random and [14] that it is random with five-fold cross-validation, while [12], [15] and [21] give sub-set sizes without saying how images were assigned. That is ordinary practice, but it

matters for datasets built from video frames. FLAME is distributed as drone footage of prescribed burns together with extracted frames [11] and the Corsican Fire Database ships annotated video sequences alongside its still images [10].

These two gaps make published accuracies hard to place side by side. Reported mean IoU runs from about 0.78 in one study of aerial imagery [11] to 0.883 in another [13], with F1 approaching 0.999 in a third [12] and architecture alone does not explain a spread that size.

2.3 Dataset Contamination and Evaluation Protocols

Outside the fire literature, near-duplicate contamination of test sets is documented and quantified. Kaufman et al. [22] gave the general formulation of leakage in data mining. In computer vision, Barz and Denzler [23] found duplicates in the training set for 3.3% of CIFAR-10 and 10% of CIFAR-100 test images and measured a 9% to 14% relative drop in accuracy once they were purged.

The effect is larger where data is naturally grouped. Yagis et al. [24] showed that splitting brain-MRI data by slice rather than by subject inflated test accuracy by 29% to 55% across four datasets and that a randomly relabeled version of one dataset still reached about 96% accuracy under slice-level splitting, against the 50% expected by chance. Tampu et al. [25] reported 5% to 30% inflation for OCT classification under image-level rather than patient-level splitting. Figueiredo and Mendes [26] treated the video case, showing that random splitting of video-sourced frames leaks through spatio-temporal correlation and proposed cluster-aware splitting.

The standard detection tool is perceptual hashing, which maps an image to a short code, so that visually similar images produce codes a small Hamming distance apart. Zauner [27] benchmarked the major constructions and McKeown and Buchanan [28] characterized the Hamming distance distributions of six widely used variants, showing how far apart unrelated images normally fall.

Recent work shows that leakage is common in visual benchmarks. Ramos et al. [29] studied data leakage across public visual datasets and found some form of it in all seven datasets they analyzed. Adimoolam et al. [30] used perceptual hashing to detect duplicates and train–validation overlap in a large geospatial building dataset. Babu et al. [31] proposed D-LeDe, which detects leakage in automotive perception data by injecting part of the test set into training and tracking the relative increase rate of mAP. The leakaudit package [32] implements a pipeline close to ours: a 64-bit difference hash, transitive grouping, a group-aware split and an audit report, with a default threshold of 5 bits. We therefore do not claim the detection method as new. We apply it to fire instance segmentation and measure its effect with repeated training runs.

2.4 Gap

Fire segmentation is unusually exposed to near-duplicate contamination and unusually unlikely to check for it. Public datasets are built from video frames or scraped from overlapping sources, exactly the condition under which a random split leaks training images into the test set. Yet, the check is standard practice neither in the fire-segmentation studies reviewed in Sub-section 2.2 nor in the datasets' own documentation. This paper applies the check, quantifies the cost of omitting it, releases a leakage-free partition and reports results on that partition.

3. Materials and Methods

3.1 Dataset

We used the Fire Instance Segmentation Dataset (FireSmokeDataset) [33], which ships COCO-format polygon annotations for two classes, fire and smoke. The public release contains 1,738 training and 500 validation images carrying 3,053 and 883 annotations.

Only the fire class was kept. This removed 74 images that contained only smoke (57 from the training release and 17 from the validation release). Smoke annotations were discarded and annotations covering less than 60 px^2 after downscaling (see below) were dropped as too small to be reliable. One image (000562.jpg) lost its only fire instance in this step and was removed. Eighty-one images larger than 1,600 px on the long side were downscaled with their annotations to bound memory use. This left 2,163 images with 3,190 fire instances. Table 1 gives the composition after partitioning and Figure 1 the distribution of instance sizes.

Table 1. Dataset composition after group-aware partitioning.

Sub-set	Images	Fire instances	Instances / image
Train	1,515	2,204	1.45
Validation	325	492	1.51
Test	323	494	1.53
Total	**2,163**	**3,190**	**1.47**

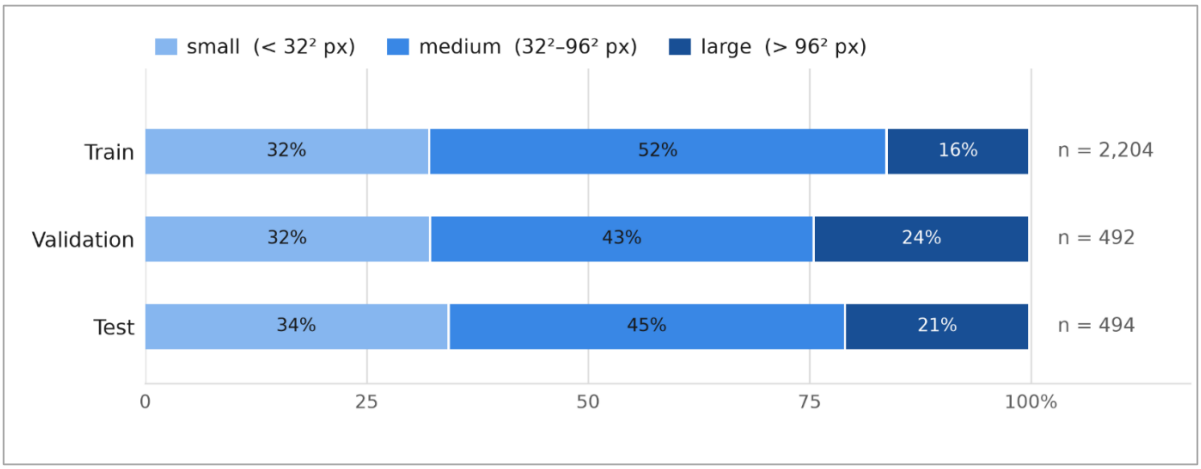

Figure 1. Distribution of fire instance sizes across the three sub-sets, under the COCO convention (small: area < 32^2 px, medium: 32^2 to 96^2 px, large: > 96^2 px). Bars are normalized to 100%. The instance count of each sub-set is given on the right.

3.2 Near-duplicate Detection and Group-aware Partitioning

Part of this corpus comes from video, so the same fire recurs across near-identical frames. Randomly splitting it would scatter frames from one event across sub-sets. The procedure below prevents that and Figure 2(a) summarizes it.

Detection: Each image was converted into grayscale, resized to 9 × 8 pixels with area interpolation and encoded as a 64-bit difference hash, in which each bit records whether the right-hand neighbor of a pixel is brighter than the pixel. Distances between all pairs were computed by mapping the bit vectors to ±1 and taking a matrix product, so that the Hamming distance is (64 − similarity) / 2. Two images within 6 bits of each other were treated as showing the same scene.

