

From haze to smoke: Weakly-supervised smoke detection

Daniel-Pau Azemar i Carnicero

Abstract

Urban areas suffer from a high level of contamination and this situation worsens every year, putting public health at severe risk at an incessant pace. Monitoring urban levels of contamination is a must, but monitoring pollution can be costly and placing different sensors in a city might be difficult. Deep learning has become a hot topic in solving computer vision tasks, including image dehazing. In this paper, we propose to go one step further and use similar techniques to detect smoke for both pollutant emitters and early stage of fires and explore the possibility of extracting partial segmentation maps using only weakly supervised training. Experiments show that our method is capable of detecting smoke with a high accuracy rate allowing also its segmentation from weakly labeled data.



Fig. 1: Example images from our experiments. On both cases: Real image (left), attention map produced by our method (middle), overprinted attention map on the image (right). The attention map indicates which regions on the image are more prone to contain smoke.

Index Terms

Air Pollution and Contamination, Image dehazing, Image de-smoking, Weakly supervision, Smoke Detection

I. INTRODUCTION

URBAN levels of contamination and pollution are a concern to many city councils. In Barcelona, the chronic exposure to excess air pollution represented at least 350 premature deaths in 2018 [1]. Air pollution is considered the first environmental risk for health [2] around the world. This is why the World Health Organization's directive 2008/50/EC on ambient air quality and cleaner air for Europe, established a limit of NO_2 emissions at an hourly average of $200 \mu\text{g}/\text{m}^3$, and a yearly average of $40 \mu\text{g}/\text{m}^3$ [3]. A limit which many major cities can not afford, resulting in big fines year after year for violating them. This problem has yet to end [4] [5] and, when applied to CO_2 and NO_2 , there is a debate if a declination in emissions will ever exist in developed countries [6]. New toxins can appear, making the curve of environment worsening to never decrease [7]. Also, environmental worsening is strictly related to countries in a current state of development, and maybe while we solve the problem in developed countries, some countries will face the same problem at a later time [8].

Monitoring air quality is therefore a necessity. However, traditional air quality monitoring needs professional sensors or equipment to be deployed at representative sites, which is expensive, intrusive, and costly, and it might not be possible for all cases. Also, these sensors provide a very limited range of action, making them insignificant when a large outdoor environment needs to be covered. An example of these large areas are industrial zones, which in Europe they can contribute up to a third of pollution that comes from humans sources (at a yearly average [9]). Monitoring them is not an easy task: high chimneys, toxins, and placing receptors can make the in-place monitoring a big burden, making it hard to assume without a large investment. Thus, pollution monitoring is vaguely performed.

The main source of pollution in industrial zones are chimneys and most of the time that pollution is in form of smoke and therefore detectable in the visible spectrum. It can spread to all directions and be identified from far away using visual sensors such as cameras. With the recent revolution of computing power and smart devices, the use of digital cameras has been a new paradigm for IoT and surveillance purposes [10], and with them, smoke detectors made a shift towards visual systems and by

Author: Daniel Azemar, daniel.azemar@e-campus.uab.cat

Supervisor: Javier Vazquez Corral, Academic Advisor, UPF

Supervisor: Anton Gomà, CEO Ctrl4Enviro, anton.goma@ctrl4enviro.com

Thesis dissertation submitted: November 2020



Fig. 2: Comparison of fog, smoke and cloud images. Both fog and clouds share features with smoke, which can either improve or decrease performance of a classifier.

2006 fire detectors using computer vision techniques were superior in performance to traditional methods [11]. Nonetheless, smoke automatic detection using computer vision is still an open challenge due to its similarities with other natural objects such as clouds, shadows, and mist, its inconsistency in motion and density, and even its complex visual pattern [12]. However, there is rarely a need for high precision, and just an approximation of what is going on with a low level of false alarms can solve most of cases.

In this project, we propose a simple solution that detects smoke while at the same time it obtains a partial segmentation of its location in the image. To do so, we base our project in previous investigations on image dehazing. These techniques consist in detecting the quantity of fog present on a simple image, with the ability to estimate the thickness of the haze and recover haze-free images. We relate this with smoke since both fog and smoke share transparency and gradient features (Figure 2). Additionally, since there is a scarce availability of public smoke datasets for detection and segmentation, we propose to perform this task with weakly-supervised training with image-label information only. Previous work on image dehazing for this latter problem proved to be successful [31]; nonetheless, newer and efficient architectures have been developed: InceptionV4, Residual Networks, Squeeze-and-Excitation models, EfficientNets, among others. We show how these new models perform better in the task of weakly-supervised fog detection, and improving the state-of-the-art in the case of haze. Then, we apply the same idea with our smoke dataset and show how these proposed models are capable of detecting smoke in a variety of images with good accuracy.

II. STATE OF THE ART

Given the specific temporal and spatial characteristics of smoke, most research in smoke detection has aimed at exploiting this information. Examples of this research are color-based discrimination techniques [13] [14], optical flow as an accumulation in orientation per blocks over the image [15], energy from video movement as a descriptor [16], wavelets and support vector machines [17], texture patterns [18], and gray level co-occurrence matrix [19], among many. Most of the problems when dealing with movement and spatial information are related to the background having large levels of noise deriving to bad segmentation and false alarms. This is the reason why some researchers exploit directly the background information in order to segment accurately the smoke and to exclude shadows, ghosts, and other micro-movement that may occur in the scene [20]. Nonetheless, all these methods still involve a high false alarm rate.

Deep learning has achieved great success in image classification and recognition due to its excellent ability of automatic learning features. This was first shown by AlexNet [21] at the *ImageNet* competition in 2012. From there, these Convolutional Neural Networks (CNN) have been a widespread tool in solving computer vision problems, including smoke detection. Frizzi et al. [22] used a custom CNN for classifying smoke and fire in images and is even capable of extracting salient features. S Khan *et al.* [23] proposed a fine-tuned VGG-16 [24] to classify smoke with and without foggy environment, resulting in good image classification, but working at image-patch level. A recent research [25] creates a dual-channel CNN to better extract smoke features, training one subnetwork to identify detailed information of the smoke such as texture, and the second subnetwork to extract contour information. Then they combine both networks that complement each other, increasing accuracy and outperforming previous works in image classification of smoke.

CNNs can work with temporal information (i.e. videos) for object detection. Throughout the years, new architectures based on CNNs emerged, and one of these architectures that can work with video is formerly known as C3DNN or C3D [26], a network in which their filters in hidden layers are of 3D shape, allowing to extract features from multiple images at the same time. Another of these architectures is named Recurrent-CNN [27] (RCNN), where information extracted from a previous frame is reused to feed the same layers at the next step. This allows the network to exhibit temporal dynamic behavior. As smoke presents strong temporal characteristics, this network is an interesting possibility for this task. Lin *et al.* [28] make use of both networks to exploit this information differently. With an improved RCNN they created multiple 3D blocks with region proposals of smoke, and then with a C3D, they classified the content using 3D features. However, these networks need

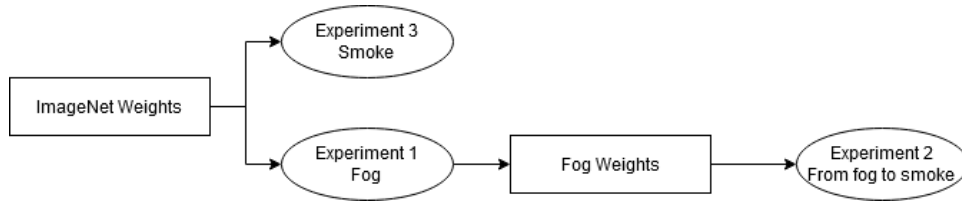


Fig. 3: Schematic of experiments that we will employ in this project.

large computational power to obtain a fast and reliable result, making them difficult to be deployed and to work in real-time scenarios.

