energies

Article

Evaluation of Deep Learning-Based Segmentation Methods for Industrial Burner Flames

Julius Großkopf [1,*,†], Jörg Matthes [1,*,†], Markus Vogelbacher [1,†] and Patrick Waibel [1,2,†]

[1] Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany; markus.vogelbacher@kit.edu (M.V.); patrick.waibel@kistler.com (P.W.)

[2] Competence Center Vision Systems, Kistler Group, 76131 Karlsruhe, Germany

* Correspondence: julius.grosskopf@gmail.com (J.G.); joerg.matthes@kit.edu (J.M.); Tel.: +49-1514614130 (J.G.)

† These authors contributed equally to this work.

Abstract: The energetic usage of fuels from renewable sources or waste material is associated with controlled combustion processes with industrial burner equipment. For the observation of such processes, camera systems are increasingly being used. With additional completion by an appropriate image processing system, camera observation of controlled combustion can be used for closed-loop process control giving leverage for optimization and more efficient usage of fuels. A key element of a camera-based control system is the robust segmentation of each burners flame. However, flame instance segmentation in an industrial environment imposes specific problems for image processing, such as overlapping flames, blurry object borders, occlusion, and irregular image content. In this research, we investigate the capability of a deep learning approach for the instance segmentation of industrial burner flames based on example image data from a special waste incineration plant. We evaluate the segmentation quality and robustness in challenging situations with several convolutional neural networks and demonstrate that a deep learning-based approach is capable of producing satisfying results for instance segmentation in an industrial environment.

Keywords: flame segmentation; image instance segmentation; afterburner chamber; combustion process control; multi-fuel swirl burner; industrial automation

Citation: Großkopf, J.; Matthes, J.; Vogelbacher, M.; Waibel, P. Evaluation of Deep Learning-Based Segmentation Methods for Industrial Burner Flames. *Energies* **2021**, *14*, 1716. https://doi.org/10.3390/en14061716

Academic Editor: Francesco Creta

Received: 28 February 2021
Accepted: 16 March 2021
Published: 19 March 2021

1. Introduction

The segmentation of flames is motivated by applications that require geometric information about flames in image or video data, see, e.g., in [1], where the authors use growing fire regions as a criterion to identify dangerous fire in outdoor and indoor scenery. Similarly, [2] use segmentations of burner flame images to derive a geometric stability measure for combustion processes. The flame stability can be used to analyze combustion and optimize process control. The procedure essentially require precise and reliable flame segmentation. In this context, the search for suitable segmentation methods for flame regions is a contribution to combustion process control that opens the door for more capable and energy-efficient combustion processes.

The issue of flame segmentation has already been addressed by other research. Many propositions use a characteristic color criterion to find flames in image frames [3–5] or a combination of color criterion and region filtering based on dynamic properties in between multiple frames of video sequences [6–9]. Other approaches include geometric criteria, such as roundness or area change [6,7], and texture criteria [7]. The computation of task-specific data properties isolates relevant image information for further classification and is often referred to as feature engineering. Simple classification methods use thresholds in the regarded feature space to determine the class label on pixel scale [10]. Ref. [2] use the Otsu threshold method for flame segmentation in an industrial context. Other methods channel multiple features into a more complex classification process such as SVM [7], Bayes classifier [4,6], neural network [3], or fuzzy logic classifier [5].

The authors of [11] demonstrate flame segmentation obtained with a level set method. The level set method optimizes a functional in the image plane to find a contour which satisfies a homogeneity criterion. The algorithm cannot be adapted easily to instance segmentation, as it struggles to separate multiple objects in homogeneous regions of overlapping flames.

The cited methods produce segmentations of flames; however, they focus on flame detection or single flame segmentation. Therefore, instance distinction between multiple flames has been neglected. None of the cited methods are capable of performing instance-level segmentation that allows to distinguish overlapping flame areas from different sources. For the purpose of combustion analysis as mentioned in [2], the distinction between flame instances is essential, because for effective combustion control a flame's properties must be analyzed with respect to its burner's source.

The recent advances in detection and segmentation with convolutional neural networks (CNNs) [12,13] suggest that instance segmentation can be performed with this approach. Algorithms like those in [14,15] achieve remarkable results in computer vision benchmarks [16] and can be considered state-of-the-art.

The authors of [17–21] make use of CNNs to detect and localize flame regions in outdoor images but do not provide segmentation.

To our knowledge, it has still not been shown whether CNN-based methods can produce flame instance segmentation results that meet the requirements of an industrial process such as in [2]. Some properties of flames, such as transparency, low contrast, and high variability in shape and texture, can even trick human visual perception and make flame segmentation more difficult than segmentation of regular objects. Consequently, we investigate the potential of CNNs for instance segmentation on overlapping flames in an industrial environment. In this research, we analyze images from the afterburner chamber of a research plant for special waste incineration. Similar observation perspectives on burner flames can be found for example in other special waste incineration facilities and also in many coal-fired power stations worldwide. For this reason, the investigations in this paper are transferable to these processes.

2. Experimental Setup and Image Acquisition

The resource for our investigation is the afterburner chamber in a research plant used for waste incineration as shown in Figure 1. Multiple burners are mounted horizontally in the chamber to treat the exhaust gas from a rotary kiln process. A camera provides continuous top-view images of the process in the visual spectrum.

Figure 1 displays example images from the afterburner chamber at different operation conditions. These and similar images are the objects of investigation in this work. We use 3000 images from 30 different sequences. In all sequences, there are two active burners. The arrangement of the active burners can be either parallel or orthogonal. When referring to a flame we use the name "flame 1" for flames propagating from the left image side and consequently "flame 2" for the other flame as labeled in Figure 1.