Grouping: Near-duplicate pairs were merged transitively by union-find, so that a run of consecutive frames forms one group even where its first and last frames are more than 6 bits apart. This gave 1,319 scene groups from the 2,164 fire images (grouping ran before the instance filter removed one image), with 1,213 images sharing a group with at least one other.

Assignment: Whole groups, not individual images, were assigned to sub-sets. We processed groups largest first and placed each one in whichever sub-set was furthest below its target share of 70/15/15. The order matters: an earlier attempt that assigned groups arbitrarily left the test sub-set with only 153 images, because several large groups happened to land together.

Verification: After partitioning, the near-duplicate search was run again across sub-set boundaries and returned zero pairs across all three boundaries (train/validation, train/test and validation/test). The check is released as a standalone script that exits non-zero if any train/test near-duplicate is found, so a reader can run it against their own partition.

The threshold of 6 bits out of 64 is a judgement rather than a derived constant. We chose it, because it grouped frames we could confirm by eye as duplicates without merging scenes that were visibly distinct. Figure 2(b) shows the distribution of pairwise Hamming distances. It has a large spike at 0 bits from exact copies (1,060 images have an exact copy), followed by a continuous tail that joins the bulk of unrelated pairs near 31 bits, so it shows no clean gap at 6 bits. Sub-section 4.2 and Table 4 therefore describe how the results change with the threshold.

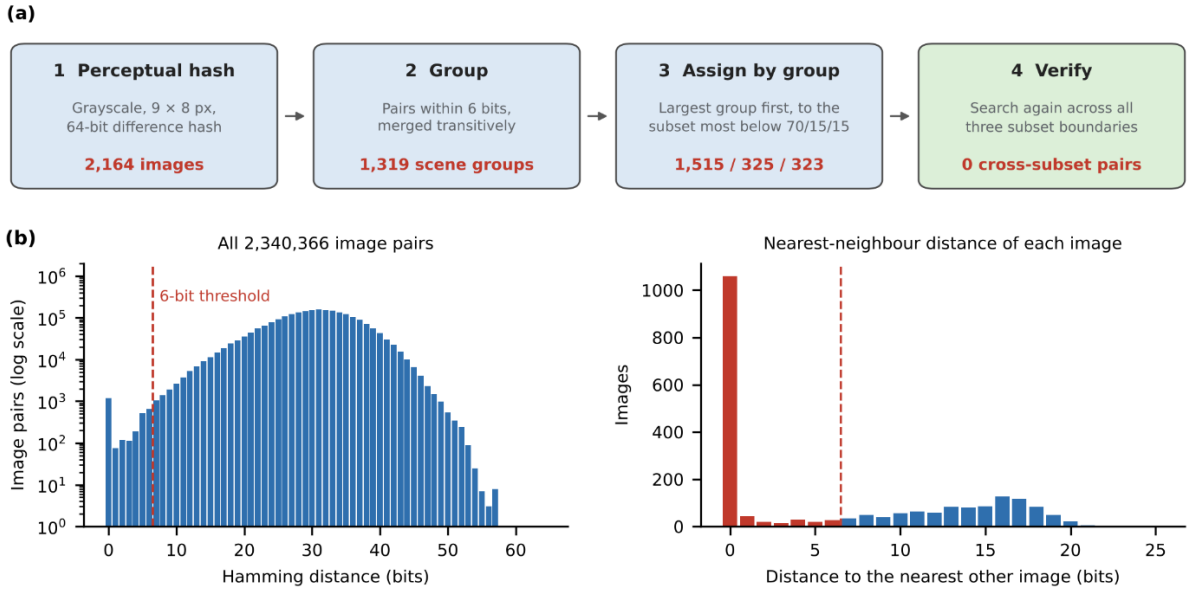

Figure 2. (a) Near-duplicate detection and group-aware partitioning. The instance filter removed one image after grouping, so the three sub-sets hold 2,163 images. (b) Hamming distances between the difference hashes of the 2,164 fire images: all pairs (left, log scale) and the distance from each image to its nearest other image (right), with images that have a near-duplicate within the threshold shown in red. The dashed line marks the 6-bit threshold.

3.3 Model

The model is Mask R-CNN with a ResNet-50 backbone and a Feature Pyramid Network neck, 43.92 M parameters in total and 43.7 M trainable. Figure 3 shows the configuration used here.

The FPN produces levels P2–P6. The region proposal network uses anchor scales of 32, 64, 128, 256 and 512 px, one per pyramid level, at three aspect ratios (0.5, 1.0, 2.0). RoIAlign pools 7 × 7 features for the box head and 14 × 14 for the mask head. The box head is two 1024-unit fully connected layers and the mask head four 3 × 3 convolutions at 256 channels followed by a 2 × 2 transposed convolution, giving 28 × 28 masks per region of interest.

Training minimizes the standard multi-task loss,

$$L = L_{cls} + L_{box} + L_{mask} + L_{obj} + L_{rpn}$$

where the mask term is a per-pixel binary cross-entropy applied only to the ground-truth class channel, which decouples mask prediction from classification and L_{obj} and L_{rpn} are the objectness and box-regression losses of the region proposal network.

3.4 Transfer Learning and Training Protocol

Weights were initialized from the COCO-pretrained Mask R-CNN, with the classification, box-regression and mask predictors re-initialized for two classes (background, fire). The ResNet-50 stem and layer1 were frozen and layer2 to layer4 and the FPN were fine-tuned. Batch-normalization statistics stayed frozen, since a batch size of 1 makes live statistics unreliable. We trained each configuration with five random seeds (42, 1, 2, 3 and 4). The seed controls the order of the training images, the augmentation and the initial weights of the new prediction heads, while the data split stays fixed. We report the mean and the sample standard deviation over the five runs. Table 2 lists the training configuration.

3.5 Data Augmentation

Training used horizontal flipping (p = 0.5) and photometric jitter (p = 0.3, with brightness, contrast and saturation varied by ±0.25 and hue by ±0.03). Vertical flipping was disabled, because flame orientation is physically meaningful. The jitter was meant to discourage reliance on absolute color.

