

Reducing False Alarms in Lightweight Indoor Fire Detection

Aya Hade^{1*}, Mustafa Kamil¹

¹Department of Computer Science, College of Computer Science and Information Technology, University of Sumer, Thi Qar, Iraq

* Email address of the Corresponding Author: ayahade@hs.uos.edu.iq
Phone number: 07733898390

Abstract

Deep learning fire detectors are almost always judged by mean average precision on a test set of fire images. That number says nothing about how often the model raises an alarm on a lamp, a candle, or a sunset. We trained six small YOLO detectors on an indoor fire dataset and found that every one of them fired on a quarter to a third of ordinary fire-like images, even though their detection scores looked healthy. This paper describes how we brought that rate down for YOLO26n, the smallest of the six and the model we deploy. We added background images with no labels to the training set, first chosen by hand and then mined automatically from a large open image collection using CLIP features and clustering. On 500 held out images of lamps, bulbs and candles the false alarm rate fell from 27.0 to 0.4 percent once both sources were combined, and on 196 images from fire-like categories that appear nowhere in training it fell from 19.4 to 1.2 percent. Distilling a larger teacher into the deployed model then recovered the accuracy given up along the way, reaching 0.9325 mAP50 with 2.5 million parameters and 5.8 GFLOPs.

Keywords: Fire detection, false alarms, background images, hard negative mining, knowledge distillation, YOLO.

تقليل الإنذارات الكاذبة في أنظمة كشف الحريق الخفيفة داخل المباني

آية هادي^{1*}, مصطفى كامل¹

¹قسم علوم الحاسوب، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة سومر، ذي قار، العراق

الخلاصة

تُقيَّم أنظمة كشف الحريق المبنية على التعلم العميق في الغالب بمقياس متوسط الدقة على مجموعة اختبار من صور الحريق، وهذا الرقم لا يخبرنا شيئاً عن عدد المرات التي يطلق فيها النموذج إنذاراً على مصباح أو شمعة أو غروب شمس. دربنا ستة نماذج YOLO صغيرة على مجموعة بيانات لحرارة المنازل، فوجدنا أن كل واحد منها يطلق إنذاراً على ما بين ربع وثلاث الصور الاعتيادية التي تشبه النار، رغم أن درجات الكشف لديها تبدو جيدة. تصف هذه الورقة كيف خفضنا هذه النسبة في النموذج YOLO26n، وهو أصغر النماذج الستة وهو النموذج الذي نشرناه. أضفنا إلى بيانات التدريب صوراً خلفية بلا تسميات، اخترنا مجموعة منها يدوياً ثم عدنا مجموعة أخرى آلياً من مكتبة صور مفتوحة كبيرة باستخدام تمثيلات CLIP والعنقدة. وعلى 500 صورة محجوزة لمصابيح وشموع انخفض معدل الإنذار الكاذب من 27.0 إلى 0.4 بالمئة بعد دمج المصدرين، وعلى 196 صورة من فئات تشبه النار لا وجود لها في التدريب انخفض من 19.4 إلى 1.2 بالمئة. ثم استخدمنا تقطير المعرفة من نموذج أكبر إلى النموذج المنشور، فاستعاد الدقة التي تنازلنا عنها وبلغ 0.9325 مع 2.5 مليون معامل و 5.8 مليار عملية فاصلة عائمة فقط.

1. Introduction

Most house fires are small when they start. A camera that notices smoke in the first minute is worth more than one that classifies a fully developed fire perfectly, which is why vision

based detection has become an active research area alongside conventional smoke alarms [1]. Cameras are already installed in many homes for other reasons, and a detector small enough to run on an embedded board can watch a room continuously without sending video anywhere. The catch is that a home alarm which goes off for no reason gets unplugged. Missing one frame out of ten is tolerable, because fire stays in view for many seconds and the alarm only has to trigger once. Alarming on the reading lamp every evening is not tolerable, no matter how high the detection score is.

This is where the published literature is thin. The recent lightweight fire detectors are built the same way and reported the same way. ESCFM-YOLO splits the network into two streams to catch small smoke on an edge device [2], EMG-YOLO targets embedded hardware [3], DSS-YOLO builds a lighter real time detector on top of YOLOv8 [4], EFA-YOLO inserts a feature attention module [7], and YOLOv11-CHBG does something similar for the eleventh generation [8]. Every one of them reports precision, recall and mean average precision on a test set in which every image contains fire, and every one of them stops there. The reference architecture for this paper, YOLO-HF, reaches 0.932 mAP50 on indoor home fire imagery [12] and reports nothing else either.

None of those numbers answer the question an installer actually asks, which is how often the system will alarm in a room where nothing is burning. A detector can score above 0.93 and still be unusable in a living room, and no table in these papers would reveal it. We looked for studies that measure this directly, on large independent sets of images containing no fire, and found very few. The closest we found treats indoor false alarms as the problem itself but classifies video clips of benign flames rather than detecting objects in frames, on a backbone far heavier than anything that fits on an embedded board [16]. Section 2 sets out what the literature does and does not measure, and Table 1 summarises it.

We started from that gap, and the first step was to measure the problem instead of assuming it away. We assembled four separate sets of images with no fire in them, drawn from different sources, and measured how often trained detectors produced a detection on each set. The result was worse than we expected. Six standard YOLO models, all trained normally and all scoring above 0.90 mAP50 on the fire test set, fired on between 26.4 and 33.7 percent of hard background images. At the operating point that allows at most one false alarm per hundred background images, most of them detected almost nothing.

The rest of the paper is about closing that gap without giving up detection accuracy. Section 2 reviews related work, and Section 2.3 states the gap precisely and says what the review is for. Section 3 describes the data, the background image sets, the mining procedure and the metrics. Section 4 reports the experiments in the order we ran them, including the ones that failed, and Sections 4.7 and 4.8 draw them together into a single ablation and a single account of the trade-off between false alarms and missed fires. Section 5 concludes.

Our main results are the following. Adding unlabelled background images to training removes most of the false alarms, but only if those images are chosen for being hard rather than at random. Background images picked by hand from a small number of categories overfit to those categories and leave the model exposed to anything outside them. Mining background images automatically from a large open collection fixes that weakness, and using both sources together is better than either alone. Distilling a larger model into the small deployed one then raises detection accuracy to the level of a much heavier published architecture while keeping the false alarm rate near one percent. We call the resulting family SAFE-Fire.

2. Related Work

2.1 *Lightweight fire and smoke detection*

Fire and smoke detection with convolutional detectors is a well covered area, and recent work is dominated by modifications of the YOLO family [5], [6]. Authors insert attention blocks, redesign the neck, or split the network into separate streams for flame and smoke, usually with the stated aim of running on a constrained device [2], [4], [7], [8]. Deployment studies place these detectors on drones and in tunnels [9], [10], and comparative studies benchmark successive YOLO versions against each other on smoke imagery [11].

The reference point for this paper is YOLO-HF [12], which builds on YOLOv5s with a parametric boosted channel attention module, space to depth downsampling and a redesigned block, and reports 0.932 mAP50 on a public indoor fire dataset. We use that dataset and its original test split, so our numbers are directly comparable.

What these papers share is the evaluation protocol. Performance is reported on fire images only. YOLO-HF, like the works it compares against, gives no measurement of behaviour on images without fire, no measurement on hardware other than a desktop GPU, and no repeated runs.