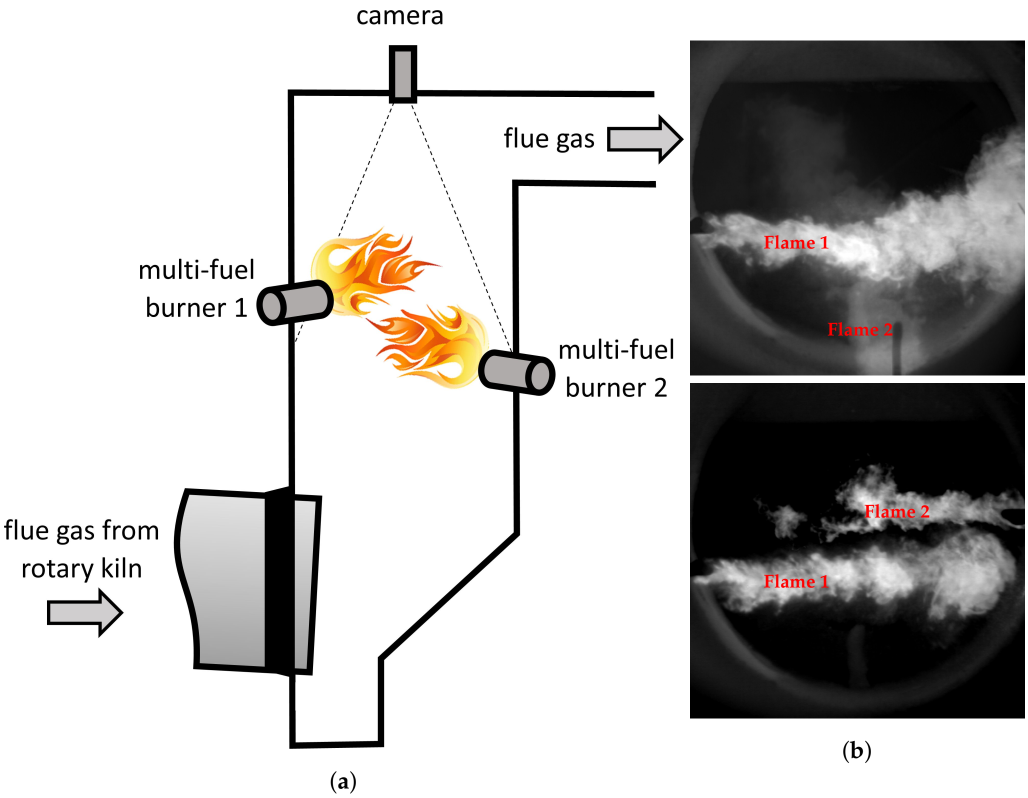
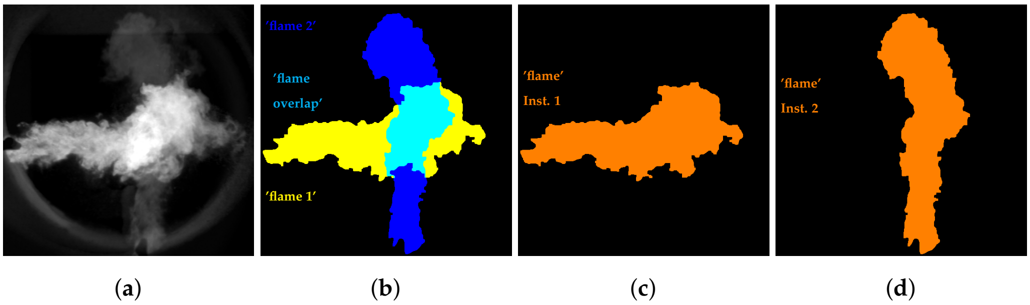


Figure 1. (**a**) Schematic of the afterburner chamber in a waste incineration plant. Gases are induced from a rotary kiln and pass through the chamber to the outlet at the top. The chamber is a vertical cylinder of about 2 m in diameter and several meters in height. At multiple positions burners are mounted horizontally in the chamber wall, producing characteristic flames. The process can be observed from top-view perspective by a camera (**b**).

3. Convolutional Neural Networks for Segmentation

From a high-level perspective, computer vision tasks can be categorized into problem types, such as image classification, object recognition, semantic segmentation, or instance segmentation. Historically, many of these problems have been tackled with feature engineering approaches. However, in the last decade, the focus of computer vision research has partially shifted to deep neural networks. The main advantage of neural networks is their ability to automatically learn relevant representations when trained with appropriate data. In contrast, feature engineering requires task-specific expert knowledge and includes significant low-level design effort to create effective algorithms. Convolutional neural networks are a subtype of neural networks which have been proven to be particularly effective in high-level computer vision tasks.

As mentioned in the introduction, we aim to obtain an individual flame segmentation for each burner in an image. This type of problem can be related to the field of instance segmentation. Therefore, naturally, we want to investigate networks that allow instance segmentation. However, in our particular case, we can also adapt semantic segmentation networks to the task. Due to the stationary burner setup in our experimental environment, we already know that there can only be a total of two unique flames in every image. Labeling each of the two flames as a separate semantic class allows flame distinction on a per-flame level. Additionally, we can introduce a third class label for the area of overlapping flames. With the prior knowledge about the total amount of flames, we formalize the task for instance segmentation and semantic segmentation with three classes as shown in Figure 2. Using CNN of both types allows us to make a direct comparison of the segmentation accuracy and inference speed between semantic and instance segmentation networks at our task of flame instance segmentation.



(**a**) (**b**) (**c**) (**d**)

Figure 2. Difference in the class partition between semantic and instance segmentation. (**a**) Flame image example, (**b**) corresponding semantic ("instance-like") ground truth, and (**c**,**d**) one partition per flame for instance segmentation. The colors encode class information. In the semantic example each flame segment is a different class ("flame 1", "flame 2", "flame overlap"), whereas for true instance segmentation each instance has its own partition of the same general class 'flame'.

As we adapt semantic segmentation networks to produce results that appear as instance segmentation, the formal difference between the methodological approach cannot always be distinguished based on visual inspection of the final segmentation results. When comparing results which show segmentations with multiple flames, we can use the term instance-like segmentation for segmentations produced by semantic networks and true instance segmentation for results produced by instance segmentation networks. One major difference between the semantic and the instance segmentation approach is that instance segmentation hypothetically allows an infinite amount of flame instances and also overlapping regions in the general case, whereas in semantic segmentation, we have to manually introduce extra classes.

In the experiments, we train existing network architectures in a supervised manner with annotated flame images from our data. We evaluate and compare the instance segmentation network Mask R-CNN [15] and the semantic segmentation network DeepLabv3+ [22].

DeepLabv3+ uses an encoder–decoder structure [23]. The encoder subnetwork extracts complex images features, whereas the decoder network is used to transform the feature representation into a segmentation and restore information at the original image resolution. For many networks, the encoder is interchangeable between different architectures. We investigate the potential of different encoder networks with different computational complexity for DeepLabv3+ [22], namely, ResNet 18, ResNet 50 [24], and Inception ResNet v2 [25]. For visual clarity, we use the abbreviations DeepLabv3+ RN18, DeepLabv3+ RN50, and DeepLabv3+ Inc.-RN.

The Mask R-CNN network [15] is a network for true instance segmentation. It has a modular architecture based on four subnetworks which perform different tasks:

- The base network is a deep neural encoder network for image feature extraction. The extracted features are shared with other more specific subnetworks. For the Mask R-CNN implementation in this investigation, we use a ResNet 101-based [24] encoder network.
- The region proposal network uses the features provided by the base network to predict regions of interest (RoIs) which likely contain a relevant object. The region proposal network uses sliding windows that mark a grid of cropped images. In the next step, the cropped images are further classified as positive or negative RoIs. The RoIs are a preprocessing stage to object instances that can be found in an image.
- The class prediction network refines and classifies all RoIs, deciding which object can be seen in the region.
- The mask prediction network predicts a binary segmentation image for each RoI.

The combination of image, RoI, binary segmentation, and class information forms the instance segmentation.