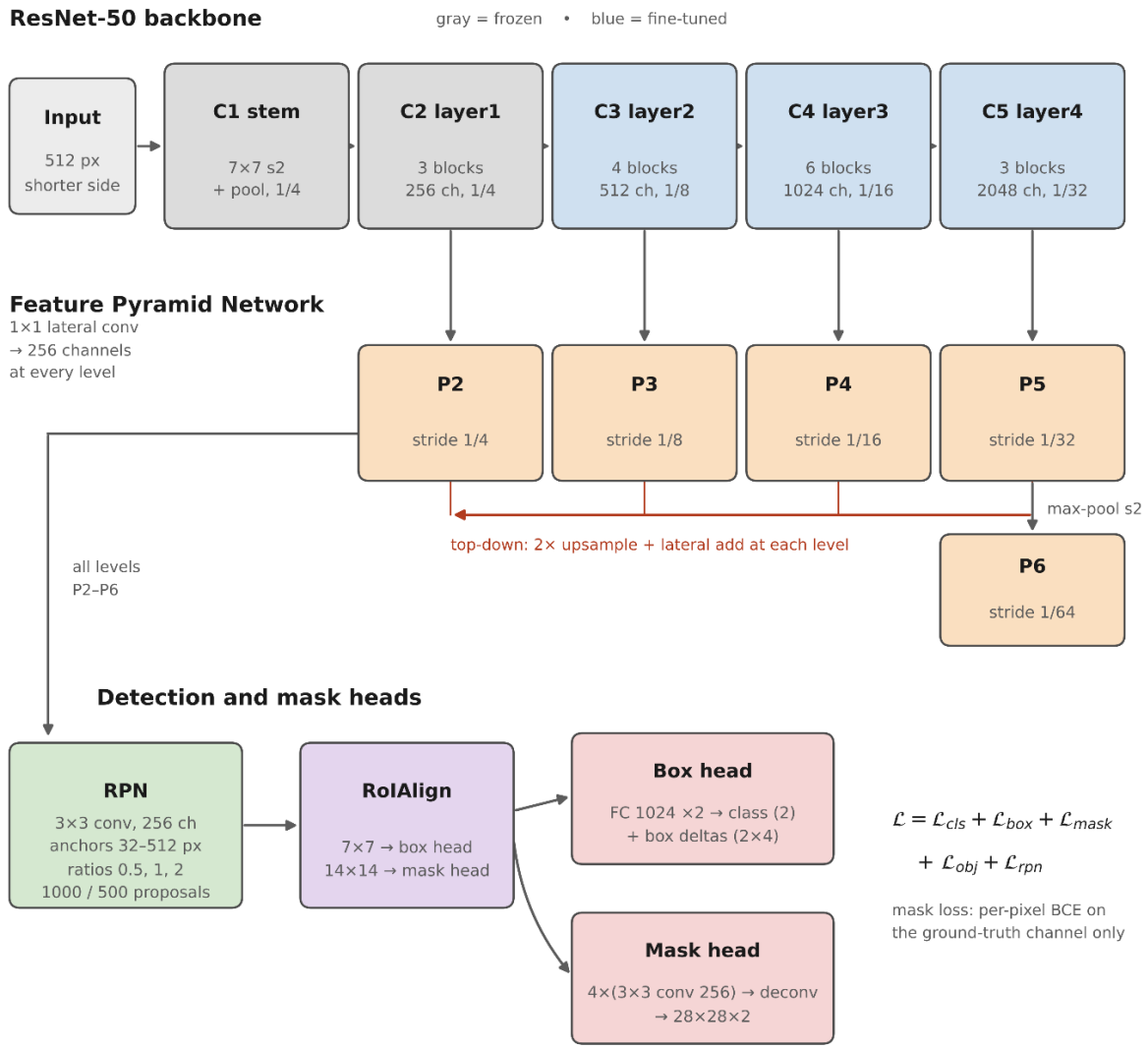

Figure 3. Mask R-CNN with a ResNet-50 FPN backbone, drawn for the configuration used in this study: 512 px shorter side, two classes, frozen stem and layer1 and the RPN proposal budget used (1,000 proposals per image in training and 500 at test time). Gray blocks are frozen. Blue backbone blocks, the FPN and the heads are fine-tuned.

Table 2. Training configuration.

Parameter	**Value**
Optimizer	SGD, Nesterov momentum 0.9
Initial learning rate	0.005
Schedule	Cosine annealing over 20 epochs
Warm-up	250 iterations, factor 0.001
Weight decay	1×10^{-4}
Gradient clipping	max-norm 10.0
Batch size	1
Input resize	shorter side 512 px, longer side $\leq$ 800 px
Precision	float32
Random seeds	42, 1, 2, 3 and 4 (five runs per partition); deterministic cuDNN
Hardware	NVIDIA RTX 3070 Laptop, 8 GB
Software	Python: 3.11.9, PyTorch: 2.13.0+cu130, torchvision: 0.28.0+cu130, CUDA: 13.0, cuDNN: 9.2
Training time	1h 19min to 1h 40min per run

3.6 Evaluation Metrics

We report three families, since the fire-segmentation literature is split between conventions:

COCO average precision for masks and boxes, reported as AP@[0.50:0.95], AP@0.50, AP@0.75 and the small/medium/large breakdown, computed with pycocotools and comparable with published Mask R-CNN results.

Pixel-level agreement at a fixed operating point: IoU, Dice, precision, recall and accuracy, pooled and averaged per image, for comparison with semantic-segmentation studies, which do not report AP.

Instance-level precision, recall and F1 at mask IoU 0.5, by greedy one-to-one matching. For each test image, predictions with confidence ≥ 0.50 are visited in order of their best mask IoU, from highest to lowest. A prediction is a true positive (TP) if its best-matching ground-truth instance has IoU ≥ 0.5 and is still unmatched. Otherwise, it is a false positive (FP). Ground-truth instances that stay unmatched are false negatives (FN). TP, FP and FN are summed over all test images and precision = TP/(TP + FP), recall = TP/(TP + FN).

Detection confidence and mask binarization thresholds were both fixed at 0.50 in advance, before any evaluation and were not tuned on the test set. Inference speed was measured at batch size 1 with CUDA synchronization, over 30 timed runs after 5 warm-up iterations.

Checkpoint selection used only the validation set. After each epoch, we computed validation mask AP and kept the checkpoint with the highest value. The test set was used only after training ended, to score that checkpoint. No test result influenced model selection, thresholds or hyper-parameters.

4. RESULTS

4.1 Training Behavior

Across the five seeds on the group-aware partition, total loss fell from an epoch average of 0.47–0.49 to 0.155–0.158 over 20 epochs. The mask term stayed the largest of the five components and finished at 0.112–0.113.

