Vol. 1 No. 1 Year. 2023

Hotspots and Smoke Detection from Forest and Land Fires Using the YOLO Algorithm (*You Only Look Once*)

Dicko Andrean [1], Mitra Unik [2], Yoze Rizki*[3]

[1,2,3] Informatics Engineering, Universitas Muhammadiyah Riau
e - mail : [1]dickoandrean@gmail.com, [2]mitraunik@umri.ac.id, *[3] yozerizki@umri.ac.id

Abstract– **The term forest and land fires is used to refer to unplanned, controlled and unwanted fires that destroy vegetated areas and their ecosystems triggered by natural or human causes . Early detection of hotspots can reduce the risk of wider forest and land fires. The use of the Deep Learning YOLO (** ***You Only Look Once*** **) algorithm is carried out to detect fire and also the smoke it produces. This study tested in 3 ways, 1) 1341 after data augmentation (496 original data), 2) 608 after data augmentation (253 original data), and 3) 1790 after data augmentation (746 original data). Detection of fire and smoke objects in the form of design, implementation and testing resulted in the YOLOv4 framework successfully producing high** ***confidence*** **of up to 97% in the second test. Based on the test results in this study, it is known that the image datasets used for training data greatly affect object detection and affect the confidence value. The more diverse the shape of the object from the image datasets, the lower the confidence value obtained.**

Key words: Forest fire; YOLO; Deep Learning; Fire; Image Processing.

I. PRELIMINARY

The contribution of forests to human life is in line with the sensitivity of forests to climate variations [1]–[3]. Together with trends in rainfall precipitation, increasing temperatures and longer dry season periods lead to increased risks of land and forest fires (karhutla), while karhutla feedback has driven climate change itself through modulation of the carbon cycle, greenhouse gas and aerosol emissions [4]–[6]. Meanwhile when vegetation burns, it releases stored carbon into the atmosphere, accelerating global warming and thereby exacerbating conditions which could lead to extreme behavior and result in greater incidents of forest fires in the years to come [3]. Forest fires are common in many places around the world where the climate is humid enough for vegetation to grow, but becomes dry during the summer period so that twigs, leaves and grass become highly flammable. The ecological role of fire in natural ecosystems can be compared to the role of fever as a protective response to human disease. According to their circumstances, both of them can be a normal systemic reaction as a disease prevention process or even deadly, thus the occurrence of small-scale fires that occur in a controlled manner is a natural ecosystem dynamic that is beneficial to the health of vegetation from an ecological point of view. On the other hand, repeated and/or large-scale fire events can prevent the tempo of vegetation standing and suppress forest regeneration so that it can become a major disaster or cause permanent damage to ecosystems [7].

monitoring , control and management of fires requires an understanding of fire interactions, as well as the ecological role of fire and some basic principles of fire [8]behavior . Fire is the rapid oxidation of a material in a chemical combustion process, usually with the evolution of heat and light; the interaction of the three components 1) fuel, 2) heat and 3) oxygen [9]. Fire is a must-have feature of forest fires. Fire in forest fires can be sourced from nature and anthropogenic. Fires of natural origin usually originate from lightning, volcanic eruptions or burning coal with a very small percentage starting with the spontaneous combustion of dry fuel such as wood chips and leaves. Meanwhile, anthropogenic sources come from human activities with a greater percentage such as cigarette fires, recreation, equipment and so on. Fire behavior in forest and land fires has a close relationship with the rate of spread and fire intensity. Fire behavior is influenced by three main components, namely topography, weather and fuel with several variables, both directly and indirectly, such as the type of vegetation, the number of fires and the time of the last fire at that location [10], [11]. While flame is the next stage of fire and is characterized by the rate of spread. Fire behavior is generally defined as the way in which fuel ignites, fire develops, spreads and exhibits other phenomena such as flame swirls [8].

Understanding the characteristics of haze in a fire environment is critical to counteracting problems associated with poor air quality, such as fires on peatlands. The term air pollution usually refers to the transfer of natural and/or synthetic hazardous materials into the atmosphere as a direct and/or indirect consequence to the environment or organisms living in the affected environment. Examples of air pollutants in general are Ozone (O3) , Lead (Pb), Nitrogen Dioxide (NO_2), Particulate Matter (PM), and Dioxide (SO2).