2.2 *False positives, negatives and reliability*

Outside the fire literature the problem is recognised. Chen et al. built COCO-FP [13] by collecting images that contain none of the target categories but on which detectors still fire, and showed that strong detectors lose a substantial amount of accuracy once such images enter the evaluation. That work measures the problem and stops there, and it does not touch fire detection or embedded deployment.

Training with negative examples is an old idea. Online hard example mining collects the highest loss regions during training and concentrates the gradient on them [14], and remains standard practice in detection. Clustering has been used to pick a diverse subset of negatives rather than simply take the highest scoring ones. What has not been asked, as far as we can tell, is whether negatives selected by a person from a handful of named categories generalise to categories that person never thought of. That question turns out to matter.

One recent paper approaches fire detection from a reliability angle. Zhang et al. treat it as a calibration problem [15], converting expected calibration error into a differentiable loss so that predicted confidences match observed correctness. That is a different target from ours. A well calibrated model can still be confidently wrong about a lamp, and we are trying to stop the wrong detection from happening at all. The closest work to ours in aim is that of Ali and Ghodrat [16], who also treat indoor false alarms as the problem worth solving and release a video dataset of benign flame sources such as vented fireplaces, tabletop gas pits, flameless candles and decorative LED flames, reporting 90.4 percent accuracy with a 3D attention network. Their setting differs from ours in three ways that matter for a device on a wall. The task is clip classification rather than detection, so there are no boxes and no localisation. The false alarms they target are real flames that happen to be safe, while ours are objects that only resemble fire. And their backbone costs about forty billion operations per clip, which is roughly eight times the budget of the model we deploy.

2.3 The gap this paper addresses

Read together, the works above have a shape. The fire detection papers compete on architecture and report their gains on fire images. The papers that take background false positives seriously either sit outside fire detection altogether [13], or address a different failure mode inside it, calibration rather than suppression [15], or work on a task and a compute budget that a wall mounted device cannot use [16]. Table 1 lays this out. The columns are not accuracy, because accuracy is where these works already agree and where they are already comparable. The columns are whether the model was ever run on images containing no fire, whether such images were used during training, and whether the reported numbers come from more than a single run.

The third column is the gap. If nobody measures behaviour on images without fire, then nobody can report improving it, and a reader has no way to tell a quiet detector from a noisy one by comparing published tables. This is why the literature review does more than list prior art here. It establishes that the quantity this paper improves is one the field has not been reporting, and an improvement in a quantity nobody reports cannot be claimed by pointing at anyone else's results. It has to be measured from scratch, on both sides. That is the order the rest of the paper follows. We build four frozen sets of images containing no fire, show that six ordinary detectors fail on all four, and only then ask what fixes it.

Table 1- What related work measures. The entry "N/R" means not reported in the paper.

Work	What it does	Run on images	Background	More than one
		without fire	images in training	run
ESCFM-YOLO [2]	Dual stream detector for edge devices	N/R	N/R	N/R
EMG-YOLO [3]	Fire detector for embedded hardware	N/R	N/R	N/R
DSS-YOLO [4]	Lightweight real time detector built on YOLOv8	N/R	N/R	N/R
EFA-YOLO [7]	Feature attention module for fire and flame	N/R	N/R	N/R
YOLOv11-CHBG [8]	Lightweight fire detector	N/R	N/R	N/R
YOLO-HF [12]	Attention and downsampling redesign, 0.932 mAP50	N/R	N/R	N/R
COCO-FP [13]	Background false positives in general detection	Yes	No, evaluation only	N/R
Zhang et al. [15]	Confidence calibration for fire detection	N/R	N/R	N/R
Ali and Ghodrati [16]	Clip classification of benign indoor flames	Yes, video of benign flames	Yes, benign flame clips	N/R
This work	Lightweight detection with mined background images	Yes, 2,596 images, four frozen sets	Yes, 3,276 images	Yes, three seeds

3. Materials and Methods

3.1 Fire dataset

We use the public Home-fire dataset released with the reference work [12], which contains 6,500 annotated images split into 3,900 for training, 1,300 for validation and 1,300 for test, with two classes, flame and smoke. The training split holds 2,978 flame and 1,834 smoke boxes; the test split holds 897 flame and 689 smoke boxes. About a fifth of the test images come from indoor security cameras and show early stage fires, which is the case we care about. The original split is kept unchanged so that our results can be compared against the published numbers on the same data.

Counted end to end, the study uses 6,500 images that contain fire or smoke and 5,872 images that contain neither. The second group splits in two and the split matters. 3,276 unlabelled background images enter training. A further 2,596 background images are used only for evaluation and were never seen by any model during training, at any stage, in any configuration reported here. Table 2 sets out every part, its size and where it came from, so that the denominator behind each rate in Section 4 can be traced to a specific set of files.

Table 2- Every image set used in this study.

Part	Role	Images	Content and source
Home-fire, train split	Training	3,900	2,978 flame and 1,834 smoke boxes [12]
Home-fire, validation split	Model selection during training	1,300	1,580 boxes [12]
Home-fire, test split	Every accuracy number in this paper	1,300	897 flame and 689 smoke boxes [12]
Hand-picked backgrounds	Training	1,638	Lamps, light bulbs and candles, Open Images train split [17]
Mined backgrounds	Training	1,638	Drawn from 12,052 detector hits over a corpus of 76,183 images [17], [19]
Union of the two	Training	3,276	The two rows above, with no overlap between them
Hard holdout	Evaluation only	500	Lamps, bulbs and candles, disjoint by file from the training pool [17]
Novel fire-like	Evaluation only	196	Lanterns, torches, flashlights, street lights, traffic lights [17]
Cross-dataset	Evaluation only	1,500	Non-fire images from D-Fire, an independent corpus [18]
Open-world mined	Evaluation only	400	Held out from the mined hits before any training budget was drawn

Figure 1 shows examples from the dataset together with their annotations. They cover the range the model has to handle: a flame in its first seconds, a fully developed room fire, smoke with no flame in view, and the low quality wide angle frames that indoor security cameras produce.

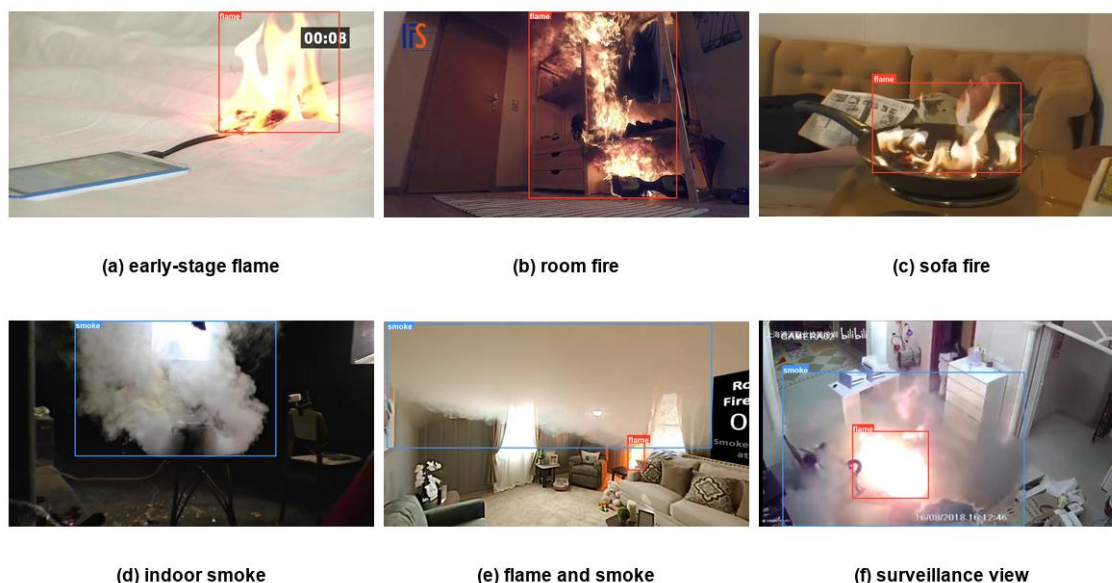


Figure -1 Examples from the Home-fire dataset with ground truth boxes.

