sensors

MDPI

Article

Augmenting Deep Learning Performance in an Evidential Multiple Classifier System

Jennifer Vandoni [1,2], Sylvie Le Hégarat-Mascle [1,*] and Emanuel Aldea [1]

[1] SATIE-CNRS UMR 8029, Paris-Sud University, Paris-Saclay University, 91405 Orsay CEDEX, France; jennifer.vandoni@gmail.com (J.V.); emanuel.aldea@u-psud.fr (E.A.)
[2] SAFRAN SA, Safran Tech, Pole Technologie du Signal et de l'Information, 78772 Magny-les-Hameaux, France
* Correspondence: sylvie.le-hegarat@u-psud.fr; Tel.: +33-169-154-036

Received: 30 September 2019; Accepted: 24 October 2019; Published: 27 October 2019

Abstract: The main objective of this work is to study the applicability of ensemble methods in the context of deep learning with limited amounts of labeled data. We exploit an ensemble of neural networks derived using Monte Carlo dropout, along with an ensemble of SVM classifiers which owes its effectiveness to the hand-crafted features used as inputs and to an active learning procedure. In order to leverage each classifier's respective strengths, we combine them in an evidential framework, which models specifically their imprecision and uncertainty. The application we consider in order to illustrate the interest of our Multiple Classifier System is pedestrian detection in high-density crowds, which is ideally suited for its difficulty, cost of labeling and intrinsic imprecision of annotation data. We show that the fusion resulting from the effective modeling of uncertainty allows for performance improvement, and at the same time, for a deeper interpretation of the result in terms of commitment of the decision.

Keywords: deep learning; ensemble classifiers; Belief Function Theory; pedestrian detection; high-density crowds

1. Introduction

Even though deep learning solutions tend to outperform the other supervised learning techniques when trained on large amounts of data, applying them effectively in presence of few labeled data is nowadays an open issue. Most of the existing works are devoted to finding the best network for applications for which huge datasets exist, but few attention is given to specific real-setting problems where training data are hard to obtain and therefore out-of-the-box networks may be impossible to be trained. Nonetheless, in recent years, many regularization techniques have been proposed to tackle the problem of overfitting, from data augmentation to early stopping and dropout, besides the traditional weight decay. These techniques used together could help in applying deep learning techniques in the presence of small datasets. In addition to these techniques, fusion with another strong classifier may be considered.

Simultaneously, a criticism that is often made of deep learning methods is the fact that they act like "black-boxes", making it hard for their users to interpret the obtained results. This limitation is highly relevant when learning from small amounts of data, where a measure of model uncertainty would be particularly important. To this extent Bayesian Neural Networks (BNNs, Bayesian NNs) offer a probabilistic interpretation of deep learning models by inferring distributions over the models' weights, allowing to measure model uncertainty, but they are usually practically limited. Recently, an ensemble-based method relying on the use of dropout at inference time has been proposed in [1] (Monte Carlo dropout), allowing to obtain several realizations sampled from the same network with randomly dropped-out units at test time, from which a confidence measure on the prediction can be derived.

Following this line of work, we intend to investigate the use of deep learning techniques in presence of small training datasets for specific applications (in our case high-density crowd pedestrian detection). A solution proposed for instance by [2] in the case of hyperspectral data is to reduce the number of weight parameters required to train the model by considering some constraints related to the physical interpretation of the weights. In this work, the type of the data (grayscale images) is not suitable for such prior constraints, we propose the use of an ensemble method that is justified according to two different reasons. Firstly, it acts as another regularization technique to mitigate the risk of overfitting; secondly, it allows us to measure the model confidence about each prediction. To this extent, we propose to work in the context of the Belief Function Theory [3–5] (BFT) to better leverage the classifier's unique properties. The evidential framework [6–10] is indeed able to naturally model the concept of *imprecision* in addition to the uncertainty value provided by the classifiers.

We thus propose an evidential Multiple Classifier System (MCS), which is in turn composed by two ensembles of classifiers. The first one, called CNN-ensemble, is an ensemble of convolutional neural networks (CNNs) derived using the Monte Carlo dropout technique. The second one, called SVM-ensemble, is an ensemble of Support Vector Machine (SVM) classifiers trained with different descriptors in an active learning (AL) Query-by-Committee fashion previously proposed in [11]. Specifically, starting from a single sensor input, i.e., an image lattice, we derive two different ensembles based on complementary classifiers. These two ensembles are then considered as different information sources, like virtual sensors.

We apply the proposed Evidential MCS to the difficult application of high-density crowd pedestrian detection for multiple reasons. Indeed, although in the last years, many efforts have been devoted to improve the performance of pedestrian detection [12], baseline methods cannot be always applied in crowded contexts because of scarce labeled data, and intrinsic differences with respect to the sparse case which may be cause of imprecision in the final detection results.

Pedestrian detection by itself is noticeably one of the most challenging categories of object detection. There exists indeed a large variability in the local and global pedestrians' appearance, due to the variety of possible body shapes, or different styles and types of clothes and accessories which may alter the silhouettes of the individuals. Besides, in real-world scenarios several people can occupy the same region, partially occluding each other, and this phenomenon becomes more prevalent as the crowd density increases.

Traditionally, in the context of supervised learning for pedestrian detection, the Histogram of Oriented Gradients (HOG) descriptor [13] has been proposed for the scope, but its performance can be easily affected by the presence of background clutter and occlusions. Alternatively, deformable part-based models [14] consider the appearance of each part of the body and the deformation among parts for detection. In parallel with the development of traditional approaches, more sophisticated methods [15] proposed a cascade Random Forest classifier with a census transform histogram visual descriptor. Recently, neural networks have been employed in the context of pedestrian detection, either in conjunction with hand-crafted features [16,17], or in a stand-alone manner [18–21]. In [22], deep features, deformation handling, occlusion handling, and classification are jointly learnt for pedestrian detection. Late fusion of multiple convolutional layers has been recently proposed in [23] relying on Region Proposal Networks (RPNs), showing that earlier convolutional layers are better at handling small-scale and partially occluded pedestrians. A Scale-Aware Fast R-CNN framework has been also proposed in [20]. Again, in presence of dense crowds, the region proposal step loses its interest as the number of targets becomes too large to be tractable. Finally, let us recall that if large datasets with dotted annotations corresponding to head's center coordinates are available for applications such as cross-scene people counting, only little attention has been given to specific scenes for the more difficult task of pedestrian detection. For all the highlighted limitations, a straightforward extension of the techniques designed for pedestrian detection in non-crowded scenes is not suitable for dealing with crowded situations.

In this work, we show that deep learning techniques can still be used in conjunction with a traditional classifier even in such difficult situations, where we are interested in specific scene analysis for which labeled data is scarce and not precise (i.e., we just have the head's center coordinates instead of bounding boxes or precise segmentation maps, contrarily to traditional pedestrian detection setting). This paper is organized as follows. Section 2 describes the proposed methodology that allows for combination of CNN-ensemble and SVM-ensemble. Firstly, we propose a fully convolutional network especially designed to recover small objects and we create a CNN-ensemble through Monte Carlo dropout. The whole output is interpreted in the Belief Function framework. Then, having described the SVM-ensemble derived from [11] work, we explain how the two ensembles are fused together in the context of BFT. Section 3 shows to which extent the proposed approach has been able to improve the overall detection results even in such a difficult setting, by analysing the obtained results. In this way, we prove that deep learning techniques can be applied also in presence of extremely small datasets for solving targeted problems, and can benefit from the fusion with another strong classifier. Finally, Section 4 draws the conclusions of this study.

2. Evidential Multiple Classifier System

In this section we go through the details of the proposed evidential MCS for pedestrian detection. This method is particularly useful in presence of scarce training data, where recent deep learning techniques may fail to obtain reliable predictions. The simultaneous use of two different ensemble of classifiers, namely a deep learning-based one and a SVM-based one, allows us to exploit their different strengths in order to obtain a robust prediction. Note that this is not straightforward, since in presence of few, strong classifiers the fusion strategy must be particularly well-designed in order to exploit their respective peculiarities, and for this reason we propose a fusion in the context of BFT which allows us to obtain a measure of localized imprecision in addition to more robust predictions.

Figure 1 shows a simplified scheme of our approach which combines two ensembles of different classifiers in an evidential framework. The CNN-ensemble is obtained by sampling the posterior distribution of the proposed FE + LFE network through Monte Carlo dropout technique [1], while the SVM-ensemble is obtained by exploiting different hand-crafted features in an active learning framework. The output of those ensembles, $\mathcal{M}_{CNN}$ and $\mathcal{M}_{SVM}$ respectively, are then combined in the context of BFT to obtain the final output map $\mathcal{M}$. In the following, after a brief introduction of the BFT that will play a key role in our method to model the concept of imprecision, we firstly detail the proposed CNN-ensemble, then we recall the SVM-ensemble, already proposed in [11], and finally, we explain their fusion procedure in the context of BFT.

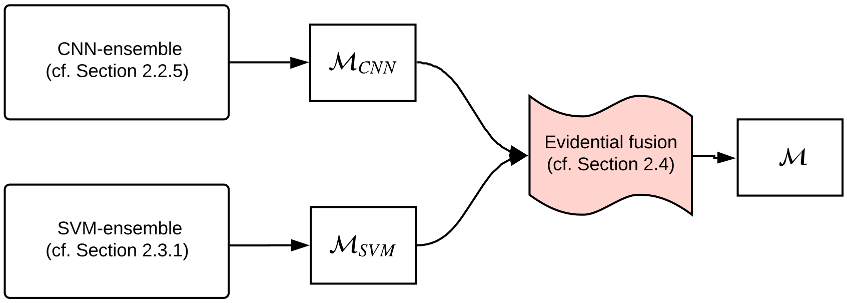


Figure 1. Proposed evidential MCS scheme.

2.1. Modeling Imprecision with BFT

To handle both the uncertainty and the related imprecision that, in our case, may come from the specific classifier and/or data used in the training process, BFT [3,4] is designed to handle compound hypotheses. If Θ denotes the set of mutually exclusive hypotheses (i.e., the discernment frame), belief functions are defined on the powerset 2^{Θ}. In our case, denoting by H and $\bar{H}$ the two singleton hypotheses, *"Head"* and *"Not Head"*, the discernment frame is $\Theta = \{H, \bar{H}\}$, and the set of hypotheses is $2^{\Theta} = \{\emptyset, H, \bar{H}, \{H, \bar{H}\}\}$.

The *mass* function noted m is the *Basic Belief Assignment* (BBA) that satisfies $\forall A \in 2^{\Theta}, m(A) \in [0, 1]$, $\sum_{A \in 2^{\Theta}} m(A) = 1$. The hypotheses for which the mass function is non-null are called *focal elements*. If only singleton hypotheses are focal elements, the BBA is called Bayesian.