Monitoring hotspots as a preventive measure in areas that are difficult to access can be done by using remote sensing technology (*drones)* or unmanned aircraft [12]. This aircraft is controlled automatically through a computer program. [13]. Drones are used in video image capture, which is then followed by *image enhancement techniques* . In using this technique, fire features can be used to efficiently detect fire such as: fire color, smoke, sparks, fire texture, fire area distribution, and edge detection. Furthermore, the resulting image from *image enhancement* can be trained by utilizing *deep learning algorithms.* namely the YOLO (*You Only Look Once*) V4 algorithm. [13]. The YOLO (*You Only Look Once*) algorithm is a *deep learning algorithm* that utilizes a convolutional neural network (CNN) to detect objects. This algorithm will divide the image into sxs sized *grids which then in each grid* will predict *the bounding box* and the class map of each *grid* . If on one *grid* predicted object, then the *grid* will predict *the bounding box* that surrounds the object. The *confidence value* will be calculated for each *bounding box* which will then be selected based on the value obtained. [14].

II. METHODS

A. Data

image data collection in the form of images containing fire and smoke such as images of fires. Collection of image data or fire images as datasets which function as training data for the deep learning method on the you only look once (YOLO) algorithm. The research data is used to train the You Only Look Once (Yolo) algorithm in detecting fire and smoke. In this study, 746 image data were collected from various sources.

B. Preprocessing

preprocessing design is the preparation of the data that has been collected before the *training stage is carried out.* In *preprocessing* there are several stages. The first stage is *labeling* , and the second is data argumentation, for the two stages.

1 Labeling: In this process hotspots and smoke will be marked on the image data that has been collected by giving a box or *bounding box* and the box will be given a name *(annotations).*

2 Data Augmentation: This step is done in order to significantly increase the diversity of data available for the training model, without actually collecting new data.

C. Training

In this process, the YOLO Training stages will be carried out using datasets. at this stage all image data will be processed and will be studied by YOLO. In the training process, a GPU (Graphic Processing Unit) is needed because it will process a large number of images. This can be done using the help of Google Colab .

D. Deploy YOLO / Python Implementation

At this stage, object detection is carried out by building a program using programming language codes and *libraries* to make it easier to provide programming language codes. In this study the programming language used is the *Python programming language* . And the *library* used is *openCV*

E. Testing

At this stage, tests are carried out in the form of calculating *Average Precision* (AP) *, Training Loss, F-Measure* , and *Sensitivity (Recall).*

Precision is the ratio of the amount of data that is correctly predicted to be positive with the overall results that are predicted to be positive. Precision can be calculated using the equation.[15]

$$Precision = \frac{TP}{FP + TP}$$

Recall is a measurement on data with the correct positive classification. Recall can be calculated using the equation.[15]

$$Recall = \frac{TP}{TP + FN}$$

F1 Score aims to compare the average precision and recall. F1 Score can be calculated using the equation.[15]

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

True Positive (TP) is a condition where the model predicts the data as yes (*TRUE*) and the actual answer is yes (*TRUE*). *False Positive* (FP) is a condition where the model predicts the data as yes (*TRUE*) and the actual answer is no (*FALSE*). *False Negative* (FN), a condition in which the model predicts data as no (*FALSE*) and the actual answer is yes (*TRUE*).[15]