3.2 Evaluation Sets and False-Alarm Metrics

A background image, in the sense used here, is an image containing no fire and no smoke. In the YOLO training format this is an image whose label file is empty, which the framework treats as pure background.

Four evaluation sets of background images were built. All four were frozen before any experiment was run, and no image in any of them was ever used for training. The hard holdout set contains 500 images of lamps, light bulbs and candles drawn from Open Images [17]. These are the obvious confusers for a fire detector. They come from the same three categories as the hand-picked training negatives but from a disjoint list of files, so this set measures whether a model learned those categories and not whether it memorised particular pictures. The novel set contains 196 images from fire-like categories that appear in no hand-picked training negative at all, among them lanterns, torches, flashlights, street lights and traffic lights. It is the set that answers the question this paper is really about, which is whether a model learned the specific categories it was shown or the general idea of a bright warm object. The cross-dataset set contains 1,500 background images from D-Fire [18], collected by a different group, from different cameras, under a different annotation policy. The open-world set contains 400 images that the mining procedure of Section 3.3 retrieved from the open corpus and that were removed from the pool before any training budget was drawn from it. It is therefore a deliberately hard set and not a neutral sample of everyday scenes, and Section 4.3 reads it accordingly.

Two kinds of disjointness matter here and they are not the same, so we state each one at the strength we can actually support. Disjointness by file holds for all four sets against all training data, it is enforced by a frozen manifest, and every one of the 2,596 evaluation images carries a SHA-256 fingerprint, so any rate reported below can be recomputed on exactly the pixels that produced it. Disjointness by category is stronger, and we claim it only for the novel set against the hand-picked source, which by construction contains nothing except lamps, light bulbs and candles. The mined source carries no category list at all, because nobody wrote one, so for that

source we state only what can be checked: the corpus was filtered of every fire related label before the detector was run over it, and no mined image appears in any evaluation set.

Figure 2 shows what the four sets look like. None of these images contains fire, and none of them was seen during training.

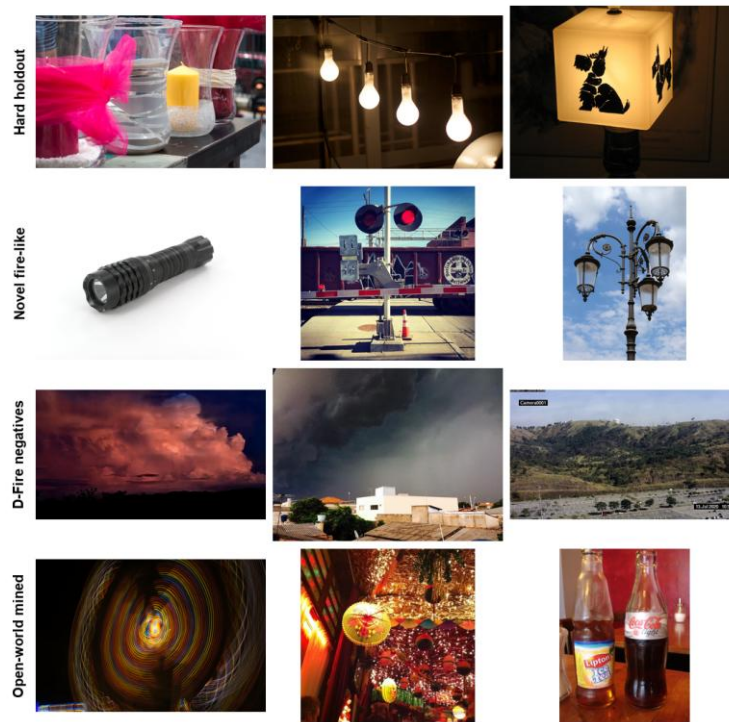


Figure - 2 Examples from the four frozen background sets used to measure false alarms.

Two numbers summarise behaviour on these sets, and both need their denominators stated. The first is the false alarm rate. The model is run over a background set at a confidence floor of 0.01, the highest confidence box in each image is taken, and an image on which the model draws no box at all scores zero. That image counts as a false alarm when its highest confidence reaches the operating threshold. The count is over images and not over boxes, so an image carrying nine spurious boxes counts once, and it ignores the predicted class, so a lamp called smoke counts exactly as a lamp called flame. For a set of N background images with per image maxima s_i and a threshold t ,

$$FPR(t) = (100 / N) \sum_i 1[s_i \geq t] \quad (1)$$

Unless a table states otherwise the threshold is 0.50, which is the value a deployed system would use, and N is the size of the individual set named in the column, so 500, 196, 1,500 or 400. We never pool the four sets into one rate. The sets differ in difficulty, and a pooled figure would hide which of them a model is failing on, which is the only thing the sets were built to reveal.

The second number puts both sides of the problem together. We sweep the threshold, find the lowest value at which the false alarm rate stays at or below one percent, and report the recall on the fire test set at that threshold, written R at FPR under 1 percent:

$$R@FPR \leq 1\% = \text{Recall}(t^*), \quad t^* = \min \{ t : FPR(t) \leq 1 \} \quad (2)$$

A model that cannot reach one percent false alarms without also suppressing its true detections scores near zero here, which is exactly the behaviour we want to expose. One point about this measure has to be explicit, because the tables can otherwise be read the wrong way. The threshold t^* is calibrated on the hard holdout set alone. It is a single number per model. Where it appears in a table beside columns for the novel, cross-dataset and open-world sets, those columns are false alarm rates at the fixed threshold of 0.50 and not at t^* , and no claim is made that a threshold calibrated on one set transfers to another. Section 4.8 measures what happens when t^* is calibrated on each set separately instead, which is the honest way to ask that question. Detection accuracy is reported throughout with mAP50 and mAP50-95 on the fire test set.

3.3 Hard-Negative Backgrounds: Hand-Picked vs. Mined

Two sources were used, and they were built differently on purpose.

The hand-picked set comes from the same three Open Images categories as the hard holdout set, drawn from the training split so that no image overlaps with the evaluation holdout. This is what a researcher does naturally: think about what confuses the model, then go and find images of it. The training pool holds 1,638 such images across lamps, light bulbs and candles.

The mined set was produced automatically. We assembled a corpus of 76,183 images by combining the Open Images test split [17] with the Places365 scene collection [19], after filtering out anything carrying a fire related label. The current detector was run over the whole corpus and every image on which it produced a detection was kept, which yielded 12,052 hits. Keeping the highest scoring hits directly would have produced a pile of near duplicates of one or two scenes, so the hits were embedded and clustered before anything was selected.

