COMBINING YOLO V5 AND TRANSFER LEARNING FOR SMOKE-BASED WILDFIRE DETECTION IN BOREAL FORESTS

A.-M. Raita-Hakola[1]*, S. Rahkonen[1], J. Suomalainen[2], L. Markelin[2], R. Oliveira[2], T. Hakala[2], N. Koivumäki[2], E. Honkavaara[2], I. Pölönen[1]

[1] , Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland - (anna.m.hakola)@jyu.fi
[2], Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute (FGI), National Land Survey of Finland (NLS), FI-00521 Helsinki, Finland - (eija.honkavaara)@nls.fi

KEY WORDS: Wildfire Detection, Forest Fire, Smoke Detection, YOLO V5, Transfer Learning, Boreal Forests

ABSTRACT:

Wildfires present severe threats to various aspects of ecosystems, human settlements, and the environment. Early detection plays a critical role in minimizing the destructive consequences of wildfires. This study introduces an innovative approach for smoke-based wildfire detection in Boreal forests by combining the YOLO V5 algorithm and transfer learning. YOLO V5 is renowned for its real-time performance and accuracy in object detection. Given the scarcity of labelled smoke images specific to wildfire scenes, transfer learning techniques are employed to address this limitation. Initially, the generalisability of smoke as an object is examined by utilising wildfire data collected from diverse environments for fine-tuning and testing purposes in Boreal forest scenarios. Subsequently, Boreal forest fire data is employed for training and fine-tuning to achieve high detection accuracy and explore benchmarks for effective local training data. This approach minimises extensive manual labelling efforts while enhancing the accuracy of smoke-based wildfire detection in Boreal forest environments. Experimental results validate the efficacy of the proposed approach. The combined YOLO V5 and transfer learning framework demonstrates a high detection accuracy, making it a promising solution for automated wildfire detection systems. Implementing this methodology can potentially enhance early detection and response to wildfires in Boreal forest regions, thereby contributing to improved disaster management and mitigation strategies.

1. INTRODUCTION

Forests are critically important to our planet. They act as carbon sinks, maintain biodiversity and provide habitats for animal and plant species. Alongside these, forests greatly impact the economy in the form of income, employment and various materials for industry.

Forest fires are devastating, both in terms of human lives, property and the public economy. One of the key factors in reducing forest fires is monitoring-based early detection. An early warning can reduce reaction time and thus reduce fire damage (Xian and Nugroho, 2022). Since the nature of a forest fire is unpredictable (Xue et al., 2022), automated wildfire surveillance methods are raising research interest. Several methods and frameworks have been developed for satellites, area surveillance cameras, wireless sensor networks and unmanned aerial vehicles (UAVs).

Surveillance and detection systems can be combinations of environmental sensors (air temperature, humidity etc.), or be based on image features using object detection and recognition methods (Xue et al., 2022). As mentioned, satellites and other technologies can provide solutions, but UAVs especially allow us to approach real-time solutions. According to (Alexandrov et al., 2019), the development of surveillance and detection methods has been widespread for about 20 years, but the practical use of UAVs has become more common near the 2020s. Similarly, recent studies (Khan et al., 2022, Gaur et al., 2020) note that UAVs are traditionally used as data collectors, but with new platform technology and carefully selected deep-learning methods, we can create systems that collect and analyse data during flights (Xue et al., 2022).

*Corresponding author

1.1 State-of-the-art and challenges of smoke detection

Wildfires can be detected using approaches from environmental parameters to image analysis. However, the first fire signal is typically visual; a small or massive column of smoke appears before visible flames or noticeable changes in distant air temperature (Mukhiddinov et al., 2022). Compared to smoke sensors, besides an alarm, an image can provide visual information about the situation to firefighters (Xue et al., 2022). From a methodological point of view, machine-learning methods and especially deep learning (Convolutional Neural Network based approached (e.g. Faster-CNN) and You Only Look Once (YOLO)) have reached the current state-of-the-art in wildfire detection (Gaur et al., 2020, Alexandrov et al., 2019, Mukhiddinov et al., 2022, Xian and Nugroho, 2022).

A known challenge related to developing deep-learning-based smoke-detection methods is the availability of suitable training data (Gaur et al., 2020). Since the data availability is limited, it is interesting to study how a generalisable object is smoke and do different environmental features affect the model's capability to adapt to different types of nature.

1.2 Smoke-related known challenges

Physically smoke is a combination of gases and airborne solid and liquid particles. Visually, smoke is a wispy object to detect. Smoke appears in different shades of colour, tends to move, change shape, and be transparent or thick; allowing environmental structures to partially or completely hide behind it. Against the sky, smoke may be confused with clouds and, depending on the environment, a surface of a lake can mirror clouds or smoke and cause more confusion.

Compared to traditional objects and detection methods, a house is most likely to be detected as a house, regardless of the surround-

ing visual environment (a desert or Boreal forest). Instead, transparent smoke in the desert can look completely different than in Boreal forests. Since wildfires are a global issue, it is important to study methods that can be generalised to different environments.

1.3 YOLO V5

YOLO V5 is an object detection method which is able to perform fast and accurately in real-time applications. YOLO V5 is an evolution of YOLO V1 to YOLO V3, incorporating iterative enhancements, being relatively close to YOLO V4, which were released close to each other (Jocher et al., 2020, Bochkovskiy et al., 2020, Xu et al., 2021). YOLO V5's improvements have resulted in performance achievements on well-known object detection dataset such as Microsoft COCO (common objects in context) (Lin et al., 2014), but unlike YOLO V4, version releases and evaluations are not published in scientific journals.