III. RESULTS AND DISCUSSION

A. Preprocessing

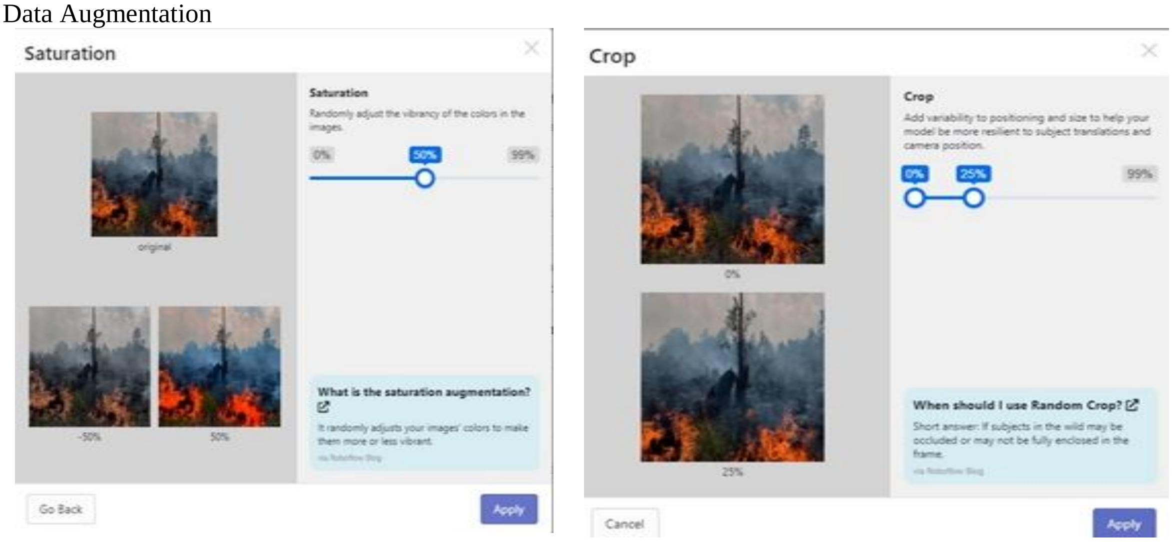
1 Image Labelling

The data that has been collected is then labeled or marked with image data. in this process marks the point of fire and smoke on the image data that has been collected. In this research, the labeling process uses an online datasets generator on the app.roboflow.com website.

Fig 1 Labeling process on datasets using the help of app.roboflow

labeling process was carried out for all datasets which totaled 746 images. The *labeling process* is carried out by determining the points of fire and smoke in the image data through the provision of boxes or *bounding boxes* and *annotations.*

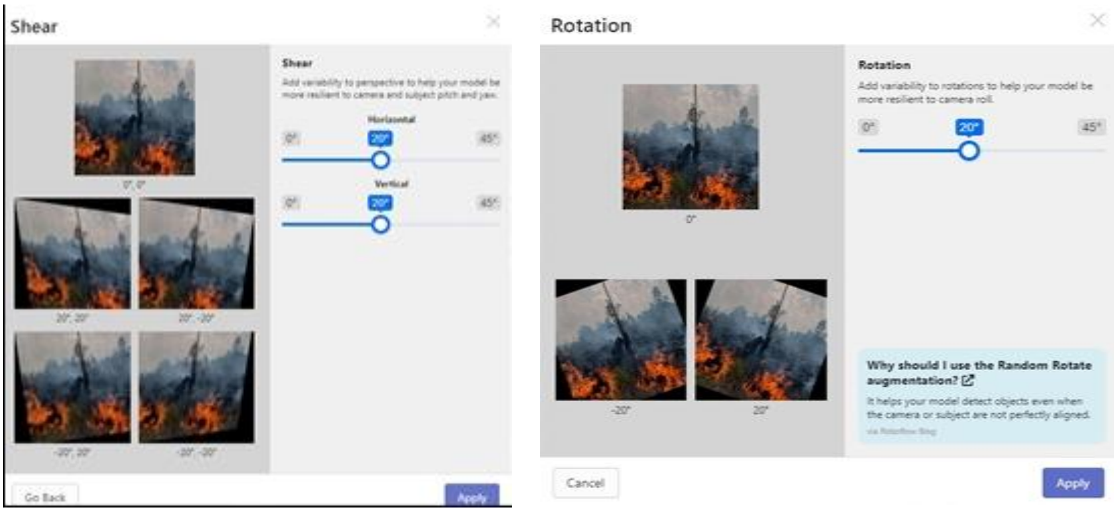
2 Data Augmentation



(a) (b)

Fig 2(a) Data Augmentation Process for Saturation (b)

In fig 2 (a) the data argumentation process is carried out to add and reduce the saturation value by 50% from the original image data so as to produce 2 new image data for each existing image data. Fig 2 (b) Data Argumentation Process The data argumentation process is carried out to cut the original image data by 25% to produce 1 new image data for each existing image data.



(a) (b)

Fig 3(a) Data Augmentation Process for Shear value (b) Data Augmentation Process for Rotation value

Fig. 3 is carried out in the process of data argumentation to change the shape of the original image to a parallelogram by shifting each corner of the original image at horizontal and vertical points by 20 degrees to produce 4 new image data for each existing image data. The data argumentation process for the Rotation 3 (b) value is carried out in the data argumentation process to rotate the original image data by 20 degrees to produce 2 new image data for each existing image data.