4. Training Data and Task-Specific Image Augmentation

The conventional training of CNNs requires annotated image data for supervised learning and additional images for evaluation. Our image acquisition disposes a raw data base of 30 video sequences with thousands of images per sequence. We randomly choose 100 images per sequence to a total of 3000 images. In the next step, we generate the ground truth annotation to all 3000 images. At the third step, we form three subsets from the 3000 images. Per sequence we select 80% images randomly (to a total of 2400) for training and 10% images (to a total of 300) for validation. Additionally, we choose another 10% images per sequence (to a total of 300) for testing. Sampling the images per sequence ensures a subset content's variety as each subset consists of equally sized image portions from each sequence.

It is difficult to train a large neural network from randomly initialized weights for our specific problem with a limited amount of labeled data available. We overcome this problem using transfer learning. Transfer learning for neural networks makes use of the observation that neural networks can learn meaningful data representations for a broader field of tasks even when originally trained for a different task, see, e.g., in [26]. We follow this procedure and initialize DeepLabv3+ RN18, DeepLabv3+ RN50, and DeepLabv3+ Inc.-RN using (ImageNet [27]) pretrained weights. For Mask R-CNN, we use weights pretrained with images from the COCO challenge for object recognition [28]. In each case, we keep only the encoder weights and reinitialize subsequent layers according to the He initialization method [29]. In our standard learning procedure, we train DeepLabv3+ RN18, DeepLabv3+ RN50, and DeepLabv3+ Inc.-RN with a reduced learning rate of 0.001 for the encoder network, and a learning rate = 0.01 for the subsequent structures. The authors of [26] reach their best results in their transfer learning experiment when also training the encoder weights with the same learning rate as for the rest of the network. We investigate if we can achieve a similar positive effect with our networks by varying the learning rate in some of our encoder networks. Therefore, we also train some of our semantic networks with a global learning rate of 0.01. In our experiments we describe these networks with the attribute -all (e.g., DeepLabv3+ RN18-all). We found that for our semantic networks the validation loss converges after 5 epochs of training which we consequently adapt to our standard learning procedure. For Mask R-CNN we train 30 epochs with a global learning rate of 0.001. We use stochastic gradient descent with momentum as optimizer during training.

We know from experience that a characteristic problem for image processing in real industrial combustion facilities can arise from the disturbance in image acquisition via pollution from the combustion process even when using an air purge system. A camera with a dirty lens produces partially stained images that challenge further image processing algorithms. In order to estimate the impact of stained images on the segmentation quality of convolutional neural networks, we artificially imitate the effect with task-specific image augmentation. To conduct augmentation, we use element-wise multiplication of an image $\boldsymbol{I}$ with a darkening image mask $\boldsymbol{A}$ to produce darkened images $\boldsymbol{I}_{dark}$. In order to adapt to different stains, we create seven different mask types: two different sizes of image-centered disks, one annulus, and sidewise obscuration from four different directions. In Figure 3, we visualize examples of augmentation masks and their application to data. In this work, we denote this procedure as *dark* augmentation. The notion dark10 with a data set (e.g., training data) describes a data set with 10% of the images obscured by augmentation. Unfortunately, we do not possess enough naturally obscured images to make a proper data set. Therefore, we use the *dark* augmentation on the test data to get enough samples for our evaluation.

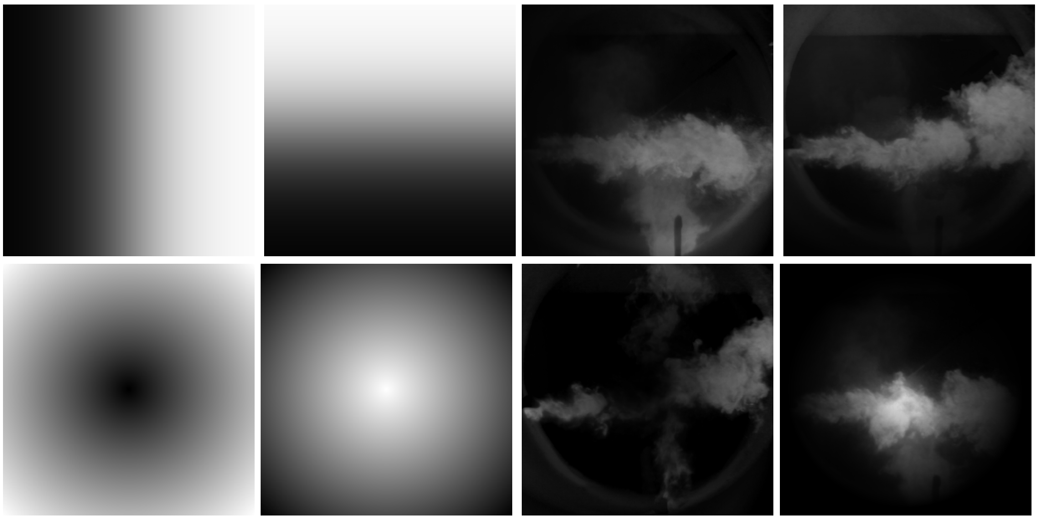


Figure 3. Visualization of a task-specific image augmentation for obscuration (*dark*). (**a**) Darkening image masks $\boldsymbol{A}$. (**b**) Obscured images $\boldsymbol{I}_{dark}$.

5. Segmentation Quality Metrics

For segmentation tasks, there are many established metrics [30] which can be used for evaluation. Different metrics emphasize different aspects of the result. Therefore, an evaluation metric should be chosen to suit to the intention of the task. For their definition of a geometric flame stability measure, [2] refer to the area of overlap of flames in between consecutive video frames. We think that a metric directly coupled to the correctly segmented flame area is a suitable measure to judge the segmentation quality for the use with a method that relies on the analysis of overlapping flame area. A commonly used metric that implements this idea is the intersection over union metric (IoU), which we therefore use as metric in our evaluation. When comparing the pixel labels between prediction and its ground truth, every pixel can either be in a true positive (TP), true negative (TN), false positive (FP), or false negative (FN) condition. From the conditions of an image, the $\mathrm{IoU_f}$ is then computed using

$$\mathrm{IoU_f} = \frac{\mathrm{TP_f}}{\mathrm{TP_f} + \mathrm{FP_f} + \mathrm{FN_f}}. \tag{1}$$

The subscript f denotes an evaluation per flame, as we want to analyze different flame instances separately. Visually, the IoU is the fraction of the area of overlap and the area of union between segmentation prediction and ground truth map.

