

Fusion of evidential CNN classifiers for image classification

Zheng Tong¹[0000-0001-6894-3521], Philippe Xu¹[0000-0001-7397-4808], and
Thierry Denœux^{1,2}[0000-0002-0660-5436]

¹ Université de technologie de Compiègne,
CNRS, UMR 7253 Heudiasyc, Compiègne, France

² Institut universitaire de France, Paris, France

zheng.tong@hds.utc.fr; philippe.xu@hds.utc.fr; thierry.denoeux@utc.fr

Abstract. We propose an information-fusion approach based on belief functions to combine convolutional neural networks. In this approach, several pre-trained DS-based CNN architectures extract features from input images and convert them into mass functions on different frames of discernment. A fusion module then aggregates these mass functions using Dempster’s rule. An end-to-end learning procedure allows us to fine-tune the overall architecture using a learning set with soft labels, which further improves the classification performance. The effectiveness of this approach is demonstrated experimentally using three benchmark databases.

Keywords: Information fusion · Dempster-Shafer theory · Convolutional neural network · Object recognition · Evidence theory

1 Introduction

Deep learning-based models, especially convolutional neural networks (CNNs) [5] and their variants (see, e.g., [9]), have been widely used for image classification and have achieved remarkable success. To train such networks, several image data sets are available, with different sets of classes and different granularities. For instance, a dataset may contain images from dogs and cats, while another one may contain images from several species of dogs. The problem then arises of combining networks trained from such heterogenous datasets. The fusion procedure should be flexible enough to allow the introduction of new datasets with different sets of classes at any stage.

In this paper, we address this classifier fusion problem in the framework of the Dempster-Shafer (DS) theory of belief functions. The DS theory [10], also known as *evidence theory*, is based on representing independent pieces of evidence by mass functions and combining them using a generic operator called Dempster’s rule. DS theory is a well-established formalism for reasoning and making decisions with uncertainty [3]. One of its applications is evidential classifier fusion, in which classifier outputs are converted into mass functions and

fused by Dempster’s rule [8,16]. The information-fusion capacity of DS theory makes it possible to combine deep-learning classifiers.

We present a modular fusion strategy based on DS theory for combining different CNNs. Several pre-trained DS-based CNN architectures, also known as *evidential deep-learning classifiers* in [12], extract features from input images and convert them to mass functions defined on different frames of discernment. A fusion module then aggregates these mass functions by Dempster’s rule, and the aggregated mass function is used for classification in a refined frame. An end-to-end learning procedure allows us to fine-tune the overall architecture using a learning set with soft labels, which further improves the classification performance. The effectiveness of the approach is demonstrated experimentally using three benchmark databases: CIFAR-10 [4], Caltech-UCSD Birds 200 [15], and Oxford-IIIT Pet [7].

The rest of the paper is organized as follows. DS theory is recalled in Section 2. The proposed approach is then introduced in Section 3, and the numerical experiment is reported in Section 4. Finally, we conclude the paper in Section 5.

2 Dempster-Shafer theory

Let $\Theta = \{\theta_1, \dots, \theta_M\}$ be a set of classes, called the *frame of discernment*. A (normalized) *mass function* on Θ is a function m from 2^Θ to $[0,1]$ such that $m(\emptyset) = 0$ and $\sum_{A \subseteq \Theta} m(A) = 1$. For any $A \subseteq \Theta$, each mass $m(A)$ is interpreted as a share of a unit mass of belief allocated to the hypothesis that the truth is in A , and which cannot be allocated to any strict subset of A based on the available evidence. Set A is called a *focal set* of m if $m(A) > 0$. A mass function is *Bayesian* if its focal sets are singletons; it is then equivalent to a probability distribution.