The embedding is CLIP ViT-B/32 [20], which gives one 512 dimensional vector per image, and it was chosen for two reasons. The first is that we did not want the detector's own features. Clustering the hits in the space of the network that produced them groups images by what that network already encodes, and what that network encodes is precisely what caused the mistakes, so the clusters would reproduce its blind spots rather than expose them. CLIP was trained on image and text pairs under a different objective, so it sorts the hits by what the scene is rather than by what the detector thinks of it. The second reason is practical. Because CLIP places images and text in one space, a cluster can be named by matching its members against a list of ordinary scene and object terms, with no manual labelling at any point. That is how Table 6 was produced, and it is the part of the procedure that told us what the detector was actually confusing for fire.

The number of clusters was not fixed by hand. We ran k-means over k in {6, 8, 10, 12, 16, 20} at a fixed random seed and scored each fit by silhouette coefficient, obtaining 0.0422, 0.0395, 0.0428, 0.0464, 0.0476 and 0.0457 in that order. Sixteen was the maximum and was taken, and the final fit was run again with more initialisations at that value. The absolute coefficients are low, which is what natural images in a 512 dimensional space give, so the sweep should be read as choosing between candidates rather than as evidence that the clusters are tight. What matters for the method is that the selection covers distinct regions of the hit set, not that the regions have clean boundaries.

The quality of the selected negatives was checked in three ways. First, the zero-shot name of every cluster was read. One cluster turned out to hold genuine fire images that had slipped

past the label filter, and it was discarded before any sampling took place. This check is not a formality: a mined negative that actually contains fire would teach the model to suppress the one thing it exists to find, and there is no metric elsewhere in this paper that would catch it cleanly. Second, 400 of the surviving hits were set aside as the open-world evaluation set before the training budget was drawn, so that mined data could be judged on images it had never contributed to. Third, the budget of 1,638 images was drawn twice under the two rules described above, and the two selections overlapped in only 236 images. Training on each separately, which Section 4.3 reports, shows whether the gain comes from the clustering itself or from the particular images one rule happened to return.

Figure 3 puts the two sources next to each other, and the contrast is the point of this section. The hand-picked images are lamps, bulbs and candles, because those are what come to mind when a person asks what looks like fire. The mined images are a steam plume, a sunlit rock face, an ice cream sign, a bedroom lamp and a flame pattern printed on a jacket. The detector scored every one of them above 0.93, and none of them belongs to a category anyone had thought to collect.

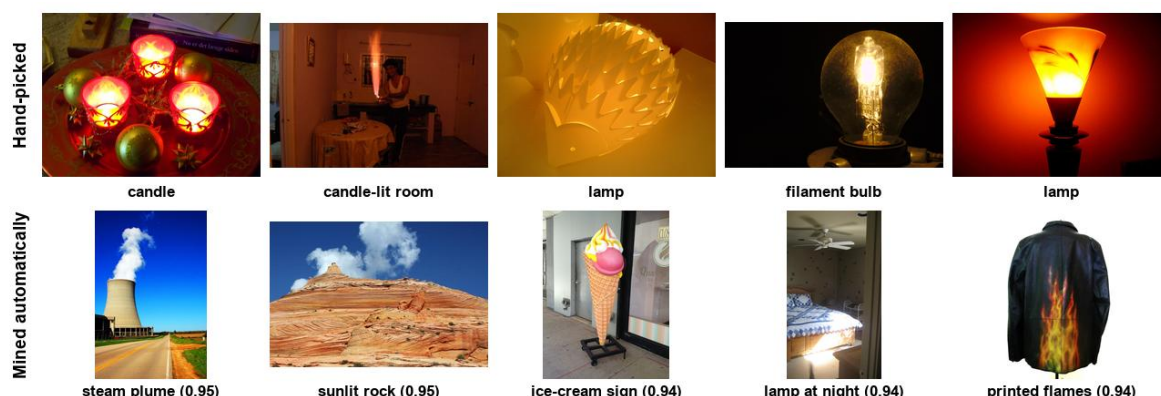


Figure -3 Hand-picked against automatically mined background images. The top row is what a researcher expects to be confusing. The bottom row is what the detector actually fired on, with its confidence in brackets.

3.4 Training setup

All models were trained with the Ultralytics framework [6] on a single NVIDIA L4 GPU. The recipe was fixed before the experiments began and never changed: 300 epochs, AdamW [23] with initial learning rate 0.001 and weight decay 0.05, cosine schedule, batch size 32 at 640 pixels, patience of 50 epochs, mosaic augmentation at full strength and copy-paste at 0.1. Weights pretrained on COCO [24] were used for initialisation. Unless a table states otherwise, every configuration was trained three times with seeds 42, 123 and 2024, and the mean is reported.

4. Results and Discussion

4.1 Choosing a base model

Six detectors were trained under the identical recipe, three nano models intended for the embedded target and three small models for reference. Table 3 gives the result.

Table 3- Baseline models, averaged over three seeds. FPR is measured on the hard holdout set.

<i>Model</i>	<i>mAP50</i>	<i>mAP50-95</i>	<i>Recall</i>	<i>R@FPR1</i>	<i>FPR (%)</i>	<i>Params (M)</i>	<i>GFLOPs</i>
<i>YOLOv8n</i>	0.9075	0.5797	0.8565	0.094	32.1	3.01	8.2
<i>YOLO11n</i>	0.9017	0.5824	0.8466	0.095	32.5	2.59	6.4
<i>YOLO26n</i>	0.9013	0.5843	0.8481	0.000	27.0	2.51	5.8
<i>YOLOv8s</i>	0.9145	0.5893	0.8650	0.123	30.5	11.14	28.7
<i>YOLO11s</i>	0.9056	0.5891	0.8605	0.027	33.7	9.43	21.6
<i>YOLO26s</i>	0.9215	0.6012	0.8782	0.000	26.4	9.95	22.5

Two things stand out. The detection scores are all respectable and all within about two points of each other, which is what a conventional comparison table would show and where most papers stop. The false alarm column tells a different story. Every model fires on a quarter to a third of the hard background images, and recall at one percent false alarms is close to zero for most of them, meaning there is no threshold at which these models are both quiet and useful.

YOLO26n was selected as the deployment model on the declared rule of highest mean mAP50-95 within the nano class. It also has the lowest computational cost and the lowest false alarm rate of the three nano models. YOLO26s was set aside as a possible teacher.

Figure 4 places the six detectors on the accuracy against compute plane. The three nano models sit together at low cost and the small models spread out at three to five times the computation, which is the region an embedded board cannot afford.

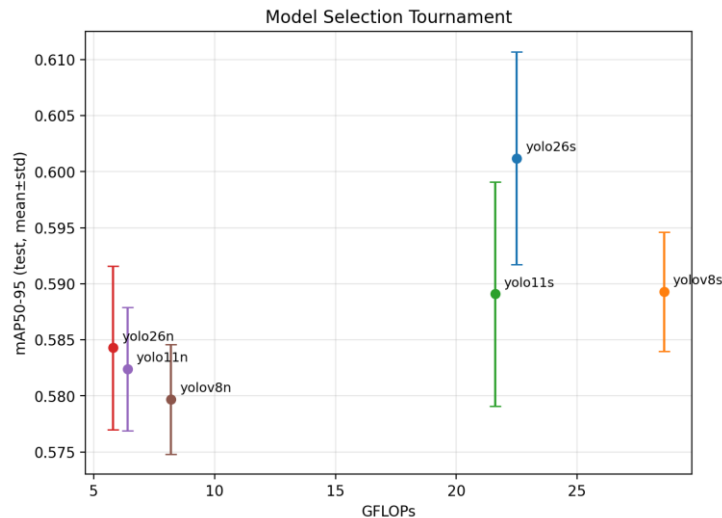


Figure -4 Model selection tournament: mean mAP50-95 on the test set against GFLOPs, with error bars over three seeds.