Then, other BF are in one-to-one relationship with m. In this particular setting in which we have only two singleton hypotheses and considering $m(\emptyset) = 0$, the *plausibility* and the *credibility* functions noted Pl and Bel respectively are defined by: $\forall A \in \{H, \bar{H}\}, Bel(A) = m(A)$ and $Pl(A) = m(A) + m(\Theta)$. Pl and Bel are dual functions: $\forall A \in 2^{\Theta}, Pl(A) = 1 - Bel(\bar{A})$ (where $\bar{A}$ denotes the complement of A with respect to Θ). Since in our case we work with only two singleton hypotheses, the equations simplify and we will introduce them directly when needed. However, we encourage the interested reader to refer to the seminal works [3,4] for a more detailed explanation about BFT foundations.

2.2. CNN-Ensemble

2.2.1. Representing Model Uncertainty in Deep Learning

Obtaining a measure of uncertainty of a model trained with deep learning techniques is not trivial. The general training of deep learning models allows us to obtain the best model parameters through backpropagation, but they are usually only point estimates. These parameters are then kept fixed at inference time in the forward pass to perform prediction. However we cannot easily know whether a trained model is certain about its output, and no classifier is able to directly provide credible or confidence intervals about its predictions.

We want the network to be able to measure predictive uncertainty, that is the confidence it has with respect to the prediction it makesThis is particularly important for applications related to real-settings such as autonomous driving or security access to critical systems, where relying on model uncertainty to adapt decision making is crucial, and generally for applications for which only a small amount of data is available for the training.

To tackle this issue, BNNs have been firstly studied extensively in [24,25] and more recently in [26–28], sometimes being referred to as variational techniques. They are based on the observation that an infinitely wide neural network with distributions placed over its weights converges to a Gaussian process [25], thus, considering finite neural networks (NNs), they are mathematically equivalent to an approximation of the probabilistic deep Gaussian process [29]. In order to obtain model uncertainty estimates, BNNs place a prior probability distribution over each networks' weight. In this way, they potentially offer robustness to overfitting during training along with uncertainty estimates about the predictions. However, the applicability of these types of models is quite limited, and they have not been largely followed up by the deep learning community. If on the one hand Bayesian probability theory offers mathematically grounded tools to reason about model uncertainty, on the other hand they come usually with prohibitive computational costs. BNNs have shown indeed to be quite difficult to work with, often requiring the optimization of many more parameters with respect to standard networks.

Recently, the authors of [1] developed a new theoretical framework by casting *dropout* (and its variants) in deep NNs as approximate Bayesian inference in deep Gaussian processes. The foundation of this theory directly provides tools to model the uncertainty without the need to change neither the model architecture nor the objective function. The authors have shown that a neural network with arbitrary depth and non-linearities, with dropout applied at every layer, is mathematically equivalent to an approximation of the probabilistic deep Gaussian process. This means that the optimal weights found through the optimization of a NN with dropout are the same as the optimal variational parameters in a Bayesian NN with the same structure. Further, this means that a network already trained with dropout *is* indeed a BNN. Moreover, this result is valid not only using dropout, but also its variants such as DropConnect [30] or multiplicative Gaussian noise [31], i.e., using any Stochastic Regularization Technique (SRT). SRTs are techniques used to regularize a deep learning model through the injection of stochastic noise directly into it (the most popular technique is dropout which switches off units with respect to a previously set probability). The intuition is that SRTs approximately integrate over the models' weights, so that they can be interpreted as performing approximate inference and, as a result, uncertainty information can be extracted.

Practically, after training the network, Monte Carlo (MC) methods are used at test time to draw samples from a Bernoulli distribution across the network's weights, by performing T stochastic forward passes through the network with dropout. This is why the method is known as *MC-dropout*. Note that this does not require any additional parametrization, and from this it is easy to derive the sample mean by averaging the results and the standard deviation that can be interpreted as predictive uncertainty. In this way, we can obtain an ensemble of classifiers composed by T different realizations given by dropping out different units of the network at each forward pass. This method has several advantages, i.e., it is easily adaptable to complex models and does not require any change to the model architecture or optimization procedure (the training is only performed once, while dropout is applied at test time), besides being very easy to implement in practice. MC-dropout has been successfully applied in different applications, from segmentation for scene understanding [32] to camera re-localization [33], allowing to model the predictive uncertainty through standard deviation of the stochastic realizations obtained with dropout.

We shall finally mention a different but related approach, that cannot be considered as approximate inference in BNNs, but which may nevertheless be used to estimate model uncertainty relying on ensemble learning. This technique builds an ensemble of deterministic models (each model in the ensemble produces a point estimate rather than a distribution) by independently training the same network on the same dataset many times with different weight initialization. Then, at inference time, an average is done in order to get a prediction, while the predictive uncertainty is measured through the variance of the outputs of all the models. Very recently, reference [34] proposed *deep ensembles* based on this idea, relying also on adversarial training [35,36] to smooth predictive distributions, treating the ensemble built in this way as a uniformly-weighted mixture model and approximating the ensemble prediction as a Gaussian whose mean and variance are the ones of the mixture respectively. However, this approach is not always suitable for a number of reasons. Firstly, training many neural networks can be a long process. Secondly, even if this approach is anyhow computationally more efficient than many Bayesian approaches presented in the previous section, its produced uncertainty estimates lack in many ways [37].

Now, we firstly detail the ground truth construction, then we describe the architecture we designed for our application, namely pedestrian detection in high density crowds, and then we formulate its Bayesian counterpart that makes use of MC-dropout to get samples from the posterior distribution over the network's weights. Finally, we move to the BFT to explain the proposed BBA allocation and combination.

2.2.2. Soft-Labeling

In our specific environment, precise labeling to perform head detection is usually impossible to achieve, due to the presence of clutter and occlusion problems that make the contour of the heads barely distinguishable from the background, in addition to the very small size of the targets. A precise definition of head borders is thus difficult even for a human operator. Therefore, commonly used datasets do not come with precise segmentation ground-truth but rather with just a list of coordinates that indicate the center of the heads. For these reasons, we investigate the problem of head detection from partially labeled data, namely where only the center of each head is dot-annotated, with only a prior knowledge about the average radius of a head in pixels (possibly with respect to its location in the image in case of strong perspective variation due to camera tilt). Note that this is a specific setting of more general case of imprecise objects definition, for which possible different imprecise shapes could be considered.

Starting from the dotted annotation, instead of simply performing a dilation with a circular structuring element centered in each annotation location, that would result in a binary map with possibly incorrect labels assigned to pixels corresponding to head's boundaries, we propose a *soft label* definition of the ground-truth map, as often done in the context of crowd density estimation [38–40]. Starting from the binary ground-truth map with 1-valued label for each head center location (x_c, y_c), we apply a cumulative Gaussian smoothing such that the ground-truth map for each head is expressed in terms of a Gaussian distribution as:

$$(x, y) \sim \eta \cdot \sum_{c \in \mathcal{C}} \mathcal{N}\left((x_c, y_c), \sigma_h\right), \tag{1}$$

where η is a scaling factor to face the class imbalance problem, while $2\sigma_h$ is the expected head radius and $\mathcal{C}$ is the set of dots annotating ground-truth heads.

We consider Gaussian distributions as they are infinitely differentiable functions presenting tails which vanish at infinity, being able to model well the imprecision on the head contour locations. We apply a cumulative Gaussian smoothing in the sense that the final ground-truth map is the sum of Gaussian distributions derived from each head center locations. The resulting map is not a probability distribution by itself, but rather the score associated to each pixel represents the sum of probabilities that any head, occluded or not, is located at that position, directly facing in this way also the problems of close and occluded heads. In presence of close heads indeed, maxima would still indicate the head center locations, while in presence of occluded heads the evidence of the partially visible head will be reinforced through the cumulative sum. Ground-truth Gaussian smoothing are finally also able to mitigate location errors in the annotated ground-truth, that could have a higher impact considering sharp-defined objects in presence of small targets. Figure 2 shows a typical example of a ground-truth map obtained with the proposed soft labels on an example image from the Makkah dataset that will be later employed to show experimental results.

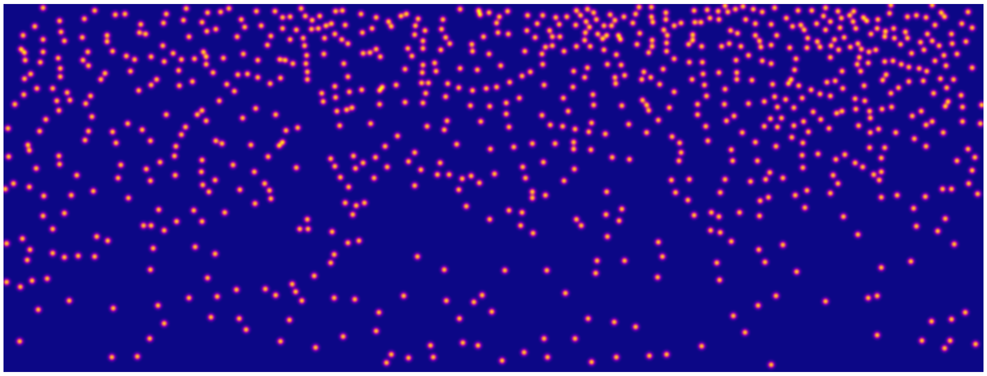


Figure 2. Typical example of a ground-truth map as cumulative Gaussian distributions, one per head. The score associated to each pixel of the ground-truth map is the sum of the contributions of each Gaussian at the given location. In the image, scores span from blue (low) to yellow (high).

2.2.3. FE + LFE Network

We choose to cast the specific problem of head detection in high-density crowds as a segmentation task, in the sense that we want to assign a different label to each different pixel of the image, depending whether or not it belongs to a head. Given an input image, we aim thus at performing *dense* prediction by estimating an output map of the same size of the input. However, attention must be payed with respect to two different concerns related to our application, namely the impossibility to obtain a precise ground-truth map and the impossibility to have huge labeled datasets at our disposal. These two aspects will be investigated in the following, as they are both possible causes of imprecision that can be modeled through the BFT.

Among the various architecture for semantic segmentation, we propose a network inspired by [41] that makes use of dilated convolution to be able to recover small objects, proposed in the field of remote sensing imagery. Note that we tried to use also UResNet [42], an encoder-decoder network inspired by both U-Net [43] and ResNet using residual blocks, but the training data at our disposal resulted to be not enough to train this type of network with a too high number of parameters.

In the context of remote sensing image analysis, the authors of [41] highlighted a major problem of segmentation in presence of small and densely aggregated objects. The use of pooling layers indeed tends to degrade the output resolution so that details of the very small objects are lost. In these situations, even the use of shortcuts like skip-connections in the U-Net could not be enough to recover small targets. Pooling layers are however important, for two different reasons. Reducing the spatial dimension of the 3D volume, they allow for a larger context consideration without increasing the receptive field of the filters, and to reduce the number of total parameters to be learned by the network.