A *refining* from a frame Θ to a frame Ω , as defined in [10], is a mapping $\rho : 2^\Theta \rightarrow 2^\Omega$ such that the collection of sets $\rho(\{\theta\}) \subset \Omega$ for all $\theta \in \Theta$ form a partition of Ω , and

$$\forall A \subseteq \Theta, \quad \rho(A) = \bigcup_{\theta \in A} \rho(\{\theta\}). \quad (1)$$

The frame Ω is then called a *refinement* of Θ . Given a mass function m^Θ on Θ , its vacuous extension $m^{\Theta \uparrow \Omega}$ in Ω is the mass function defined on frame Ω as

$$m^{\Theta \uparrow \Omega}(B) = \begin{cases} m^\Theta(A) & \text{if } \exists A \subseteq \Theta, \quad B = \rho(A), \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

for all $B \subseteq \Omega$. Two frames of discernment Θ_1 and Θ_2 are said to be *compatible* if they have a common refinement Ω .

Two mass functions m_1 and m_2 on the same frame Ω representing independent items of evidence can be combined conjunctively by *Dempster’s rule* [10] defined as follows:

$$(m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{\sum_{B \cap C \neq \emptyset} m_1(B) m_2(C)} \quad (3)$$

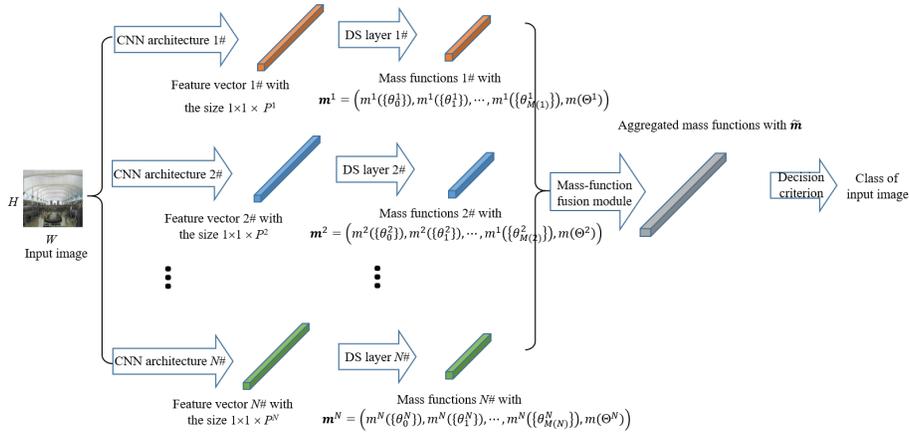


Fig. 1: Architecture of a MF-ECNN classifier.

for all $A \subseteq \Omega$, $A \neq \emptyset$, and $(m_1 \oplus m_2)(\emptyset) = 0$. Mass functions m_1 and m_2 can be combined if and only if the denominator in the right-hand side of (3) is positive. The operator \oplus is commutative and associative. The mass function $m_1 \oplus m_2$ is called the *orthogonal sum* of m_1 and m_2 . Given two mass functions m^{Θ_1} and m^{Θ_2} on compatible frames Θ_1 and Θ_2 , their orthogonal sum $m^{\Theta_1} \oplus m^{\Theta_2}$ is defined as the orthogonal sum of their vacuous extensions in their common refinement Ω : $m^{\Theta_1} \oplus m^{\Theta_2} = m^{\Theta_1 \uparrow \Omega} \oplus m^{\Theta_2 \uparrow \Omega}$.

3 Proposed approach

In this section, we describe the proposed framework for fusion of evidential deep learning classifiers. The overall architecture is first described in Section 3.1. The end-to-end learning procedure is then introduced in Section 3.2.

3.1 Overview

The main idea of the proposed approach is to combine different pre-trained evidential CNN classifiers by plugging a mass-function fusion module at the outputs of these CNN architectures. The architecture of the proposed approach, called *mass-fusion evidential CNN (MFE-CNN) classifier*, is illustrated in Figure 1 and can be defined by the following three-step procedure.