4.2 Effect of Negative-Image Quantity and Hardness

Background images were added to the training set at two ratios relative to the 3,900 fire images, and a control was run in which the same number of images was drawn at random from COCO instead of being chosen for difficulty. Table 4 shows the outcome.

Table 4- Effect of adding background images to training.

<i>Setting</i>	<i>mAP50</i>	<i>R@FPR1</i>	<i>FPR hard (%)</i>	<i>FPR open-world (%)</i>
<i>None (baseline)</i>	0.9013	0.000	27.0	32.4
<i>15 percent, hard</i>	0.9141	0.750	2.00	15.0
<i>30 percent, hard</i>	0.9025	0.800	1.07	12.0
<i>30 percent, random</i>	0.8865	0.000	9.40	5.25

The jump from the baseline to any hard setting is large, and detection accuracy does not suffer for it. The random control is the informative row. Adding the same volume of ordinary photographs leaves the model still firing on 9.4 percent of hard backgrounds, with recall at one percent false alarms still at zero, so the gain does not come from having more images. It comes from having the right ones. Two caveats belong with this table. The 15 percent setting and the random control were each run once, so they are read here as a direction and not as a spread, while the 30 percent hard setting that the rest of the paper builds on was run three times and gave 1.0, 0.8 and 1.4 percent on the hard set. And the second ratio is nominally 30 percent but is 29.6 percent in fact: the recipe asked for 1,671 images and the pool of the three chosen categories held 1,638 in total. For the same reason we had intended to test a 45 percent ratio and could not, because the request was truncated to the same 1,638 images, which would have made it a duplicate of the 30 percent run rather than a fourth point.

4.3 Hand-Picked Negatives Fail to Generalize

The hand-picked model looks good on the set it was designed for, firing on only 1.07 percent of the hard holdout images. On open-world images the same model still fires 12 percent of the time. Table 5 compares it against a model trained on the same number of mined images. Both sampling rules from Section 3.3 were trained and evaluated at seed 42 before one was carried forward. Spreading the budget evenly across clusters gave 0.9022 mAP50 with 2.60 percent false alarms on the hard set and 0.7011 recall at a one percent budget. Taking the highest confidence hits from each cluster gave 0.8871 mAP50 with 1.40 percent and 0.7516. The second rule was carried forward on the strength of the last two of those numbers, and the mined results below use it.

Table 5- Hand-picked against mined background images. Both use 1,638 background images.

<i>Source</i>	<i>mAP50</i>	<i>R@FPR1</i>	<i>FPR hard (%)</i>	<i>FPR novel (%)</i>	<i>FPR cross (%)</i>	<i>FPR open (%)</i>
<i>Hand-picked</i>	0.9025	0.800	1.07	3.06	9.58	12.0
<i>Mined</i>	0.8987	0.790	1.00	1.36	3.09	1.42

On detection accuracy and on recall at one percent false alarms the two are level, and on the hard holdout they are level as well. Everywhere else the mined model is ahead. On the novel categories it fires on 1.36 percent against 3.06, and on the independent D-Fire negatives on 3.09 percent against 9.58. On the open-world set the gap is a factor of 8.5, and that one comes with a caveat we would rather state than have pointed out to us. The open-world images were retrieved by the mining procedure itself, so they are nearer in kind to the mined training data than to the hand-picked training data, and part of that particular factor is a property of how the set was built. The novel and D-Fire sets carry no such advantage. Neither was produced by the mining pipeline, D-Fire was not produced by us at all, and the mined model is ahead on

both by wide margins. That is where the generalization claim rests, and it does not need the open-world number to stand up.

The explanation is straightforward once the numbers are laid out. The hand-picked set was drawn from three categories, and the hard holdout set was drawn from those same three categories. Testing one on the other measures how well the model learned those categories, not how it will behave in a house. The mining procedure never had a category list. It asked the model where it was wrong across 76,183 images, and the sixteen clusters it produced, summarised in Table 6, are dominated by reflections and glare, glowing screens, orange vehicles, red flowers, food and people in warm coloured clothing, none of which we had thought to include by hand.

Table 6- What the detector confused for fire, from clustering the 12,052 mined hits into sixteen groups and merging groups with the same label.

<i>Confuser category</i>	<i>Images</i>
<i>Reflections and glare</i>	1,883
<i>People and warm-coloured clothing</i>	3,061
<i>Orange vehicles and objects</i>	2,904
<i>Glowing screens and monitors</i>	1,092
<i>Lamps and light fixtures</i>	636
<i>Orange or red clothing</i>	632
<i>Bright windows and indoor lighting</i>	626
<i>Red and orange flowers</i>	624
<i>Food and fruit</i>	594

This is the finding we consider most useful to others working on the same problem. Choosing negatives by intuition produces a model that is quiet in the ways you imagined and noisy in the ways you did not, and the only way to see it is to evaluate on categories you never selected.

4.4 Combining Hand-Picked and Mined Negatives

Since the two sets were built by unrelated procedures, we tried their union, 3,276 images after removing duplicates. Table 7 gives the comparison.

Table 7- Combining both background sources, averaged over three seeds.

<i>Source</i>	<i>mAP50</i>	<i>R@FPR1</i>	<i>FPR hard (%)</i>	<i>FPR novel (%)</i>	<i>FPR cross (%)</i>	<i>FPR open (%)</i>
<i>Hand-picked</i>	0.9025	0.800	1.07	3.06	9.58	12.0
<i>Mined</i>	0.8987	0.790	1.00	1.36	3.09	1.42
<i>Both</i>	0.9006	0.826	0.40	1.19	3.64	1.00

The union gives the best recall at one percent false alarms of the three and the lowest false alarm rate on three of the four sets, at the same detection accuracy, and it has the tightest spread across seeds. The two sources are complementary rather than redundant. The hand-picked images cover the obvious confusers densely, the mined images cover the long tail, and neither covers the other.

4.5 Input Resolution and Model Scaling

Raising the input size is the usual way to buy accuracy. Table 8 shows what it bought here.

Table 8- Input resolution, single runs at 640, 768 and 960 pixels.

<i>Resolution</i>	<i>mAP50</i>	<i>mAP50-95</i>	<i>Flame AP50</i>	<i>Smoke AP50</i>	<i>GFLOPs</i>
640	0.9049	0.5837	0.9348	0.8751	5.8
768	0.9123	0.5986	0.9447	0.8800	7.5
960	0.9070	0.6004	0.9248	0.8892	11.7

The gain at mAP50 peaks at 768 pixels and falls back at 960, while mAP50-95 keeps climbing and smoke accuracy improves steadily with resolution. Under the long training schedule used here the benefit of high resolution is smaller than short-schedule experiments suggest, and it costs twice the computation, so 640 was kept for the deployed model.