Yet another important criterion in industrial environments is the robustness of a process. We think that the quality of a flame segmentation algorithm for industrial application is not only determined by the highest mean in $\mathrm{IoU_f}$, but also by the minimum performance on single images over the entire data set. For this reason, we present a new metric for robustness evaluation, which we call critical image rate $\mathrm{i_{crit}}$. By

$$\mathrm{i_{crit}} = \frac{\mathrm{N_{crit}}(\alpha_{\mathrm{crit}})}{\mathrm{N_{Test}}} \tag{2}$$

we define $\mathrm{i_{crit}}$ by the proportion of images from a data set with an evaluation score in a specific metric below a critical threshold α_{crit}. α_{crit} denotes a threshold value in a segmentation evaluation metric that allows to separate a set of segmentations into subsets of high and low quality. As we use the $\mathrm{IoU_f}$ as primary evaluation metric, we use an $\mathrm{IoU_{f,crit}}$ as our α_{crit}, but other segmentation metrics could be used instead. $\mathrm{N_{crit}}$ represents the number of critically rated images out of a data set and $\mathrm{N_{Test}}$ denotes the total amount of images in the test set.

We use the per flame $\mathrm{IoU_f}$ and an $\mathrm{IoU_{f,crit}}$ based critical image rate $\mathrm{i_{crit}}$ as segmentation quality metrics.

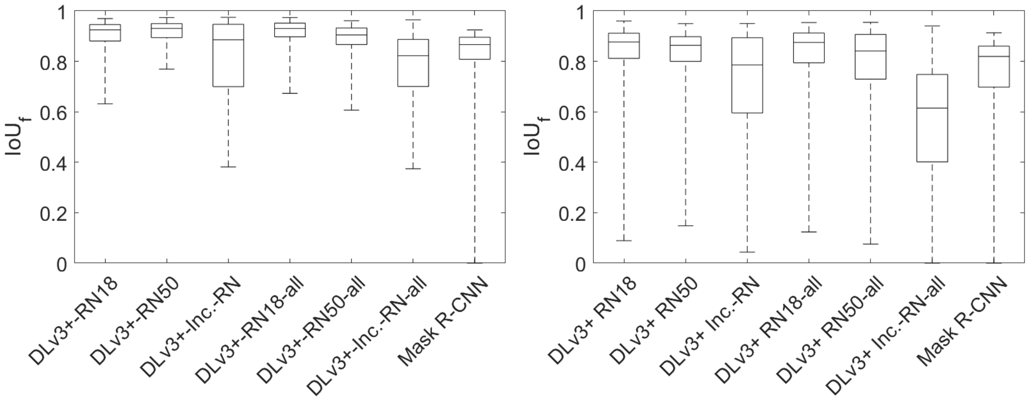
6. Experiments

In our experiments, we investigate the performance of the different networks for segmentation of our flame images. We follow a general procedure for training and evaluating neural networks. For each network, we start with pretrained weights and retrain with our selected training and validation data until validation loss converges. In the next step, we conduct segmentation interference with the network models on the 300 images in our test set. Finally, we evaluate the segmentations and analyze the results.

6.1. Instance Segmentation

In our first experiment, we investigate general instance segmentation of the burner images with several CNNs. We evaluate and compare DeepLabv3+ RN18, DeepLabv3+ RN50, DeepLabv3+ Inc.-RN, DeepLabv3+ RN18-all, DeepLabv3+ RN50-all, DeepLabv3+ Inc.-RN-all, and Mask R-CNN. We compute flame 1 and flame 2 $\mathrm{IoU_f}$ separately, because we assume that in most applications, a high segmentation quality of flame 1 cannot compensate a low segmentation quality of flame 2, and vice versa.

Figure 4 depicts the $\mathrm{IoU_f}$ distribution of the test images for each network. The $\mathrm{IoU_f}$ results of flame 1 are higher than for flame 2. We can possibly relate the difference to more difficult visual properties of flame 2. In many images, flame 2 has low contrast to its surroundings. This is an implication of the different fuel conditions and therefore a natural variation of the combustion process. We present the results in the order of median $\mathrm{IoU_f}$ of flame 1 starting at the lowest score. DeepLabv3+ Inc.-RN-all (median $\mathrm{IoU_f}$: Flame 1 = 0.823, Flame 2 = 0.615), Mask R-CNN (median $\mathrm{IoU_f}$: Flame 1 = 0.866, Flame 2 = 0.820) DeepLabv3+ Inc.-RN (median $\mathrm{IoU_f}$: Flame 1 = 0.886, Flame 2 = 0.786) have lower median $\mathrm{IoU_f}$ and also lower minimal $\mathrm{IoU_f}$ compared to the other networks. DeepLabv3+ RN50-all (median $\mathrm{IoU_f}$: Flame 1 = 0.905, Flame 2 = 0.841), DeepLabv3+ RN18 (median $\mathrm{IoU_f}$: Flame 1 = 0.925, Flame 2 = 0.877), DeepLabv3+ RN18-all (median $\mathrm{IoU_f}$: Flame 1 = 0.930, Flame 2 = 0.875), and DeepLabv3+ RN50 (median $\mathrm{IoU_f}$: Flame 1 = 0.931, Flame 2 = 0.864) and similarly achieve the highest median $\mathrm{IoU_f}$ scores in our evaluation.



(**a**) (**b**)

Figure 4. $\mathrm{IoU_f}$ on test set for instance segmentation. Each box shows the $\mathrm{IoU_f}$ of the segmentations obtained with different convolutional neural networks (CNNs). The boxed line represents the median, whereas the boxes itself include the 25 to 75 percentiles. The whiskers indicate the most extreme values. (**a**) Evaluation of flame 1. (**b**) Evaluation of flame 2.

Contrary to expectations, a larger backbone network does not increase the median $\mathrm{IoU_f}$ results in our experiments. The Inception ResNet v2 backbone network has more weight parameters and performs more operations [31] than ResNet 50, but in Figure 4 DeepLabv3+ Inc.-RN scores lower $\mathrm{IoU_f}$ results than DeepLabv3+ RN50. Furthermore, the use of a global learning rate of 0.01 does not have a clear positive effect on the IoU_f distribution of DeepLabv3+. In our experiment, the median IoU_f slightly increases for DeepLabv3+ RN18-all (vs. DeepLabv3+ RN18), but it decreases for DeepLabv3+ Inc.-RN-all (vs. DeepLabv3+ Inc.-RN) and DeepLabv3+ RN50-all (vs. DeepLabv3+ RN50).

In Figure 5, we use the metric i_crit to visualize the robustness of the instance segmentation results. In instance segmentation, we consider an image to be critical if either flame 1 or flame 2 has an IoU_f below the $\text{IoU}_\text{f,crit}$ threshold. Applying the logical "or" classifies each image by its worst contributing element and is rather strict. A low sum of i_crit over all samples indicates good general performance and a flat curve in the intervals of a low $\text{IoU}_\text{f,crit}$ indicates robustness. For instance, segmentation the i_crit of DeepLabv3+ Inc.-RN, DeepLabv3+ Inc.-RN-all, and DeepLabv3+ RN50-all steadily increases over all thresholds. The i_crit curve of Mask R-CNN increases abruptly between $\text{IoU}_\text{f,crit} = 0.0$ and $\text{IoU}_\text{f,crit} = 0.05$, then it continues almost constant until $\text{IoU}_\text{f,crit} \approx 0.55$. We analyze the step increase in Section 6.3 to understand the phenomenon. DeepLabv3+ RN18, DeepLabv3+ RN18-all, and DeepLabv3+ RN50 achieve low i_crit in the low $\text{IoU}_\text{f,crit}$ region. With these three networks, we obtain instance segmentation $\text{IoU}_\text{f} \geq 0.7$ on both flames for more than 90% of our test images which is our best results with respect to robustness. At high thresholds $\text{IoU}_\text{f,crit} > 0.7$ the i_crit increases rapidly across all networks.