Step 1: An input image is classified by N pre-trained DS-based CNN architectures [12]. The n -th CNN architecture, $n = 1, \dots, N$, extracts a feature vector from the input, as done in a probabilistic CNN [5]. The vector is then fed into an evidential distance-based neural-network layer for constructing mass functions, called *the DS layer* [1]. Each unit in this layer computes a mass function on the frame of discernment Θ^n composed of $M(n)$ classes

$\theta_1^n, \dots, \theta_{M(n)}^n$ and an “anything else” class θ_0^n , based on the distance between the feature vector and a prototype. The mass on Θ^n is larger when the feature vector is farther from the prototype. The mass functions computed by each of the hidden units are then combined by Dempster’s rule. Given the design of the DS layer, the focal sets of mass function m^n are the singletons $\{\theta_k^n\}$ for $k = 1, \dots, M(n)$ and Θ^n . The outputs after this first step are the N mass functions m^1, \dots, m^N defined on N compatible frames $\Theta^1, \dots, \Theta^N$.

Step 2: A mass-function fusion module aggregates the N mass functions by Dempster’s rule. Let Ω be a common refinement of the N frames $\Theta^1, \dots, \Theta^N$. A combined mass function \tilde{m} on Ω is computed as the orthogonal sum of the N mass functions $\tilde{m} = m^1 \oplus \dots \oplus m^N$. This final output of the mass-function fusion module represents the total evidence about the class of the input image based on the outputs of the N CNN classifiers.

Step 3: The pignistic criterion [11][2] is used for decision-making: the mass function \tilde{m} is converted into the pignistic probability $BetP_{\tilde{m}}$ as

$$BetP_{\tilde{m}}(\{\omega\}) = \sum_{\{A \subseteq \Omega: \omega \in A\}} \frac{\tilde{m}(A)}{|A|},$$

for all $\omega \in \Omega$, and the final prediction is $\hat{\omega} = \arg \max_{\omega \in \Omega} BetP_{\tilde{m}}(\omega)$.

3.2 Learning

An end-to-end learning procedure is proposed to fine-tune all the parameters in a MFE-CNN classifier using a learning set with soft labels, in order to improve the classification performance. In the procedure, the learning sets of different pre-trained CNN architectures are merged into a single one. As some labels become imprecise after merging, they are referred to as *soft labels*. For example, the “bird” label in the CIFAR-10 [4] database becomes imprecise when the database is merged with the Caltech-UCSD Birds 200 database containing 200 bird species [15]. To fine-tune the different classifiers using a learning set with soft labels, we define a label as a nonempty subset $A \in 2^\Omega \setminus \emptyset$ of classes an image may belong to. Label A indicates that the true class is known to be one element of set A , but one cannot determine which one specifically if $|A| > 1$.

In the fine-tuning phase, all parameters in a MFE-CNN classifier are initialized by the parameters in the pre-trained CNNs. Given a learning image with nonempty soft label $A \subseteq \Omega$ and output pignistic probability $BetP_{\tilde{m}}$, we define the loss as:

$$\mathcal{L}(BetP_{\tilde{m}}, A) = -\log BetP_{\tilde{m}}(A) = -\log \sum_{\omega \in A} BetP_{\tilde{m}}(\omega). \quad (4)$$

This loss function is minimized when the pignistic probability of soft label A equals 1. The gradient of this loss w.r.t all network parameters can be back-propagated from the output layer to the input layer.

Table 1: Lists of classes in the CIFAR-10, CUB, Oxford databases. The notations θ_0^2 and θ_0^3 stand for the “anything else” class added to the frames of the CUB and Oxford databases.

Frame	Class
CIFAR-10	airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.
CUB	cardinal, house wren, . . . , (200 species of birds), θ_0^2 .
Oxford	bengal, boxer, . . . , (37 species of cats and dogs), θ_0^3 .
Common frame	airplane, automobile, deer, frog, horse, ship, truck, cardinal, house wren, . . . , (200 species of birds), bengal, boxer, . . . , (37 species of cats and dogs).

4 Experiment

In this section, we study the performance of the above fusion method through a numerical experiment. The databases and metrics are first introduced in Section 4.1. The results are then discussed in Section 4.2.