Resolution was more useful for building a teacher. Training YOLO26s at 960 pixels on the combined background set produced the strongest model in the study. Averaged over three seeds it reached 0.9285 mAP50 and 0.6193 mAP50-95, with recall of 0.881 at a one percent false alarm budget and a false alarm rate of 0.33 percent on the hard set. Its three runs were 0.9346, 0.9287 and 0.9223 mAP50, and the first of them, seed 42, is the checkpoint used as the teacher in Section 4.6. Both figures appear in this paper and they are not in conflict: 0.9285 is the mean of the three runs and 0.9346 is the single run that was distilled from. At 9.95 million parameters and 46 GFLOPs at 960 pixels the model is far outside the embedded budget, so it was used as a teacher rather than as a product. Scaling further to YOLO26m at the same resolution and the same seed gave 0.9272 mAP50 against 0.9346 for YOLO26s, which suggests the dataset is not large enough to support the extra capacity.

4.6 Knowledge Distillation into the Deployed Model

At this point the deployed model reached 0.90 mAP50, about three points below the published reference. Knowledge distillation [25] was used to close that gap. We used the feature based form that has been standard in detection since Chen et al. [26], with the 960 pixel teacher supervising the 640 pixel student through the three neck layers that feed the detection head. Everything else was held fixed: same architecture, same background set, same schedule, same seed. Table 9 reports the result.

The student is trained against the sum of the ordinary detection loss and a distillation term:

$$L = L_{det} + \lambda_{dis} L_{KD} \quad (3)$$

where the first term is the usual YOLO objective, itself a weighted sum of the box regression, classification and distribution focal losses:

$$L_{det} = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{dfl} L_{dfl} \quad (4)$$

and the second compares the features that student and teacher produce at the three neck outputs that feed the detection head:

$$L_{KD} = (1/|L|) \sum_l (1/C_l H_l W_l) \| \phi_l(F_S^l) - F_T^l \|^2 \quad (5)$$

In (5) the two feature maps are those of the student and of the teacher at level l , taken over the three pyramid levels P3, P4 and P5, while C , H and W are the channel and spatial dimensions of that level. The function applied to the student map is a 1×1 convolution that lifts its channel count to the teacher's, since a nano model is narrower than a small one. The teacher is frozen and run in inference mode on the same batch as the student, so the two feature maps share spatial dimensions and only the channels need aligning. Gradients reach the student and

the adaptation layers only. The weight in (3) is the single knob we vary; the framework calls it the distillation gain.

The teacher was the seed 42 checkpoint of the YOLO26s model trained at 960 pixels on the combined background set, the run that reached 0.9346 mAP50 in Section 4.5. The student was YOLO26n at 640 pixels. Everything other than the distillation term was held identical to the baseline run: the same 3,900 fire images and 3,276 background images, the same 300 epoch AdamW schedule, the same batch size and the same seed. Only the distillation weight was changed, at 6 and at 12. A two epoch trial run was used first to confirm the distillation path was active in the framework version we used, since a silent failure would have cost a full night of training. One detail is worth recording: the distilled run kept improving until epoch 223, while the baseline stopped improving at epoch 103 under the same budget and the same patience setting, which suggests the teacher's features carried signal that the labels alone did not.

Table 9- Effect of distillation on the deployed model.

<i>Configuration</i>	<i>mAP50</i>	<i>mAP50-95</i>	<i>Recall</i>	<i>R@FPR1</i>	<i>FPR hard (%)</i>	<i>FPR open (%)</i>
<i>No distillation</i>	0.8987	0.5778	0.8346	0.824	0.8	1.25
<i>Distillation, weight 12</i>	0.9242	0.6188	0.8578	0.804	1.0	1.50
<i>Distillation, weight 6</i>	0.9325	0.6250	0.8819	0.798	1.2	2.75

The accuracy gain is larger than we expected. Published distillation results for this model family report improvements of a few tenths of a point on COCO, and here the same mechanism gave 3.4 points of mAP50 and 4.7 points of mAP50-95. The likely reason is dataset size. COCO has over a hundred thousand training images and this dataset has 3,900, so the teacher supplies a supervision signal that a small training set cannot provide on its own. The distilled student also kept improving for far longer, reaching its best epoch at 223 against 103 for the undistilled run under the same budget.

There is a cost, and it runs in the opposite direction to the accuracy. The weaker distillation term is the more accurate setting and it is also the noisier one. At weight 6 the model reaches 0.9325 mAP50 and fires on 1.2 percent of the hard set and 2.75 percent of the open-world set. At weight 12 it gives back eight tenths of a point of mAP50 and comes down to 1.0 and 1.5 percent on the same two sets. Recall at a one percent budget moves the same way, 0.798 at weight 6 against 0.804 at weight 12. The direction makes sense once the teacher is taken into account. The teacher was trained on the same background images and is itself very quiet, at 0.33 percent on the hard set, so the harder the student is pushed to reproduce the teacher's features the more of that quiet arrives with them. A lighter term leaves the student freer to fit the fire data, which buys accuracy and gives back some of the silence. The weight therefore behaves as a dial between raw accuracy and reliability, and a model can be moved along it without retraining from scratch or changing the architecture. We deploy weight 6, because the false alarm rate it gives is already inside the budget we set for the system and the accuracy is worth having.

4.7 Ablation Study

The sections above introduce one change at a time, which is the order the work was done in but not the easiest order to read. Table 10 puts every step in one place, measured on the same

frozen evaluation sets, at the same operating threshold of 0.50, on the same fire test split. The missed detection column is one minus recall and is included because it is the quantity a user of the system actually pays for.

Table 10- Every step of the method under identical test conditions.

Configuration	mAP50	Recall	Missed (%)	R@FPR1	FPR hard (%)	FPR novel (%)	FPR cross (%)	FPR open (%)
Baseline, no background images	0.9013	0.8481	15.2	0.000	27.00	19.39	11.80	32.42
Hand-picked backgrounds	0.9025	0.8439	15.6	0.800	1.07	3.06	9.58	12.00
Mined backgrounds	0.8987	0.8339	16.6	0.790	1.00	1.36	3.09	1.42
Both sources	0.9006	0.8366	16.3	0.826	0.40	1.19	3.64	1.00
Both, distillation weight 12	0.9242	0.8578	14.2	0.804	1.00	3.06	1.93	1.50
Both, distillation weight 6	0.9325	0.8819	11.8	0.798	1.20	1.53	1.60	2.75

Read down the false alarm columns and the shape is clear. Almost all of the gain arrives with the first hard negatives, and the later steps redistribute it rather than add to it. Hand-picked images take the hard set from 27.0 to 1.07 percent but leave the novel set at 3.06 and the open-world set at 12.0. Mined images give up nothing on the hard set and bring the other three down with it. The union is the lowest of the four on three sets out of four, and it is also the only configuration whose recall at a one percent budget passes 0.82. Distillation is a separate axis and should not be read as part of the same trend. It does not improve the false alarm rate, and on two sets it slightly worsens it, but it is the only step that improves detection accuracy and the only step that takes the missed detection rate below the baseline, from 15.2 percent to 11.8.

4.8 False Alarms, Recall and Missed Detections

Section 3.2 gives the definition of a false alarm. This section gives the arithmetic behind the headline figures, because a rate quoted without its denominator cannot be checked. The hard holdout set has 500 images. The baseline YOLO26n produced at least one box at confidence 0.50 or above on 138 of them at seed 42, on 128 at seed 123 and on 139 at seed 2024, which is 27.6, 25.6 and 27.8 percent and a mean of 27.0. The deployed model produced a box on 6 of the same 500, which is 1.2 percent. The novel set has 196 images. The baseline fired on 36, 37 and 41 of them across the three seeds, a mean of 19.4 percent, and the model trained on both background sources fired on 4, 1 and 2, a mean of 1.19 percent. The cross-dataset set has 1,500 images and the open-world set has 400. Every percentage in this paper is a count of images over one of those four denominators, and no other denominator is used anywhere.