As mentioned, the challenges of smoke detection methods relate to data availability and the appearance of the smoke. However, according to the Ultralytics documentation for YOLO (Jocher, 2023) training a YOLO V5 model from scratch requires a minimum of 1500 images and 10 000 instances per class to be accurately labelled and 10% of background images. Ultralytics offers five YOLO V5 models (sizes N,S,M, L and X), where the main difference is the model's size, speed and expected accuracy. The smallest is the fastest, but the achieved accuracy (Mean average precision mAP) is reported to be higher with larger models.

We approached the data-availability challenge by selecting pre-trained YOLO V5 (version 6.0) models in sizes S,M and L which were pre-trained with COCO dataset (Jocher, 2023). These models are trained and tested to recognise 80 object types in their common environments. The COCO data excludes smoke and wildfire scenes.

In this study, we will concentrate on smoke in different environments and see how a YOLO V5 algorithm detects smoke using a transfer learning approach and data sets captured in visually differentiating environments. The aim is to minimise the size of the models and create an approach that could later on easily be deployed in different fire scenes.

1.4 Hypothesis, research questions and expected results

Our hypothesis is that an image-based, pre-trained object detection method for wildfire smoke will need locally collected and annotated data for training in order to generalise regardless of changing environmental features. As a known need exists for training data, we will perform several exams to answer the following questions; *"A. How general object is a wildfire smoke in a visually strongly changing environment?"* and *"B. When we are using a pre-trained model, how much do we need locally collected data to generalise the wildfire model for fire detection in Boreal forests?"*

This study is the first opening of our three-year studies of wildfire surveillance and detection methods, reaching for real-time solutions that can be applied to UAVs. The aim is to develop a smoke-detection-based lightweight model based on YOLO V5 and transfer learning. As outcomes of the first article, we will propose an architecture and methods for locally adaptable Wildfire detection YOLO V5 models and set the baselines for the data acquisition and annotation needs for fine-training the models to adapt to the new environments. We will perform the tests with novel RGB data collected with UAV flights in the summer of 2022 in Finland from five forest restoration burnings. This new Boreal forest fire data and its annotations will be released later.

The paper is organised as follows. After the introduction, section 2. introduces the methods and data. The results and discussion are in Sections 3. and 4. and the paper is finalised in the conclusions in Section 5.

2. MATERIAL AND METHODS

At first, the YOLO V5 transfer learning framework is introduced. In the following subsections, we will seek experiments A and B with methodological and data-related details.

2.1 YOLO V5 architecture

YOLO architecture is often described as a structure containing backbone, neck and head. Figure 1 is a heavily simplified illustration of the structure and its main features, explained as follows.

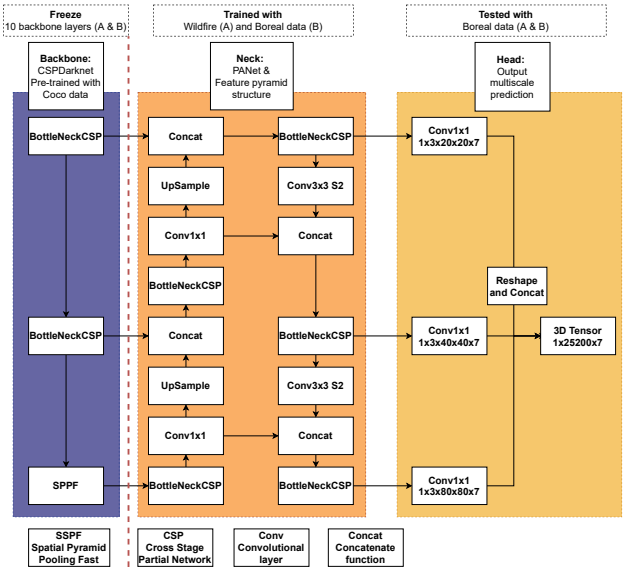

Figure 1: The YOLO v5 transfer learning architecture. A YOLO model can be divided into backbone, neck and head structures. Our approach uses the weights of a pre-trained model by freezing the backbone and fine-training the model starting from the neck.

The YOLO V5 architecture incorporates Cross Stage Partial Network (CSPNet) Darknet (CSPDarknet53) as its backbone, which effectively addresses the issue of repetitive gradient information in large backbones (Nepal and Eslamiat, 2022). By integrating gradient changes into the feature map, it achieves benefits such as improved inference speed, increased accuracy, and reduced model size. The architecture structure has bottlenecks and skip connections, ensuring the efficiency (Xu et al., 2021, Nepal and Eslamiat, 2022).

To enhance information flow, YOLO V5 utilises a path aggregation network (PANet) as its neck. PANet includes a feature pyramid network (FPN) with both bottom-up and top-down layers. This FPN improves the propagation of low-level features and enhances object localisation accuracy in lower layers (Nepal and Eslamiat, 2022, Jocher et al., 2020).

Spatial pyramid pooling fast (SPPF) eliminates the fixed size constraint of the network, being part of the backbone (Figure 1 and Table 1). Upsample (Table 1) is employed for upsampling the previous layer fusion using nearest neighbour interpolation, while Concat (Figure 1) is a slicing layer used to segment the previous

layer. The last three Conv layers serve as detection modules located in the network's head.

Similar to its predecessors Yolov4 and Yolov3, Yolov5's head generates three sets of feature maps for multi-scale predictions. This enables the efficient detection of objects of varying sizes (Nepal and Eslamiat, 2022, Jocher et al., 2020, Xu et al., 2021).