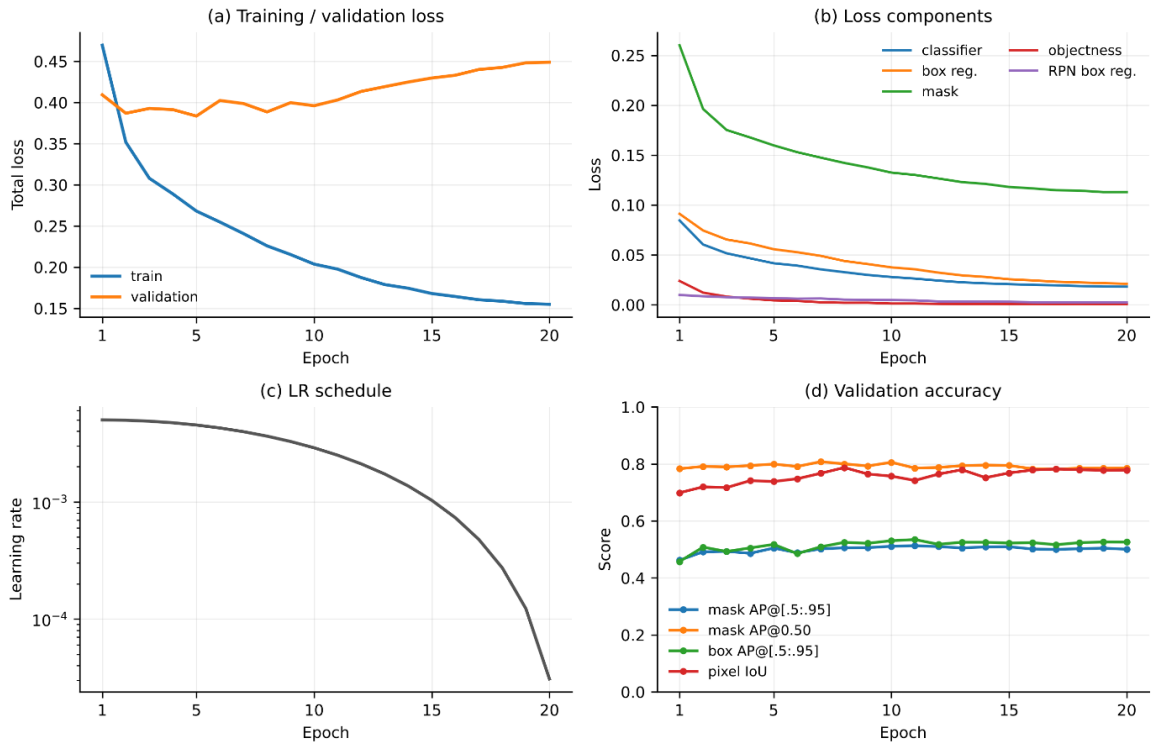

Figure 4. Training diagnostics over 20 epochs for the median run (seed 42). (a) Training and validation total loss. (b) The five individual loss components. (c) The cosine learning-rate schedule on a log scale. The 250-iteration warm-up falls inside epoch 1. (d) Validation accuracy under four metrics. Training loss falls steadily while validation loss drifts upward after epoch 5, yet validation mask AP in (d) stays between 0.49 and 0.51 over the same span.

Validation mask AP peaked between epochs 10 and 16 (0.518 ± 0.004) and those checkpoints produced every group-aware result below. The random-split runs were selected in the same way. Figure 4 collects the training diagnostics of the median run by test mask AP (seed 42).

4.2 What the Leakage Cost

Every number that follows depends on the partitioning decision, so it is worth showing what difference it makes. We trained the same model five times on each partition, with seeds 42, 1, 2, 3 and 4, changing only the assignment of images to sub-sets (Table 3).

Table 3. Effect of the partitioning strategy on measured accuracy, mean ± SD over five seeds. Model, hyper-parameters, augmentation, schedule and seeds were identical across the two partitions. The random partition placed 324 images in test and the group-aware one 323. The difference comes from one image that the instance filter removed.

	Random partition	Group-aware partition
Train ↔ test near-duplicate pairs	639	0
Test images with a near-copy in training	159 / 324 (49.1%)	0 / 323 (0%)
Best validation mask AP	0.583 ± 0.004	0.518 ± 0.004
Test mask AP@[0.50:0.95]	0.561 ± 0.005	0.524 ± 0.005
Test mask AP, leaked test images (159)	0.632 ± 0.008	—
Test mask AP, unleaked test images (165)	0.497 ± 0.005	—
Relative inflation	+7.1% ($p < 0.001$)	—

Inflation is 0.037 AP on test or 7.1% in relative terms (Welch's t-test, $t = 11.0$, $p < 0.001$) and larger on validation at 0.066 AP. Every random-split run (0.554–0.568) scored above every group-aware run (0.518–0.530) and the gap is about seven times the seed-to-seed standard deviation.

To separate leakage from differences between the two test sets, we split the random test set into 159 images with a training near-duplicate (leaked) and 165 images without one (unleaked) and scored the same random-split models on both. They reached 0.632 ± 0.008 mask AP on leaked images and 0.497 ± 0.005 on unleaked ones (paired t-test, $t = 32.3$, $p < 0.001$). On unleaked images, the random-split models did not beat the group-aware models (0.497 against 0.524), so the higher random-split score comes from the leaked images.

Table 4. Sensitivity of grouping and leakage to the Hamming threshold.

Bits	Scene groups	Largest group	Random split: test images with a train near-copy (of 324)	6-bit split: train–test pairs (test images)
0	1,502	10	134 (41.4%)	0 (0)
2	1,439	13	138 (42.6%)	0 (0)
4	1,384	74	148 (45.7%)	0 (0)
5	1,352	101	154 (47.5%)	0 (0)
6	1,319	174	159 (49.1%)	0 (0)
7	1,272	262	163 (50.3%)	8 (6)
8	1,210	341	171 (52.8%)	77 (22)
10	1,080	520	193 (59.6%)	398 (56)
12	921	751	215 (66.4%)	1,422 (91)

Table 4 shows how the leakage estimate depends on the hash threshold. For each threshold, we rebuilt the groups and re-checked both partitions. For any threshold from 4 to 8 bits, 45.7% to 52.8% of the random test images have a training near-copy. At 5 bits, the default of the leakaudit package [32], the value is 47.5%. Our 6-bit partition has no cross-sub-set pairs up to 6 bits, but 6 test images have a training match at 7 bits and 22 at 8 bits. Removing these images did not lower the group-aware mask AP (0.530 ± 0.006 and 0.535 ± 0.007, against 0.524 ± 0.005), so these borderline pairs do not act as leakage. Above 8 bits, transitive merging starts to chain unrelated scenes and the largest group grows from 174 images at 6 bits to 751 at 12 bits.

4.3 Accuracy on the Leakage-free Partition

Table 5 reports the full metric set on the held-out partition of 323 images and 494 fire instances, as mean ± SD over five seeds. Box and mask AP@0.50 are almost identical, 0.810 against 0.808, so at the loose threshold outlining the flame is no harder than boxing it. Figure 5 shows the precision–recall curves of the median run (seed 42).

Table 5. Test-set performance, 323 images and 494 fire instances, mean ± SD over five seeds. Latency is the median over nine runs and throughput is its reciprocal. Pixel accuracy is omitted: at 0.98 it is dominated by background and carries no information about the task.