In order to enlarge the receptive field of the filters going deeper with the layers, without degrading the output resolution nor increasing the filters' size (thus increasing the number of parameters), dilated convolutions can be exploited. Linearly increasing the dilation factor through the layers' chain will result in an exponential enlargement of the receptive field that is therefore able to capture larger context and recover smaller objects. Context information is indeed crucial in recovering small objects, as pointed out in [44].

The authors of [41] however noticed that aggressively increasing dilation factors through the network's layers in a straightforward way is detrimental in aggregating local features. Dilation causes weights to skip information between cells, and this results in a bad modelisation of the structure of small objects, presenting grid patterns in the final output. To solve this problem, they propose a network without pooling layers that conversely concatenates a Front End (FE) module of increasing dilation factors, with a Local Feature Extraction (LFE) module of decreasing dilation factors, arranged in a symmetrical way. The FE module is thus able to consider larger context for small objects detection, while the LFE module enforces the spatial consistency of the output by gathering spatial information decreasing the dilation size.

The Front End architecture employed in [41] is a VGG network [45] deprived of the final classifier layer (to obtain a fully convolutional network) and deprived of the max pooling layers. Instead of the latter, dilation factors are increased at the corresponding network depth. Then, the LFE module keeps invariant the number of filters and the kernel size, while decreasing the dilation factors up to one, in a specular way with respect to the FE.

The only drawback of such an architecture is that with respect to the U-Net it needs more memory to perform backward and forward passes, because feature maps at each layer have always the same size as the original input since there is no pooling operation. Thus, we propose a modified architecture (shown in Table 1) that keeps the memory use manageable by reducing the number of filters at each convolutional layer. The choice of the reduction of the number of filters per layer has nevertheless two purposes. On the one hand, it allows us to fulfill hardware constraints, while on the other hand it helps in preventing the network to overfit, as we know we are going to train it with small datasets. To this extent, also batch normalization has been added on top of each convolutional layer, for faster convergence [46] by reducing internal covariate shift.

Table 1. Detailed architecture of the proposed network inspired by [41], where F is the number of filters and D is the dilation factor to perform dilated convolutions. It is possible to notice the symmetric structure of the dilations whose factor increases in the Front End (FE) module, allowing us to increase the receptive field, and decreases in the Local Feature Extraction (LFE) module, aggregating local features to obtain spatial consistency in the output map. Note that each convolutional layer is followed by batch normalization (except the last layer) and ReLU activation function.

Layers	
FE	Conv 3×3, $F = 16$, $D = 1$
	Conv 3×3, $F = 32$, $D = 1$
	Conv 3×3, $F = 32$, $D = 2$
	Conv 3×3, $F = 64$, $D = 2$
	Conv 3×3, $F = 64$, $D = 3$
LFE	Conv 3×3, $F = 64$, $D = 2$
	Conv 3×3, $F = 64$, $D = 2$
	Conv 3×3, $F = 64$, $D = 1$
	Conv 3×3, $F = 64$, $D = 1$
	Conv 1×1, $F = 1$, $D = 1$

Note that to obtain a "simpler" network to prevent overfitting we could have conversely decreased the number of total layers. However, we preferred to reduce the number of filters per layer for two reasons. Firstly, decreasing the number of layers would have prevented the network to learn more complex features. Secondly, we would have not entirely exploited the benefit of increasing/decreasing dilation factors. Moreover, we found that adding many layers with a dilation factor of 3 was too heavy, and for this reason we preferred to add a single central layer with such a big dilation (however, this depends on the expected size of the targets).

Another modification to the usual structure of fully convolutional networks regards the last layer. Usually, the layers' chain terminates with a convolutional layer, without any activation function that would apply a non-linear transformation which is usually not needed. In our specific application however, the desired output values are real values which are possibly positively unbounded (a pixel's score is indeed the sum of contributions given by all the surrounding Gaussian distributions representing surrounding heads), while at the same time loosing their meaning as long as they take values under zero. By definition, being a cumulative sum of Gaussian distributions, the ground-truth maps obtained through the soft labeling procedure contain values which are always greater than or equal to zero. In this way, by simply integrating the map over a region of interest we can obtain the number of people present in that area. By setting the last layer as a convolutional one however, we would allow the network to possibly produce negative output values which would be the origin of noise in the estimation of the number of people, complementary application to pedestrian detection in the context of crowd macroscopic analysis [40].

For this reason, we propose to use an activation function in the last layer which bounds the values at zero, like the sigmoid or the ReLU. For example a sigmoid activation as last layer is used in [47], to constrain the output values between 0 and 1. For our particular application however, the sigmoid is not suited for two main different reasons. Firstly, since the sigmoid function tends toward zero without really reaching it, punctual noise would become more evident. Secondly, since it saturates at 1, we loose the meaning of "cumulative" output: when two heads are one next to the other, they would result in a single large blob and the score of each pixel would no more represent the cumulative sum of surrounding heads contributions.

On the contrary, we propose to use a ReLU activation function in the last layer. It has the effect of a threshold, setting all the negative values to zero. Nevertheless, since it is integrated inside the network, it has beneficial effects on backpropagation convergence with respect to a simple post-processing thresholding. In this way, the network learns easily to return zero for background pixels, being able at the same time to suppress a part of the background noise. The local density estimation is therefore also enhanced, since the network looses its tendency to compensate between low and high values adding noise, as highlighted in [40].

2.2.4. Bayesian FE + LFE Network

Now, we want to formulate the Bayesian counterpart of the proposed FE + LFE network that uses MC-dropout to get samples from the posterior distribution over the network's weights. Note that the "Bayesian" prefix refers here to the Bayesian probability theory, and not to the Belief Function framework meaning which simply stands for null-mass BBAs except for singleton hypotheses.

For the definition of the Bayesian FE + LFE we follow the formulation of Bayesian SegNet [32], which is built upon SegNet [48] network for pixel-wise segmentation.

Given a list of training inputs $\mathbf{x}$ and corresponding outputs y, we are interested in finding the posterior distribution over the network's convolutional weights, $\mathbf{W}$:

$$p(\mathbf{W}|\mathbf{x}, y), \tag{2}$$

However, this posterior distribution is generally intractable, and needs to be approximated [49]. To this extent, variational inference can be used, that allows us to approximate the intractable posterior through a function $q(\mathbf{W})$ over the network's weights. This function is learned by minimizing the Kullback-Leibler (KL) divergence between this approximating distribution and the actual posterior:

$$\mathcal{D}_{KL}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{x}, y)). \tag{3}$$

As proposed in [50], given a CNN with L layers of dimension $K \times K$, we define the approximating variational distribution $q(\mathbf{W}_i)$ for every convolutional layer i with units j as:

$$\begin{aligned} \mathbf{W}_i &= \mathbf{M}_i \cdot \text{diag}([z_{i,j}]_{j=1}^{K_i}), \\ z_{i,j} &\sim \text{Bernoulli}(p_i^{\text{drop}}) \qquad i = 1, \dots, L, \quad j = 1, \dots, K_{i-1}, \end{aligned} \tag{4}$$

where $z_{i,j}$ are Bernoulli distributed random variables with probabilities p_i^{drop} (i.e., dropout probabilities), and $\mathbf{M}_i$ contains the variational parameters to optimize. Note that dropout probabilities p_i^{drop} could also be optimized, but as in [32] we kept them fixed to an equal constant value found through validation. In this way, as proven in [50], we obtain the approximate model of the Gaussian process.

Now, we train the network and we sample the posterior distribution over the weights using dropout at test time, performing T different forward passes through the network. As a result for a given testing image we obtain T different realization maps $\hat{\mathcal{M}}_1, \dots, \hat{\mathcal{M}}_T$, output of different dropout-perturbed versions of the original network. Classically, the mean map $\mathcal{M}_\mu$, given by the mean value evaluated independently for each pixel, is interpreted as the final prediction map, while the standard deviation map $\mathcal{M}_\sigma$ is interpreted as an estimate of the predictive uncertainty. However, we propose once again to work in the BF framework, that we consider more suited to model the specific imprecision of each different realization obtained with dropout. In the following, we will explain the proposed BBA allocation for every realization that will allow us to perform a robust fusion among them as well as to obtain evidential measures of predictive uncertainty for every pixel of the final output map.

2.2.5. CNN-Ensemble BBA Allocation and Combination

While being an easy yet mathematically grounded approach to obtain a measure of uncertainty out of any kind of deep network, MC-dropout has some drawbacks. Firstly, for practical reasons, often we can perform only a limited number of forward passes with dropout, and the mean value could not robust in presence of outliers. Secondly, as reported in [37], the obtained uncertainty is not calibrated (it can scale differently for different datasets) and usually underestimated (variational inference is known indeed to underestimate predictive variance).

Leaving partially apart the mathematical ground in favor of an analysis more adapted to our specific setting, we note that the median is a more robust estimator than the average in presence of outliers. In the same way, instead of relying on the standard deviation, we can better employ the Median Absolute Deviation (*MAD*) [51], which is a robust measure of the variability of a univariate sample of quantitative data, more resilient to outliers than the standard deviation.

In our context, given the T realization maps, the *MAD* map $\mathcal{M}_{MAD}$ is defined as:

$$\mathcal{M}_{MAD} = \text{median}\left(\left| \hat{\mathcal{M}}_i - \text{median}\left(\{\hat{\mathcal{M}}\}_1^T \right) \right| \right), \tag{5}$$

where median $\left(\{\hat{\mathcal{M}}\}_1^T \right)$ is the median over all the T realizations.

Finally, we propose to rely on the Belief Function framework to obtain a better estimation of the model imprecision given the T realizations that can be interpreted as different sources for an evidential combination. BBA allocation is then performed as follows.

We have T maps $\hat{\mathcal{M}}_1, \ldots, \hat{\mathcal{M}}_T$ which correspond to the T output realizations obtained with forward passes through the network with dropout, and we are interested in finding a BBA allocation to perform the combination among them and derive evidential measures of imprecision.

Firstly, we can derive Bayesian BBA maps $\mathcal{M}_1^{\mathcal{B}}, \ldots, \mathcal{M}_T^{\mathcal{B}}$, where a BBA is associated to each pixel $\mathbf{x}$ of each realization, so that we obtain T maps of BBAs $\{m_{\mathbf{x},t}^{\mathcal{B}}, \mathbf{x} \in \mathcal{P}\}$, where $\mathcal{P}$ is the pixel domain and $t = 1, \ldots, T$. These Bayesian BBAs maps are 4-layers images where each layer corresponds to the mass values of any hypothesis in $\{\emptyset, H, \bar{H}, \Theta\}$. $\mathcal{M}_t^{\mathcal{B}}(A)$ corresponds to the layer image associated to hypothesis A for the realization (source) t. Note that in this preliminary Bayesian BBA allocation, layer images corresponding to non-singleton hypotheses are null, by definition. So, for each source t, with $t = 1, \ldots, T$:

$$\begin{cases} \mathcal{M}_t^{\mathcal{B}}(\emptyset) & = & \{0\}_{\mathbf{x} \in \mathcal{P}}\,, \\ \mathcal{M}_t^{\mathcal{B}}(H) & = & \hat{\mathcal{M}}_t, \\ \mathcal{M}_t^{\mathcal{B}}(\bar{H}) & = & 1 - \hat{\mathcal{M}}_t, \\ \mathcal{M}_t^{\mathcal{B}}(\Theta) & = & \{0\}_{\mathbf{x} \in \mathcal{P}}\,. \end{cases} \tag{6}$$

Now, we want to take into account the reliability of the pixel-wise prediction given by every source in order to perform a pixel-wise tailored discounting. Note that this would be impossible in the probabilistic framework; moreover, we are not just computing an overall source discounting, but rather each pixel of each source will be discounted differently on the basis of its reliability.