Table 1 reveals the details of layers and parameters used in experiments A and B. *Conv* represents a convolution layer and module *C3* consists of three cascaded convolution layers with different bottlenecks. *Anchors* are pre-defined bounding boxes with specific heights and widths. These capture the scale and aspect ratio characteristics of distinct classes of objects that are observable in images. The Anchors in Table 1 are anchors for the COCO dataset, being re-evaluated for the smoke images automatically within the YOLO V5 implementation (Jocher, 2023).

Architecture and parameters, pre-trained models			
Parameters	Model S	Model M	Model L
Classes	80	80	80
Model depth multiple	0.33	0.67	1.0
Layer channel multiple	0.50	0.75	1.0
Anchors			
P3/8	[10,13, 16,30, 33,23]		
P4/16	[30,61, 62,45, 59,119]		
P5/32	[116,90, 156,198, 373,326]		

YOLOv5 v6.0 backbone	
Layers	[from, number, module, args]
0-P1/2	[-1, 1, Conv, [64, 6, 2, 2]]
1-P2/4	[-1, 1, Conv, [128, 3, 2]]
	[-1, 3, C3, [128]]
3-P3/8	[-1, 1, Conv, [256, 3, 2]]
	[-1, 6, C3, [256]]
5-P4/16	[-1, 1, Conv, [512, 3, 2]]
	[-1, 9, C3, [512]]
7-P5/32	[-1, 1, Conv, [1024, 3, 2]]
	[-1, 3, C3, [1024]]
9	[-1, 1, SPPF, [1024, 5]]
YOLOv5 v6.0 head	
Layers	[from, number, module, args]
0-P1/2	[-1, 1, Conv, [64, 6, 2, 2]]
	[-1, 1, Conv, [512, 1, 1]]
	[-1, 1, nn.Upsample, [None, 2, 'nearest']]
Cat backbone P4	[[-1, 6], 1, Concat, [1]]
	[-1, 3, C3, [512, False]]
	[-1, 1, Conv, [256, 1, 1]]
	[-1, 1, nn.Upsample, [None, 2, 'nearest']]
Cat backbone P3	[[-1, 4], 1, Concat, [1]]
17 (P3/8-small)	[-1, 3, C3, [256, False]]
	[-1, 1, Conv, [256, 3, 2]]
Cat head P4	[[-1, 14], 1, Concat, [1]]
20 (P4/16-medium)	[-1, 3, C3, [512, False]]
	[-1, 1, Conv, [512, 3, 2]]
Cat head P5	[[-1, 10], 1, Concat, [1]]
23 (P5/32-large)	[-1, 3, C3, [1024, False]]
Detect(P3, P4, P5)	[[17, 20, 23], 1, Detect, [nc, anchors]]

Table 1: The architecture, layers and parameters of the pre-trained Ultralytics YOLO V5 version 6.0 S-M models. Upper part describes the parameters of the S,M and L models and COCO dataset's anchors. The lower part describes the network architecture as it is described in the Ultralytics implementation.

To be short, the image undergoes feature extraction using CSP-Darknet53, followed by feature fusion using PANet. The final results are presented as 3D feature tensor, produced by the head layer.

2.2 YOLO V5 transfer learning approach

As mentioned, YOLO models require a high amount of images. Those need to be annotated frame-by-frame, which is time-taking and expensive. There are several tools to work with, some of those are automated, but since the partly transparent smoke is a difficult object to border with confusion possibilities to clouds etc., the annotation process may need to be supervised frame by frame by a human. Therefore, we are using pre-trained models and a transfer learning technique to decrease the amount of training and validation data and related frame-by-frame annotation work.

YOLO V5 architecture allows layers to be frozen, which means that the weights of the frozen layers are not changed during the fine-training phase. In our experiments, we froze the backbone, denoted as layers 0-9, and used the wildfire and Boreal forest fire data for generalising the models to detect smoke.

A red dotted line in Figure 1 visualises the frozen line between the backbone and neck layers, enabling the YOLO V5 models to be trained using considerably less training data than a model trained from scratch. Table 2 shows the data specs in experiments A and B, which are used in the neck and head layers in our transfer learning approach.

2.2.1 Parameters, optimisation and workflow The models used hyperparameters which were optimized for YOLO V5 COCO training (details can be obtained from the repository: (Jocher, 2023)). The loss function (Equation (1)) of a YOLO V5 model is a combination of multiple components:

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \tag{1}$$

Objectness loss (L_{obj}) evaluates the confidence of the predicted bounding boxes containing objects. It employs binary cross-entropy loss to measure the agreement between the predicted objectness score and the ground truth label. Classification loss (L_{cls}) performs multi-class object detection, calculating the cross-entropy loss between the predicted class probabilities and the true class labels. The third component, localisation loss (L_{loc}) measures the discrepancy between the predicted bounding box coordinates and the ground truth box coordinates.

According to (Jocher, 2023), notation λ in Equation (1) represents the balance between the three prediction layers P3 (small objects),P4 (medium objects) and P5 (large objects). Objectness, classification and localisation losses mentioned in Equation (1) are balanced as shown for object loss in Equation (2), using predefined weights [4.0,1.0,0.4].

$$L_{obj} = 4.0 L_{obj}^{small} + 1.0 L_{obj}^{medium} + 0.4 L_{obj}^{large} \tag{2}$$

The tests for experiments A and B were repeated several times using an early stopping method to obtain the optimal amount of epochs and batch size, which were eventually set to 120 and 12, respectively.