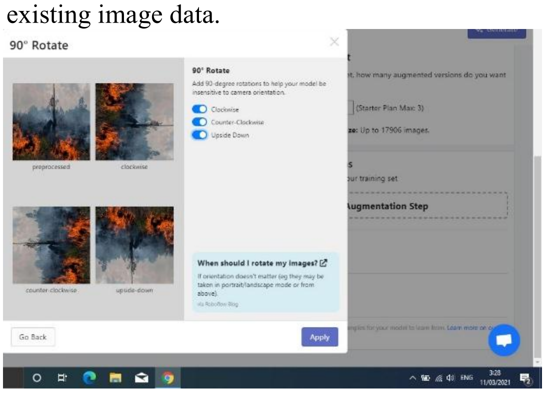


Fig 4 Data argumentation process for 900 Rotate Value

In Fig. 4, the data augmentation process is carried out to rotate left, right, and down the original image data so as to produce 3 new image data for each existing image data.

B. Training Data

Datasets that have been *labeled,* then proceed to the *YOLO Training stage.* In the training process, the YOLO algorithm is trained to detect hot spots and smoke. The *training* process requires a GPU (*Graphics Processing Unit)* with high specifications, therefore the *training process* in this study uses the help of Google Colab . In the training process using the help of Google Colab there are several stages in the form of cuDNN configuration , *Installing* Darknet, extracting *datasets* , *training* configuration , and YOLO *Training* .

1 Configure cuDNN

In the cuDNN configuration on Google Colab , the first step is to check the GPU hardware provided by Google Colab . The following is the syntax for checking Nvidia GPU hardware:

```
!nvidia-smi
```

Fig 4 Syntax check GPU hardware

The output results in Fig 4 will display the type of GPU obtained from Google Colab . Then cuDNN will be configured according to GPU type using the following syntax:

```
%env compute capability=75
```

Fig 5 CuDNN configuration syntax

In Fig 5, the cuDNN configuration has a value of 75, this value can be changed according to the type of GPU found on Google Colab .

2 Darknet Installing

Before performing the darknet installation, the process of creating a darknet folder is carried out as a place to accommodate darknet *clone files* using the following syntax:

```
%cd /content/drive/MyDrive/yolotrain/v5.1
%rm -rf darknet
```

Fig 6Darknet folder creation syntax

Darknet files can be *cloned* using the following syntax:

```
!git clone https://github.com/roboflow-ai/darknet.git
```

Fig 7Darknet file cloning syntax

After cloning darknet, the process of creating a " MakeFile " can be done using the following syntax:

```
%cd /content/drive/My Drive/yolotrain/v5.1/darknet
%rm Makefile
```

Fig 8Syntax for Creating a Makefile

For the installation process of darknet files that have been cloned, a compilation process is carried out with the following syntax:

```
%cd darknet/
!sed -i 's/OPENCV=0/OPENCV=1/g' Makefile
!sed -i 's/GPU=0/GPU=1/g' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/g' Makefile
!sed -i "s/ARCH= -gencode arch=compute_75,code=sm_75" Makefile
!make
```

Fig 9Darknet compilation syntax

3 Data Training Configuration

In the training configuration, the number of epochs is determined. In this study, the epochs were set at 4000 epochs. The training configuration is carried out using the following syntax:

```
...
num_classes = file_len('train/_darknet.labels')
max_batches = num_classes*2000
...
```

Fig 10Training configuration syntax

To write down the number of *epochs* , *you* can see in Figure 4.15 it is explained that *num _ classes* are meant to mean that the number of objects detected on this number is 2 objects, namely fire and smoke objects which are then multiplied by 2000 to produce 4000 *epochs*

4 YOLO Training

To do YOLO V4 training, use the following syntax:

```
!./darknet detector train data/obj.data cfg/custom-yolov4-tiny-detector.cfg yolov4-tiny.conv.29 -dont_show -map
```