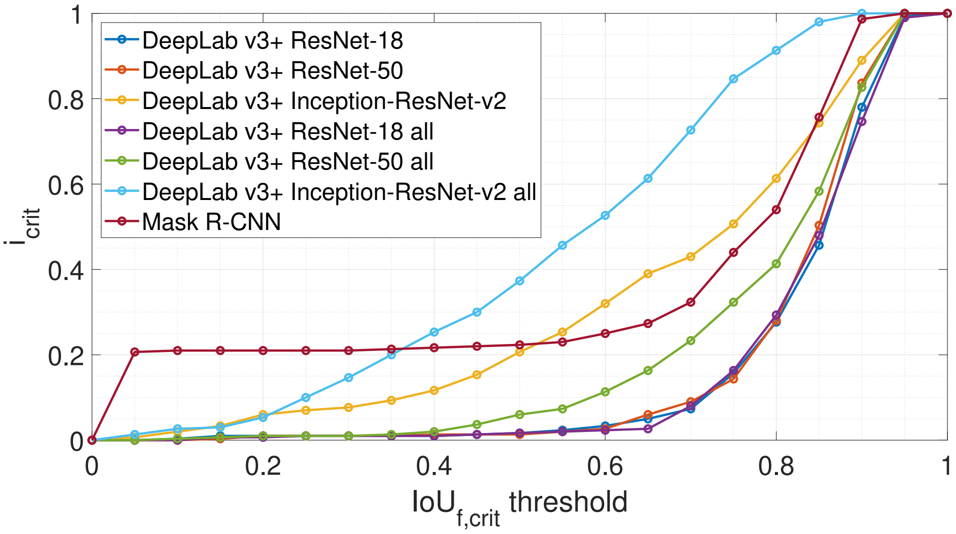
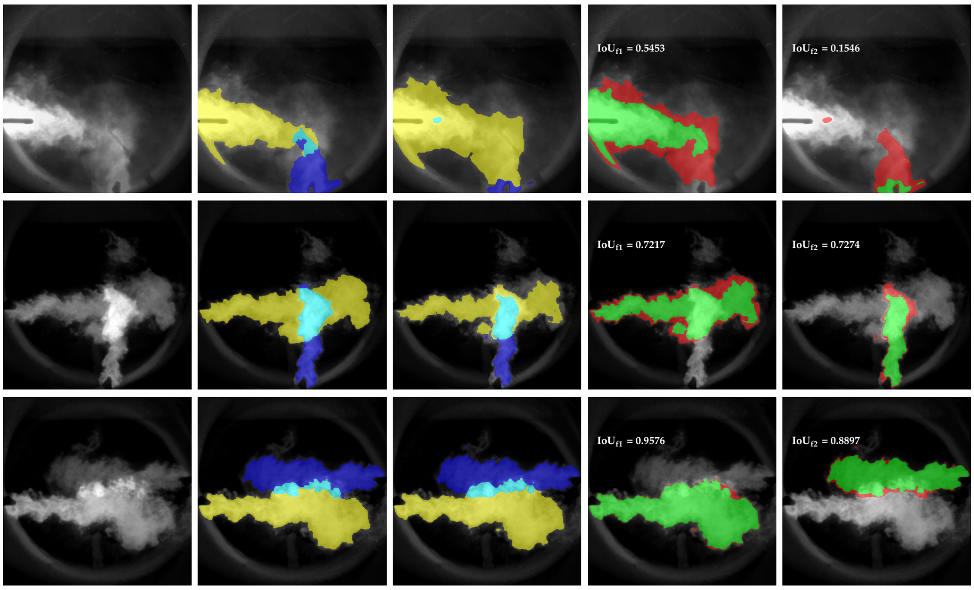


Figure 5. i_crit Curves over incremental $\text{IoU}_\text{f,crit}$ thresholds for instance flame segmentation. For each segmentation network we sample i_crit at 21 $\text{IoU}_\text{f,crit}$ levels from 0.0 to 1.0. At increasing $\text{IoU}_\text{f,crit}$, the fraction of critical Images i_crit rises, resulting in a characteristic curve for each segmentation method.

The image gallery in Figure 6 visualizes increasing segmentation qualities from top to bottom. At the top example the network does not split the area between flame 1 and flame 2 correctly. The middle example represents the segmentation quality that we reach for more than 90% of the test images. As shown in the middle and bottom examples, we can successfully perform instance-like segmentation of flames with overlapping regions.

We also record the inference processing speed of the networks. DeepLabv3+ inference is executed within a MATLAB (Mathworks, ver. R2019b) environment and Mask R-CNN network we use a Python implementation. We use GPU-accelerated execution with a Nvidia GeForce GTX 1050 Ti GPU. The results in Table 1 show that the fastest network is DeepLabv3+ RN18 with an average processing time of 0.149 seconds. Our ranking of the tested networks with respect to speed is analogous to the backbone network benchmark speed evaluation of [31]. A processing speed of about 1 s to 0.149 s is not fast enough for video processing, but we assume it can be sufficient for an online process control in many applications, where the properties of the process change at a slower rate.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 6. Gallery of instance flame segmentation with CNNs. From top to bottom, the figure shows different image series with increasing IoU$_f$ scores at our evaluation. (**a**) An original image example in our test data set. (**b**) The corresponding ground truth and (**c**) segmentation inference with a CNN. The segmentation in the upper two series have been obtained with DeepLabv3+ Inc.-RN. whereas the series at the bottom is produced with DeepLabv3+ RN18. (**d**,**e**) Vsualizations of the differences between the ground truth and the prediction of flame 1 (d) and flame 2 (e). The green mark-up indicates matching pixels between ground truth and prediction, whereas red mark-up indicates mismatched pixels.

Table 1. Inference speed of convolutional segmentation networks.

Network	Avg. Time	Max. Time
DeepLabv3+ RN18	0.149 s	0.227 s
DeepLabv3+ RN50	0.270 s	0.291 s
DeepLabv3+ Inc.-RN.	1.044 s	1.134 s
Mask R-CNN (RN101)	0.722 s	0.756 s

6.2. *Effect of Image Augmentation*

In this section, we investigate the effect of image augmentation on flame instance segmentation. For this purpose, we train two separate models for each network type. For one model, we use dark10 image augmentation, as introduced in Section 4 on the training data, and in another case, we train without any augmentation. Furthermore, we can also test each of the two differently trained models on two different sets of test data, one without augmentation and one with dark10 augmentation. This comprises a total of four different evaluation scenarios which we can compare for each network. We further use the notation (train: 0aug, test: dark10) to characterize a scenario where a CNN model has been trained with images without augmentation and evaluated on dark10 augmented test images.