2.2.2 Data details and visualisations The wildfire smoke dataset was released in 2022 (Dwyer, 2022). The data is annotated for YOLO V5, containing images mainly from desert-like environments and wildfire smoke. All available wildfire smoke data was used in this study; the data was divided into training and validation portions. Figure 2 visualises the different scenes of wildfire smoke data and Boreal forest fire data.

The Boreal forest fire data was captured during four forest restoration burnings in the summer 2022. The restoration burnings took place in Finnish towns Evo (E25.18555556, N61.22805556), Heinola (E26.44250000, N61.30083333), Karkkila (E23.97805556, N60.64222222) and Ruokolahti (E28.92222222, N61.35055556). The weather conditions and locations differed, and the captured data contains both, close-range and long-range videos. The data was collected using unmanned aerial vehicles (drones), and phantom p4 action camera's RBG sensor (4096 x 2160 resolution).

(a)

(b)

Figure 2: Data samples: a) typical scenes in wildfire smoke data and b) typical scenes in Boreal forest fire data. The environmental difference is noticeable between these smoke object detection data sets.

Experiment A	
Classes	"smoke", "big smoke"
Models	S | M | L
Model size	14 MB | 41 MB | 81 MB
Layers	157 | 212 | 267
Parameters	7 012 822 | 20 852 934 | 46 108 278
Training data	wildfire flame
Training data class	"smoke"
Background images	10 %
Training size	590
Validation size	147
Test data	Ruokolahti
Test data classes	"big smoke" | "smoke"
Test data size	101 | 101
Experiment B	
Classes	"smoke", "big smoke"
Trials	1 | 2 | 3 | 4 | 5|
Models	S | S | S | S | S |
Training data	Boreal forest fire
Training data classes	"big smoke" | "smoke"
Background images	10 %
Training size	390 | 700 | 1010 | 1320 | 1630
Validation size	159 | 159 | 159 | 159 | 159
Test data name	Ruokolahti
Test data classes	"big smoke" | "smoke"
Test data size	101

Table 2: Experiments A and B data and model details. The tests were run separately for "big smoke" and "smoke" annotations.

In experiment A, the models S, M and L were fine-trained using wildfire smoke data and the models were tested using newly collected Finnish Boreal forest fire data. The aim was to examine how generalisable object is smoke in different environments and compare the performance of S, M and L models.

Experiment B utilised different amounts of Boreal forest fire data for training S models. The training data was collected from three different flight campaigns and the test data was the same as in experiment A (Ruokolahti). Experimental details are in Table 2.

2.2.3 The data pre-processing The RGB images were converted from mp4 format to .JPG images using a Python script. The script converted every 48th frame (approximately one frame per two seconds) from the videos. Images were manually checked. Any image containing artificial effects, General Data Protection Regulation (GDPR) related elements (identifiable humans, car register plates or homes, etc.) were removed. The remaining data were manually annotated for two classes "big smoke" and "smoke" using makesense.ai web tool (makesense, 2023). Big smoke surrounded the whole smoke cloud, smoke was a collection of smaller smoke elements, containing no visible background elements. Since the smoke is the first sign of a fire, the flames were left unannotated.

Ruokolahti data was excluded from the training and validation. The test data for all experiments of A and B were selected randomly. The amount of test data was 101 images and it contained 10% images with no smoke ("background" in Table 2).

Training sets of experiment B were randomly selected from Evo, Heinola and Karkkila images. The amount of data in different models is denoted in Table 2. Training data contained 10% background frames.

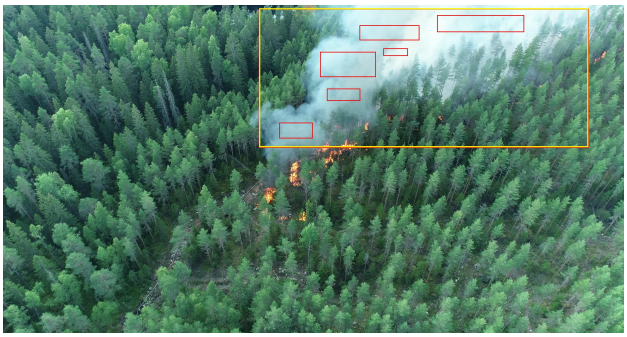

Figure 3: Boreal forest fire data annotation example. Yellow bounding box: "big smoke", red boxes: "smoke".

An annotation example is shown in Figure 3. The yellow bounding box represents annotations with "big smoke" labels, one instance per image, and the red boxes are "smoke" annotation instances. The wildfire data in experiment A was annotated similarly to our "big smoke". The tests were performed twice for each experiment A model; the results were obtained using "big smoke" and "smoke" annotations for all S, M and L size models. Experiment B performed using both classes. The five models of the five trials were eventually two-class classifiers for "smoke" and "big smoke".

3. RESULTS

The results are introduced experiment-wisely. We obtained the results using Python programming language, pre-trained Ultralytics PyTorch version of YOLO v5 (6.0) (Jocher et al., 2020). The results were computed using 28 core Linux server for non-parallel computing, x86_64.