To measure this latter, we take inspiration from the *MAD*. For each source t, we compute a discounting coefficient map $\Gamma_t \ : \ \{\gamma_{\mathbf{x},t}\}_{\mathbf{x} \in \mathcal{P}}$ such that a different coefficient $\gamma_{\mathbf{x},t}$ is associated to every pixel of each source,

$$\Gamma_t = \alpha \left(1 - \left(\left|\hat{\mathcal{M}}_t - \text{median}\left(\{\hat{\mathcal{M}}\}_1^T\right)\right|\right)\right). \tag{7}$$

In this way, we discount more the pixels whose value is more distant to the median value among the T realizations, since they are supposed to be less representative (even possibly outliers). The α parameter is a scaling factor which allows us to control the amount of discounting.

Applying the proposed discounting, we derive the following BBA maps for every source t: $\forall A \in \{H, \bar{H}\}$,

$$\begin{cases} \mathcal{M}_t(\emptyset) &= \{0\}_{\mathbf{x}\in\mathcal{P}}\,, \\ \mathcal{M}_t(A) &= \Gamma_t \star \mathcal{M}_t^{\mathcal{B}}(A), \\ \mathcal{M}_t(\Theta) &= \{1\}_{\mathbf{x}\in\mathcal{P}} - \mathcal{M}_t(H) - \mathcal{M}_t(\bar{H}), \end{cases} \tag{8}$$

where $M_1 \star M_2$ represents the Hadamard product between matrices M_1 and M_2.

To combine the T different maps to obtain a single output map $\mathcal{M}$ with BBAs associated to each pixel $\mathbf{x}$, i.e., $\{m_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{P}}$, we use the conjunctive combination rule [3]. Note that using this rule instead of a cautious rule (e.g., [52]) devoted to non-independent sources, we consider that the different dropout realizations correspond to cognitively independent sources even if sampled from the same distribution.

In our case where $|\Theta| = 2$, the analytic result using the conjunctive combination rule may be easily derived: $\forall A \in \{H, \bar{H}\}$,

$$\begin{cases} m_{\mathbf{x}}(A) &= \sum\limits_{\substack{(B_1,\ldots,B_T)\in\{A,\Theta\}^T, \\ \exists t\in[1,T] s.t. B_t = A}} \prod_{t=1}^T m_{\mathbf{x},t}(B_t)\,, \\ m_{\mathbf{x}}(\Theta) &= \prod_{t=1}^T m_{\mathbf{x},t}(\Theta)\,, \\ m_{\mathbf{x}}(\emptyset) &= 1 - m_{\mathbf{x}}(H) - m_{\mathbf{x}}(\bar{H}) - m_{\mathbf{x}}(\Theta)\,. \end{cases} \tag{9}$$

The result is thus a four-layer map $\mathcal{M}_{CNN}$ of BBAs $m_{\mathbf{x}}$, that can be used to derive evidential measures of uncertainty about the network prediction. To this extent, we can obtain the ignorance map as $\mathcal{M}(\Theta)_{CNN}$, that represents the remaining ignorance which has been decreased by the combination but not completely solved, indicating a lack of sufficient information during training to perform a reliable prediction. Likewise, $\mathcal{M}(\emptyset)_{CNN}$ is often interpreted as a conflict map [53], and presents higher values for pixels whose prediction completely disagrees through the various realizations.

Finally, in every pixel $\mathbf{x}$ the decision is taken from $m_{\mathbf{x}}$. Pignistic probability may be used to give a probabilistic interpretation to the BBAs. Since in our setting $|\Theta| = 2$, the $\text{BetP}(H)$ map can be computed as: $\forall A \in \{H, \bar{H}\}$,

$$BetP_{\mathbf{x}}(A) = \frac{1}{1 - m_{\mathbf{x}}(\emptyset)} \left(m_{\mathbf{x}}(A) + \frac{m_{\mathbf{x}}(\Theta)}{2}\right). \tag{10}$$

This allows us to assign a $BetP_{\mathbf{x}}(H)$ value to the resulting BBA associated to each pixel $\mathbf{x}$ that will be differently normalized on the basis of its conflict value, $m_{\mathbf{x}}(\emptyset)$.

To illustrate the benefit of the explained BBA allocation for the CNN-ensemble, Table 2 proposes a toy example where MC-dropout is applied to sample the posterior distribution obtaining T = 4 realizations, for two different pixels $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, discounting coefficients $\gamma_{\mathbf{x},t}$ are derived using Equation (7), setting $\alpha = 0.5$. After having performed the conjunctive combination among the discounted BBAs, $BetP_{\mathbf{x}}(H)$ and $m_{\mathbf{x}}(\Theta)$ are shown for the two pixels $\mathbf{x}_1$ and $\mathbf{x}_2$. The posterior distribution sampled for pixel $\mathbf{x}_1$ presents similar values with respect to the one sampled for pixel $\mathbf{x}_2$, so that all the realizations are close to the median value and thus we obtain high discounting coefficients that reflect in reliable BBAs that do not need to be much discounted. Conversely, $\mathbf{x}_2$ presents a sampled distribution which is more spread out, so that more discounting (i.e., lower discounting coefficients) is applied. This fact reflects in higher value of ignorance for $\mathbf{x}_2$, that may be interpreted as higher predictive uncertainty.

Table 2. Example of different values obtained sampling the posterior distribution with MC-dropout technique with T = 4, for two different pixels $\mathbf{x}_1$ and $\mathbf{x}_2$, along with the corresponding discounting coefficient $\gamma_{\mathbf{x},t}$ obtained with Equation (7) setting $\alpha = 0.5$. After having performed the conjunctive combination among the discounted BBAs, $BetP_{\mathbf{x}}(H)$ and $m_{\mathbf{x}}(\Theta)$ results are shown for the two pixels $\mathbf{x}_1$ and $\mathbf{x}_2$.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	**Median**	$\gamma_{\mathbf{x},1}$	$\gamma_{\mathbf{x},2}$	$\gamma_{\mathbf{x},3}$	$\gamma_{\mathbf{x},4}$	$BetP_{\mathbf{x}}(H)$	$m_{\mathbf{x}}(\Theta)$
$\mathbf{x}_1$	0.8	0.8	0.82	0.82	0.81	0.99	0.99	0.99	0.99	0.87	0.06
$\mathbf{x}_2$	0.01	0.99	0.27	0.73	0.5	0.51	0.51	0.77	0.77	0.5	0.2

Now, instead of directly taking decision based on Equation (10), we keep $\mathcal{M}_{CNN}$ aside for a later fusion with another ensemble of classifiers composing the proposed evidential MCS.

2.3. SVM-Ensemble

The second ensemble of the proposed evidential MCS is based on SVM classifiers and it is obtained through active learning, which we find complementary to deep networks being particularly adapted in situations where we have a specific problem and an extremely small training set. Through active learning indeed, we are able to select the most informative training samples in order to reduce at minimum the number of samples needed yet maximizing the overall detection performance.

The SVM-ensemble here employed has been previously proposed in [11], where an evidential Query-By-Committee active learning strategy is designed in order to exploit different detectors based on different descriptors (gradient, texture and orientation features) in order to train an ensemble of SVM-based classifiers with limited amount of data. In that work, the BFT is exploited both for the sources combination and for the new samples selection. The result of the source combination indeed is a BBA associated to every unlabeled sample, that intrinsically contains conflict and ignorance components.

We now briefly recall the BBA allocation proposed in [11] for the sake of clarity, as the BBA map $\mathcal{M}_{SVM}$ resulting from the SVM-ensemble fusion will be the second input of the proposed evidential MCS.

2.3.1. SVM-Ensemble BBA Allocation and Combination

In the context of SVM-based high density crowds pedestrian detection, imprecision can arise in two different and complementary ways: in the derivation of posterior probability values from SVM decision scores, and later, from the spatial layout of the detections in the output image space.

Two successive discounting steps are thus processed on the initial Bayesian BBAs derived from the learned sigmoids during Platt's calibration [54]. Firstly, having learned the sigmoid σ_i of classifier i by logistic regression, BBAs are defined to model the imprecision due to possible errors in the calibration, by applying erosion $\mathcal{E}_w$ and dilation δ_w operators in the 2D space where SVM calibration scores are projected with respect to their label, with structuring element w. Then, the mass on Θ is increased by discounting the previous BBAs, by performing a morphological opening operation γ_a, this time in the image space, to take into account neighbor pixels information based on the assumption that they are likely to belong to the same class.

Specifically, with $s_{\mathbf{x}}$ being the SVM score associated to pixel $\mathbf{x}$:

$$\begin{cases} \forall \mathbf{x} \in \mathcal{P}, \widetilde{m}_{\mathbf{x},i}(H) & = & (\mathcal{E}_w \circ \sigma_i)(s_{\mathbf{x}}), \\ \forall \mathbf{x} \in \mathcal{P}, \widetilde{m}_{\mathbf{x},i}(\bar{H}) & = & 1 - (\delta_w \circ \sigma_i)(s_{\mathbf{x}}), \\ \forall \mathbf{x} \in \mathcal{P}, \widetilde{m}_{\mathbf{x},i}(\Theta) & = & 1 - \widetilde{m}_{\mathbf{x},i}(H) - \widetilde{m}_{\mathbf{x},i}(\bar{H}). \end{cases}$$

where $\mathcal{P}$ is the pixel domain, σ_i is the learned sigmoid for classifier i and $(\mathcal{E}_w \circ \sigma_i)$ and $(\delta_w \circ \sigma_i)$ its eroded and dilated results with a (flat) structuring element of width w, applied in the score space. Then, in the image space,

$$\begin{cases} \mathcal{M}_i(\emptyset) & = & \{0\}_{x\in\mathcal{P}}\,, \\ \forall A \in \{H, \bar{H}\}\,, \mathcal{M}_i(A) & = & \gamma_a\left(\widetilde{\mathcal{M}_i}(A)\right), \\ \mathcal{M}_i(\Theta) & = & \{1\}_{x\in\mathcal{P}} - \mathcal{M}_i(H) - \mathcal{M}_i(\bar{H}), \end{cases}$$

where $\mathcal{M}_i(A)$ is the layer image associated to hypothesis A, $\forall A \in 2^\Theta$, and γ_a is the opening operator of parameter a applied in the image domain.