Nowadays there are generalistic datasets for object recognition [29]. Unfortunately, these are not usually enough to solve specific problems such as smoke detection. However, CNNs can be trained with a lack of supervision information and obtain a desirable result. This process is named weakly supervised training [30]. This type of training consists of feeding the network with inaccurate, inexact, or incomplete data, in such a way that the network will learn to complete the lack of information, and also learn the important features. Galdran *et. al.* [31] used a weakly-supervised method for the task of fog detection that learns to predict haze presence using only image-level labels. They re-train a chopped CNN (InceptionV3 [32]; originally trained for ImageNet) until the last layer with receptive fields, and then instead of upsampling, they apply a 1×1 convolution to the resulting activation map, followed by a sigmoid function to obtain a prediction on each image patch. This approach gives a great accuracy in the task of fog detection, and due to its spatial similarities with smoke (transparencies, color, gradients) could be re-trained for smoke detection. This way we only require a single label per-image, knowing if the image contains some smoke. This type of training is much easier when dealing with image semantic segmentation, since dataset construction requires only image-label data.

For all these reasons, we build upon the work of Galdran *et. al.* [31]. Nonetheless, the architecture used in their work has had some revisions [33], and as stated before, newer CNNs architectures are appearing being way more accurate. Some of them, after InceptionV3, are InceptionV4 [33] which construct a deeper network resulting in better performance; ResNets [36], which introduce a skip-connections to attain effective training eradicating the problem of vanishing gradients; SENets [34], which introduce the idea of using per-channel information to excite the network at channel-wise level; and finally EfficientNet [35] that proposes a formula for effectively scaling a network, creating a novel efficient architecture using all previous discoveries on efficiency regarding neural network computing.

III. MATERIAL AND METHODS

A. From fog to smoke transfer learning

Transfer learning is a common tool when not enough data is available during training process. It consists in using another, larger, dataset for training your network, and then use the learned weights to initialize your network when training with the smaller dataset. Better results can be obtained when both datasets share objects with common spatial features, since optimal weights will be closer on both cases. Given the spatial and temporal characteristics of smoke and fog, we can see how they share common features (see Figure 2): their color, gradient and transparency throughout the image are similar. This is due the fact that actually, they are the same object but from different points of view. Therefore, as fog datasets are larger than smoke ones, we hypothesize that better results can be obtained if we pre-train our models using common and public fog-like datasets. We will split this project in two parts: first, we will pre-train our models with fog datasets using weights from Imagenet [21]; and then, use the resultant weights to train with our smoke dataset. This way, if we pre-train our models with this first step, we will be able to see first, a comparison of wellness of proposed models with an already-known problem; and then, either if our hypothesis of pre-training was successful. Let us note, however, that our methods for fog detection are already based on transfer learning from the ImageNet dataset (see Figure 3).

B. Image Classifiers

An image-classifier chopped keeping good compromise between last-layer spatial resolution and enough parameters to do classification can be used to detect fog [31] and segment it. The resultant network specification looks as figure 4, where all the feature maps obtained from the last layer of the classification model are being convoluted with a 1×1 matrix in-depth, and then a pooling layer classifies the image. We can use the resultant attention map as a segmentation since every patch of that layer would correspond to a small part of the image itself. A cautionary remark we should note here is that this approach can give false positive segmentation results. For example, if an opaque gas object is floating around the sky with no trees nor industry around, it should be classified as a cloud. But if it is coming out from trees or a chimney, it should be classified as smoke. If we balance well the dataset, in order to have both clouds near trees, or fog near trees, at the same time that we have smoke that propagates towards the sky, we can expect this effect to diminish.

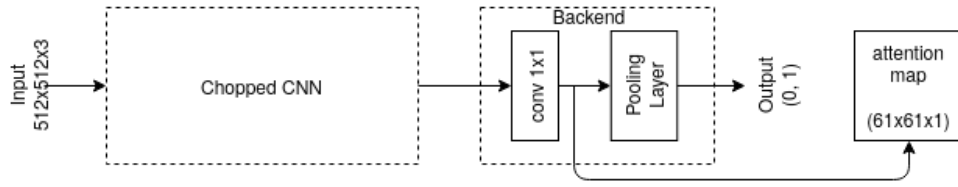


Fig. 4: Schematic of the network architecture proposed in this project.

In the original paper [31], Galdran *et. al.* use a network named InceptionV3 [32]. In this project we will explore this architecture, but also the architectures explained below:

1) *Residual Networks*: One problem that arose from deep networks is the vanishing gradients one. Weights receive an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. When the deepness of a network increases, the path that this update travels through the network gets larger, making them vanishes i.e. close to zero, making no difference in the resultant weight update. Residual Networks [36] solve this problem by putting skip-connections through layers, creating a highway for these weights to be updated without losing the gradients. ResNet networks were first proposed for image classification [36], and since then it has become an standard the use of skip-connections for training even-larger networks [37].

Their core is composed basically by the same idea as common convolutional architectures such as AlexNet [21] or VGG [24], where multiple layers composed of small convolution filters are appended one next to the other. In this work we will consider the first ResNet proposed: ResNet-18, ResNet-50, and ResNet-152. In table I we show initial phases of these three networks. As larger the network gets, the more filters are in these layers, helping the network to better classify the image, but we need to remember that deepness can annihilate our ability to segment through weakly-supervised training since there will be a mash-up of visible features on every patch of the image.

2) *Inception Networks*: Inceptions [38] are networks that uses the concept of "Network in-network" by Lin *et. al.* [39]. Larger convolutions are factorized into smaller ones obtaining lesser computation power and the same accuracy. They exploit this concept so if we look at some blocks of the network (figure 5) we can see that all convolutional layers are small and as large as 3×3 . Newer versions exploit deepness and skip-connections.

3) *Squeeze-and-Excitation Networks*: Hu *et. al.* [34] were the first to investigate the relationship between channels, creating the term of *Squeeze-and-Excitation* (SE) block, that adaptatively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. An SE block concept is when a layer (denoted as U) has a transformation F_{tr} (e.g. a convolution). This transformation is then subject to the following pipeline: first, a *squeeze* operation F_{sq} is being made that produces a per-channel descriptor of U . Then, an *excitation* F_{ex} process is applied to the resultant channel-wise descriptor, and modulates it to create a per-channel modulation weights. These weights are then re-applied to U and can be fed directly into the subsequent layers of the network.

F_{sq} ends being a global spatial information per-channel descriptor, and F_{ex} the adaptative recalibration of these channels to capture the importance that every channel has in different situations. They propose F_{sq} to be a global average pooling to create these channel-wise statistics, and F_{ex} to be a gating mechanism formed by two fully-connected layers. The output of F_{sq} is then re-scaled and multiplied by U to form the output.