3.1 Experiment A

Based on the results of experiment A, the proposed transfer learning approach seemed to work. Figure 5 shows an example of the training and validation losses. Subfigure Figure 4a and Table 3 confirm that all of the S, M and L models were capable of generalising over the *wildfire smoke validation data.*

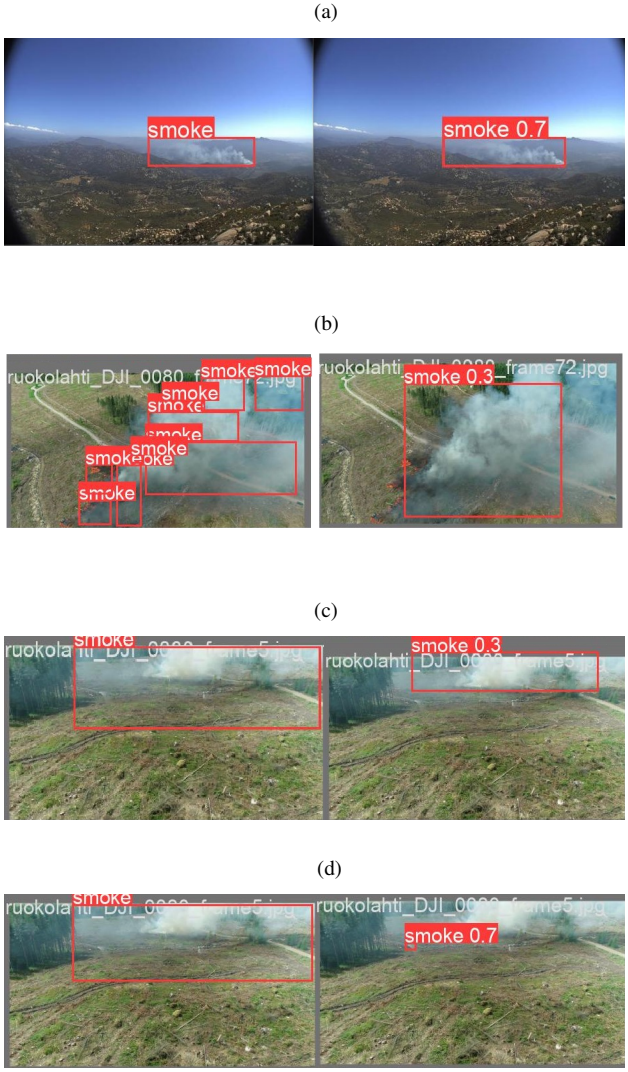

Figure 4: Examples of experiment A predictions: a) M model, trained with wildfire smoke data, prediction dta: wildfire smoke (validation result). b) S model, trained with wildfire smoke data, tested with Ruokolahti and ”smoke” annotations. c) M model, trained with wildfire smoke data, tested with Ruokolahti and ”big smoke” annotations. d) L model, trained with wildfire smoke data, tested with Ruokolahti and ”big smoke” annotations.

However, as expected the models were not able to generalise well with *Boreal forest fire* data. Figures 4b and 4c visualise examples of the test data prediction bounding boxes. All models struggled with ”smoke” annotations, performing better with ”big smoke” annotations. The size of the model did not have a meaningful impact on the results, resulting to perform experiment B using the S model, which was the fastest to train (Table 4) and achieved the best results with Ruokolahti data and ”big smoke” annotations (Table 3).

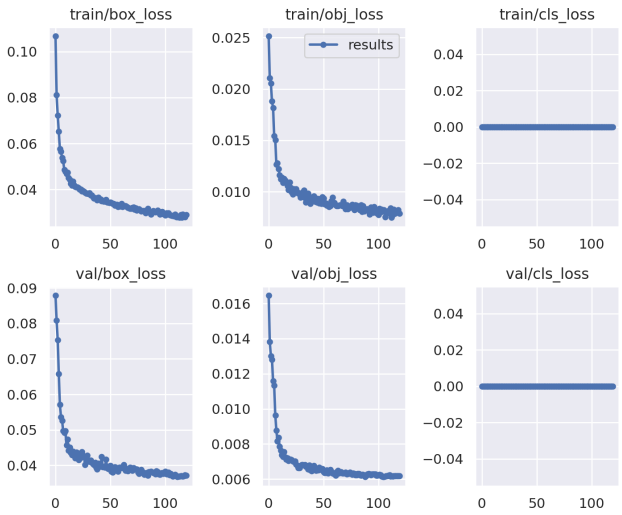

Figure 5: Experiment A, training and validation losses of the S model. Results confirm that the model was able to generalise for the wildfire validation data. Models S, M and L loss curves behaved similarly.

Validation results **Wildfire, 147 images, 147 ”smoke” instances**				
	Precision	Recall	mAP95	mAP50-95
Model S	0.90	0.88	0.92	0.49
Model M	0.94	0.93	0.93	0.51
Model L	0.93	0.92	0.93	0.51
Test results **Ruokolahti, 101 images, 794 small ”smoke” instances**				
	Precision	Recall	mAP95	mAP50-95
Model S	0.05	0.06	0.03	0.008
Model M	0.041	0.035	0.022	0.0055
Model L	0.027	0.14	0.017	0.0044
Test results **Ruokolahti, 101 images, 101 ”big smoke” instances**				
	Precision	Recall	mAP95	mAP50-95
Model S	0.25	0.15	0.094	0.019
Model M	0.0072	0.05	0.0042	0.0009
Model L	0.031	0.020	0.017	0.0035

Table 3: Wildfire training data data: 590 images, validation data: 147 images, 147 ”smoke” instances. Ruokolahti test data: 101 images, with 101 ”big smoke” instances 794 ”smoke instances”.

Training time	
Model S	1.39 h
Model M	4.51 h
Model L	5.72 h

Table 4: The training times of experiment A models. Model S was trained faster than the larger models, using a similar amount of training and validation data.