Metric	Value	Metric	Value
Mask AP@[0.50:0.95]	0.524 ± 0.005	Pixel IoU	0.739 ± 0.010
Mask AP@0.50	0.808 ± 0.017	Pixel Dice	0.850 ± 0.006
Mask AP@0.75	0.604 ± 0.013	Pixel precision	0.887 ± 0.022
Mask AP small	0.301 ± 0.013	Pixel recall	0.817 ± 0.009
Mask AP medium	0.613 ± 0.013	Mean per-image IoU	0.793 ± 0.004
Mask AP large	0.670 ± 0.006	Instance precision @0.5	0.819 ± 0.031
Box AP@[0.50:0.95]	0.539 ± 0.004	Instance recall @0.5	0.785 ± 0.015
Box AP@0.50	0.810 ± 0.010	Instance F1 @0.5	0.801 ± 0.010
Latency (ms)	37.2	Throughput (FPS)	26.9

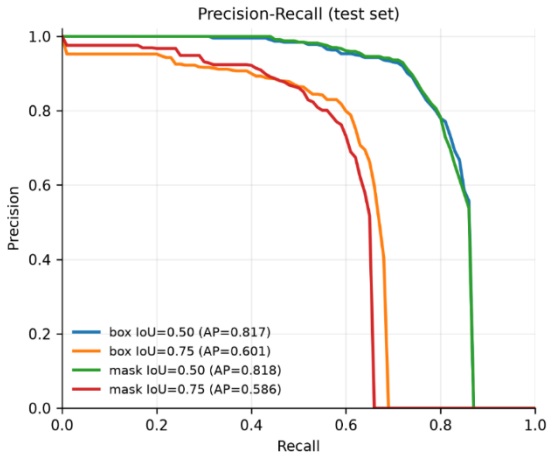

Figure 5. Precision–recall curves on the test set for the median run (seed 42), for boxes and masks at IoU thresholds 0.50 and 0.75. The box and mask curves at 0.50 are nearly superimposed (AP 0.817 and 0.818).

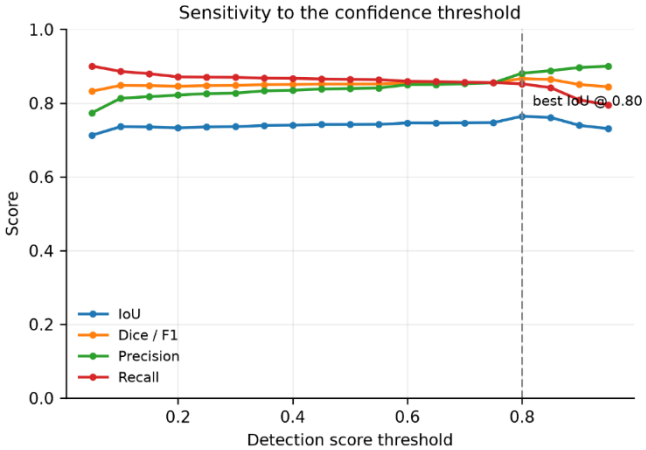

Figure 6. Pixel-level precision, recall, Dice and IoU across confidence thresholds on the validation set, for the median run (seed 42). As the threshold increases, precision rises and recall drops. The dashed line marks the threshold with the highest IoU (0.765 at 0.80).

We tested confidence thresholds from 0.05 to 0.95 on the validation set. The curve is flat near its peak, as Figure 6 shows. The best threshold changed from seed to seed (0.10, 0.40, 0.60, 0.80 and 0.85), so we kept the conventional 0.50 that we fixed before any evaluation.

Overall averages hide the distribution, so Figure 7 shows the spread of scores for individual images. The results are heavily left-skewed. For the median run, the median score is 0.838, well above the mean of 0.788. Across the five runs, the median is 0.835–0.844 and the mean 0.788–0.800. A small number of very poor results pulls the average down. We examine these failures in Sub-sections 5.3 and 5.4.

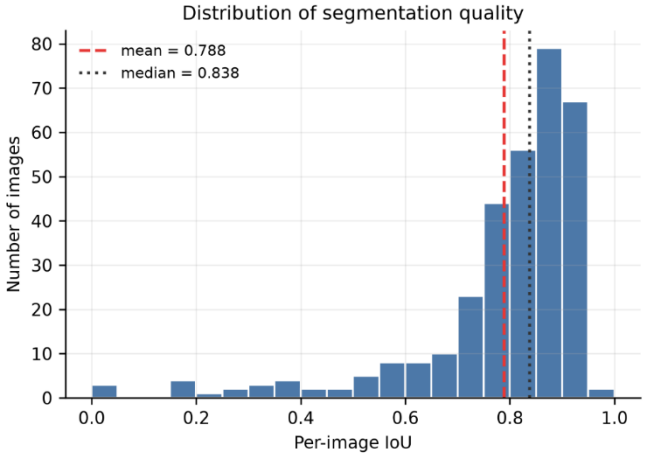

Figure 7. The distribution of per-image IoU scores across the 323 test images for the median run (seed 42). A half of the images score above 0.838, but the mean is only 0.788. This difference is caused by a small number of severe failures dragging the average down, rather than by poor performance across the board.

4.3 Comparison with Published Results

Table 6 places our results next to those of published fire-segmentation studies. The numbers are not directly comparable: the studies use different datasets and metric definitions and most do not describe how they split their images (Sub-section 2.2). We therefore show them as context, not as a ranking. Our pixel-level scores (IoU 0.739, Dice 0.850) are somewhat below the range reported for aerial imagery and only our study reports COCO mask AP, so the instance-segmentation result of Guan et al. [15] cannot be placed on the same scale. Moving from the group-aware split to the random split raised our own pixel IoU from 0.739 to 0.799 (mean of five seeds), a change of about 0.06, which is as large as many of the gaps in this table.

Table 6. Results of fire-segmentation studies. Values are copied from each paper and use each paper's own dataset, split and metric definitions. P = precision, R = recall, [a] mean IoU, [b] mean of per-image IoU.

Study (dataset)	Model	P	R	F1 / Dice	IoU	Mask AP
Guan et al. [15] (FLAME)	MaskSU R-CNN	0.919	0.888	0.903	0.823[a]	—
Shamsoshoara et al. [11] (FLAME)	U-Net	0.920	0.839	0.878	0.782	—
Perrolas et al. [13] (aerial)	U-Net + quad-tree	—	—	—	0.883[b]	—
Ghali et al. [12] (FLAME)	TransUNet	—	—	0.999	—	—
This work (FireSmokeDataset)	Mask R-CNN R50-FPN	0.887	0.817	0.850	0.739 (0.793[b])	0.524

5. Discussion

5.1 What the Leakage Result Means

The 3.7 AP point difference does not mean that previous studies are wrong. It means that results from randomly split, video-derived data cannot be directly compared with results from a group-aware split of the same data. Researchers often treat a difference of two or three AP points as evidence that one model is better than another. Our tests show that changing how the data is split can cause a gap of this size on its own, about seven times the seed-to-seed noise, even when the model stays the same. This makes the

splitting method a core part of the result, rather than a minor setup detail.