4) *Efficient Networks*: CNNs are usually over-parametrized, making them susceptible to overfit. This is due the common tendency into scaling up a CNN to achieve better accuracy. Common ways are scaling by depth [36] or width [40] or even

layer name	18-layers	50-layers	158-layers
input	$512 \times 512 \times 3$		
conv1	7×7		
max pool	3×3		
conv2.x	$\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$
conv3.x	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 8$
output	$64 \times 64 \times 512$		

TABLE I: Two initial core phases for ResNet architectures that will be used. As we will see later, in fact we are chopping them, so we could have called them ResNet-10 instead of ResNet-18 for example, but that would have led to confusion since we are using original architecture's names.

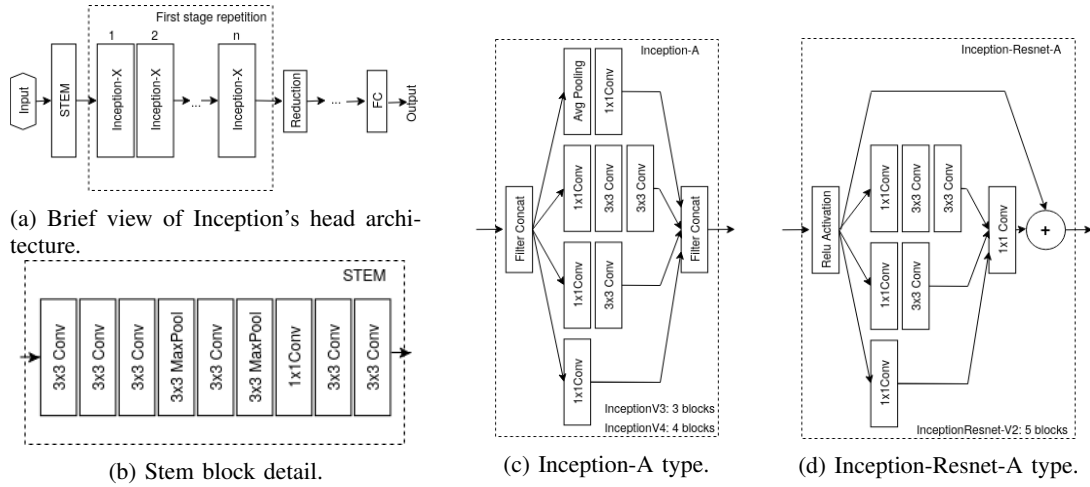


Fig. 5: Summary of important parts on *Inception* architecture analyzed. (a) depicts the overall behavior. (b) the first stage *stem*. (c) and (d) the repetition blocks and how many times they are repeated through different architectures. Inception-A is present on InceptionV3 and V4, and Inception-Resnet-A is present in InceptionResnetV2.

scaling up input size [41]. All previous works have shown hints that scaling neural networks in one way or another leads to an improve in accuracy. Tan *et. al.* [35] were the first to empirically quantify the relationship among all the three scaling parameters of the network: width, depth, and resolution. They proposed a *compound model scaling* method that relates these factors to effectively increase network size using a common coefficient ϕ :

$$\begin{aligned}
 \text{depth: } d &= \alpha^\phi \\
 \text{width: } w &= \beta^\phi \\
 \text{resolution: } r &= \gamma^\phi \\
 \text{s.t.: } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
 \alpha \geq 1, \beta &\geq 1, \gamma \geq 1
 \end{aligned} \tag{1}$$

Where α , β , and γ are constants that should to be determined using grid search. ϕ is the user-specified parameters to control how many more resources are available. Furthermore, they proceed creating a baseline efficient network (see Table II) based on these constraints, where their main block is based on *mobile inverted bottleneck* MBConv [42] [43], in which they added Squeeze-and-Excitation (see III-B3).

From here, they scale the network with different ϕ , and obtain EfficientNet-B1 to B7 [35].

C. Fine-tuning for weakly-supervised segmentation training

The last layer with receptive fields has a dimension of 61×61 . The networks that we use, firstly pretrained with *Imagenet* [58], are chopped accordingly to keep the same dimension output in the last layer. In table III we can see proposed networks and corresponding layers where these last receptive fields do not get lower than the mentioned dimension. We chop the networks at those layers and proceed to stack them with a convolution 1×1 to reduce the dimensionality to one attention map. If we pool directly on this map and train the network accordingly to our data, the patches that we can extract in this attention map

Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3	224×224	32	1
2	MBConv1, $k3 \times 3$	112×112	16	1
3	MBConv6, $k3 \times 3$	112×112	24	2
4	MBConv6, $k5 \times 5$	56×56	40	2
5	MBConv6, $k3 \times 3$	28×28	80	3
6	MBConv6, $k5 \times 5$	14×14	112	3
7	MBConv6, $k5 \times 5$	14×14	192	4
8	MBConv6, $k3 \times 3$	7×7	320	1
9	Conv 1×1 & Pooling & FC	7×7	1280	1

TABLE II: Structure of EfficientNet-B0, i.e., EfficientNet base model.

Name	Chopped at Layer		#Params
	#	Keras Name	
InceptionV3	87	mixed2	993k
InceptionV4	131	concatenate_7	1879k
InceptionResnetV2	261	block35_10_ac	1676k
Resnet18	86	add_4*	688k
Resnet50	190	add_7*	1450k
Resnet152	190	add_11*	2576k
Senet152	1045	activation_55*	3525k
Seresnet18	74	add_4*	683k
Seresnet50	156	add_7*	1613k
Seresnet152	208	add_11*	2877k
EfficientNet-B0	75	block3b_add*	68k
EfficientNet-B1	117	block3c_add*	112k
EfficientNet-B2	117	block3c_add*	138k
EfficientNet-B3	117	block3c_add*	169k
EfficientNet-B4	147	block3d_add*	278k
EfficientNet-B5	189	block3e_add*	472k
EfficientNet-B6	219	block3f_add*	667k

TABLE III: Comparison of networks used and where did we chopped them. * Output of 64×64 .

will contain the most relevant information to discriminate. Depending on the network, every pixel of this receptive field will be the result of the information contained in a certain $n \times n$ patch on the original image. This receptive field is connected to a special pooling layer to classify the image, named Shifted Weighted Average pooling.

1) *Shifted Weighted Average-Pooling*: As shown in section III-C, smoke characteristics -such as fog ones- are usually not equally distributed in the different parts of the image, and thus, common pooling layers may not tackle this problem with good effectiveness. As described in [44], the Shifted Weighted Average pooling layer (SWA) can tackle two different problems at once. This layer can be described as:

$$\mathcal{P}(I) = k + \frac{1}{n} \sum_{i=1}^n w_i \cdot p_i^I \quad (2)$$

Where I is the image, and p_i^I is a certain patch from the output of previous layer, with index i across n total patches. Every p_i^I is multiplied by weight w_i . Finally, k is used as an arbitrary learnt regulator that helps increase (or decrease) the number of total patches needed to classify I as positive.

The weight acts as a map of interest per region of the image. This might be useful when determining if an image contains smoke or not, since smoke-like parts of an image are concentrated in the center-and-upper rows of it due to its tendency of propagating vertically. Since clouds will be present during training, we can expect that in order to avoid false-negatives, this layer will also make more relevant middle parts of the image.

SWA models both the facts that some regions are more relevant, and also that a minimum region can be accepted to consider the image positive. With smoke, both ideas can be interesting, and assuming that the training dataset is well distributed, the network will decide how to modify the values during training. Let us note than in the case in which these features where not useful, the SWA layer reduces to the *Global Average Pooling* when $k = 0$, and when the weights have a constant value.