3.2 Experiment B

This experiment was performed fine-training the S model five times (trials 1-5), using different amounts of Boreal forest fire training data (details in Table 2). The results were obtained for both annotations. Similarly, as in experiment A, the backbone was frozen. The training and validation losses had similar trend

than the models of the experiment A, confirming that the models were not noticeably overfitting.

Test results Boreal forest fire, 794 "smoke" instances				
	Precision	Recall	mAP95	mAP50-95
Trial 1	0.15	0.083	0.048	0.012
Trial 2	0.17	0.096	0.061	0.016
Trial 3	0.29	0.10	0.088	0.021
Trial 4	0.33	0.091	0.085	0.021
Trial 5	0.24	0.11	0.074	0.019
Test results Boreal forest fire, 101 "big smoke" instances				
	Precision	Recall	mAP95	mAP50-95
Trial 1	0.83	0.76	0.77	0.43
Trial 2	0.91	0.80	0.81	0.44
Trial 3	0.90	0.78	0.80	0.48
Trial 4	0.92	0.76	0.78	0.50
Trial 5	0.94	0.78	0.80	0.50

Table 5: All five trials were performed using frozen backbone and Boreal forest fire data. During the trials, the amount of training data was increased by 310 images from 390 (Trial 1) to 1630 images (Trial 5). The results were obtained using the same Ruokolahti test data as in experiment A

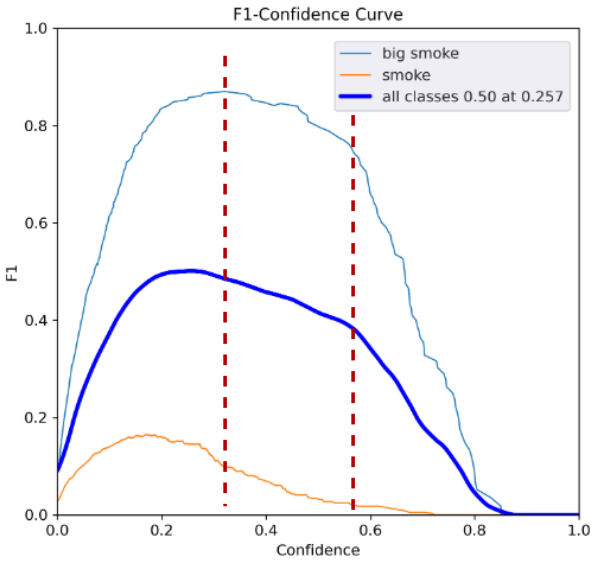

Figure 6: A visualisation of experiment A, Trial 5 results. The relation between F1-score and confidence. F1 score can be interpreted as the balance between precision and recall. Red lines show the optimal confidence threshold levels; the highest F1 results can be achieved using a 0.3-0.4 confidence level, while a threshold close to 0.6 may be an optimal choice, being not too strict. Being over 0.6, the F1 score will decrease. The most interesting curve is visualised with light blue, representing the "big smoke annotations."

As we see from the results, Trial 5 achieved the highest scores. Figure 6 visualises Trial 5 F1-Confidence results. The light blue curve visualises the "big smoke" class results, the orange "smoke" class results and the dark blue line both classes. Numerical results in Table 5 and orange curve in Figure 6 confirms that the "smoke" annotation is not suitable for smoke detection. Annotations that covered the entire smoke area could be detected remarkable better than the partial smoke annotations. The model's performance evaluation in F1-Confidence curve visualisation show that the level of confidence reached the best results between 0.3 and 0.4, but the level could be set up to 0.6. An example of a test image with drawn prediction bounding boxes and confidence levels is in Appendix A.

4. DISCUSSION

4.1 Annotations

Since YOLO annotations are lines in a text file (class info and bounding box coordinates), it is easy to use a Python script to extract the wanted annotation type for each test. Therefore, a single image can be annotated several ways, and the annotation style can be selected and the results separately evaluated during the implementation.

We tested two approaches, "smoke" and "big smoke". "Smoke" annotation was partial (shown in Figure 3). The idea was to include only samples smoke into the smaller bounding boxes. Each image contained several instances of "smoke". "Big smoke", which performed noticeable better, was a large rectangle, which contained the whole smoke cloud and some parts of the surroundings. Annotations of the Boreal Forest fire images were performed hand-drawn. As a full time job, it took approximately one week for 900 images to be annotated when the annotation goal was set to 25 images per hour. This annotation was for two classes (approx 9-10 bounding boxes per one image), indicating that simple one class "big smoke" annotation could be reasonable faster task to perform.

4.2 Numerical results

All models in both experiments were trained using the proposed transfer-learning approach. Figure 5 is an example of training and validation losses. The visualisation is from experiment A (S model), but all the rest training and validation processes followed similar trend, without alerting signs of overfitting. The validation results of experiment A (test and validation data: Wildfire smoke) are high. The model seemed to learn detect smoke objects. However, the tests with Ruokolahti data returned low results. As main outcome, the model S was selected to be used in the experiment B, since it performed relatively well agains the larger models and it was the fastest to train. Another result indicated the annotation style differences; the "big smoke" results outperformed "smoke" ones.

Experiment B used the transfer-learning method and Boreal forest fire data. Five S-models were trained using different amounts of training data. The increment between models was 310 images. "Smoke" conditions were very difficult to detect, even though the model was trained to identify two smoke classes. Trials 4 and 5 were the most successful, but the accuracy metrics were very low.