Fig 11 YOLO V4 Training Syntax

In this study, the YOLO V4 training process took approximately 1 hour to 2 hours

C. Deploy Yolo/ Python Implementation

After the training process and obtaining the files in the form of cfg , weights and coco.names , the next step is to generate programming code using the Python language to implement the YOLO V4 algorithm in detecting fire and smoke objects through video captured by cameras. In implementing python the first thing to do is to import the opencv and numpy libraries . Video image is used. In Figure 4.18 is the code for video capture, Source code pad Fig 12 with a value of 0 is the value for real time video capture using a camera device on a PC or laptop. The value 0 can be replaced using the file directory path to retrieve video via a video file or can be replaced with an IP address if using an IP webcam.

```
cap = cv2.VideoCapture(0)
```

Fig 12 video capture source code

Next, the class name label extraction is performed. Extract class name label. In this study, we will extract the class name label for the object of fire and smoke. The output results from the YOLO V4 training process in the form of cfg files and weights will be retrieved using the source code in Fig 13.

```
cfg = './cfg/custom-yolov4-tiny-detector.cfg'
weights = './backup/custom-yolov4-tiny-
detector_best.weights'
net = cv2.dnn.readNetFromDarknet(cfg,weights)
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)
```

Fig 13 loads YOLO V4 source code

Next, the process of detecting fire and smoke objects is carried out as well as providing a *bounding box* to the detected object and calculating the *confidence value.* After the object detection process is carried out and the bounding box is given to the detected object, the video that has been detected is displayed.

D. Testing

In this study, 3 test models were carried out with the following details:

Table 1Test Model

Model	**Datasets**	**Number of Training Data**	**Number of Testing Data**	**Information**
Testing 1	1341 after data augmentation (496 original data)	1266	25	The data contains images of forest fires
Test 2	608 after data augmentation (253 original data)	531	25	The data contains images of candle flames
Test 3	1790 after data augmentation (746 original data)	1566	75	The data contains images of forest fires and candle flames

1 Testing using *datasets* totaling 1341 after augmentation of forest fire image data.

In testing the YOLO V4 algorithm, a *training process was carried out* using 1341 *datasets after image data augmentation where the data contained images of forest fires.* the following results were obtained:

Table 2

YOLO V4 test results Using datasets 1341 after augmentation of forest fire image data

Parameter	Results
True Positive (TP)	49
False Positives (FP)	64
False Negatives (FN)	106
Precision	0.433628
recall	0.316129
F1-Score	0.365671

Then the *average value* the *loss* obtained from the *training datasets* 1341 after augmenting the forest fire image data is as follows:

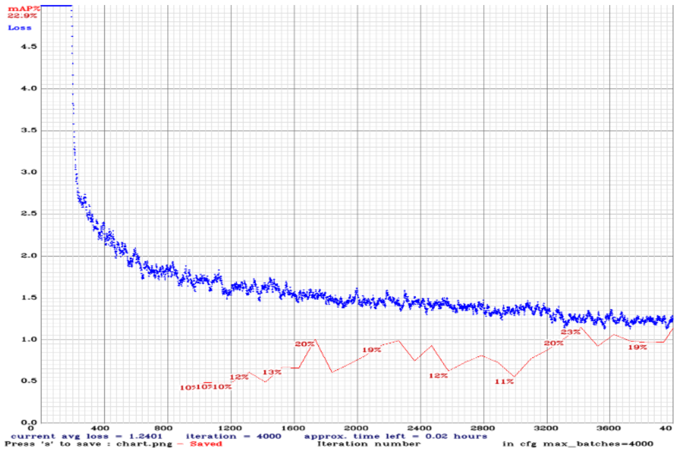


Fig 14The AVG Loss diagram in training uses 1341 datasets after augmenting forest fire image data

Based on the diagram in Fig. 14 it is known that the value of the *Average Loss* in this test is 1.2401. then the best mAP value is found in the epoch between epoch 3200 and epoch 3600 which is 23%.
Furthermore, the YOLO V4 algorithm that has been *trained* using 1341 after augmenting forest fire image data is tested using forest fire videos and *real -time testing* in detecting candle fire points. The following results are obtained:

Fig 15 The results of forest fire video detection testing use datasets 1341 after augmenting forest fire image data

In testing using forest fire videos, the algorithm can detect fire and smoke objects and has a fairly high *confidence value* , reaching a value of 50% in detecting forest fire fire objects.