IV. EXPERIMENTS

Experiments were designed to evaluate the effectiveness of the proposed networks over weakly supervised datasets. We trained the networks using *Early Stopping* with 30 epochs of patience, by ending the training when validation accuracy stopped improving. Also, we applied a reduction on the learning rate (LR) when the validation accuracy reached a plateau with patience of 12 epochs, multiplying the LR by 0.1, initialized in all experiments at 1. Since datasets are of binary-type, we regressed over a binary cross-entropy loss function i.e., for every $y \in Q$:

$$H_p(Q) = -\frac{1}{N} \sum_{i=1}^N (y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))) \quad (3)$$

Being Q our set of samples with size N , and $p(y)$ the probability of a random sample y of being class 1.

To determine the wellness of our experiments we used different metrics that helped us to identify the best classifier for our case. First, we considered the *accuracy* of the classification, in order to analyze how both positive and negative samples are

discriminated by our network. Since we are dealing with smoke detectors, we need to have a good *recall* (i.e., how many smoke situations are correctly identified as smoke), but also lowering as much as possible the *False Positive Ratio* (FPR, i.e., specificity, how many non-smoke situations end up being selected as smoke). This is why we use a measure that combines both features, the Receiver Operator Characteristic (ROC) to see how good the classifier is with false positives and false negatives calculating the *Area Under the Curve* (AUC) of the classification.

The first experiment is what we consider the baseline, and allowed us to perform a fair comparison between the proposed chopped models and the InceptionV3 considered in the work of Galdran *et. al.* [31]. In short, we therefore test the Galdran implementation of InceptionV3 with InceptionV4, InceptionResNet, ResNet18, Resnet50, Resnet152, SEResNet18, SEResNet50, SEResNet152, SENet152, and EfficientNet-B0 to EfficientNet-B6 (EfficientNet-B7 was excluded due to implementation problems). All these networks were presented in section III. We performed transfer learning in all experiments, in particular, all these networks were originally trained for the *Imagenet* [58] dataset.

In the second and third experiments is where we focus on the smoke detection problem. In the second experiment we perform transfer learning using the weights learnt for haze in Experiment 1. In the third experiment we follow the same procedure as in the first one, but for the smoke datasets, i.e., we perform transfer learning directly for the ImageNet weights (i.e., in this case we do not use the intermediate haze prediction).

A. Data collection and preprocessing

One of the problems for smoke detection algorithms is the scarce availability of smoke datasets. A particularity of this, is that due to its direct application in surveillance, many of them are comprised only of static motion videos [45], [46].

To accomplish our goal of training a network that can detect smoke in multiple scenarios, we found the need of creating a realistic dataset that comprises both smoke and possible objects that can create confusion. For this project different datasets were collected and separated to be manipulated with ease. Two datasets were collected for our experiments, both containing positive and negative samples to avoid overfitting:

1) *Fog dataset*: As stated in Section III-C, we aim at pre-training our networks using transfer learning from previously trained weights using a fog dataset, since fog shares spatial features with smoke, and there are already works obtaining good accuracy for the fog case. In this work, we use the same of [31], which is composed of the NYU [47] and FADE [48] datasets (totalling 2K positives and 2K negative samples) for training and validation, and a random selection of 130 positive and 130 negative samples from Reside- β [49] for testing.

2) *Smoke dataset*: For smoke, we sampled and gathered images from different datasets. Most of them contained few samples, thus, a wide variety of sources are collected. For training and validation, we used the smoke images from the Fire and Smoke Dataset [50] (487 positive, 258 negative), the Smoke Dataset from Github [51] (143 positive). Also, we included an artificial wildfire dataset that was captured through a UAV [52] (99 positive, 369 negative). A Google Image search containing the words "industrial area" followed by "polygon", "pollutant", and "smoke" for positive samples; and "urban area" followed by "landscape", "skyline", "city", "town", "mountain" for negative samples were also added (287 positive, 162 negative). Finally, 229 negative samples were extracted from Places [53] dataset, and 207 negative from Harvard's Cloud dataset [54]. Testing comprises a Wildfire [55] dataset containing real images from wildfire extinction events provided by the Institute of Research of Smoke detection of Croatia (88 positive), a natural mountain walk dataset [56] (87 negative), a test set from [52] (41 positive, 88 negative), and custom images of an industry (40 positive). Totalling, we have 1131 positive and 1225 negative for training and validation, which contains smoke in different situations such as mountains and city landscapes as well as indoor and outdoor situations. We also have 169 positive and 175 negatives for testing.

B. Data augmentation

Since data provided for both datasets is scarce, we decided to use a strong data augmentation pipeline. The tool used [57] is able to do strong stochastic augmentation resulting in much larger datasets with slightly altered images. The data augmentation is a modification of the default "complex" augmentation described on their project main page. In a nutshell, we apply not only basic transformations such as vertical flip, crop, padding, and affine, but also we apply to noise, gaussian filtering, dropout schemes, hue and saturation modifiers, linear contrast, or even rain and solar glares. We checked, and in almost all of the modified cases, smoke was still detectable (Fig. 6).

V. RESULTS

A. Comparison of our models to the state-of-the-art (InceptionV3 [31])

1) *Experiment 1 with fog*: We tested all the proposed networks of Table III with the SWA pooling layer for the fog datasets (Section IV-A1). As stated before, these trainings are initialized from previous weights obtained for the *Imagenet* [58]. Table IV presents the results. We can see how the newly versions of Inception improve accuracy, and not only that, but they also lower the FPR and increase Recall, resulting in a better score on AUC. We can also see that Residual Networks also obtain better score in AUC, since AUC is not based on classification, but on the intrinsic probability of every sample being of one



Fig. 6: Some examples for data augmentation over the *wildfire* dataset. From left to right: Random color blurred glares and affine translation with edge mirroring; gaussian blur, rain and desaturation; random noise and affine translation with edge mirroring.

Name	Acc	AUC	FPR	Recall
InceptionV3	90.22	95.08	9.55	90.00
InceptionV4	90.77	95.56	9.23	90.77
InceptionResNetV2	91.92	97.14	5.69	94.62
ResNet18	90.38	96.54	10.53	89.23
ResNet50	90.77	96.01	9.84	90.77
ResNet152	87.69	96.04	14.49	84.62
SEResNet18	91.92	96.48	10.79	88.46
SEResNet50	91.15	97.25	9.77	90.00
SEResNet152	92.31	97.54	8.96	90.77
EfficientNet-B0	89.23	94.77	13.04	86.15
EfficientNet-B1	93.46	96.41	10.49	88.46
EfficientNet-B2	90.00	95.78	12.32	86.92
EfficientNet-B3	90.38	94.91	9.30	90.77
EfficientNet-B4	92.69	96.13	8.89	90.77
EfficientNet-B5	90.00	96.17	13.89	84.62
EfficientNet-B6	91.92	96.78	8.40	91.54

TABLE IV: Measurements of networks performance for Experiment 1.

class or another, but in this case, the FPR and Recall can end up being worse. We see how skip connections boost AUC in all cases, being those who use them, better than those who do not. If we go further and add Squeeze-and-Expand on ResNets (SEResNets), we see that they outperform the previous networks proving the effectiveness of using per-channel information. Finally, we see how EffectiveNet are obtaining comparable results, even better, than previous works, using way less parameters (see Table III).