In Figure 7, we visualize the i_{crit} curves of all four evaluation scenarios for the network DeepLabv3+ RN50. The curve (train: 0aug, test: 0aug) is the bottom-line scenario that we have already analyzed in Section 6.1. When we evaluate the same model with augmented test data (train: 0aug, test: 10dark), the i_{crit} increases, marking the highest curve at most IoU$_{f,crit}$ thresholds in Figure 7. The comparison between (train: 0aug, test: 0aug) and (train: 0aug, test: 10dark) allows us to estimate how darkened test image content affects the results produced by a model which has not been trained with similar images. We report the largest difference at medium IoU$_{f,crit}$ thresholds between 0.45 and 0.85 with all different networks we test. This reveals the negative influence of the darkened image content, when the model has not been adapted by specific training with augmented image data.

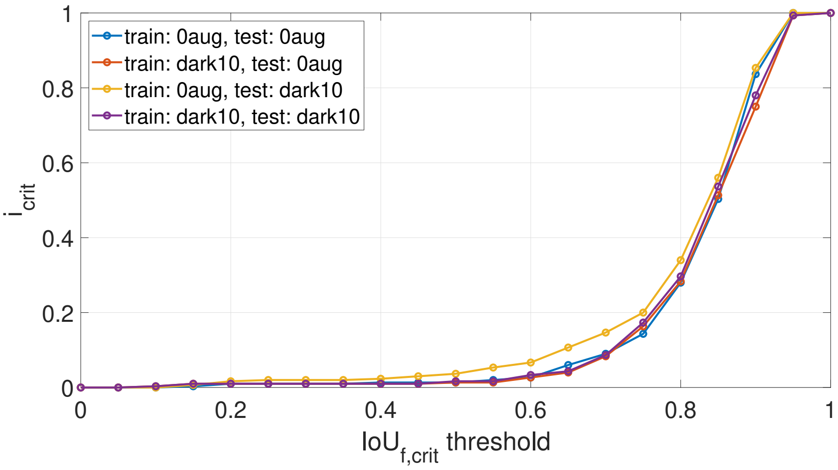


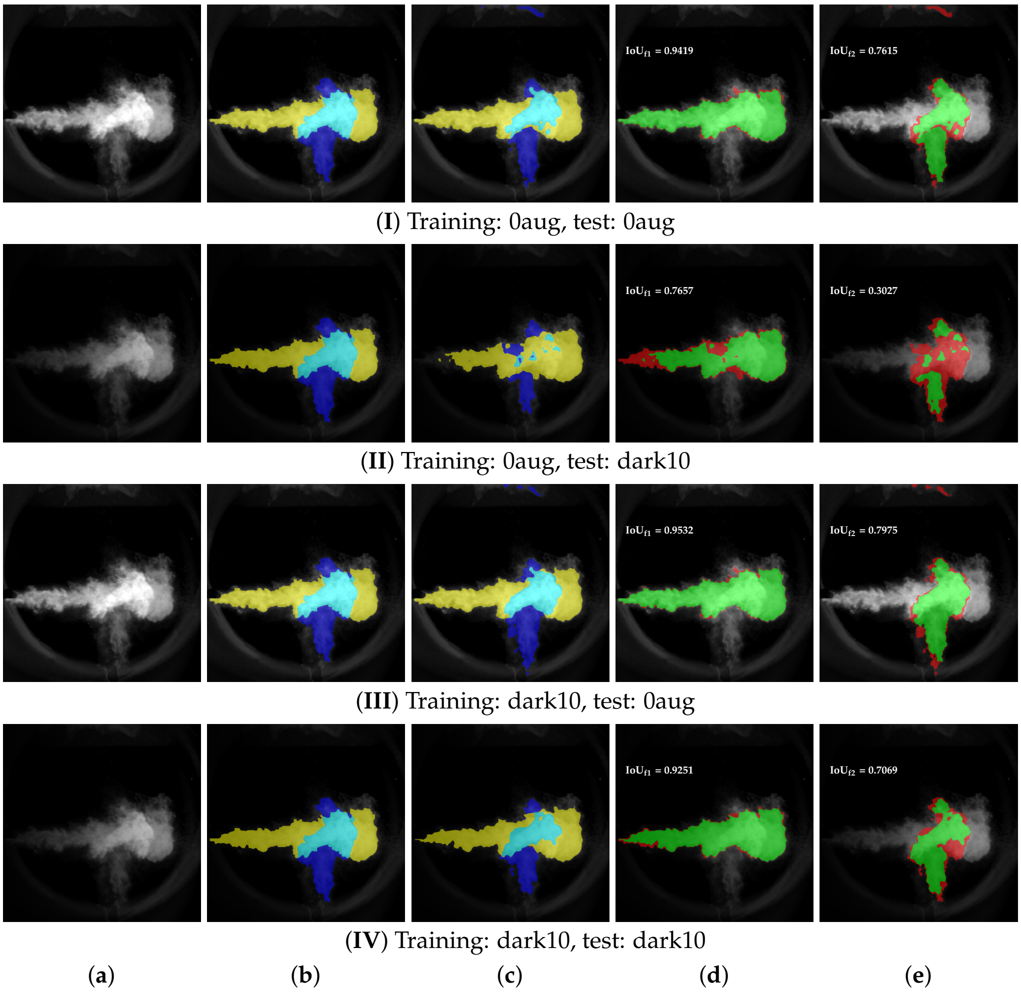
Figure 7. i_{crit} Curves over incremental thresholds for DeepLabv3+ RN50, for instance, segmentation at different augmentation scenarios. We sample i_{crit} at 21 $IoU_{f,crit}$ levels from 0.0 to 1.0. At increasing $IoU_{f,crit}$, the fraction of critical Images i_{crit} rises, resulting in a characteristic curve. The different curves visualize different augmentation scenarios for the network DeepLabv3+ RN50.

On the other hand, when applying dark10 augmentation during training, we report an i_{crit} curve lower than the scenario (train: 0aug, test: dark10) in both test scenarios (train: dark10, test: 0aug) and (train: dark10, test: dark10)at most $IoU_{f,crit}$ thresholds. In general, the difference of i_{crit} between augmented and unaugmented test data is much smaller between our models which use augmented training data as compared to training with unaugmented data. This indicates an equal adaption of the model to unaugmented and augmented test data. Compared to the bottom-line scenario (train: 0aug, test: 0aug), we do not observe a one-sided positive or negative effect of the training data augmentation on the i_{crit} achieved with our models. In our experiment, the i_{crit} with (train: dark10, test: 0aug) and (train: dark10, test: dark10) is higher or lower than (train: 0aug, test: 0aug) at different $IoU_{f,crit}$ thresholds.