The results for "big smoke" were promising. High accuracy and recall scores were achieved in all trials, and mAP scores were good. In the first trial, the model trained with 390 images achieved a precision score of 0.83 and a recall score of 0.76. In the best-performing trial 5, the model trained with 1630 images achieved a precision score of 0.94 and a recall score of 0.78.

As expected, the results improved as the amount of data increased. The need for more than 10 000 images for good results does not apply in our transfer-learning approach. As an example, good results can be obtained using over 350 annotated images, which can be annotated by hand approximately in 7 hours, and faster when using AI-reinforced annotation tool (makesense, 2023). Results over 0.91 precision required 700 "big smoke" annotated images.

Figure 6 visualises the relation between F1-score and confidence. F1 score can be interpreted as the balance between the precision and recall. The confidence threshold has the opposite affect on recall and precision; a confidence threshold increment will cause decreasing recall curve while the precision curve raises up. Therefore, it is useful to evaluate F1 score and evaluate the models' optimal confidence level against it. We can see from the F1-Confidence curve that the optimal confidence threshold value for providing us the maximum F1 score is between 0.3 and 0.4. However, since this threshold might be too strict, the optimal confidence level could be set close to 0.6, which still provides a reasonable F1 score. When exceeding the 0.6 threshold value, the results will decrease fast-pace.

4.3 How general object is a smoke in a visually strongly changing environment?

As expected, smoke seemed to be an object that is difficult to detect in a changing environment. An example in Appendix A (Figure A.1) show that a trained model can detect smoke in it's different colours and densities, but at the same time we can see, that those smaller smoke bounding boxes have low confidence levels.

A "big smoke" instances are easier to detect, and those contains usually smoke and no-smoke features since the smoke does not fill in the whole bounding box. Therefore the model might have been learning the environmental features around the smoke and fine features that can be seen under the transparent smoke. This might partly explain why the models trained with wildfire data did not generalise to Boreal forest environment; visually the desert images differ strongly from the green Boreal forests.

4.4 When we are using a pre-trained model, how much do we need locally collected data to generalise the wildfire model for fire detection in Boreal forests?

In our transfer-learning approach, even the first trial's model was able to generalise well, using only 390 annotated images. However, YOLO models are utilising deep neural networks, which benefits from large amount of data. Based on this experiment, a process where a wildfire detection model is to be trained in a new environment, requires minimum of hundreds to a few thousand locally collected and annotated images, but reasonable less images than a model trained from the scratch. The images of the training data should be captured from different distances, angles, and include some non-smoke background frames.

4.5 In practice

When a applying a YOLO V5 model into a real-time scene, based on these results, it is possible to train a customised model relatively fast. It takes minimum of one UAV flight over the fire scene, a python script to extract jpg images and some annotation work. If the annotation approach is one "big smoke" instance per image, the annotations can be manually performed relatively fast; approximately 50 images per hour. An automated process may decrease the annotation time remarkably. After annotations, it takes one to three hours to train and the model is ready to be implemented into the platform and it's pipeline.

4.6 In future

A forest fire may need days, weeks or even years of attention after the forest fire has been extinguished since the peat may still be invisibly burning under the moss, and the fire may intensify again.

One option for after-surveillance might be the use of UAVs and YOLO models. In an ideal situation, there could be a model that is light enough to be implemented in a platform that can be attached to a drone. This model could be quickly trained using local fire data, captured during the active burning phases.

While developing real-time systems, there are payload, energy and computational capacity limitations to exceed, which need to be solved. The next steps of this study could be model optimisation and a choice of a suitable platform, such as Tiny ML shield. After the components are selected and an optimised model is ready, the system needs to be implemented into a good framework with a pipeline which delivers the results to the end users. All of the mentioned steps require further research.

As earlier mentioned, this is an opening study of a three year project. Besides the upcoming engineering and model optimisation studies, we will later release the annotated Boreal forest fire data to ease the known lack of remotely sensed, annotated forest and wildfire data.

In Finland, Aerial forest fire surveillance has been organised at the request of the authorities by flying clubs and entrepreneurs. The country is covered by 22 flight routes. As the population ages, the number of pilots may decrease from the current level, making autonomous solutions necessary. Large, autonomous fixed-wing UAVs could be one solution to carry out these firefighting flights (Finnish Regional State Administrative Agency, 2023).

4.7 Science and YOLO V5

The equations, architecture description and algorithms of YOLO V5 are not published in a peer-reviewed academic channels. The author of Yolo V5 has developed documentation and explanations in the repository, however, the approach of the repository is more engineering than scientific.

As a disclaimer, the technical and mathematical details of this article relies on the YOLO v5 documentation and scientific articles written by others than the original authors (Jocher et al., 2020) and to the releases of the earlier YOLO versions, which are more accurately presented and reviewed. This lack of peer reviewed theories might leave some space for theoretical misinterpretations, but on the other hand, it does not change on the results and conclusions of this study.

5. CONCLUSIONS

A smoke is a difficult object to detect in a changing environment. A transfer-learning approach, utilising pre-trained YOLO V5 model, can be re-trained using only a few hundreds to a few thousand images to detect smoke, but the images should be collected in a similar environment than the detection environment to be.

In the future, YOLO V5 -based light-weight and fast object detection approaches could serve in UAV-based wildfire surveillance tasks. The system could be relatively fast applied into a new environment or fire scene.

ACKNOWLEDGEMENTS

With warm thanks to our assistants (Waleed Akhtar) for annotating and (Anni Tarvainen) for analysing the literature.

This study is funded by the Academy of Finland (Grant No. 348009 and 346710).