2) *Experiment 2, from fog to smoke*: This experiment consisted in training with smoke the weights obtained for fog in Experiment 1. Results are shown in Table V-left. We can notice a significant decrease in accuracy when compared with the previous fog experiment. This can be happening for diverse reasons. On one hand, smoke presents higher complexity patterns than fog, and the last layer of the networks might not see enough information on the image patch presented. Also, it can be happening that our datasets need more sources to enable a good generalization of situations, since most of them just present smoke with common backgrounds such as trees or factories, and that might tend the networks to overfit for these objects (as we will discuss in Section V-B). Another thing that might be related is that our SWA layer might not be the best pool-option on this situation, since averaging a whole image containing only one little part with smoke can lead to a wrong classification. As with the previous case, in Section V-B we will see how this averaging can be causing difficulties.

Pure convolutional with skip-connections (ResNet) seem to worsen the results during classification, raising the highest false alarms and the lowest recall. Its deepness also decrease its accuracy, and that might be due to overfitting problems, since ResNet152 almost doubles its parameters compared to ResNet50, and at the same time ResNet50 doubles them from ResNet18 as we chopped them (see table III). Exciting these networks with SE blocks (see Section III-B3) seems to improve all the results, but as we will discuss in Section V-B, excitation brings other meaningful problems to segmentation.

In the other hand, EfficientNets still obtain some of the best results using narrow-like architectures in term of parameters, making them the best networks when training using smoke images. As bigger the EfficientNet is, better its classification rates, making them a network that also can avoid overfitting.

3) *Experiment 3, pure smoke training*: Our final experiment consisted in training the networks for the smoke detection case using weights from *Imagenet* [58]. The results of this experiment are shown in Table V-right. The first interesting result when comparing the results of these experiment to those of Experiment 2 is that pre-trainign for fog does not help all the networks in the same way. ResNet architectures present an inverse accuracy pattern proportional to its depth when compared to the

Name	Acc	AUC	FPR	Recall	Name	Acc	AUC	FPR	Recall
InceptionV3	81.40	95.32	25.12	59.69	InceptionV3	85.26	96.98	19.68	71.32
InceptionV4	84.91	95.75	21.32	67.44	InceptionV4	89.12	95.86	13.54	82.17
InceptionResNetV2	82.81	94.37	23.38	63.57	InceptionResNetV2	82.11	84.22	24.14	62.02
ResNet18	77.89	90.86	27.54	55.81	Resnet18	70.18	86.13	34.89	36.43
ResNet50	76.14	87.02	29.82	49.61	Resnet50	74.74	90.22	31.25	45.74
ResNet152	74.39	83.17	30.70	48.84	Resnet152	77.54	86.64	29.09	50.39
SEResNet18	82.81	84.40	22.28	66.67	Seresnet18	81.40	91.24	24.88	60.47
SEResNet50	80.35	87.60	25.73	58.91	Seresnet50	80.70	90.69	25.60	58.91
SEResNet152	79.30	93.96	27.23	55.04	Seresnet152	82.81	90.34	23.68	63.57
EfficientNet-B0	84.91	95.00	20.11	70.54	EfficientNet-B0	82.81	92.18	22.84	65.12
EfficientNet-B1	83.51	88.07	20.54	70.54	EfficientNet-B1	83.86	94.12	21.94	66.67
EfficientNet-B2	83.16	93.25	23.00	64.34	EfficientNet-B2	85.26	90.30	20.31	69.77
EfficientNet-B3	85.61	91.50	19.25	72.09	EfficientNet-B3	83.16	93.33	22.73	65.12
EfficientNet-B4	85.69	93.85	18.82	72.87	EfficientNet-B4	93.33	96.66	9.94	86.82
EfficientNet-B5	86.67	94.53	18.95	72.09	EfficientNet-B5	91.93	96.17	10.65	86.05

TABLE V: Measurements of networks performance for both Experiment 2 (left) and Experiment 3 (right).

models pre-trained using fog. This time though, depth helps ResNet to improve their result. That might be happening because in the training of experiment 2 some remnants of inner network weights that helped to classify fog, where untouched during training with smoke, worsening the results.

Pure Inception architectures improve their accuracy without fog pre-train, but when using residual layers the accuracy is roughly the same. Lower architectures of EfficientNets present roughly the same accuracy, but then, bigger architectures tend to classify better smoke. If we see the layer where they are chopped (Table III), EfficientNet-B4 has one extra layer than the previous architecture, being it a possible indicator that the deepness of EfficientNet helped resolve this task better. For EfficientNet-B4, we obtain a False Positive Rate lower than 10%, and the best recall; but although it is the best network with this dataset, these results are not yet acceptable for industrial applications, indicating that we might need to perform further research in this direction.

B. Qualitative Results

Our approach allows the networks to predict presence of smoke from image-level ground-truth, creating attention maps where excited regions are those that lead to a positive prediction due to the backend pooling mechanism. This has the advantage that active regions are learned from data, avoiding any manual input of specifying where smoke is [31], and also, these attention maps are performing a detection of the regions in which the smoke is presented, that can be visualized (Figure 7). There are some qualitative differences among all networks. We will discuss the disadvantages that they can bring up in following section.

C. Failure Cases

Although we obtained good AUC, classification was still lower than expected. Here we analyze qualitatively, some of the problems that might have appeared during training, that can lead to confusion when weakly-segmentating our results:

1) *Problem with correlating smoke with other structures.*: In figure 8 we can see how InceptionV4 in Experiment 3 behaves with some images of our test set. Attention maps show high excitation where smoke is present. Nonetheless, other parts of the image trigger the neurons as well. As explained in previous sections (III-B, V-A3), this might be due to the fact that some networks expect larger regions of the image to be smoke, in order to be classified. Although this, highest excitation tend to be smoke.

2) *Problem with using pretrained weights from fog.*: In figure 9 we can see another interesting example, that can happen when pre-training with fog. Architectures like ResNet50 tend to classify only those regions that showed transparency. This might happen because our training dataset contained high samples of smoke with transparency parts, leading some of our networks to use only those regions for classification. More samples containing both smoke and fog might help mitigate this problem.

3) *Problem detecting only smoke.*: In these example we show the results from the same network as before (ResNet50) but trained with smoke directly using weights from Imagenet. In figure 10 present some images that were False Negatives in the classification. The network successfully detected where smoke is, but perhaps a lack of excitation, or using a wrong backend lead the network to classify negatively these images. An idea to fix this problem may deal with replacing the pooling mechanism, or extending the network.