In summary, the training with dark10 augmentation positively affects the i_{crit} test results of our networks when evaluating with augmented test data. At the same time, training with dark10 augmentation has an ambiguous effect on the i_{crit} with unaugmented test data.

In Figure 8, we depict the described effect on the image level. The bottom-line scenario (train: 0aug, test: 0aug) is a typical example from our test data with regular IoU_f results in the range close to the median for both flame 1 and flame 2. However, darkening the test image with an augmentation decreases the IoU_f on both flames and also disturbs the visual segmentation quality for the model trained with unaugmented data (train: 0aug, test: dark10). On the other side, the model (train: dark10, test: dark10) trained with augmented data achieves higher IoU_f of both flames and better visual segmentation quality on the augmented test image than the model trained with unaugmented data.

We demonstrate that we can positively affect evaluation results on darkened image data with a specific dark10 image augmentation. However, we also find it notable that all network variations reach at least a similar curve on dark10 augmented test images. The low i_{crit} in the range of low $IoU_{f,crit}$ thresholds in Figure 7 shows that the effect of darkening augmentation does not produce fully degenerated segmentation results. This is even the case for the scenario (train: 0aug, test: dark10), in which the network model is unaware of the augmentation used in the test data. This indicates that the flame representation learned by all networks is already resilient to a certain degree to the degradation induced by the dark10 augmentation.



(**I**) Training: 0aug, test: 0aug

(**II**) Training: 0aug, test: dark10

(**III**) Training: dark10, test: 0aug

(**IV**) Training: dark10, test: dark10

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 8. Effect of image augmentation on segmentation. I–IV show series of an image at different conditions of training or test data darkening augmentation as specified underneath each series. In every series image, subfigure (**a**) shows an original image example in our test data set. (**b**) The corresponding ground truth and (**c**) segmentation inference with a CNN. (**d**,**e**) Visualizations of the differences between the ground truth and the prediction of flame 1 (**d**) and flame 2 (**e**). The green mark-up indicates matching pixels between ground truth and prediction, whereas red mark-up indicates mismatched pixels.

6.3. Failure Cases and Future Work

In Figure 5, we have seen that we obtain a significant amount of segmentations below $\text{IoU}_{\text{f,crit}} < 0.05$ on test images with the network Mask R-CNN. The phenomenon is related to the issue that Mask R-CNN sometimes detects additional (false-positive) flame instances or misses them (false-negative), e.g., in Figure 9.

By visual inspection we find that the false-positive flames correspond to images with disconnected flame bodies. The ambiguous flame bodies are a natural property of flame images and hence additional flame segmentations are difficult to avoid. One approach to deal with additional flame instances could be a postprocessing procedure that eliminates superfluous instances based on application specific criteria, such as size.

On the other hand, we have identified two sources that contribute to false-negative flame instances. First, we have identified a critical role of non-maximum suppression based on IoU, which is used in the Mask R-CNN pipeline to filter out redundant object instances. In some cases of images with high flame overlaps, a flame is accidentally removed from the final results. This can be avoided by fine-tuning the threshold parameters for the non-maximum suppression at the cost of more false positive instances. In other examples, flames are not found by the region proposal network.

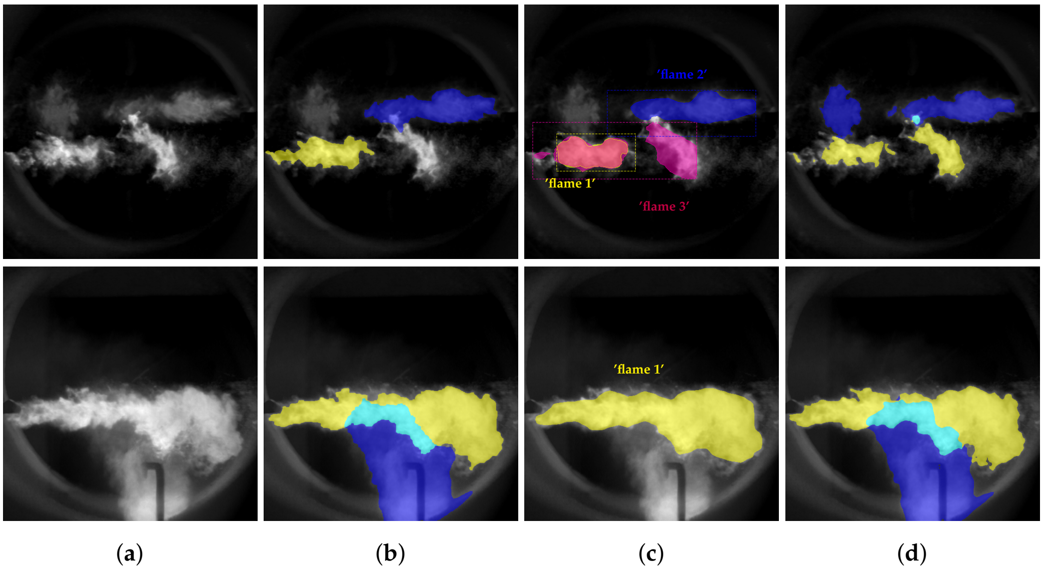


Figure 9. Challenging scenarios with Mask R-CNN in instance segmentation. (**a**) The original flame image, (**b**) the ground truth, (**c**) the segmentation inference with Mask R-CNN, and (**d**) the segmentation inference with DeepLabv3+ RN18. In the top example, Mask R-CNN detects a third flame instance (purple) on top of flame 1 (yellow, but visually adds up with purple to pink). In the bottom image, only flame 1 is detected by Mask R-CNN.

With the semantic networks we segment the correct amount of two flames in all test images. As semantic segmentation networks are limited to a predetermined maximum amount of flames instances by the class definition, it is impossible to segment false positive instances. Fort the case of false negative flame instances, we can try to explain the success with the differences between Mask R-CNN and the semantic networks. As mentioned previously, we report two sources for missing predictions with Mask R-CNN, first the non-maximum suppression and second missing predictions at the region proposal network stage. The semantic networks do not use these structures. It seems to be an easy task to find all flame regions without these elements. When we analyze the examples in Figure 9, we can see that our best semantic model DeepLabv3+ RN18 also segments the two additional flame bodies in the image content contrarily to our ground truth. However, DeepLabv3+ RN18 considers the areas as extensions of flame 1 and flame 2. We see this as an indication that the network has learned a generalized flame representation, as it detects these flame regions based on image features without explicit learning from our ground truth. On the other hand, it indicates that the network has not learned a representation which allows the class distinction based on area connection as labeled in the ground truth.