The suppression is not free, and Table 10 is where the price shows. Recall on the fire test set falls from 0.8481 for the baseline to 0.8439 with hand-picked images and to 0.8339 with mined ones, so the missed detection rate rises from 15.2 to 16.6 percent while the false alarm rate on the hard set falls from 27.0 to 1.0. In plain terms that is one and a half extra missed detections per hundred fire images, bought at twenty six fewer false alarms per hundred

background images. Distillation then removes the deficit rather than merely offsetting it. The deployed model misses 11.8 percent, which is lower than the baseline that had never seen a background image at all, so on this dataset the final system is not paying for its quiet in missed fires when the two are measured at the same threshold.

A budget calibrated on one set does not transfer to another, and Table 11 measures how far it fails to. For every model and every background set we found the lowest threshold at which that set's false alarm rate is at or below one percent, then read the recall on the fire test set at that threshold. The hand-picked model is the clearest case in the table. Held to one percent on the hard holdout, which is built from the same three categories it trained on, it keeps 0.833 recall. Held to one percent on any of the other three sets it keeps between 0.341 and 0.440. The mined model stays between 0.696 and 0.796 on all four. This is the same generalisation failure that Table 5 shows in the false alarm columns, seen from the recall side, and it is the sharper of the two views because recall at a fixed budget is what a deployed system would actually deliver.

Table 11- Recall on the fire test set when the threshold is calibrated independently on each background set.

<i>Configuration</i>	<i>Hard holdout</i>	<i>Novel</i>	<i>Cross-dataset</i>	<i>Open-world</i>
<i>Baseline, no background images</i>	0.305 (0.92)	0.305 (0.93)	0.355 (0.89)	0.305 (0.92)
<i>Hand-picked backgrounds</i>	0.833 (0.42)	0.372 (0.88)	0.440 (0.86)	0.341 (0.89)
<i>Mined backgrounds</i>	0.772 (0.64)	0.793 (0.58)	0.696 (0.76)	0.796 (0.57)
<i>Both sources</i>	0.835 (0.35)	0.623 (0.78)	0.645 (0.76)	0.779 (0.55)
<i>Both and distillation</i>	0.838 (0.51)	0.482 (0.84)	0.816 (0.56)	0.701 (0.73)

The deployed model deserves an honest reading of this table. It is the strongest of the five on the hard holdout at 0.838 and on the cross-dataset set at 0.816, and it is the weakest of the three mined configurations on the novel set at 0.482. Holding it to one percent on categories it has never seen costs it roughly half its recall. We report that rather than choosing a threshold that hides it, and it is the result in this paper we would most want a follow up study to improve.

There is a harder limit than any of these tables shows, and it belongs here rather than in a later section. We kept four photographs of small indoor fires outside every split. The baseline detected all four, at maximum confidences of 0.81, 0.85, 0.87 and 0.96. The mined model dropped three of them to 0.006, 0.004 and 0.14, and the deployed model to 0.005, 0.025 and 0.020, while the fourth stayed above 0.93 for every model we trained. Three small indoor flames are pushed below any threshold a person would set, which is a failure of exactly the kind this system exists to prevent. The cause is the background images and not the distillation, because the mined model with no distillation at all fails on the same three. Suppressing things that look like fire and detecting a small fire early are closer to the same problem than we would like them to be, and on this evidence the method reported here trades some of the second for the first.

4.9 Comparison with the State of the Art

Table 12 places the final models next to the published architecture on the same test split.

Table 12- Comparison with YOLO-HF on the same dataset and the same test split.

<i>Model</i>	<i>mAP50</i>	<i>mAP50-95</i>	<i>Recall</i>	<i>Flame AP50</i>	<i>Smoke AP50</i>	<i>Params (M)</i>	<i>GFLOPs</i>
<i>YOLOv5s</i>	0.904	0.603	0.833	-	-	7.02	-
<i>YOLOv5m</i>	0.915	0.623	0.855	-	-	20.9	-
<i>YOLO-HF</i>	0.932	0.645	0.879	0.963	0.901	6.59	31.5
<i>Ours, deployed</i>	0.9325	0.6250	0.8819	0.9486	0.9164	2.51	5.8
<i>Ours, teacher</i>	0.9285	0.6193	0.8801	0.9579	0.8991	9.95	46.2

The deployed model reaches the same mAP50 as the reference architecture using 38 percent of its parameters and 18 percent of its computation, with recall level and smoke accuracy higher. It remains behind on mAP50-95, where the reference model's redesigned downsampling produces tighter boxes than ours. Neither published model has a false alarm measurement to compare against, which is the point Section 2.3 makes and the reason the comparison in this table can only be made on the left hand columns. Ours fires on 1.2 percent of hard background images and 2.75 percent of open-world images, against 27.0 percent on the hard set for the same architecture trained conventionally. Figure 5 shows the two families on the false alarm against recall plane.

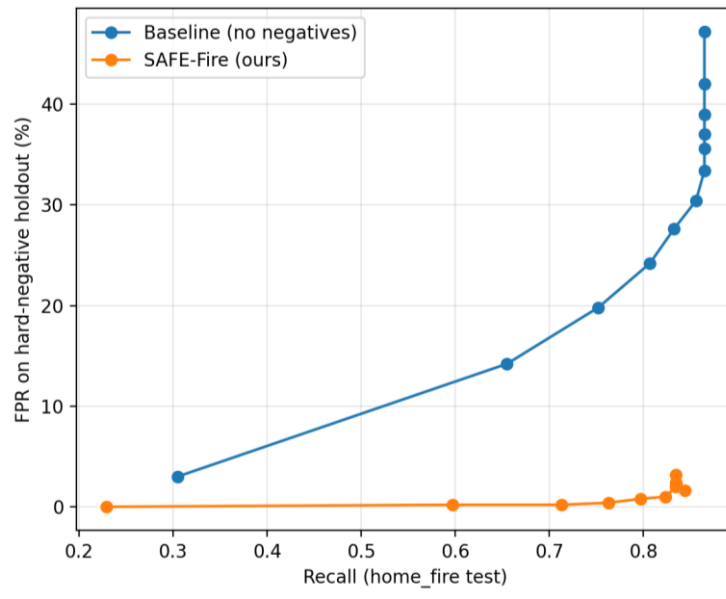


Figure -5 False alarm rate on the hard holdout set against recall on the fire test set.

4.10 Negative Results

Four things were tried and dropped, and we report them because the negative result is useful.

A background suppression term was added to the loss, penalising the maximum detection confidence produced on background images. At a small weight it did nothing measurable. At a larger weight it was unstable across seeds, giving recall at one percent false alarms of 0.724, 0.249 and 0.782 on three runs of the same configuration, so a single run would have looked like a success. The mined background images reach the same goal through the data rather than the loss, and without the instability.

Weight averaging across seeds usually costs nothing and occasionally helps [27], but here it produced a model that detected nothing at all. Averaging the parameters of three independently trained YOLO26 networks collapsed the output entirely. We attribute this to the one-to-one detection head used in this generation, which does not appear to tolerate interpolation between separately converged solutions.