4.1 Experimental settings

Three databases are considered in this experiment: CIFAR-10 [4], Caltech-UCSD Birds-200-2010 (CUB) [15], and Oxford-IIIT Pet [7]. The CIFAR-10 database was pre-split into 50k training and 10k testing images. For the CUB (6,033 images) and Oxford (7,349 images) databases, we divided each database into training and testing sets with a ratio of about 1:1. The training and testing sets keep the ratio of about 1:1 in each class. In the fine-tuning procedure, the frames of the three databases are refined into a common one, as shown in Table 1. Thanks to the “anything else” classes, the three frames are compatible. After merging the three databases, there are 56,692 training samples and 16,690 testing samples for, respectively, fine-tuning and performance evaluation.

For a testing set T with soft labels, the *average accuracy* is defined as

$$AA(T) = \frac{1}{|T|} \sum_{i \in T} \mathbb{1}_{A(i)}(\hat{\omega}(i)), \quad (5)$$

where $A(i)$ is the soft label of sample i , $\hat{\omega}(i)$ is the predicted class, and $\mathbb{1}_{A(i)}$ is the indicator function of set $A(i)$.

The three CNNs used for the three datasets have the same FitNet-4 [9] architecture with 128 output units. The numbers of prototypes in the DS layers for the CIFAR-10, CUB, and Oxford databases are, respectively, 70, 350, and 80. We compared the proposed classifier to four classifier fusion systems with the same CNN architectures:

Probability-to-mass fusion (PMF) [16]: we feed the feature vector from each CNN n into a softmax layer to generate a Bayesian mass function on Θ^n . The mass functions from the three CNNs are then combined by Dempster’s rule. (It should be noted that the vacuous extension of each Bayesian mass function in the common refinement Ω is no longer Bayesian.)

Table 2: Test classification accuracies of different classifiers. “E2E” stands for fine-tuned classifiers. E- and P-FitNit-4 are the evidential and probabilistic CNN classifiers before fusion.

	Classifier	CIFAR-10	CUB	Oxford	Overall
Before fusion	E-FitNit-4	0.936	0.934	0.898	-
	P-FitNit-4	0.935	0.926	0.895	-
After fusion	MFE	0.950	0.934	0.901	0.936
	PMF	0.941	0.927	0.898	0.929
	BF	0.938	0.911	0.889	0.923
	E2E MFE	0.955	0.935	0.902	0.940
	E2E PMF	0.946	0.927	0.899	0.932
	E2E BF	0.938	0.913	0.891	0.924
	E2E EFC	0.931	0.928	0.887	0.921
E2E PFC	0.938	0.936	0.903	0.930	

Bayesian-fusion (BF) [14]: the feature vector from each CNN is converted into a Bayesian mass function on the common frame Ω , by equally distributing the mass of a focal set to its elements; the obtained Bayesian mass functions are combined by Dempster’s rule. This procedure is equivalent to Bayesian fusion.

Probabilistic feature-combination (PFC) [6]: the three feature vectors are concatenated to form a new vector of length 384, which is fed into a softmax layer to generate the probability distribution on the common frame.

Evidential feature-combination (EFC): feature vectors are concatenated as in the above PFC approach, but the aggregated vector is fed into a DS layer to generate an output mass function. The dimension of the aggregated vector and the number of prototypes are, respectively, 192 and 400, to obtain optimal performance of the EFC-CNN classifier.

4.2 Results

Table 2 shows the test average accuracies of the evidential and probabilistic classifiers trained from each of the three datasets, as well as the performances of the different fusion strategies (with and without fine tuning) on each individual dataset, and on the union of the three datasets.

Looking at the performance of the MFE strategy, we can see that, after fusion, the accuracies on the CIFAR-10 and Oxford databases increase, but the accuracy on the CUB database does not change. As shown in Table 3, the accuracies for the “cat”, “dog”, “bird” classes on the CIFAR-10 database increase, but the ones of other classes do not change after fusion. These observations show that the proposed approach makes it possible to combine CNN classifiers trained from heterogenous databases to obtain a more general classifier able to recognize classes from any of the databases, without degrading the performance of the individual classifiers, and sometimes even yielding better results for some classes.

Table 3: Test accuracy before and after information fusion on CIFAR-10.