In our investigation of instance segmentation of flames, we reach better results with instance-like segmentation from the semantic DeepLabv3+ network than with true instance segmentation with the network Mask R-CNN. We have related the main differences of the results to the challenge of false positive and false negative predictions. However, we think that from a general perspective the approach of Mask R-CNN has a theoretical advantage over the instance-like semantic segmentation performed with semantic networks. As it is not limited to a fixed number of flames, Mask R-CNN could anticipate to segment images with more flames including overlaps, without retraining, whereas for the same task a semantic network would need a retraining along with a work over of the ground truth to adapt to new classes. Furthermore, as Mask R-CNN does not learn each flame as separate class, it is also supposed to learn a more general representation of flames, which could make it more adaptive to different process conditions and environments. We want to outline that, despite the challenges with Mask R-CNN, the concept of true instance segmentation remains an attractive goal for flame instance segmentation. It is open to future work to show whether our challenges can be solved either with modifications to Mask R-CNN or other instance segmentation networks.

7. Conclusions

We evaluated several recent CNN architectures for the task of instance segmentation on combustion images in an industrial plant. With the best networks, we obtained experimental instance segmentation results with an $\mathrm{IoU}_f \geq 0.7$ of both flames on more than 90% of our test images. We also tested the networks on images with simulated darkened regions that imitate an optical lens soiling typical in combustion processes. We could show that the network segmentation is robust to this type of disturbance and can further be adapted with specific image augmentation. In summary, we provided evidence that with CNN-based segmentation methods, remarkable results can be achieved even on difficult images with overlapping flame regions. Our contribution is a verification of the capability of CNN-based methods for instance segmentation of burner flames. With regard to its applications, we think that these methods can be used in the context of closed loop combustion process control to give leverage on process capability and energy efficiency.

Author Contributions: All listed authors have contributed to Conceptualization and methodology of this research. The project was administrated by J.M. J.M. and P.W. have acquired the image data resources. It was processed and visualized by M.V. and J.G. All authors have contributed to the analysis of the results. The original draft was section-wise prepared by all authors and cross-reviewed by all other authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the research program "Energy Efficiency, Materials and Resources" of the German Helmholtz Association. We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data that has been used in this paper is available at https://doi.org/10.5281/zenodo.4453599. The data includes datasets from the training and testing of the neural networks, segmentation results and supplements for plotting.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, R.; Uddin, J.; Corraya, S. Real-Time Fire Detection Using Enhanced Color Segmentation and Novel Foreground Extraction. In Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 28–30 September 2017; pp. 488–493.
2. Matthes, J.; Waibel, P.; Vogelbacher, M.; Gehrmann, H.-J.; Keller, H. A New Camera-Based Method for Measuring the Flame Stability of Non-Oscillating and Oscillating Combustions. *Exp. Therm. Fluid Sci.* **2019**, *105*, 27–34. [CrossRef]
3. Zhong, Z.; Wang, M.; Shi, Y.; Gao, W. A Convolutional Neural Network-Based Flame Detection Method in Video Sequence. *Signal Image Video Process.* **2018**, *12*, 1619–1627. [CrossRef]
4. Li, Y.; Wang, L. Flame Image Segmentation Algorithm Based on Motion and Color Saliency. In *Communications, Signal Processing, and Systems*; Liang, Q., Liu, X., Na, Z., Wang, W., Mu, J., Zhang, B., Eds.; Springer: Singapore, 2015; pp. 184–191.
5. Celik, T.; Ozkaramanli, H.; Demirel, H. Fire Pixel Classification using Fuzzy Logic and Statistical Color Model. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, Honolulu, HI, USA, 15–20 April 2007; Volume 1, pp. 1205–1208. [CrossRef]
6. Borges, P.; Izquierdo, E. A Probabilistic Approach for Vision-Based Fire Detection in Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 721–731. [CrossRef]
7. Zhao, J.; Zhong, Z.; Han, S.; Qu, C.; Yuan, Z.Y.; Zhang, D. SVM Based Forest Fire Detection Using Static and Dynamic Features. *Comput. Sci. Inf. Syst.* **2011**, *8*, 821–841. [CrossRef]
8. Jamali, M.; Samavi, S.; Nejati, M.; Mirmahboub, B. Outdoor Fire Detection Based on Color and Motion Characteristics. In Proceedings of the 2013 21st Iranian Conference on Electrical Engineering, ICEE 2013, Mashhad, Iran, 14–16 May 2013. [CrossRef]
9. Wang, S.; He, Y.; Zou, J.; Duan, B.; Wang, J. A Flame Detection Synthesis Algorithm. *Fire Technol.* **2014**, *50*, 959–975. [CrossRef]
10. Zhang, Z.; Zhao, J.; Yuan, Z.; Zhang, D.; Han, S.; Qu, C. Color Based Segmentation and Shape Based Matching of Forest Flames from Monocular Images. In Proceedings of the 2009 International Conference on Multimedia Information Networking and Security, Wuhan, China, 18–20 November 2009; Volume 1, pp. 625–628. [CrossRef]
11. Fan, H.; Cong, Y.; Xia, Y.; Shao, W.; Wang, Y. Flame Front Detection and Curvature Calculation Using Level Set. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2014; pp. 2918–2922.

12. Li, Z.; Yang, W.; Peng, S.; Liu, F. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *arXiv* **2020**, arXiv:2004.02806.
13. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
14. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully ConnectedCRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
16. Lin, T.Y.; Patterson, G.; Ronchi, M.R.; Cui, Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; et al. COCO Challenge Website. Available online: https://cocodataset.org (accessed on 8 March 2020).
17. Frizzi, S.; Kaabi, R.; Bouchouicha, M.; Ginoux, J.; Moreau, E.; Fnaiech, F. Convolutional Neural Network for Video Fire and Smoke Detection. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 877–882.
18. Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep Convolutional Neural Networks for Forest Fire Detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application, Guangzhou, China, 30–31 January 2016; pp. 568–575. [CrossRef]
19. Dunnings, A.J.; Breckon, T.P. Experimentally Defined Convolutional Neural Network Architecture Variants for Non-Temporal Real-Time Fire Detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1558–1562.
20. Zhang, Q.; Lin, G.; Zhang, Y.M.; Xu, G.; Wang, J.J. Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images. *Procedia Eng.* **2018**, *211*, 441–446. [CrossRef]
21. Aktaş, M.; Bayramçavuş, A.; Akgün, T. Multiple Instance Learning for CNN Based Fire Detection and Localization. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–8.
22. Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; Volume 11211, pp. 963–983._49. [CrossRef]
23. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
25. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
26. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How Transferable Are Features in Deep Neural Networks? In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
27. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
28. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision 2014, Zurich, Switzerland, 6–12 September 2014.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 7–13 December 2015.
30. Fernandez-Moral, E.; Martins, R.; Wolf, D.; Rives, P. A New Metric for Evaluating Semantic Segmentation: Leveraging Global and Contour Accuracy. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1051–1056.
31. Bianco, S.; Cadene, R.; Celona, L.; Napoletano, P. Benchmark Analysis of Representative Deep Neural Network Architectures. *IEEE Access* **2018**, *6*, 64270–64277. [CrossRef]