5.2 The Model Finds Fire Reliably but Disagrees about Its Edges

The 0.20 gap between AP@0.50 (0.808) and AP@0.75 (0.604) says that the model locates fire reliably and then disagrees with the annotator about its exact extent. Some of that is the model and some is the object. A flame has no crisp boundary, since luminance fades into smoke and haze, so the correct outline is ambiguous and two annotators would not draw it alike. The labels also have gaps: Sub-section 5.4 shows that in 10 of the 15 over-segmented test images, the extra mask covers real flames that the ground truth leaves out. We ran no inter-annotator study, which is a limitation, but mask AP 0.524 on flame is not the same achievement as 0.524 on cars.

5.3 Small Fire Is Where the Model Is Weakest

Mask AP falls from 0.670 on large instances to 0.301 on small ones, a 2.2-fold drop that appeared in every seed (2.12 to 2.35). A hard case, 000473.jpg, is a forest fire hidden behind smoke with only a small patch visible at the base. The model missed it completely in one of the five runs and segmented it well (IoU 0.78 to 0.81) in the other four. For early warning, this is the wrong way round, since an incipient fire is small, distant and often hidden by smoke.

Figure 8 shows the opposite end. The highest-scoring images each hold a single, well-separated flame with a clear boundary, which is the easy case rather than the one that matters in practice.

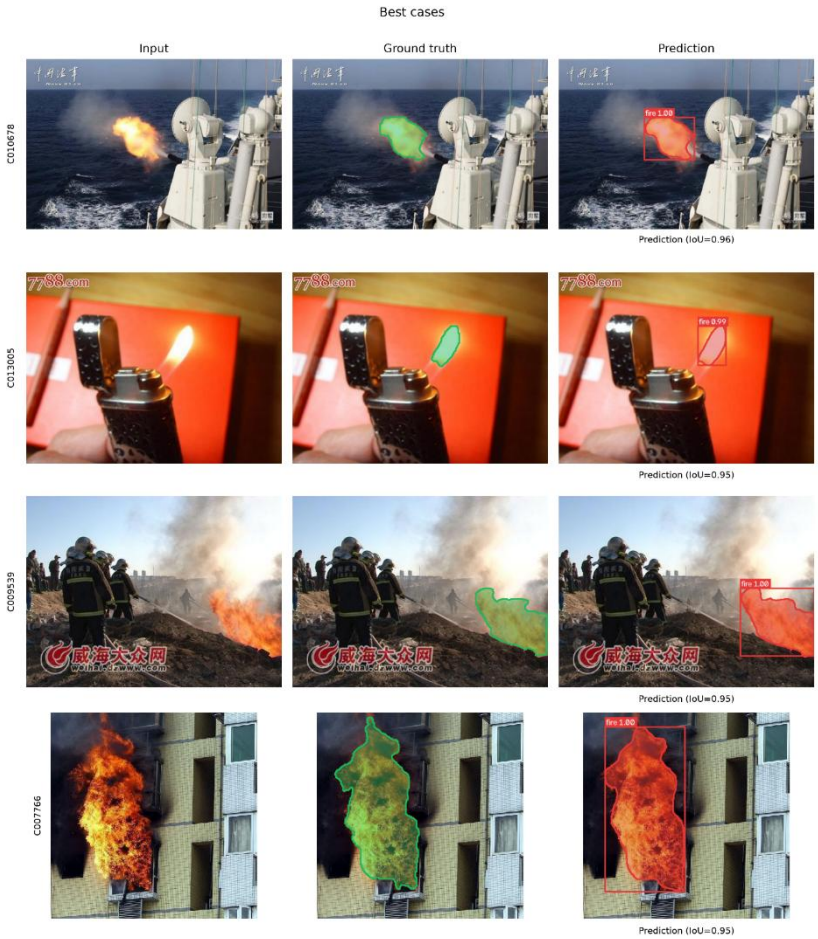

Figure 8. The four highest-scoring test images of the median run (seed 42), with IoU 0.95 to 0.96, showing input, ground truth and prediction. The ground truth is shown in green and the prediction in red with its box and confidence score. In every case, the flame is well separated from its surroundings.

5.4 Error Analysis: Over-segmentation

Across the five runs, 5 to 13 of the 323 test images (1.5% to 4.0%) reached recall above 0.70 with precision below 0.50, the signature of finding the flame and continuing past it. In total, 15 different

images were flagged and we inspected each of them (Table 7, Figure 9). In 10 images, the extra mask covers real flames that the ground-truth mask does not include. In 3 images, it covers glowing smoke or a fire-lit background. Only 2 images show objects lit by fire next to the flame.

The two fire-lit cases are the wax body of a candle and the orange jacket of a firefighter next to a flame. They show that warm color can still mislead the network, as with color-driven methods [7], but in our results, this error is rare. Most of these over-segmentations are labeling gaps rather than model mistakes, so the measured precision underestimates the true precision. Table 7 groups the 15 images by cause and Figure 9 shows one example of each cause.

Table 7. Over-segmented test images across five runs, by cause.

Cause	Images	Examples
Flame not included in the ground truth	10	000523, C001130, C004979, C003680
Glowing smoke or fire-lit background	3	C011818, 000007, C006632
Object lit by fire	2	C013660 (candle), 000459 (jacket)

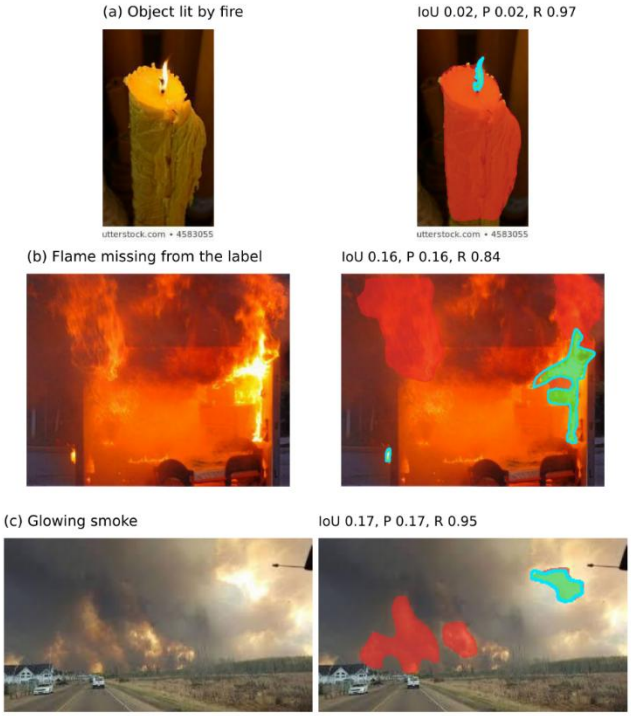

Figure 9. Typical over-segmentation cases from the median run (seed 42), one for each cause in Table 7. Left: input. Right: prediction. Green = predicted and labeled as fire, red = predicted, but not labeled, cyan line = ground-truth outline. (a) Object lit by fire (C013660). (b) Real flame missing from the label (000523). (c) Glowing smoke (C011818).