	Classifier	aero	mobile	bird	cat	deer	dog	frog	horse	ship	truck
Before fusion	E-FitNit-4	0.976	0.961	0.936	0.865	0.910	0.899	0.944	0.932	0.965	0.973
	P-FitNit-4	0.984	0.974	0.913	0.843	0.904	0.875	0.958	0.947	0.981	0.974
After fusion	E2E MFE	0.978	0.961	0.981	0.937	0.915	0.961	0.945	0.935	0.965	0.973
	E2E PMF	0.984	0.975	0.950	0.872	0.910	0.908	0.958	0.947	0.982	0.974
	E2E BF	0.985	0.975	0.919	0.860	0.910	0.890	0.959	0.948	0.982	0.975

Table 4: Examples of mass functions (MF's) before and after fusion by the MFE strategy. (Only some masses before and after fusion are shown for lack of space).

Instance/label	Before fusion			MF on Ω
	MF from CIFAR-10	MF from CUB	MF from Oxford	after fusion
 /bird	$m(\{\text{airplane}\}) = 0.506$	$m(\{\text{caspinan}\}) = 0.698$	$m(\{\text{samyod}\}) = 0$	$m(\{\text{airplane}\}) = 0.101$
	$m(\{\text{bird}\}) = 0.382$	$m(\{\text{horned grebe}\}) = 0.109$	$m(\{\text{pyrenees}\}) = 0.001$	$m(\{\text{caspinan}\}) = 0.672$

	$m(\theta^1) = 0.065$	$m(\theta_0^2) = 0.098$	$m(\theta_0^3) = 0.905$	$m(\Omega) = 0.007$
 /caspinan	$m(\{\text{airplane}\}) = 0.009$	$m(\{\text{caspinan}\}) = 0.423$	$m(\{\text{samyod}\}) = 0$	$m(\{\text{caspinan}\}) = 0.415$
	$m(\{\text{bird}\}) = 0.823$	$m(\{\text{horned grebe}\}) = 0.452$	$m(\{\text{pyrenees}\}) = 0.001$	$m(\{\text{horned grebe}\}) = 0.450$

	$m(\theta^1) = 0.092$	$m(\theta_0^2) = 0.084$	$m(\theta_0^3) = 0.951$	$m(\Omega) = 0.009$
 /byssinian	$m(\{\text{cat}\}) = 0.742$	$m(\{\text{caspinan}\}) = 0.002$	$m(\{\text{byssinian}\}) = 0.412$	$m(\{\text{byssinian}\}) = 0.414$
	$m(\{\text{dog}\}) = 0.131$	$m(\{\text{horned grebe}\}) = 0$	$m(\{\text{bengal}\}) = 0.503$	$m(\{\text{bengal}\}) = 0.505$

	$m(\theta^1) = 0.032$	$m(\theta_0^2) = 0.931$	$m(\theta_0^3) = 0.038$	$m(\Omega) = 0.005$
 /keeshond	$m(\{\text{cat}\}) = 0.158$	$m(\{\text{albatross}\}) = 0.001$	$m(\{\text{rogdoll}\}) = 0.682$	$m(\{\text{rogdoll}\}) = 0.369$
	$m(\{\text{dog}\}) = 0.705$	$m(\{\text{horned grebe}\}) = 0$	$m(\{\text{keeshond}\}) = 0.254$	$m(\{\text{keeshond}\}) = 0.485$

	$m(\theta^1) = 0.058$	$m(\theta_0^2) = 0.975$	$m(\theta_0^3) = 0.001$	$m(\{\text{cat}\}) = 0.021$

Table 4, which shows examples of mass functions computed by the different classifiers, allows us to explain the good performance of the MFE fusion strategy. The first example from the CIFAR-10 database is misclassified using only the mass function from the classifier trained from this dataset, but the decision is corrected after the evidential fusion because the mass function provided by the classifier trained from the CUB data supports some bird species. In contrast, for images misclassified by the CUB mass function, such as the second instance, the other two individual classifiers do not provide any useful information to correct the decisions. Consequently, the classification performance on the CUB database is not improved. Besides, the mass function from the CIFAR-10 classifier sometimes includes useful information for classifying samples into one species of cat or dog, such as the final instance, even though the number of such examples is small. This phenomenon is responsible for the small change in the performance on the Oxford database.