As in [11,55,56], a spatial Gaussian structuring element fitted in a window of radius a is used, to better take into account the spatial consistency.

The final result is thus a BBA map $\mathcal{M}_{SVM}$, that can be used either for decision through Equation (10), or in conjunction with the previously defined $\mathcal{M}_{CNN}$ to compose the evidential MCS.

2.4. Evidential MCS

Until now we have proposed two different ensemble-based methods based on two different classifiers, namely SVM and CNN. In order to obtain the final MCS, we intend to perform a fusion between the two ensembles. Note that this is not straightforward, since in presence of few, strong classifiers the fusion strategy must be particularly well-designed in order to exploit their respective strengths. Moreover, in this work a noticeable difficulty comes from the unbalanced performance between the two ensembles which makes most of fusion schemes not improving results derived from CNN approach alone.

Figure 3 shows the overall flowchart of the final evidential MCS. Starting from the initial pool of samples $\mathcal{U}$, we perform in a parallel way the SVM-based active learning procedure to select the most informative samples to be added to the set of labeled training samples $\mathcal{L}$, while at the same time we train the FE+LFE network on the entire set $\mathcal{U}$.

In order to build the SVM-ensemble, the evidential QBC active learning procedure consists of the following steps:

- Training the different SVM classifiers based on different features (e.g., HOG [13], LBP [57], DAISY [58] and GABOR [59] are employed in [11]);
- Performing BBA allocation for each pixel of each source, taking into account possible imprecision in the score calibration procedure and in the image space, and combining them through the conjunctive combination rule;
- Selecting the new samples to be added to the SVM training set $\mathcal{L}$ based on evidential entropy disagreement measures.

At the end of the evidential Query-by-Committee procedure, the result is a single four-layers BBA map ($\mathcal{M}_{SVM}$) with a BBA associated to each pixel that intrinsically contains evidence of belonging to H and $\bar{H}$, i.e., $\mathcal{M}_{SVM}(H)$ and $\mathcal{M}_{SVM}(\bar{H})$ respectively, as well as a component of ignorance ($\mathcal{M}_{SVM}(\Theta)$) which is not solved through the combination and a component of conflict ($\mathcal{M}_{SVM}(\emptyset)$) that arises through the combination itself.

Regarding the second component of the MCS, namely the CNN-ensemble, it also consists of several steps:

- Training the FE + LFE network (Section 2.2.3) based on the relatively small training dataset $\mathcal{U}$;
- Applying MC-dropout procedure at inference time to obtain the T realizations, as explained in the previous Section 2.2.4;

- Performing BBA allocation for each realization to model the network's predictive uncertainty about each pixel's prediction, and combining them through the conjunctive combination rule (cf. Section 2.2.5).

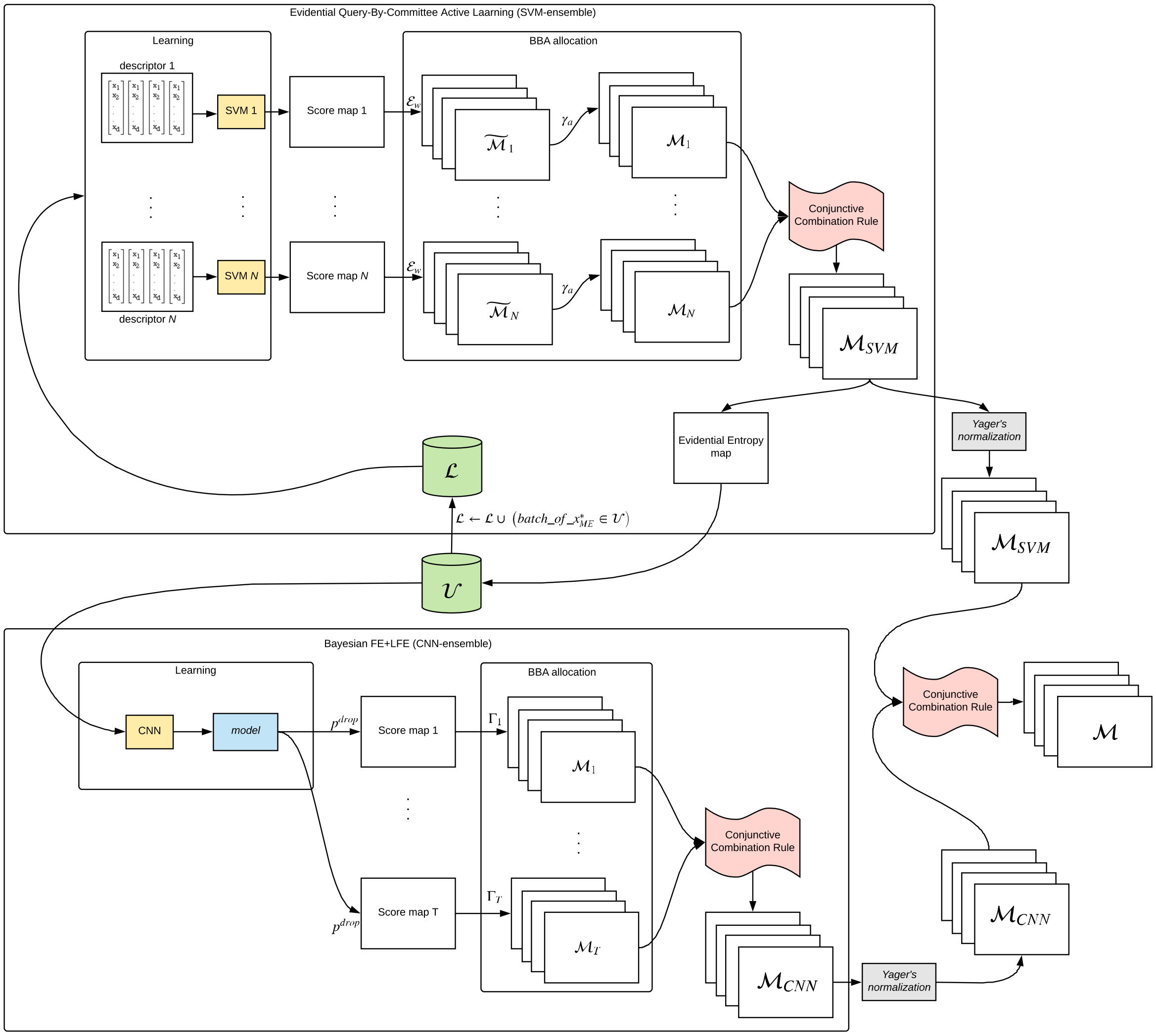


Figure 3. Proposed evidential Multiple Classifier System flowchart.

The output of the proposed evidential CNN-ensemble is thus a single four-layers BBA map ($\mathcal{M}_{CNN}$), where each pixel contains evidence of belonging to $\emptyset, H, \bar{H}, \Theta$ respectively. Here we interpret the ignorance value related to each pixel as the model's predictive uncertainty about it, being able thus to model the imprecision in addition to the uncertainty value provided by the network.

After having combined a relatively high number of sources through the conjunctive rule both to obtain $\mathcal{M}_{SVM}$ and $\mathcal{M}_{CNN}$, we note that they both contains non-negligible masses on the empty set representing the conflict while the mass on the Θ focal element naturally decreases thanks to the conjunctive combinations. This could lead to disproportionate values of conflict with respect to the masses on the other focal elements. To solve this issue, classically Dempster's rule is adopted or a normalization of the BBAs is lately performed, but in this way the conflicting mass would be equally spread over the remaining hypothesis. Instead, as done in [53], we focus on the normalization included in Yager's combination rule [60] that, in the absence of knowledge about the conflict origin, transfers it to the ignorance component.

Finally, the conjunctive combination rule is performed between the normalized $\mathcal{M}_{SVM}$ and $\mathcal{M}_{CNN}$, obtaining the final BBA map $\mathcal{M}$ which can be used either for decision, computing the associated $\text{BetP}(H)$ map with Equation (10), and to obtain a measure of the imprecision about the final prediction, naturally given by $\mathcal{M}(\Theta)$.

3. Results

3.1. Dataset

The proposed algorithm was tested on images acquired at one of the largest scale, high-density locations accessible for study, namely the holy Muslim pilgrimage taking place in Makkah, Saudi Arabia. The data was acquired at peak density during Hajj [61] in 2012. The camera used for recording is a robotic camera (AVT Guppy PRO) mounted statically in order to observe the high-density pilgrim crowd, and providing gray-level regular images (visible spectrum). The input data consists thus in a video sequence of the crowd (at a frame-rate of 8 Hz), but for the detection task we use individual images. For the training, calibration and evaluation of the head detectors, we use 35 images extracted at distant moments (in order to establish a full level of independence among the images used). Each image instance contains in the analyzed Region of Interest (corresponding roughly to the lower half of the scene) a high number of objects to detect (about 900–1000 heads) due to the high density.

The use of an ensemble, along with the discussed regularization techniques, allows us to apply a deep learning-based solution even in presence of a very small dataset. In particular, as training set we use the pool of data available for the active learning solution in [11] for choosing the new samples to add to the training set, noted $\mathcal{U}$, i.e., the pool of unlabeled samples for the active learning which corresponds to image patches roughly including 2000 different heads. Note that in the traditional active learning an oracle is supposed to answer about the true label of a sample only when it has been chosen by the algorithm to be added to the set, so that $\mathcal{U}$ is indeed a pool of yet unlabeled samples. In our case, the pool is not unlabeled as we dispose of ground-truth maps, nevertheless for consistency we keep the notation "$\mathcal{U}$". Note also that the active learning solution do not use all the available data in $\mathcal{U}$, but selects only 2000 positive or negative samples out of it (i.e., pixels belonging or not to a head). Nonetheless, we consider this a rather fair compromise since the training data available to the two methods is a-priori the same (while the active learning procedure chooses the most informative samples, the deep learning strategy uses all the available pixels of the training images in a fully convolutional architecture setting). Then, image patches containing roughly others 2000 heads (different from the training ones) are used for validation, and the rest of the dataset is used for testing.

The difficulty of the detection task results from multiple factors that we briefly introduce in the following paragraph. Figure 4 shows a patch from an image of the dataset, highlighting the difficulty of the problem since the heads are barely visible and many occlusions occur. We performed a dotted annotation in the head centers for the training images, such that the ground-truth so obtained can be used either as an *oracle* to assign the correct label to the samples selected for querying by AL (Active Learning), or in order to evaluate the loss for training the fully convolutional network. Even though in Makkah the crowd follows a general direction, there is a significant degree of head appearance variability due to gender, type of head cover, and most importantly, to the various degrees of occlusion coupled with the small size of the targets. For annotating a single image (clicking on the heads exhaustively), a human annotator requires typically half a day of work, and approximately 20% of the heads are so difficult to annotate that the human needs to look in the previous and the next frames in order to take a head/not head decision (something which our algorithm cannot do, as it performs the detection only in the current frame). As it is possible to see from the image, another problem in this type of scenes is the high data imbalance between positive and negative samples (i.e., pixels belonging or not to a head, respectively), stressing the importance of finding an effective strategy to select significant samples for SVM on one side, and exploiting at maximum the information of the few positive pixels with dilated convolutions for the fully convolutional deep architecture on the other side.