Averaged over the five runs, the model produced 387.6 true positives, 86.4 false positives and 106.4 false negatives against 494 ground-truth instances, so missed instances were slightly more common than spurious ones. Hard-negative mining on fire-lit scenes is a possible remedy for the few fire-lit cases, but we did not test it in this study.

5.5 Inference Speed and Limitations

At 37.2 ms per image (median latency over nine runs or 26.9 FPS) on a single laptop-class GPU, the model runs close to standard video frame rates (25 to 30 FPS) without dedicated hardware. This is one configuration on one device, excluding acquisition and decoding, so an upper bound rather than a general efficiency claim. We excluded one outlier speed measurement (6.7 FPS, about four times slower than the other nine runs).

The study uses still images and ignores temporal cues. We segmented one fire class and discarded the smoke annotations and GPU memory limited us to batch size 1 at 512 px. Two limitations are specific

to the leakage analysis. Perceptual hashing finds visual near-duplication, not semantic recurrence, so two photographs of the same fire from different viewpoints are not grouped and our 7.1% is a lower bound. The 6-bit threshold is a judgement and Table 4 shows how the results change with it. We also did not run a formal inter-annotator study and Sub-section 5.4 suggests that label gaps lower the measured precision.

6. CONCLUSION

This paper presented a leakage-free evaluation of Mask R-CNN with a ResNet-50 FPN backbone for pixel-level fire segmentation in still images. Public fire datasets are built largely from video and a random partition of one such dataset places near-duplicates of 49.1% of test images into the training set. Trained with the same configuration and five seeds on both partitions, the model scored a mask AP of 0.561 ± 0.005 on the contaminated split against 0.524 ± 0.005 on the group-aware one, an inflation of 7.1% and more on the validation set that selected the checkpoint. The protocol is part of the result, not a detail of the setup.

On the leakage-free partition of 323 test images, the model reached a mask AP@[0.50:0.95] of 0.524, AP@0.50 of 0.808 and pixel IoU of 0.739, at about 27 frames per second on a laptop-class GPU. The main weakness is small fire: accuracy drops 2.2-fold on small fire regions relative to large ones, the wrong way round for early warning. Over-segmentation is rare and most of it covers flames that the labels miss rather than objects lit by fire.

Evaluating on a verified split costs little, but changes what the reported numbers mean and reporting several seeds shows whether a difference is real. Hard-negative mining on fire-lit scenes and a more complete labeling of the test set are possible next steps that we did not test here. We release the partition manifest and the leakage-detection script, so that future work on this dataset can be compared on a clean and shared basis.

Data Availability: The scripts used to group near-duplicates, split the data and audit it for leakage are available at [34], together with the partition manifest listing the exact contents of each sub-set. We will release the full training and evaluation code upon publication.

REFERENCES

[1] W. Phillips III, M. Shah and N. da Vitoria Lobo, "Flame Recognition in Video," Pattern Recognition. Letter, vol. 23, no. 1–3, pp. 319–327, DOI: 10.1016/S0167-8655(01)00135-0, Jan. 2002.

[2] T.-H. Chen, P.-H. Wu and Y.-C. Chiou, "An Early Fire-detection Method Based on Image Processing," Proc. of the 2004 Int. Conf. on Image Processing (ICIP), pp. 1707–1710, DOI: 10.1109/ICIP.2004.1421401, Singapore, Nov. 2004.

[3] T. Çelik and H. Demirel, "Fire Detection in Video Sequences Using a Generic Color Model," Fire Safety Journal, vol. 44, no. 2, pp. 147–158, DOI: 10.1016/j.firesaf.2008.05.005, Feb. 2009.

[4] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay and A. E. Çetin, "Computer Vision Based Method for Real-time Fire and Flame Detection," Pattern Recognition Letter, vol. 27, no. 1, pp. 49–58, Jan. 2006.

[5] Y. H. Habiboğlu, O. Günay and A. E. Çetin, "Covariance Matrix-based Fire and Flame Detection Method in Video," Machine Vision and Applications, vol. 23, no. 6, pp. 1103–1113, DOI: 10.1007/s00138-011-0369-1, Nov. 2012.

[6] B. C. Ko, K.-H. Cheong and J.-Y. Nam, "Fire Detection Based on Vision Sensor and Support Vector Machines," Fire Safety Journal, vol. 44, no. 3, pp. 322–329, DOI: 10.1016/j.firesaf.2008.07.006, 2009.

[7] D. Y. T. Chino, L. P. S. Avalhais, J. F. Rodrigues and A. J. M. Traina, "BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis," Proc. of the 2015 28th SIBGRAPI Conf. on Graphics, Patterns and Images, pp. 95–102. doi: 10.1109/SIBGRAPI.2015.19, Aug. 2015.

[8] O. Ronneberger, P. Fischer and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, vol. 9351, N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with Atrous Separable Convolution for Semantic Image Segmentation," Proc. of the European Conf. on Computer Vision (ECCV 2018), vol. 11211, pp. 833–851, DOI: 10.1007/978-3-030-01234-2_49, 2018.

[10] T. Toulouse, L. Rossi, A. Campana, T. Celik and M. A. Akhloufi, "Computer Vision for Wildfire Research: An Evolving Image Dataset for Processing and Analysis," Fire Safety J., vol. 92, pp. 188–194, DOI: 10.1016/j.firesaf.2017.06.012, Sep. 2017.

[11] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé and E. Blasch, "Aerial Imagery Pile Burn Detection Using Deep Learning: The FLAME Dataset," Computer Networks, vol. 193, p. 108001, DOI: 10.1016/j.comnet.2021.108001, Jul. 2021.

[12] R. Ghali, M. A. Akhloufi and W. S. Mseddi, "Deep Learning and Transformer Approaches for UAV-based Wildfire Detection and Segmentation," Sensors, vol. 22, no. 5, p. 1977, DOI: 10.3390/s22051977, Mar. 2022.

[13] G. Perrolas, M. Niknejad, R. Ribeiro and A. Bernardino, "Scalable Fire and Smoke Segmentation from Aerial Images Using Convolutional Neural Networks and Quad-tree Search," Sensors, vol. 22, no. 5, p. 1701, DOI: 10.3390/s22051701, Feb. 2022.

[14] H. Harkat, J. Nascimento, A. Bernardino and H. Thariq Ahmed, "Assessing the Impact of the Loss Function and Encoder Architecture for Fire Aerial Images Segmentation Using DeepLabv3+," Remote Sensing, vol. 14, no. 9, p. 2023, DOI: 10.3390/rs14092023, Apr. 2022.