Comparing the test accuracies of the MFE classifiers with and without the end-to-end learning procedure as shown in Table 2, we can see that fine tuning further boosts the overall performance, as well as the performance on the CIFAR-10 and Oxford databases. Thus, the fine-tuning procedure improves the classification accuracy of the proposed architecture, and can be seen as a way to improve the performance of CNN classifiers. This is because the end-to-end

learning procedure adapts the individual classifiers to the new classification problem. More specifically, before fusion, the CNN classifiers are pre-trained for the classification tasks with the frames of discernment before refinement. The proposed end-to-end learning procedure fine-tunes the parameters in the CNN and DS layers to make them more suitable to the classification task in the refined frame.

Finally, Table 2 sheds some light on the relative performance of different classifier fusion strategies. The PMF fusion strategy also improves the performance of the probabilistic CNNs trained on each of the three databases, but it is not as good as the proposed method. In contrast, the Bayesian fusion strategy (BF) degrades the performance of the individual classifiers, which shows that the method is not effective when the numbers of classes in the different frames are very unbalanced. The relatively low accuracies obtained of the two feature fusion strategies (E2E EFC and E2E PFC) show that this simple fusion method is less effective than the other ones. All in all, the proposed evidential fusion strategy outperforms the other tested methods on the datasets considered in this experiment.

5 Conclusion

In the study, we have proposed a fusion approach based on belief functions to combine different CNNs for image classification. The proposed approach makes it possible to combine CNN classifiers trained from heterogeneous databases with different sets of classes. The combined classifier is able to classify images from any of these databases while having at least as good performance as those of the individual classifiers on their respective databases. Besides, the proposed approach makes it possible to combine additional classifiers trained from new datasets with different sets of classes at any stage. An end-to-end learning procedure further improves the performance of the proposed architecture. This approach was shown to outperform other decision-level or feature-level fusion strategies for combining CNN classifiers. Future work will consider combining evidential fully convolutional networks for pixel-wise semantic segmentation [13].

References

1. Denoeux, T.: A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **30**(2), 131–150 (2000)
2. Dencoux, T.: Decision-making with belief functions: a review. *International Journal of Approximate Reasoning* **109**, 87–110 (2019)
3. Dencoux, T., Dubois, D., Prade, H.: Representations of uncertainty in artificial intelligence: Beyond probability and possibility. In: *A Guided Tour of Artificial Intelligence Research*, vol. 1, chap. 4, pp. 119–150. Springer Verlag (2020)
4. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)

5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* **521**(7553), 436–444 (2015)
6. Nguyen, L.D., Lin, D., Lin, Z., Cao, J.: Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–5. IEEE (2018)
7. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition. Providence, Rhode Island (2012)
8. Quost, B., Masson, M.H., Dencœux, T.: Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules. *International Journal of Approximate Reasoning* **52**(3), 353–374 (2011)
9. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: 3rd International Conference on Learning Representations. San Diego, USA (2015)
10. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
11. Smets, P.: Constructing the pignistic probability function in a context of uncertainty. In: Henrion, M., Schachter, R.D., Kanal, L.N., Lemmer, J.F. (eds.) *Uncertainty in Artificial Intelligence 5*, pp. 29–40. North-Holland, Amsterdam (1990)
12. Tong, Z., Xu, P., Denoeux, T.: An evidential classifier based on Dempster-Shafer theory and deep learning. *Neurocomputing* **450**, 275–293 (2021)
13. Tong, Z., Xu, P., Dencœux, T.: Evidential fully convolutional network for semantic segmentation. *Applied Intelligence* (2021), <https://doi.org/10.1007/s10489-021-02327-0>
14. Wei, Q., Dobigeon, N., Tournet, J.Y.: Bayesian fusion of multi-band images. *IEEE Journal of Selected Topics in Signal Processing* **9**(6), 1117–1127 (2015)
15. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
16. Xu, P., Davoine, F., Bordes, J.B., Zhao, H., Dencœux, T.: Multimodal information fusion for urban scene understanding. *Machine Vision and Applications* **27**(3), 331–349 (2016)