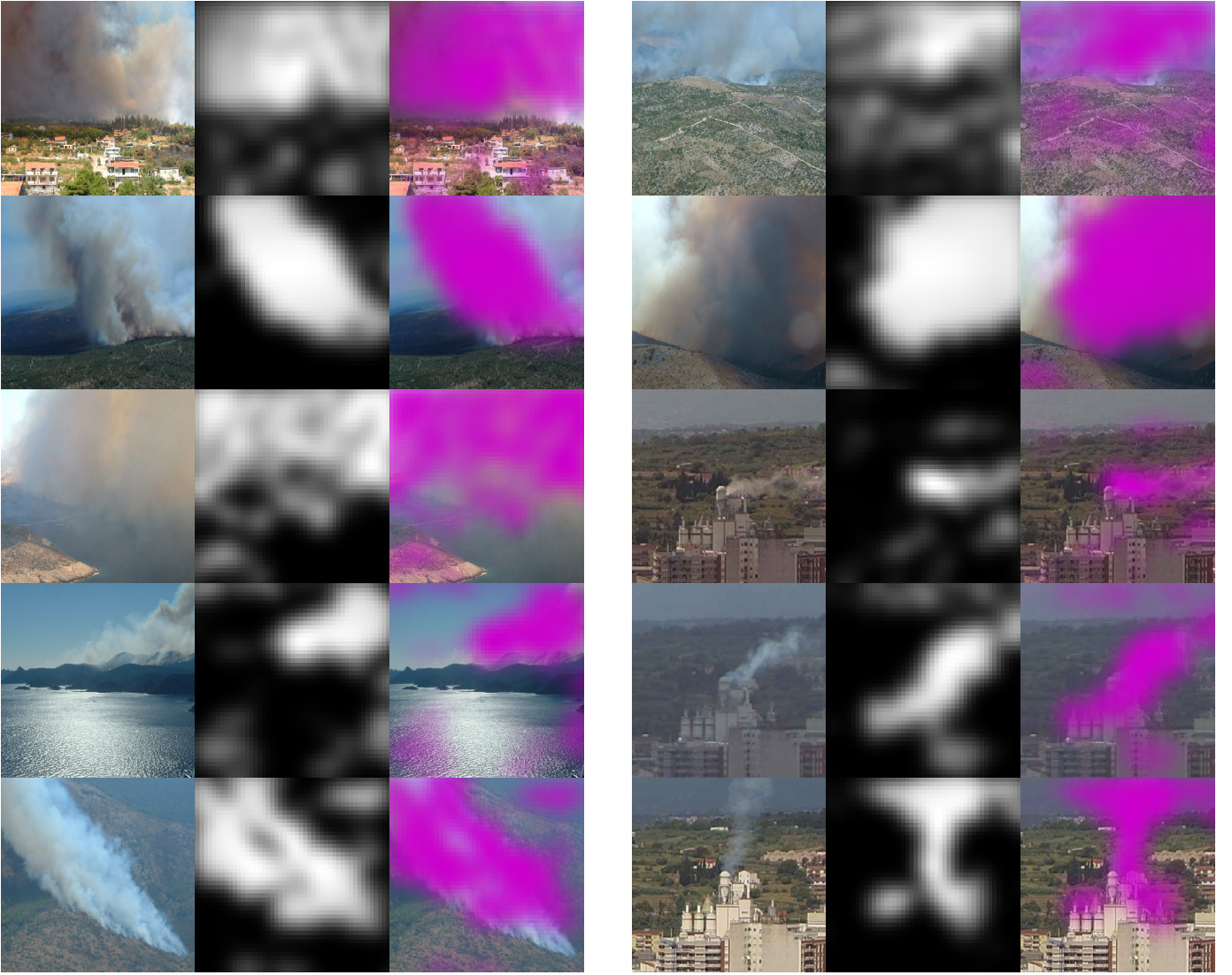


Fig. 7: Visual responses of our methods in Experiment 3. In rows from top to bottom: InceptionV4, ResNet152, EfficientNet-B4, and the two last rows from ResNet50.



Fig. 8: Visual responses using InceptionV4 of Experiment 3.

4) *Over excitation through channels:* A problem arises when using excitation mechanisms for segmentation of images. Networks that uses SE blocks tend to detect regions without smoke, but that can lead to a positive classification (for example, trees or industries, see Figure 11). The problem that comes when we using excitation mechanisms is that it may burn out the excitation of attention maps (as shown in Figure 11), where we show images of positive samples from SEResNet-18.

D. Experiment discussion

We have seen different problems that may happen when dealing with smoke detection, and solving all of them at the same time is not an easy task with the proposed architectures. In a common CNN architecture, neurons do not see most of image information until last layers. Since we chopped them, a neuron in the attention map produced can only see one region of the



Fig. 9: Visual responses using ResNet50 of Experiment 2.



Fig. 10: Visual responses using ResNet50 of Experiment 3.

image, and as discussed in Section III-A, other objects have some characteristics that can lead to confusion when classifying smoke. This results in a neuron in our attention map not being sure at 100% of the patch it is seeing of being a positive sample (see Figure 2). Thus, that neuron from the attention map tends to not be saturated (i.e., being a value extremely high), expecting other neurons to be activated too. That leads to wrong classification when smoke is present in tiny parts of the image (i.e., far away from the camera). The regulatory parameter k in the SWA pooling layer seems not to be high enough. It may be happening because our training images contain more smoke than test ones, since architectures seems to rely more on smoke information itself and expect it to be in enough patches of the image in order to be averaged as positive sample. This is a hint that we might be suffering from *dataset bias*, a common problem when generating new datasets, where distribution of possible samples is not well balanced between training and test.

Solving the above explained problem might require burning these regions, but would at the same time lead to wrong classification of images of clouds/fog near trees. Thus, networks tend to increase their need to use other parts of the image, worsening the weakly segmentation results (Figure 8). Nonetheless, if we only need a good weakly-segmentator, without need of classification results, ResNet50 from figure 10 would be a good candidate, since it has ended just using smoke information without relying strongly of other image-related features. On the other hand, if we need both features at the same time (classification and segmentation), our trained InceptionV4 from Experiment 3 might be the best solution. Other combinations could be using the best classifier (EfficientNet-B4 from Experiment 3) and our best segmentator (some ResNet from experiment 3) to do both tasks separately.

VI. CONCLUSIONS

We have been able to improve previous results on weakly-supervised fog detection using newly and state-of-the-art architectures. We also tested a wide variety of networks and saw that, although some of them can classify better, segmentation differs a lot when using different classifiers. Problems related with the visual complexity of smoke lead to a difficult weakly-supervised segmentation from image-level information only, making the classifier to create a high false alarms rate and worsening the recall. But, if we take only positive samples though, we have been able to train interesting image segmentators such as ResNet50 from Experiment 3, that leads to positive response only where smoke is present. We believe that our ResNet50 has shown good image-segmentation results (for further examples, refer to Appendix A). We have also seen that the backend plays a crucial role when creating this segmentation map and that using a good backend can be a challenging problem when it comes to segmentate smoke in a weakly-supervised way. SWA presented some flaws, that we are studying for further improvements.

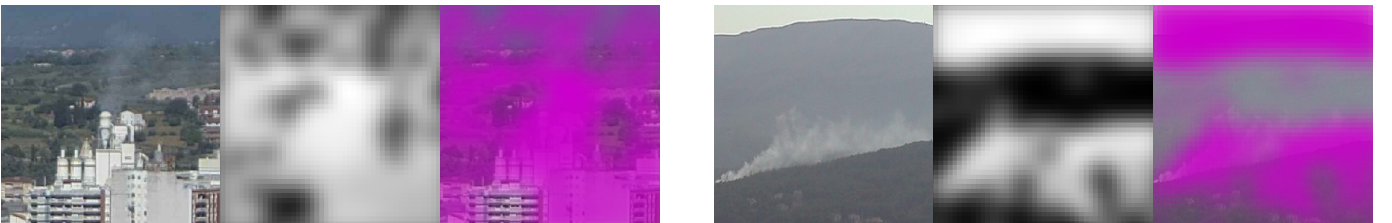


Fig. 11: Visual responses using SEResNet-18 of Experiment 3.