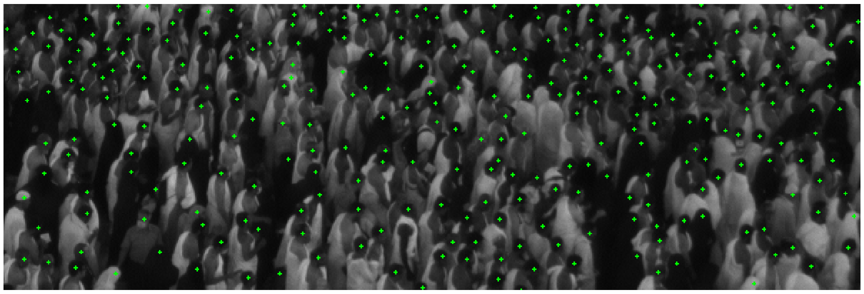


Figure 4. Patch with ground-truth dotted annotation.

3.2. Evaluation Method

Regarding the evaluation procedure of the pedestrian detection map results, we choose to evaluate our method on the basis of two different measures, which do not depend on any threshold and at the same time are suited for imbalanced data, namely Area Under Precision-Recall Curve (AUPRC) and Precision-Recall Break Even Point (PRBEP). This latter in particular is a useful operative threshold value, corresponding to the threshold for which Precision is equal to Recall, the number of false positive detections (fp) is equal to the number of false negatives (fn) since $Precision = \frac{tp}{tp+fp}$ and $Recall = \frac{tp}{tp+fn}$ with tp the number of true positive detections. These two metrics are computed on the $\mathrm{BetP}(H)$ map, applying non maxima suppression (NMS) at every threshold to identify the targets, setting the radius of a head to $r = 3$, with $2r + 1$ minimum distance between two maxima (head centers) in order to avoid overlapping detections. The value of the radius has been set empirically and depends on the dataset.

3.3. Training Setting

Fully convolutional networks usually cast segmentation as a dense *classification* problem, in the sense that each pixel is assigned to a given class (where the number of classes is discrete). For this reason, they commonly employ loss functions suited for this task, e.g., cross-entropy (weighted, in case of class imbalance). We are rather interested in performing *regression*, since after the cumulative Gaussian soft labeling the pixels of the ground-truth map (and thus our desired output) are not labeled with their class but rather with a real value resulting from the accumulation of head distributions. Since output values are not discrete (disregarding the unavoidable discretization of the Gaussian function over the pixel domain) and possibly not bounded, we choose to use a MSE loss, as an efficient pixelwise estimate of the distance between two 2D maps.

Note that the parameter η of Equation (1) is equivalent to weight the loss for the positive class for classification problem employing weighted cross-entropy loss. The parameter η is particularly important since higher its value, higher the impact of each single pixel belonging to a head in the loss function, and must be set taking into account the expected crowd density (lower the density higher its value, as per-pixel class imbalance would be more relevant). Considering that a head diameter spans between 8 and 12 pixels, pixel-level class imbalance issue is solved by setting $\eta = 150$ in Equation (1) (value empirically obtained through validation).

The network is trained by using an Adam stochastic optimizer [62], with a learning rate of 7×10^{-3} for the proposed FE + LFE (exact values have been found through validation). The weights of the convolutional layers are initialized with the Kaiming He method [63], which has proven to be particularly adapted for deep networks relying on ReLU activation functions. Early stopping with a patience of 20 epochs is used in order to terminate the learning process when the networks stop improving on the validation set. This contributes also to mitigating the risk of overfitting.

Regarding the hyperparameters used to obtain the SVM-ensemble through active learning, we refer the reader to the previous work [11].

3.4. CNN-Ensemble Results

Before showing the results of the final evidential MCS, we firstly investigate the benefit of the proposed evidential CNN-ensemble over traditional methods, using the proposed FE + LFE network.

In order to obtain the CNN-ensemble, we applied MC-dropout method. Dropout is added in the central layers as in [32], i.e., before and after the bottleneck layer with dilation factor equal to 3 (see Table 1). The probability of dropout p^{drop} is set to 0.2 since the default value of 0.5 resulted to be detrimental for the final result. The number of realizations T is fixed to 10 through validation, as increasing it do not show any significant performance improvements.

Figure 5 shows the PR-curves obtained with the active learning solution based on the use of a committee of SVMs [11] (denoted as "SVM-ensemble") with the deep learning-based solutions obtained training the network on the same limited amount of data. Specifically, after training the FE + LFE network, "CNN" refers to the output map obtained with the traditional forward pass to perform inference. "CNN-ensemble Mean" and "CNN-ensemble Fusion" refer instead to the use of MC-dropout to obtain the ensemble, combining the members through the traditional average operator and with the proposed evidential approach respectively. Table 3 provides quantitative values for PRBEP and AUPRC with the same names notation.

As it is possible to see both from Figure 5 and from Table 3, deep learning-based solutions tends to outperform SVM-based one, most noticeably regarding the precision values. Nevertheless, SVM has been trained with a chosen fraction of the available samples pool $\mathcal{U}$, with respect to deep learning-based methods that are able to exploit all the available data.

The use of the CNN-ensemble to perform inference rather than the usual forward pass is beneficial especially in increasing the recall values, meaning that the ensemble is able to retrieve more heads. Note that there is not a great difference between the mean output map $\mathcal{M}_\mu$ (CNN-ensemble Mean) and the $\text{BetP}(H)$ map after having performed the fusion of the T realizations (CNN-ensemble Fusion), although this latter is slightly better. However, having defined a BBA allocation for each realization allows us to have a final BBA map that can be easily combined together with the BBA map given by the SVM-ensemble.

Figure 6 shows the final output maps for a given image patch, considering the traditional forward pass for inference in Figure 6b with respect to the CNN-ensemble based output maps in Figure 6c,e, respectively obtained through the proposed BBA allocation and evidential fusion, and through the classical average of the T realizations. Figure 6d,f instead, represent the ignorance map after the evidential conjunctive combination and the classical standard deviation map respectively, which can be interpreted as a measure of predictive uncertainty. The predictive uncertainty map obtained with the proposed evidential method is clearly more precise in the localization of areas where the model is uncertain, while the standard deviation map although being useful is less localized and noisier.

This behaviour is even better highlighted in Figure 7, where for a given little image patch we see the associated output map of CNN-ensemble Fusion, i.e., $\text{BetP}(H)$, the ignorance map obtained as $\mathcal{M}(\Theta)$ and the traditional standard deviation map. The ignorance map is far less noisier than the standard deviation map, as highlighted especially in the area of the green box. Moreover, the standard deviation map provides misleading high values in some areas (e.g., the highlighted red box), where the network ensemble correctly predict no heads (and indeed the associated ignorance is correctly low). The areas of the ignorance map with high values on the contrary are more localized, and generally correspond to difficult cases such as occluded and low-contrasted heads.

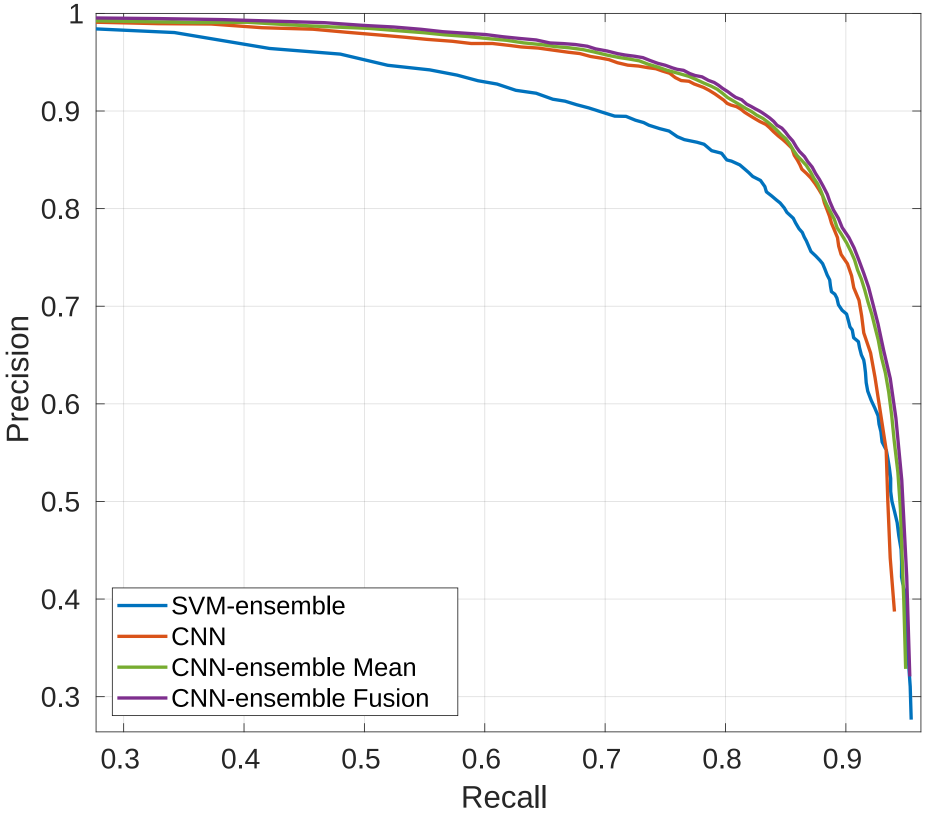


Figure 5. PR-curves of SVM-ensemble and deep learning solutions. All the classifiers disposed of the same amount of (limited) data for the training.

Table 3. Precision-Recall Break Even Point and Area Under Precision-Recall Curve with the different architectures trained on the same limited amount of data.