REFERENCES

Alexandrov, D., Pertseva, E., Berman, I., Pantiukhin, I. and Kapitonov, A., 2019. Analysis of machine learning methods for wildfire security monitoring with an unmanned aerial vehicles. In: 2019 24th conference of open innovations association (FRUCT), IEEE, pp. 3–9.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection.

Dwyer, B., 2022. Wildfire smoke dataset. Roboflow universe repository, open dataset BY-NC-SA-4.0 licence. *https://universe.roboflow.com/brad-dwyer/wildfire-smoke/dataset/1*, (1.Apr. 2023).

Finnish Regional State Administrative Agency, 2023. Finnish aerial forest fire surveillance. *https://avi.fi/en/services/government-agencies/guidance-and-advice/aerial-forest-fire-surveillance*, (30.June. 2023).

Gaur, A., Singh, A., Kumar, A., Kumar, A. and Kapoor, K., 2020. Video flame and smoke based fire detection algorithms: A literature review. Fire technology 56(5), pp. 1943–1980.

Jocher, G., 2023. Ultralytics documentation, YOLO V5 Architecture summary, version 6.0. *https://docs.ultralytics.com/yolov5/tutorials/architecture_description/*,(1.Jun.2023).

Jocher, G., Changyu, L., Hogan, A., , L. Y., changyu98, Rai, P. and Sullivan, T., 2020. ultralytics/yolov5: Initial Release. Zenodo. Provided by the SAO/NASA Astrophysics Data System.

Khan, A., Hassan, B., Khan, S., Ahmed, R. and Abuassba, A., 2022. Deepfire: A novel dataset and deep transfer learning benchmark for forest fire detection. Mobile Information Systems.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, pp. 740–755.

makesense, 2023. Annotation tool. www.makesense.ai, an open annotation tool under GPLv3 licence. *https://www.makesense.ai/*, (5.June. 2023).

Mukhiddinov, M., Abdusalomov, A. B. and Cho, J., 2022. A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5. Sensors.

Nepal, U. and Eslamiat, H., 2022. Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. Sensors.

Xian, T. K. and Nugroho, H., 2022. Forest fire detection for edge devices. In: 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), pp. 1–4.

Xu, R., Lin, H., Lu, K., Cao, L. and Liu, Y., 2021. A forest fire detection system based on ensemble learning. Forests.

Xue, Z., Lin, H. and Wang, F., 2022. A small target forest fire detection model based on yolov5 improvement. Forests.

APPENDIX (A)

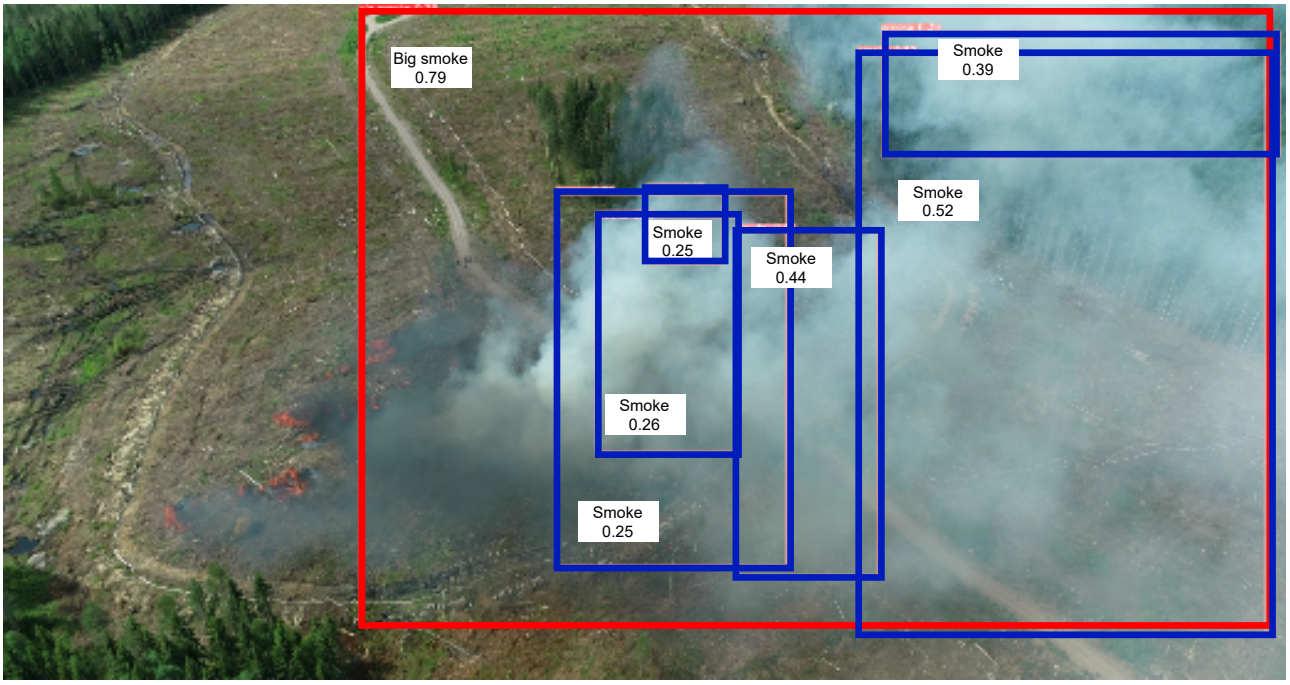

Figure A.1: Experiment B: An example of Trial 5 model's predictions. Red bounding box visualises the "big smoke" and blue boxes "smoke" class predictions. A white label in the boxes represents the confidence levels.