VII. FUTURE WORK

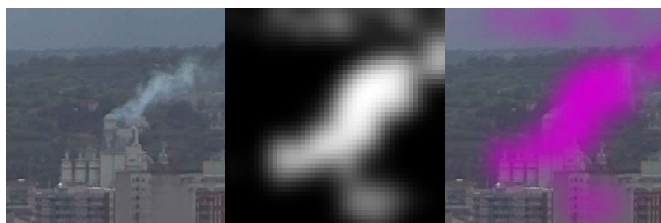
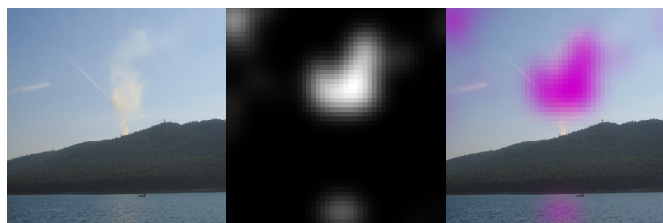
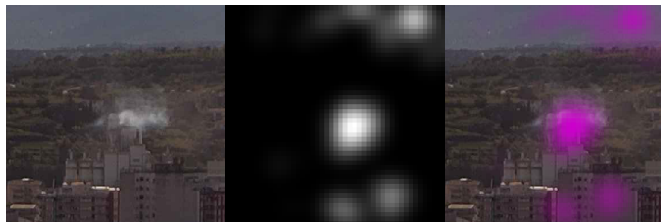
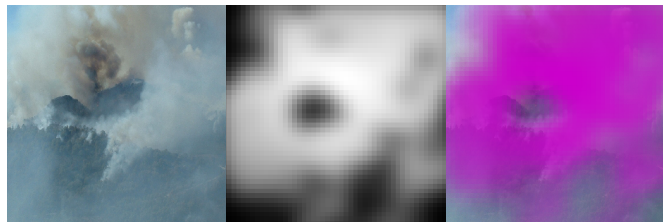
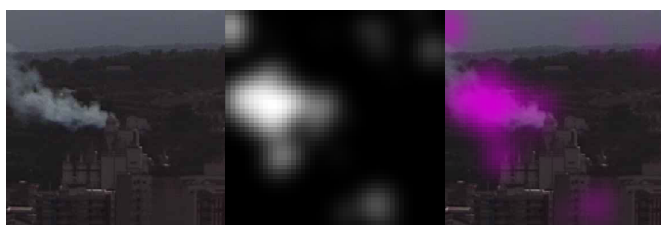
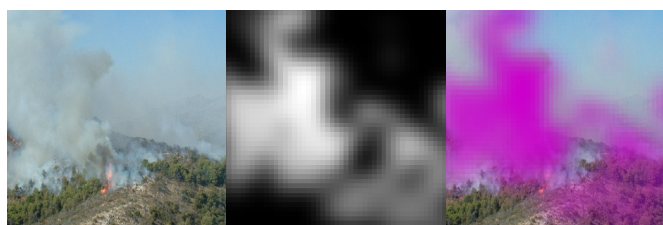
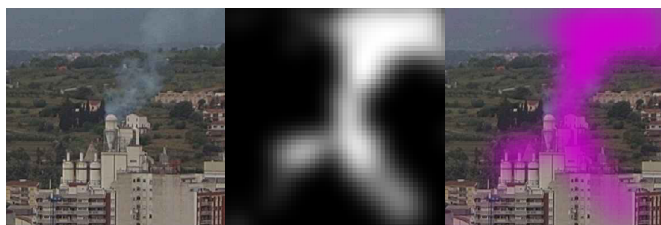
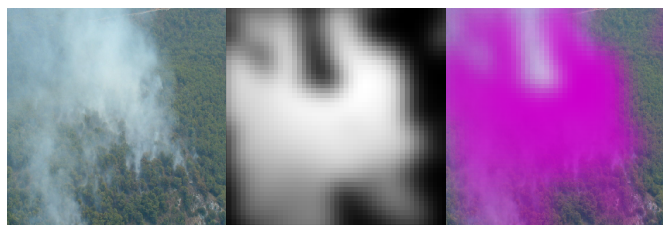
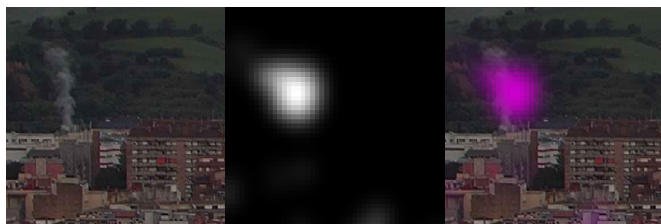
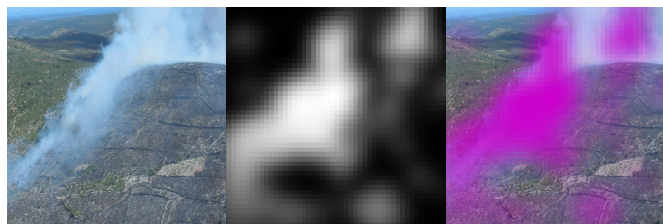
Smoke presents challenging characteristics. Our dataset was not solid enough to reach acceptable results for industrial monitorization. Thus, we believe that more data needs to be gathered for training networks in smoke detection. Recently, Hsu *et. al.* [59] have published a new dataset containing almost 200 videos of image-labeled information of smoke industrial pollutants. Using these, and other video sources for smoke detection [60] [61] [62] might help improve the performance. This said, these datasets may well be enough for the classification case, but for segmentation even larger datasets might be in need. Nowadays, simulation algorithms are improving a lot, and particle-based simulations are getting extremely better. Tools like *mantaflow* [63] for fluid simulation start to be a good alternative to simulate some low-resolution high-amount of data, even videos. That is why our team has started a project to keep improving these results, using synthetic data with these type of tools.

Also, smoke presents strong temporal characteristics that makes it easier to recognize. If we solve the missing datasets problems, works like [28] [26], exploring RCNNs with newly EfficientNet, using a new approach such as EfficientNet-LSTM [64] or a 3D-EfficientNet could effectively exploit temporal characteristics. This way, we could also improve weakly-supervised smoke detection if we employ temporal characteristics from video-based labels. With the weights extracted from every network proposed, if a 3D architecture is proposed, we could use inflation techniques for already-learned 2D filters to create 3D filters and start from here for the C3DNN.

ACKNOWLEDGMENT

I would like to thank Javier Vázquez Corral and Anton Gomà for their help, suggestions and support they gave me throughout the process of this project. Also thank to every friend, family and colleague who supported during this tough times; specially to my long-time university friend Richard Segovia.