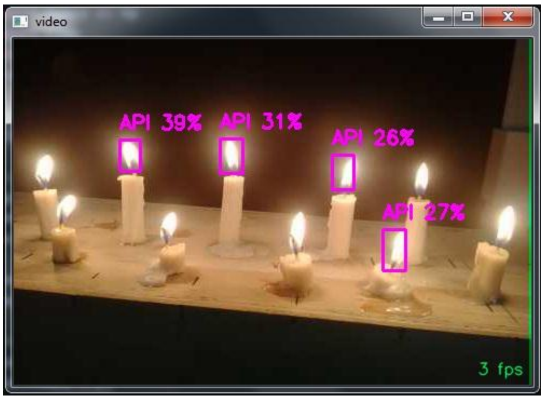


Fig 16 The results of real-time candle fire detection tests use datasets 1341 after augmenting forest fire image data

In the candle fire detection test using 1341 *datasets* after augmenting forest fire image data, the YOLO V4 algorithm is able to recognize hotspots on candles, but not all hotspots are detected, as shown in Fig. 16 only a few hotspots are detected and also the resulting *confidence value* quite low, namely below 40%.

2 The test uses 608 datasets after augmenting the candle flame image data
Subsequent tests were carried out by testing the YOLO V4 algorithm which had been carried out in the *training process* using 608 *datasets* after augmenting the fire image data on candles. The following results were obtained:

Table 3
YOLO V4 test results Using 608 datasets after the augmentation of the fire image data on candles

Parameter	**Results**
True Positive (TP)	85
False Positives (FP)	22
False Negatives (FN)	28
Precision	0.794392
recall	0.752212
F1-Score	0.772727

Then the *average loss value* obtained from the results of the 608 *training datasets* after the augmentation of the candle flame image data is as follows:

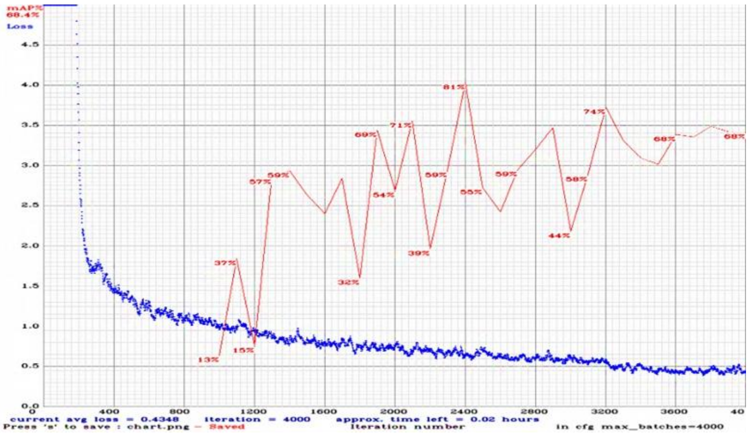


Fig 17 AVG Loss diagram in training using datasets 604 after augmentation of forest fire image data

Based on the diagram in Figure 17 it is known that the value of the *Average Loss* in this test is 0.4348. then the best mAP value is found in epoch 2400 which is 81%.
Furthermore, the YOLO V4 algorithm which has been *trained* using 608 after augmentation of forest fire image data is tested using forest fire videos and *real -time testing* in detecting candle fire points. The following results are obtained:

Fig 18 The results of forest fire video detection testing use datasets 608 after augmenting forest fire image data

In testing using *datasets* 608 after the augmentation of candle flame image data as *training data* , in detecting hotspots in forest fire videos the YOLO V4 algorithm has not been able to recognize hotspots in forest fires. as seen in Figure 18 there is no *bounding box* given.

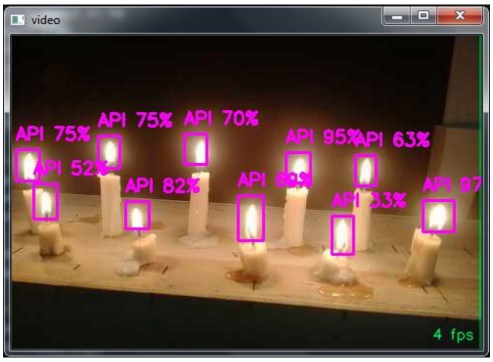


Fig 19 The results of real-time candle fire detection tests use datasets 608 after augmenting forest fire image data

In testing using the *608* dataset after augmenting the candle flame image data as *training data* , in the detection of hot spots in candle flame videos the YOLO V4 algorithm is able to recognize hotspots very well. as seen in Fig. 19 all the fire points on the candle can be detected and have a high *confidence value* of up to 97%.