Test time augmentation is not supported by this model family in the version used, and the framework falls back to single scale inference, so it returned results identical to ordinary evaluation. Finally, scaling the teacher from YOLO26s to YOLO26m at the same resolution reduced mAP50 slightly, from 0.9285 to 0.9272, so there is no benefit to be had from extra capacity on a training set of this size.

5. Conclusions

A fire detector that scores well on fire images can still be unusable, and the standard evaluation protocol will not show it. Six models trained conventionally on a public indoor fire dataset all passed 0.90 mAP50 while alarming on a quarter to a third of ordinary images containing no fire.

Adding unlabelled background images to training removes most of that behaviour, but the choice of images decides everything. Random photographs do almost nothing. Images chosen by hand work well on the categories the researcher thought of and leave three times the false alarms on an independent corpus and eight times on open-world images. Mining the images automatically, by running the detector over a large open corpus and using CLIP clustering to select a diverse subset, closes that hole, and using both sources together beats either one. The same failure appears from the recall side: held to one percent false alarms on a set it was not tuned for, the hand-picked model keeps about 0.4 of its recall and the mined model keeps about 0.8.

Distillation from a larger teacher then raised the deployed model from 0.90 to 0.9325 mAP50, matching a published architecture that needs almost three times the parameters and six times the computation, while keeping false alarms near one percent. The strength of the distillation term acts as a usable dial between accuracy and reliability.

Two limits should be read next to those results. The gains are measured on background sets we built ourselves, and although two of the four owe nothing to this work, none of them is a house. The second limit is sharper. The suppression reaches too far in at least one direction, and three small indoor flames that the baseline detected comfortably are pushed below any usable threshold by the mined negatives. A detector that is quiet in a living room and blind to a wastebasket fire at two metres has solved the wrong half of the problem. Closing that gap is where we would put the next effort.

The work reported here is confined to still images and a desktop GPU. Measuring latency, power and accuracy on the embedded target, and combining the detector with heat and gas sensors so that the two modalities support each other, is the subject of ongoing work.

Acknowledgements

The authors thank the Department of Computer Science at the University of Sumer for its support.

Disclosure and conflict of interest

The authors declare that they have no conflicts of interest.

References

- [1] A. Elhanashi, S. Essahraui, P. Dini, and S. Saponara, "Early fire and smoke detection using deep learning: a comprehensive review of models, datasets, and challenges," *Applied Sciences*, vol. 15, no. 18, art. 10255, 2025, doi: 10.3390/app151810255.
- [2] J.-C. Park, M. Kim, S.-M. Choi, and G.-W. Kim, "ESCFM-YOLO: a lightweight dual-stream architecture for real-time small-scale fire smoke detection on edge devices," *Applied Sciences*, vol. 16, no. 2, art. 778, 2026, doi: 10.3390/app16020778.
- [3] L. Xiao, W. Li, X. Zhang, H. Jiang, B. Wan, and D. Ren, "EMG-YOLO: an efficient fire detection model for embedded devices," *Digital Signal Processing*, vol. 156, art. 104824, 2025, doi: 10.1016/j.dsp.2024.104824.
- [4] H. Wang, X. Fu, Z. Yu, and Z. Zeng, "DSS-YOLO: an improved lightweight real-time fire detection model based on YOLOv8," *Scientific Reports*, vol. 15, art. 8963, 2025, doi: 10.1038/s41598-025-93278-w.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- [6] G. Jocher and J. Qiu, *Ultralytics YOLO (Version 8) [Software]*, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [7] W. Pan, X. Wang, and W. Huan, "EFA-YOLO: an efficient feature attention model for fire and flame detection," *arXiv:2409.12635*, 2024, doi: 10.48550/arXiv.2409.12635.
- [8] Y. Jiang, P. Liu, Y. Han, and B. Xiao, "YOLOv11-CHBG: a lightweight fire detection model," *Fire*, vol. 8, no. 9, art. 338, 2025, doi: 10.3390/fire8090338.
- [9] M. F. S. Titu, M. A. Pavel, G. K. O. Michael, H. Babar, U. Aman, and R. Khan, "Real-time fire detection: integrating lightweight deep learning models on drones with edge computing," *Drones*, vol. 8, no. 9, art. 483, 2024, doi: 10.3390/drones8090483.
- [10] C. Li, B. Zhu, G. Chen, Q. Li, and Z. Xu, "Intelligent monitoring of tunnel fire smoke based on improved YOLOX and edge computing," *Applied Sciences*, vol. 15, no. 4, art. 2127, 2025, doi: 10.3390/app15042127.
- [11] I. Karydis and M. Avlonitis, "Smoke detection on the edge: a comparative study of YOLO algorithm variants," *Signals*, vol. 6, no. 4, art. 60, 2025, doi: 10.3390/signals6040060.
- [12] B. Peng and T.-K. Kim, "YOLO-HF: early detection of home fires using YOLO," *IEEE Access*, vol. 13, pp. 79451-79466, 2025, doi: 10.1109/ACCESS.2025.3566907.
- [13] M. Chen et al., "From COCO to COCO-FP: a deep dive into background false positives for COCO

detectors," arXiv:2409.07907, 2024, doi: 10.48550/arXiv.2409.07907.

- [14] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 761-769, doi: 10.1109/CVPR.2016.89.
- [15] Z. Zhang, X. Zhou, and X. Gong, "Learning to calibrate for reliable visual fire detection," arXiv:2502.09872, 2025, doi: 10.48550/arXiv.2502.09872.
- [16] M. M. E. H. Ali and M. Ghodrat, "Toward reliable fire detection in indoor CCTV footage: reducing false alarms from benign flames using 3D attention CNNs," IEEE Access, vol. 13, pp. 144092-144103, 2025, doi: 10.1109/ACCESS.2025.3598919.
- [17] A. Kuznetsova et al., "The Open Images Dataset V4: unified image classification, object detection, and visual relationship detection at scale," International Journal of Computer Vision, vol. 128, pp. 1956-1981, 2020, doi: 10.1007/s11263-020-01316-z.
- [18] P. V. A. B. de Venancio, A. C. Lisboa, and A. V. Barbosa, "An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices," Neural Computing and Applications, vol. 34, no. 18, pp. 15349-15368, 2022, doi: 10.1007/s00521-022-07467-z.
- [19] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: a 10 million image database for scene recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 6, pp. 1452-1464, 2018, doi: 10.1109/TPAMI.2017.2723009.
- [20] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. 38th Int. Conf. Machine Learning (ICML), PMLR 139, 2021, pp. 8748-8763, doi: 10.48550/arXiv.2103.00020.
- [21] S. P. Lloyd, "Least squares quantization in PCM," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137, 1982, doi: 10.1109/TIT.1982.1056489.
- [22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics, vol. 20, pp. 53-65, 1987, doi: 10.1016/0377-0427(87)90125-7.
- [23] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. Int. Conf. Learning Representations (ICLR), 2019, doi: 10.48550/arXiv.1711.05101.
- [24] T.-Y. Lin et al., "Microsoft COCO: common objects in context," in Proc. European Conf. Computer Vision (ECCV), 2014, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.
- [25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015, doi: 10.48550/arXiv.1503.02531.
- [26] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in Advances in Neural Information Processing Systems (NIPS), vol. 30, 2017, pp. 742-751.
- [27] M. Wortsman et al., "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in Proc. 39th Int. Conf. Machine Learning (ICML), PMLR 162, 2022, pp. 23965-23998, doi: 10.48550/arXiv.2203.05482.