	SVM-Ensemble	CNN	CNN-Ensemble Mean	CNN-Ensemble Fusion
PRBEP	0.81	0.85	0.85	**0.86**
AUPRC	0.86	0.89	**0.90**	**0.90**

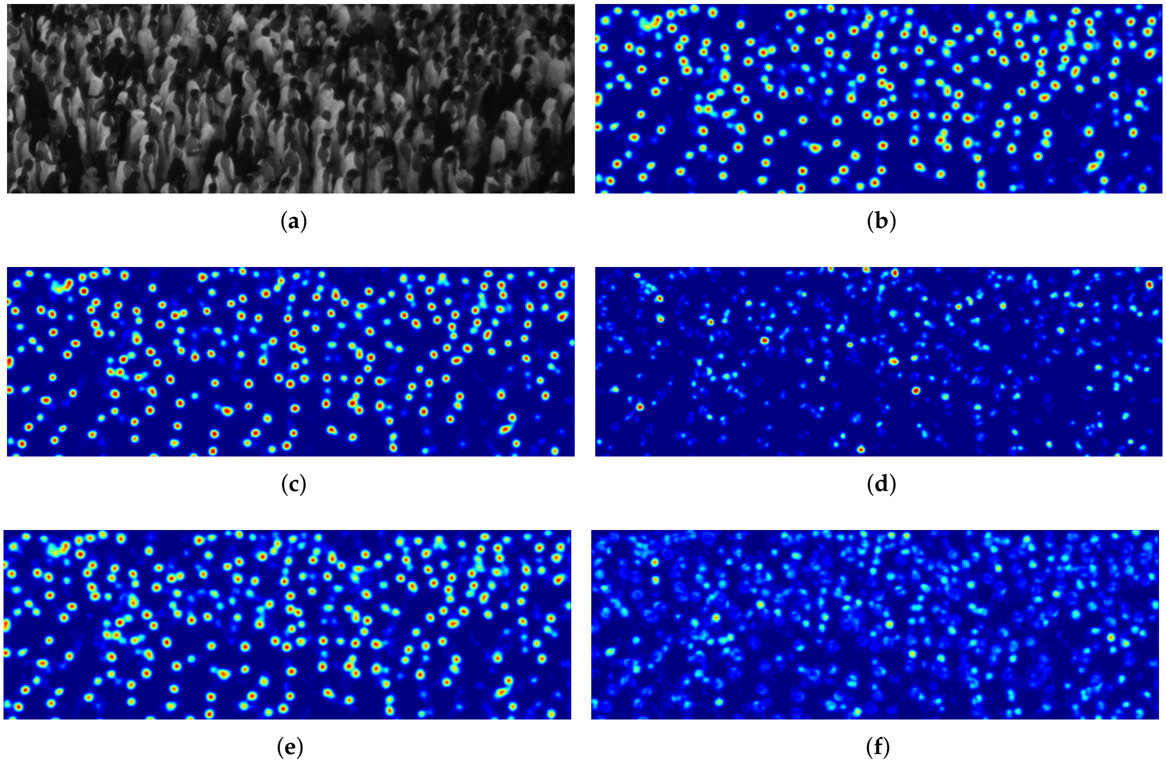

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 6. Output maps on a testing image patch with the deep learning solutions trained on the same amount of limited data, as well as model's predictive uncertainty outputs through traditional standard deviation and proposed evidential ignorance. (**a**) Image patch; (**b**) Output map of FE + LFE; (**c**) Output map of CNN-ensemble Fusion, i.e., $\mathrm{BetP}(H)$; (**d**) Ignorance map of CNN-ensemble Fusion, i.e., $\mathcal{M}(\Theta)$; (**e**) Output map of CNN-ensemble Mean, i.e., $\mathcal{M}_\mu$; (**f**) Standard deviation map of CNN-ensemble, i.e., $\mathcal{M}_\sigma$.

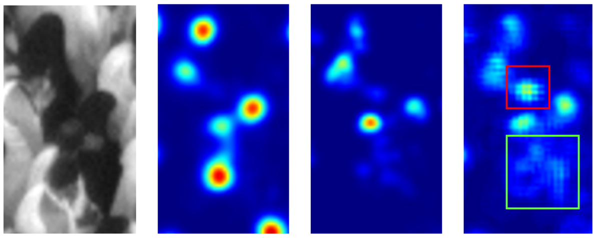
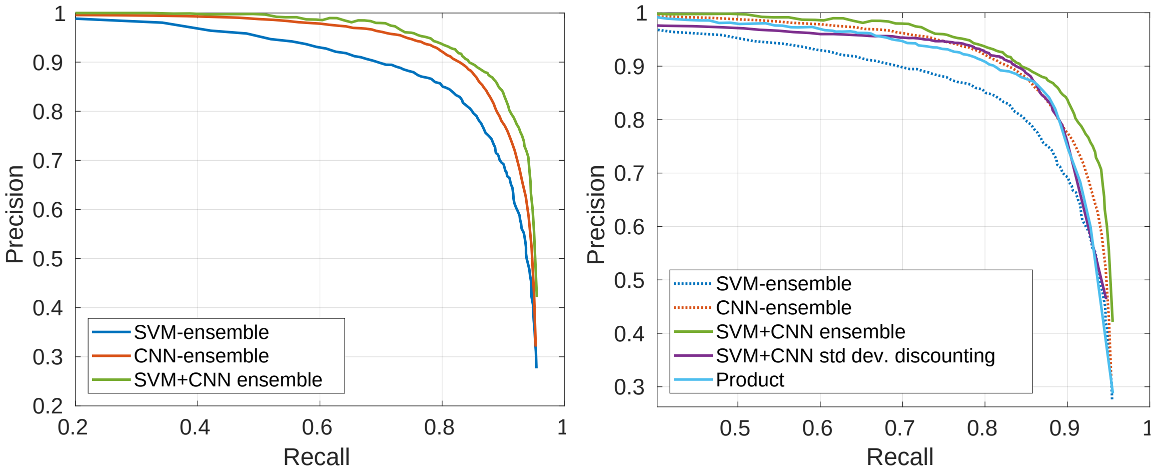


Figure 7. From left to right: (**a**) Example of an image patch (with increased contrast for clarity) with (**b**) associated output map of CNN-ensemble Fusion, i.e., $\mathrm{BetP}(H)$, (**c**) ignorance map (proposed) and (**d**) traditional standard deviation map respectively. Failure cases of standard deviation map are highlighted in red and green boxes.

3.5. Evidential MCS Results

To illustrate the benefit of the final evidential MCS, Figure 8a shows the Precision Recall (PR) curve of the proposed approach described with the flowchart reported in Figure 3, where the SVM-ensemble and CNN-ensemble BBA output maps are combined together after Yager's normalization. PR-curves of SVM-ensemble and CNN-ensemble alone are reported as well, to show the improvement obtained thanks to their fusion.



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

Figure 8. (**a**) PR-curves of SVM-ensemble and CNN-ensemble, along with their combination SVM+CNN ensemble; (**b**) Comparison in terms of PR-curves of the proposed SVM+CNN ensemble with respect to product of BetP(H) maps given by the two ensembles, and a fusion between the SVM-ensemble BetP(H) map with the result of a simple discounting performed on the mean map $\mathcal{M}_\mu$ based on the standard deviation values in $\mathcal{M}_\sigma$.

Figure 8b shows the comparison of the proposed approach with respect to two other strategies, namely the fusion between SVM-ensemble and the result of a simple discounting performed on the mean map $\mathcal{M}_\mu$ based on the standard deviation values in $\mathcal{M}_\sigma$, and the product of BetP(H) maps (interpreted as probability maps) given by the two ensembles. The two initial sources SVM-ensemble and CNN-ensemble are reported as well with dotted lines. Values of PRBEP and AUPRC for the considered approaches are then detailed in Table 4.

Table 4. Precision-Recall Break Even Point and Area Under Precision-Recall Curve of the BetP(H) result with the proposed MCS composed by SVM+CNN ensemble, as well as a comparison with respect to product of BetP(H) maps given by the two ensembles, and a fusion between the SVM-ensemble BetP(H) map with the result of a simple discounting performed on the mean map $\mathcal{M}_\mu$ based on the standard deviation values in $\mathcal{M}_\sigma$. SVM-ensemble and CNN-ensemble performances are reported as reference.

	SVM-Ens.	**CNN-Ens.**	**SVM+CNN Ens.**	**SVM+CNN Std Dev. Disc.**	**Product**
PRBEP	0.81	0.86	**0.87**	0.86	0.86
AUPRC	0.86	0.90	**0.92**	0.89	0.90

Both from the PR-curves and from the values reported in the table, we can see that the evidential fusion of the two ensembles preceded by Yager's normalization resulted to be the best approach. Conversely, both the product of probabilities and the simpler discounting method fail to exploit all the available information so that the final result do not improve on CNN-ensemble or rather worsen it. This is due to the fact that, being already a map of BBAs obtained after the fusion of the T realizations, CNN-ensemble's BetP(H) map is more informative than the mean map on which we apply a hand-crafted discounting (even though tailored with respect to standard deviation).

This proves that, in presence of few labeled data, the joint use of two classifiers (in our case SVM and CNN) is able to reach competitive performance.

Figure 9 provides visual results obtained testing the proposed evidential MCS on a given image patch, in terms of $\mathrm{BetP}(H)$ output map, detection map at the threshold corresponding to the PRBEP, and the final ignorance map of the system. The obtained $\mathrm{BetP}(H)$ map presents well-localized and well-shaped detections. Regarding the ignorance map, which we interpret as the global system's predictive uncertainty, we notice that it presents higher values in the surrounding of the heads. This is due to the fact that we applied Yager's normalization before the combination of the two ensembles based on the different classifiers, reversing the conflict mass (which is higher at the border of the heads) on the compound set. Thus, a part of the ignorance is not solved with the final combination resulting in the obtained map. Nevertheless, disregarding from the high values on the head's borders, the map is interesting in that it highlights the regions where none of the classifiers (nor the SVM nor the deep learning-based one) were able to give a committed answer about the predicted pixel's value.

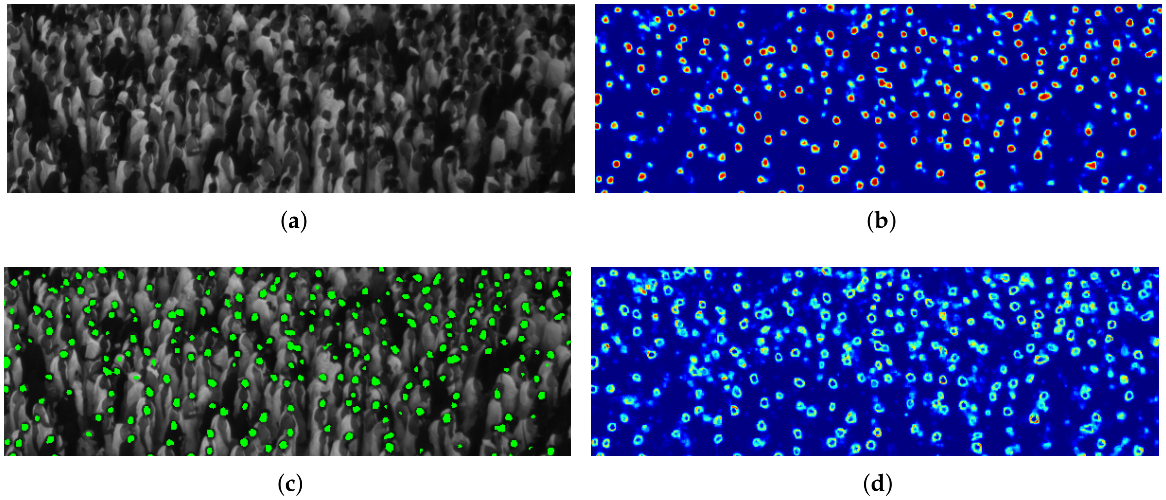


Figure 9. Visual results obtained testing the proposed evidential MCS on an image patch, in terms of $\mathrm{BetP}(H)$ output map, detection map at the threshold corresponding to the PRBEP, and the final ignorance map of the system. Output maps on a testing patch image with the deep learning solutions trained on the same amount of limited data, as well as model's predictive uncertainty outputs through traditional standard deviation and proposed evidential ignorance. (**a**) Image patch; (**b**) Output map of SVM+CNN ensemble, i.e., $\mathrm{BetP}(H)$; (**c**) Detections at PRBEP threshold; (**d**) Ignorance map of SVM+CNN ensemble, i.e., $\mathcal{M}(\Theta)$.