3 The test uses datasets totaling 1790 after augmenting image data of forest fires and candle fires

Subsequent tests were carried out by testing the YOLO V4 algorithm which had been carried out in the *training process* using the *1790* dataset after augmenting the image data of forest fires and fire on candles. The following results were obtained:

Table 4
YOLO V4 test results Using 1790 datasets after
image data augmentation of forest fires and fire on candles.

Parameter	**Results**
True Positive (TP)	141
False Positives (FP)	67
False Negatives (FN)	166
Precision	0.677885
recall	0.459283
F1-Score	0.547572

Then the *average loss value* obtained from the results of the 1790 *training datasets* after augmenting the image data of forest fires and images of candle flames is as follows:

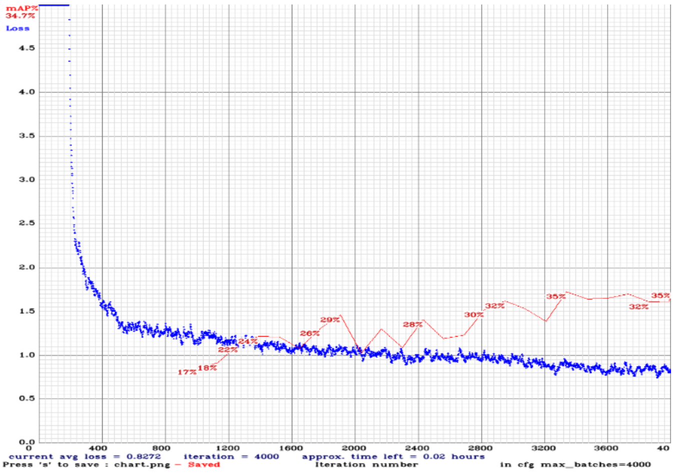


Fig 20 AVG Loss diagram in training using datasets 1790 after augmentation of image data of forest fires and candle flames

Furthermore, the YOLO V4 algorithm which has been trained using 496 image data of forest fires is tested using forest fire videos and testing in real time in the detection of candle fire points. The following results are obtained:

Fig 21 The results of forest fire video detection testing using datasets 1790
after augmenting forest fire image data

In testing using the *1790* dataset after augmenting image data of forest fires and candle fires as *training data* , in detecting fire spots in forest fire videos the YOLO V4 algorithm is able to recognize hotspots but has a fairly low *confidence value* , which is below 40%.

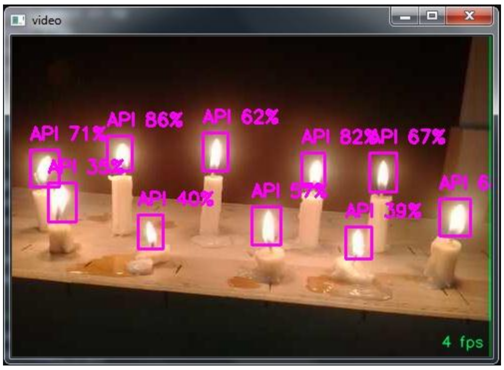


Fig 22 The results of real-time candle fire detection tests use the 1790 dataset
after augmenting forest fire image data

In testing using the 1790 datasets after augmenting image data of forest fires and candle fires as training data, in detecting hotspots in video candle flames the YOLO V4 algorithm is able to recognize points of fire on candles with a fairly high confidence value, but the confidence value in this test is not very high. high when compared to the 2nd model test.

IV. CONCLUSION

After conducting research related to the detection of fire and smoke objects in the form of design, implementation and testing of applications that have been made, the following conclusions can be drawn:

1 Prevention of forest and land fires can be done by monitoring hotspots using the *You Only Look Once* (YOLO) V4 algorithm to detect objects of fire and smoke because the YOLO V4 algorithm has good results. However, in the process the YOLO V4 has a few problems when it detects in *real time* , namely the *delay* in video processing from the camera, this is because the specifications of the PC / Laptop used in this study are not capable enough to do the *rendering process* .