[15] Z. Guan, X. Miao, Y. Mu, Q. Sun, Q. Ye and D. Gao, "Forest Fire Segmentation from Aerial Imagery Data Using an Improved Instance Segmentation Model," Remote Sensing, vol. 14, no. 13, p. 3159, DOI: 10.3390/rs14133159, Jul. 2022.

[16] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," Proc. of the 2017 IEEE Int. Conf. on Computer Vision (ICCV), pp. 2980–2988, DOI: 10.1109/ICCV.2017.322, Venice, Oct. 2017.

[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, DOI: 10.1109/CVPR.2016.90, Las Vegas, USA, Jun. 2016.

[18] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 936–944. doi: 10.1109/CVPR.2017.106, Honolulu, Jul. 2017.

[19] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," Proc. of the European Conference on Computer Vision (ECCV 2014), vol. 8693, pp. 740–755, DOI: 10.1007/978-3-319-10602-1_48, 2014.

[20] S. F. A. Abuowaida et al., "A Novel Instance Segmentation Algorithm Based on Improved Deep Learning Algorithm for Multi-object Images," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 1, pp. 74–88, DOI: 10.5455/jjcit.71-1603701313, Mar. 2021.

[21] R. Ghali et al., "Wildfire Segmentation Using Deep Vision Transformers," Remote Sensing, vol. 13, no. 17, p. 3527, DOI: 10.3390/rs13173527, Sep. 2021.

[22] S. Kaufman, S. Rosset, C. Perlich and O. Stitelman, "Leakage in Data Mining: Formulation, Detection and Avoidance," ACM Trans. Knowl. Discov. Data (TKDD), vol. 6, no. 4, pp. 1–21, Dec. 2012.

[23] B. Barz and J. Denzler, "Do We Train on Test Data? Purging CIFAR of Near-duplicates," Journal of Imaging, vol. 6, no. 6, p. 41, DOI: 10.3390/jimaging6060041, Jun. 2020.

[24] E. Yagis et al., "Effect of Data Leakage in Brain MRI Classification Using 2D Convolutional Neural Networks," Scientific Reports, vol. 11, no. 1, p. 22544, DOI: 10.1038/s41598-021-01681-w, Nov. 2021.

[25] I. E. Tampu, A. Eklund and N. Haj-Hosseini, "Inflation of Test Accuracy Due to Data Leakage in Deep Learning-based Classification of OCT Images," Scientific Data, vol. 9, no. 1, p. 580, Sep. 2022.

[26] R. B. D. Figueiredo and H. A. Mendes, "Analyzing Information Leakage on Video Object Detection Datasets by Splitting Images into Clusters with High Spatiotemporal Correlation," IEEE Access, vol. 12, pp. 47646–47655, DOI: 10.1109/ACCESS.2024.3383047, 2024.

[27] C. Zauner, Implementation and Benchmarking of Perceptual Image Hash Functions, Upper Austria Univ. of Applied Sci., Hagenberg, [Online], Available: http://phash.org/docs/pubs/thesis_zauner.pdf, 2012.

[28] S. McKeown and W. J. Buchanan, "Hamming Distributions of Popular Perceptual Hashing Techniques," Forensic Sci. Int. Digit. Investig., vol. 44, p. 301509, DOI: 10.1016/j.fsidi.2023.301509, Mar. 2023.

[29] P. Ramos et al., "Data Leakage in Visual Datasets," Proc. of the IEEE/CVF Int. Conf. on Computer Vision Workshops (ICCVW), pp. 6309–6319, DOI: 10.1109/ICCVW69036.2025.00661, Honolulu, USA, 2025.

[30] Y. K. Adimoolam, C. Poullis and M. Averkiou, "Data Leakage Detection and De-duplication in Large Scale Geospatial Image Datasets," Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 72–81, 2026.

[31] M. A. A. Babu, S. K. Pandey, D. Durisic, A. C. Koppisetty and M. Staron, "D-LeDe: A Data Leakage Detection Method for Automotive Perception Systems," Proc. of the 11th Int. Conf. on Vehicle Technology and Intelligent Transport Systems (VEHITS), pp. 210–221, DOI: 10.5220/0013476700003941, 2025.

[32] K. Patel and S. A. Shah, leakaudit: detect and audit train/test leakage from near-duplicate images. (Jul. 2026). R package version 0.1.0. CRAN. [Online]. Available: https://CRAN.R-project.org/package=leakaudit

[33] Hankwenyx, "Fire-instance Segmentation-dataset," GitHub, [Online], Available: https://github.com/hankwenyx/Fire-Instance-Segmentation-Dataset, Oct. 2025.

[34] Abdullah, "Leakage-free Data Partitioning for Instance Segmentation," GitHub, [Online], Available: https://github.com/abdullah-shekha/leakagefree-firesegmentation.

ملخص البحث:

تُبنى مجموعات بيانات الحرائق المتاحة للجمهور جزئياً من مقاطع فيديو، ممّا يؤدّي إلى تكرار المشهد ذاته مرّاتٍ عديدة. وعند تقسيم هذه الصّور عشوائياً، قد تظهر نُسخ متطابقة تقريباً في كلّ من مجموعتي التّدريب والاختبار، وهو ما يجعل مستوى الدّقة يبدو أفضل ممّا هو عليه في الواقع.

في هذه الدّراسة قمنا باستخدام تقنية التّجزئة، وجمعنا الإطارات المتشابهة بصرياً في عدد من مجموعات المشاهد، ثم أجرينا التّقسيم بناءً على هذه المجموعات بدلاً من الصّور الفردية. وتبين أنّه في حالة التّقسيم العشوائي، احتوت مجموعة الاختبار على نُسخ متطابقة تقريباً من بيانات التّدريب بنسبة 49.1%، وكان متوسط دقّة القناع أعلى بنسبة 7.1% مقارنة بالتّقسيم الخالي من تسريب البيانات، وذلك بناءً على خمس تجارب تدريبية.

وبالنّسبة لمجموعة الاختبار الخالية من تسريب البيانات، بلغت سرعة المعالجة حوالي 27 إطاراً في الثّانية على وحدة معالجة رسوميات في حاسوب محمول. وتتمثّل نقطة الضّعف الرّئيسية في التّعامل مع الحرائق الصّغيرة، حيث يكون أداء النّموذج أقلّ بكثير في تحديد مناطق الحرائق الصّغيرة مقارنة بالمناطق الكبيرة (بلغت دقّة القناع 0.301 في المتوسّط مقابل 0.670).

ويعدّ التّقسيم المفّرط أمراً نادراً؛ إذ يَظهر في 1.5% إلى 4.0% من صور الاختبار في كلّ دورة تشغيل، ومعظم حالاته تتعلّق بوجود لهب فعلي لم تتضمنه البيانات المرجعية.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).