4. Conclusions

In this work, we proposed an evidential Multiple Classifier System which, based on the joint use of two heterogeneous classifier ensembles, is able to reach a higher level of performance that would otherwise be accessible only by using larger amounts of annotated training data. On the first hand, our approach underlines the importance of modeling uncertainty in fusion rules for decision systems for extracting additional information from prior knowledge and training data. Secondly, the pedestrian detection application we considered shows the practical interest of such methods for deploying classifiers with a lower burden in terms of required labeling, which is often a costly and time-consuming process. Ultimately however, the main aim of our work is not necessarily to provide a better classifier, but mostly to provide a robust approach that works in presence of limited training datasets and which is also able to give insights related to the interpretability of deep learning-based methods, addressing the limitations raised by standard deep learning architectures, which tend to perform as "black boxes".

In future studies, we intend to tackle cross camera detection, a task which is far from trivial at high densities due to scene scale and illumination changes and to significant variations in occluded pedestrian appearance. At the same time, an easy fine-tuning process among different views is vital from a practical standpoint. Since our method requires less training data in order to reach a good performance, we intend to investigate whether the proposed strategy may be extended as well during a retraining step, in order to minimize thus the amount of costly annotations required in order to deploy a detection system in additional views.

Author Contributions: J.V. implemented the algorithms, ran the experiments and wrote the manuscript; E.A. and S.L.H.-M. supervised the research and edited the manuscript.

Funding: This work was funded by ANR grant ANR-15-CE39-0005 and by QNRF grant NPRP-09-768-1-114.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
BNN	Bayesian Neural Network
BFT	Belief Function Theory
MCS	Multiple Classifier System
AL	Active Learning
HOG	Histogram of Oriented Gradients
RPN	Region Proposal Network
SVM	Support Vector Machine
BBA	Basic Belief Assignment
NN	Neural Network
SRT	Stochastic Regularization Technique
MC	Monte Carlo
FE	Front End
LFE	Local Feature Extraction
MAD	Median Absolute Deviation
AUPRC	Area Under Precision-Recall Curve
PRBEP	Precision-Recall Break Even Point
NMS	Non Maxima Suppression
PR-curve	Precision Recall curve

References

1. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 1050–1059.
2. Makantasis, K.; Doulamis, A.D.; Doulamis, N.D.; Nikitakis, A. Tensor-based classification models for hyperspectral data analysis. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6884–6898. [CrossRef]
3. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976; Volume 1.
4. Smets, P.; Kennes, R. The transferable belief model. *Artif. Intell.* **1994**, *66*, 191–234. [CrossRef]
5. Denoeux, T. 40 years of Dempster-Shafer theory. *Int. J. Approx. Reason.* **2016**, *79*, 1–6. [CrossRef]
6. Kallel, A.; Le Hégarat-Mascle, S. Combination of partially non-distinct beliefs: The cautious-adaptive rule. *Int. J. Approx. Reason.* **2009**, *50*, 1000–1021. [CrossRef]
7. Jousselme, A.; Maupin, P. Distances in evidence theory: Comprehensive survey and generalizations. *Int. J. Approx. Reason.* **2012**, *53*, 118–145. [CrossRef]

8. Ma, L.; Destercke, S.; Wang, Y. Online active learning of decision trees with evidential data. *Pattern Recognit.* **2016**, *52*, 33–45. [CrossRef]
9. Lachaize, M.; Le Hégarat-Mascle, S.; Aldea, E.; Maitrot, A.; Reynaud, R. Evidential framework for Error Correcting Output Code classification. *Eng. Appl. Artif. Intell.* **2018**, *73*, 10–21. [CrossRef]
10. Pellicanò, N.; Aldea, E.; Le Hégarat-Mascle, S. Geometry-Based Multiple Camera Head Detection in Dense Crowds. In Proceedings of the 28th British Machine Vision Conference (BMVC)—5th Activity Monitoring by Multiple Distributed Sensing Workshop, London, UK, 4–7 September 2017.
11. Vandoni, J.; Aldea, E.; Le Hégarat-Mascle, S. Evidential query-by-committee active learning for pedestrian detection in high-density crowds. *Int. J. Approx. Reason.* **2019**, *104*, 166–184. [CrossRef]
12. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 743–761. [CrossRef]
13. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
14. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]
15. Braik, M.; Al-Zoubi, H.; Al-Hiary, H. Pedestrian detection using multiple feature channels and contour cues with census transform histogram and random forest classifier. *Pattern Anal. Appl.* **2019**. [CrossRef]
16. Hosang, J.; Omran, M.; Benenson, R.; Schiele, B. Taking a deeper look at pedestrians. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4073–4082.
17. Tian, Y.; Luo, P.; Wang, X.; Tang, X. Pedestrian detection aided by deep learning semantic tasks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5079–5087.
18. Zhang, S.; Benenson, R.; Omran, M.; Hosang, J.; Schiele, B. How Far are We from Solving Pedestrian Detection? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1259–1267.
19. Zhang, L.; Lin, L.; Liang, X.; He, K. Is faster R-CNN doing well for pedestrian detection? In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 443–457.
20. Li, J.; Liang, X.; Shen, S.; Xu, T.; Feng, J.; Yan, S. Scale-aware fast R-CNN for pedestrian detection. *IEEE Trans. Multimed.* 2017, *20*, 985–996. [CrossRef]
21. Sermanet, P.; Kavukcuoglu, K.; Chintala, S.; LeCun, Y. Pedestrian detection with unsupervised multi-stage feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 3626–3633.
22. Ouyang, W.; Zhou, H.; Li, H.; Li, Q.; Yan, J.; Wang, X. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1874–1887. [CrossRef] [PubMed]
23. Ujjwal, U.; Dziri, A.; Leroy, B.; Bremond, F. Late Fusion of Multiple Convolutional Layers for Pedestrian Detection. In Proceedings of the 15th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018.
24. MacKay, D.J. A practical Bayesian framework for backpropagation networks. *Neural Comput.* **1992**, *4*, 448–472. [CrossRef]
25. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer Science & Business Media: New York, NY, USA, 2012; Volume 118.
26. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight Uncertainty in Neural Network. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 1613–1622.
27. Graves, A. Practical variational inference for neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Granada, Spain, 12–15 December 2011; pp. 2348–2356.
28. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *Statistics* **2014**, *1050*, 10.

29. Damianou, A.; Lawrence, N. Deep gaussian processes. In Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics; Scottsdale, AZ, USA, April 29–May 1, 2013; pp. 207–215.
30. Wan, L.; Zeiler, M.; Zhang, S.; Le Cun, Y.; Fergus, R. Regularization of neural networks using dropconnect. In Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 1058–1066.
31. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
32. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv* **2015**, arXiv:1511.02680.
33. Kendall, A.; Cipolla, R. Modelling uncertainty in deep learning for camera relocalization. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4762–4769.
34. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 6402–6413.
35. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
36. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arxiv:1412.6572.
37. Gal, Y. Uncertainty in Deep Learning. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2016.
38. Lempitsky, V.; Zisserman, A. Learning to count objects in images. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 6–9 December 2010; pp. 1324–1332.
39. Sindagi, V.A.; Patel, V.M. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognit. Lett.* **2018**, *107*, 3–16. [CrossRef]
40. Vandoni, J.; Aldea, E.; Le Hégarat-Mascle, S. Evaluating Crowd Density Estimators Via Their Uncertainty Bounds. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4579–4583. [CrossRef]
41. Hamaguchi, R.; Fujita, A.; Nemoto, K.; Imaizumi, T.; Hikosaka, S. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1442–1450.
42. Guerrero, R.; Qin, C.; Oktay, O.; Bowles, C.; Chen, L.; Joules, R.; Wolz, R.; Valdés-Hernández, M.; Dickie, D.; Wardlaw, J.; et al. White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks. *NeuroImage Clin.* **2018**, *17*, 918–934. [CrossRef] [PubMed]
43. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
44. Hu, P.; Ramanan, D. Finding tiny faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1522–1530.
45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
46. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
47. Iglovikov, V.; Shvets, A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *arXiv* **2018**, arXiv:1801.05746.
48. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
49. Denker, J.S.; Lecun, Y. Transforming neural-net output levels to probability distributions. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Denver, Colorado, 26–29 November 1991; pp. 853–859.

50. Gal, Y.; Ghahramani, Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv* **2015**, arXiv:1506.02158.
51. Hoaglin, D.C.; Mosteller, F.; Tukey, J.W. *Understanding Robust and Exploratory Data Analysis*; Number Sirsi i9780471384915; Wiley-Interscience: Hoboken, NJ, USA, 2000.
52. Denœux, T. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artif. Intell.* **2008**, *172*, 234–264. [CrossRef]
53. Lachaize, M.; Le Hégarat-Mascle, S.; Aldea, E.; Maitrot, A.; Reynaud, R. Evidential split-and-merge: Application to object-based image analysis. *Int. J. Approx. Reason.* **2018**, *103*, 303–319. [CrossRef]
54. Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.* **1999**, *10*, 61–74.
55. Vandoni, J.; Aldea, E.; Le Hégarat-Mascle, S. An evidential framework for pedestrian detection in high-density crowds. In Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
56. Vandoni, J.; Le Hégarat-Mascle, S.; Aldea, E. Belief Function Definition for Ensemble Methods-Application to Pedestrian Detection in Dense Crowds. In Proceedings of the 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 2481–2488.
57. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
58. Tola, E.; Lepetit, V.; Fua, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 815–830. [CrossRef]
59. Li, M.; Bao, S.; Dong, W.; Wang, Y.; Su, Z. Head-shoulder based gender recognition. In Proceedings of the International Conference on Image Processing (ICIP), Melbourne, VIC, Australia, 15–18 September 2013; pp. 2753–2756.
60. Yager, R.R. On the Dempster-Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137. [CrossRef]
61. Aldea, E.; Kiyani, K.H. Hybrid focal stereo networks for pattern analysis in homogeneous scenes. In Proceedings of the Asian Conference on Computer Vision (ACCV), Singapore, 1–5 November 2014; Springer: Cham, Switzerland, 2014; pp. 695–710.
62. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
63. 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 (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1026–1034.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).