2 YOLO V4 is an algorithm that is quite effective in detecting hotspots and smoke that occurs in forest and land fires, because YOLO V4's ability to detect hotspots and smoke has good results and detection can be done in real time.

3 Based on the test results in this study, it is known that the image datasets used for *training data* greatly affect object detection and affect the confidence value. The more diverse the shape of the object from the image datasets, the lower the confidence value obtained. .

V. BIBLIOGRAPHY

[1] N. A. Ulya, E. A. Waluyo, S. Lestari, and B. T. Premonoi, “Peat Swamp Forest Degradation: Impacts, Affected Communities and Losses,” *E3S Web of Conferences*, vol. 68, no. 1, pp. 1–7, 2018, doi: 10.1051/e3sconf/20186803007.

[2] G. Dieterle, “Sustainable Forest Management in a Changing World: a European Perspective,” *Agriculture*, vol. 19, pp. 83–91, 2009, doi: 10.1007/978-90-481-3301-7.

[3] M. D. Flannigan, B. D. Amiro, K. A. Logan, B. J. Stocks, and B. M. Wotton, “Forest fires and climate change in the 21ST century,” *Mitig Adapt Strateg Glob Chang*, vol. 11, no. 4, pp. 847–859, 2006, doi: 10.1007/s11027-005-9020-7.

[4] B. Langmann, B. Duncan, C. Textor, J. Trentmann, and G. R. van der Werf, “Vegetation fire emissions and their impact on air pollution and climate,” *Atmos Environ*, vol. 43, no. 1, pp. 107–116, 2009, doi: 10.1016/j.atmosenv.2008.09.047.

[5] B. Berwyn, “How Wildfires Can Affect Climate Change (and Vice Versa),” *INSIDECLIMATE NEWS*, 2018. https://insideclimatenews.org/news/23082018/extreme-wildfires-climate-change-global-warming-air-pollution-fire-management-black-carbon-co2 (accessed Jan. 15, 2020).

[6] Z. A. Holden *et al.*, “Decreasing fire season precipitation increased recent western US forest wildfire activity,” *Proc Natl Acad Sci U S A*, vol. 115, no. 36, pp. E8349–E8357, 2018, doi: 10.1073/pnas.1802316115.

[7] G. Eftychidis, “Forest and Range Fires,” in *Encyclopedia of Natural Hazards*, P. T. Bobrowsky, Ed. Dordrecht: Springer Netherlands, 2013, pp. 346–359. doi: 10.1007/978-1-4020-4399-4_146.

[8] C. M. Countryman, “The fire environment concept,” p. 15, 1972.

[9] NWCG, “Fire,” *NWCG Glossary Of Wildland Fire, PMS 205*. https://www.nwcg.gov/term/glossary/fire (accessed Apr. 01, 2020).

[10] B. L. Estes, E. E. Knapp, C. N. Skinner, J. D. Miller, and H. K. Preisler, “Factors influencing fire severity under moderate burning conditions in the Klamath Mountains, northern California, USA,” *Ecosphere*, vol. 8, no. 5, 2017, doi: 10.1002/ecs2.1794.

[11] L. Syaufina, *Kebakaran hutan dan lahan di Indonesia: perilaku api, penyebab, dan dampak kebakaran*. Bayumedia Pub., 2008.

[12] M. Unik and Sri Nadriati, “Overview: Random Forest Algorithm for PM2.5 Estimation Based on Remote Sensing,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 422–430, Dec. 2022, doi: 10.37859/coscitech.v3i3.4380.

[13] M. L. Nazilly, B. Rahmat, and E. Y. Puspaningrum, “IMPLEMENTASI ALGORITMA YOLO (YOU ONLY LOOK,” *Jurnal Informatika dan Sistem Informasi*, vol. 1, no. 1, pp. 81–91, 2020.

[14] J. S. D. R. G. A. F. Redmon, “(YOLO) You Only Look Once,” *Cvpr*, 2016, doi: 10.1109/CVPR.2016.91.

[15] K. Gunadi, E. Setyati, and J. S. Surabaya, “Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network,” *Jurnal Infra*, vol. 8, no. 1, pp. 295–301, 2020.