APPENDIX A OTHER VISUAL RESPONSES FROM RESNET-50



REFERENCES

- [1] M. Rico, L. Font, J. Airmon, M. Mari, and A. Gomez, "Informe de la qualitat de l'aire de barcelona," Agència de la Salut Pública de Barcelona, 2018.
- [2] C. A. Pope III and D. W. Dockery, "Health effects of fine particulate air pollution: lines that connect," *Journal of the air & waste management association*, vol. 56, no. 6, pp. 709–742, 2006.
- [3] Council of European Union, "Directive 2008/50/ec of the european parliament and of the council on ambient air quality and cleaner air for europe," Official Journal of the European Union, May 2008, <https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX%3A32008L0050>.
- [4] M. Jiang, E. Kim, and Y. Woo, "The relationship between economic growth and air pollution—a regional comparison between china and south korea," *International Journal of Environmental Research and Public Health*, vol. 17, no. 8, p. 2761, 2020.
- [5] G. M. Grossman and A. B. Krueger, "Environmental impacts of a north american free trade agreement," National Bureau of economic research, Tech. Rep., 1991.
- [6] K. Uchiyama, "Environmental kuznets curve hypothesis," in *Environmental Kuznets curve hypothesis and carbon dioxide emissions*. Springer, 2016, pp. 11–29.
- [7] A. M. Rasli, M. I. Qureshi, A. Isah-Chikaji, K. Zaman, and M. Ahmad, "New toxics, race to the bottom and revised environmental kuznets curve: The case of local and global pollutants," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 3120–3130, 2018.
- [8] "Least developed countries (ldcs)," United Nations. *Economic Analysis*. Department of Economic and Social Affairs, <https://www.un.org/development/desa/dpad/least-developed-country-category.html>.
- [9] F. Karagulian, C. A. Belis, C. F. C. Dora, A. M. Prüss-Ustün, S. Bonjour, H. Adair-Rohani, and M. Amann, "Contributions to cities' ambient particulate matter (pm): A systematic review of local source contributions at global level," *Atmospheric environment*, vol. 120, pp. 475–483, 2015.
- [10] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.
- [11] D. T. Gottuk, J. A. Lynch, S. L. Rose-Pehrsson, J. Owrutsky, and F. W. Williams, "Video image fire detection for shipboard use," *Fire safety journal*, vol. 41, no. 4, pp. 321–326, 2006.
- [12] C. Brunner, T. Peynot, T. Vidal-Calleja, and J. Underwood, "Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire," *Journal of Field Robotics*, vol. 30, no. 4, pp. 641–666, 2013.
- [13] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming, "Video fire smoke detection using motion and color features," *Fire technology*, vol. 46, no. 3, pp. 651–663, 2010.
- [14] T. K. T. Nguyen and J.-M. Kim, "Multistage optical smoke detection approach for smoke alarm systems," *Optical Engineering*, vol. 52, no. 5, p. 057001, 2013.
- [15] F. Yuan, "A fast accumulative motion orientation model based on integral image for video smoke detection," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 925–932, 2008.
- [16] W. Ye, J. Zhao, S. Wang, Y. Wang, D. Zhang, and Z. Yuan, "Dynamic texture based smoke detection using surfacelet transform and hmt model," *Fire Safety Journal*, vol. 73, pp. 91–101, 2015.
- [17] H. Maruta, A. Nakamura, and F. Kurokawa, "A new approach for smoke detection with texture analysis and support vector machine," in *2010 IEEE International Symposium on Industrial Electronics*. IEEE, 2010, pp. 1550–1555.
- [18] Y. Chunyu, Z. Yongming, F. Jun, and W. Jinjun, "Texture analysis of smoke for real-time fire detection," in *2009 Second International Workshop on Computer Science and Engineering*, vol. 2. IEEE, 2009, pp. 511–515.
- [19] F. Yuan, "Video-based smoke detection with histogram sequence of lbp and lbpv pyramids," *Fire safety journal*, vol. 46, no. 3, pp. 132–139, 2011.
- [20] S. Calderara, P. Piccinini, and R. Cucchiara, "Vision based smoke detection system using image energy and color information," *Machine Vision and Applications*, vol. 22, no. 4, pp. 705–719, 2011.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [22] S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 877–882.
- [23] S. Khan, K. Muhammad, S. Mumtaz, S. W. Baik, and V. H. C. de Albuquerque, "Energy-efficient deep cnn for smoke detection in foggy iot environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9237–9245, 2019.
- [24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [25] F. Zhang, W. Qin, Y. Liu, Z. Xiao, J. Liu, Q. Wang, and K. Liu, "A dual-channel convolution neural network for image smoke detection," *Multimedia Tools and Applications*, pp. 1–17, 2020.
- [26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.
- [27] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375.
- [28] G. Lin, Y. Zhang, G. Xu, and Q. Zhang, "Smoke detection on video sequences using 3d convolutional neural networks," *Fire Technology*, vol. 55, no. 5, pp. 1827–1847, 2019.
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [30] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.
- [31] A. Galdran, P. Costa, J. Vazquez-Corral, and A. Campilho, "Weakly supervised fog detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 2875–2879.
- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.
- [34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [35] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [37] —, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.
- [38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [39] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
- [40] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.
- [41] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu et al., "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *Advances in neural information processing systems*, 2019, pp. 103–112.

- [42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [43] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.
- [44] P. Costa, A. Campilho, B. Hooi, A. Smailagic, K. Kitani, S. Liu, C. Faloutsos, and A. Galdran, "Eyequal: Accurate, explainable, retinal image quality assessment," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 323–330.
- [45] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545–1556, 2015.
- [46] K. Dimitropoulos, P. Bampoutis, and G. Nikos, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, pp. 339–351, 02 2015.
- [47] C. Ancuti, C. O. Ancuti, and C. De Vleeschouwer, "D-hazy: A dataset to evaluate quantitatively dehazing algorithms," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2226–2230.
- [48] L. K. Choi, J. You, and A. C. Bovik, "Referenceless prediction of perceptual fog density and perceptual image defogging," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3888–3901, 2015.
- [49] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2018.
- [50] A. Upadhyay, "Fire and smoke dataset," <https://www.kaggle.com/ashutosh69/fire-and-smoke-dataset>, 2020, online; accessed 13-Nov-2020.
- [51] rekon, "Smoke detection via semantic segmentation using baseline u-net model and image augmentation in keras," <https://github.com/rekon/Smoke-semantic-segmentation>.
- [52] Q. Zhang, "Smoke detection based on scene parsing and salient segmentation," <http://smoke.ustc.edu.cn/datasets.htm>.
- [53] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [54] P. Liu, "Cirrus Cumulus Stratus Nimbus (CCSN) Database," 2019. [Online]. Available: <https://doi.org/10.7910/DVN/CADDPD>
- [55] D. K. Toni Jakovcovic, "Image and video databases from wildfire observers and smoke recognition," Wildfire Observers and Smoke Recognition of Croatia, http://wildfire.fesb.hr/index.php?option=com_content&view=article&id=62&Itemid=72.
- [56] G. Xu, Y. Zhang, Q. Zhang, G. Lin, Z. Wang, Y. Jia, and J. Wang, "Video smoke detection based on deep saliency network," *Fire Safety Journal*, vol. 105, pp. 277–285, 2019.
- [57] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," <https://github.com/aleju/imgaug>, 2020, online; accessed 01-Feb-2020.
- [58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [59] Y.-C. Hsu, T.-Y. Hu, P. Dille, S. Prendi, R. Hoffman, A. Tshlars, R. Sargent, I. Nourbakhsh *et al.*, "Rise video dataset: Recognizing industrial smoke emissions," *arXiv preprint arXiv:2005.06111*, 2020.
- [60] G. Lin, Y. Zhang, Q. Zhang, Y. Jia, G. Xu, and J. Wang, "Smoke detection in video sequences based on dynamic texture using volume local binary patterns," *TIIS*, vol. 11, no. 11, pp. 5522–5536, 2017.
- [61] B. University, "Fire and smoke dataset," <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/>, 2020, online; accessed 16-Nov-2020.
- [62] C. Vision and K. Pattern Recognition Laboratory, Keimyung University, "Kmu fire smoke database," <https://cvpr.kmu.ac.kr/>, 2020, online; accessed 16-Nov-2020.
- [63] N. Thuerey and T. Pfaff, "Mantaflow," 2016.
- [64] X. Yin, D. Wu, Y. Shang, B. Jiang, and H. Song, "Using an efficientnet-1stm for the recognition of single cow's motion behaviours in a complicated environment," *Computers and Electronics in Agriculture*, vol. 177, p. 105707, 2020.