MSFA-Net: A convolutional neural network based on multispectral filter arrays for texture feature extraction

Anis Amziane, Olivier Losson, Benjamin Mathon, Ludovic Macaire

MSFA-Net: A convolutional neural network based on multispectral filter arrays for texture feature extraction

Anis Amziane*, Olivier Losson, Benjamin Mathon and Ludovic Macaire

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

ARTICLE INFO

Keywords:
Multispectral imaging
Texture feature extraction
Multispectral filter array
Supervised classification
Precision farming

ABSTRACT

Multispectral snapshot cameras fitted with a multispectral filter array (MSFA) acquire several spectral bands in one shot and provide a raw mosaic image in which a single channel value is available at each pixel. Texture features are classically extracted from fully-defined images that are estimated by demosaicing. Such an estimation may however cause spatio-spectral artifacts. Moreover, texture feature extraction becomes computationally inefficient and yields to high-dimensional features as the number of bands increases. In this paper, we propose an original approach based on a convolutional neural network called MSFA-Net to capture spatio-spectral interactions in raw images at reduced computation costs. Experiments of multispectral image classification and outdoor image segmentation show that the proposed approach outperforms several hand-crafted and deep learning-based feature extractors.

1. Introduction

Increasing the number of bands to enhance spectral resolution is a goal of multispectral imaging. Multispectral cameras embed several optical filters so that material surfaces are observed in several spectral bands. Depending on the type of filters that sample the incident light (radiance), multispectral images may contain spectral information from the visible (VIS), the near infrared (NIR), and/or the short wave infrared domain. "Multishot" devices [1] build an image by stacking several successive frames. Oppositely, "snapshot" devices provide a multispectral image from a single shot [2]. Multi-sensor snapshot devices use dichroic prisms to split the incoming beam onto multiple sensors, hence are expensive and can only sample few spectral bands. Single-sensor snapshot devices embed a multispectral filter array (MSFA) laid over the sensor, like the widely-used Bayer filter array in color imaging, to spatio-spectrally sample incoming radiance according to the photosensor locations. Each filter of the MSFA is sensitive to a specific narrow spectral band, so that each pixel of the acquired raw image represents a single band. The missing other ones are computed by demosaicing to recover the fully-defined multispectral image [3]. Outdoor applications require illumination-independent spectral signatures. To this end, reflectance images are classically computed from demosaiced radiance ones, and texture features extracted from them.

Because demosaicing generates artifacts and increases computation costs, some authors propose to directly process raw images for reflectance estimation [4] or feature extraction [5]. In [6], texture features based on the local binary pattern (LBP) operator are directly extracted from raw images. In the same goal, we here exploit deep learning advantages and design a convolutional neural network (CNN) that acts as a texture feature extractor from raw images.

This paper is organized as follows. In Sec. 2, we first discuss the classical approach for multispectral texture feature extraction and review some state-of-the-art texture descriptors. Then, we introduce our approach for texture feature extraction from raw images. In Secs. 3 and 4, we evaluate the proposed method against state-of-the-art ones for multispectral image classification and segmentation. Finally, conclusions are drawn in Sec. 5.

2. Multispectral texture feature extraction

2.1. Texture features from multispectral images

To classify texture images provided by a single-sensor snapshot camera that samples B^2 bands through an MSFA, one usually estimates fully-defined (B^2-channel) images from raw images by demosaicing, then computes texture features [7]. The simplest demosaicing scheme uses a weighted bilinear interpolation filter to estimate the $B^2 - 1$ values that miss at each pixel from those available at its neighbors for the same respective bands [8]. Each neighbor is associated with a weight that depends on its spatial distance to the considered pixel. This method only exploits intra-channel spatial correlation to estimate missing values. To improve the estimation, demosaicing should use inter-channel correlation or, if the latter is low, the correlation between each channel and the pseudo-panchromatic image (PPI) [9]. The PPI is first estimated from the raw image thanks to an averaging filter. Then its sharpness is improved using local directional variations of raw values. Finally, the PPI is analyzed by an iterative procedure

Among local texture features, those based on the LBP operator and its variants have been widely used for their robustness against illumination, rotation, and scale [10]. They have also been extended to the multispectral domain using vector approaches [11]. Considering spectral correlation

*Corresponding author
anisamziane6810@gmail.com (A. Amziane); olivier.losson@univ-lille.fr (O. Losson); benjamin.mathon@univ-lille.fr (B. Mathon); ludovic.macaire@univ-lille.fr (L. Macaire)
ORCID(s): 0000-0002-2480-9205 (A. Amziane)

between all bands provides state-of-the-art texture classification performance, but is computationally greedy and yields high-dimensional LBP features that are neither memory efficient nor easily interpretable [6]. The relative spectral difference occurrence matrix (RSDOM) [12] performs spectral differences using the Kullback-Leibler pseudo-divergence measure to extract low-dimensional texture features from multispectral images as a multi-dimensional probability density function. Such hand-crafted texture features have recently been overshadowed by deep learning techniques based on CNNs [13, 14]. For instance, the SegNet model has been extensively and successfully applied to segment scene images [14], and to analyze multispectral (RGB-NIR) images in the context of weed detection [15]. The ResNet model and its variants [16] are advanced deep CNNs that use residual blocks to improve the classification performance and avoid the vanishing gradient issue.

CNNs are commonly used as feature extractors and classifiers. Hidden layers perform feature extraction, and the output one (usually a softmax function in multi-class case) turns features from the last hidden layer into probabilities for prediction. Some authors propose to use CNNs only as feature extractors. Donahue et al. [17] show that features provided by deep hidden layers, especially the last two ones, are highly discriminant and provide astonishing classification performances when combined with a supervised classifier. Zhou et al. [18] use deep features of the last hidden (fully-connected) layer to train a linear support vector machine (SVM) classifier for scene classification. Razavian et al. [19] conduct different recognition tasks (e.g., object detection, visual instance and fine-grained recognition) using the Overfeat CNN model. In each experiment, features of the first fully-connected layer are L_2 normalized and used to train an SVM classifier to perform predictions. The 2D-CNN called S-CNN is used to extract features from multispectral images [20, 21]. These features are used by an SVM classifier to perform multispectral band selection and face recognition. This deep scheme outperforms state-of-the-art hand-crafted descriptors such as HOG, LBP, and SIFT [20]. In spite of their performances, deep learning-based approaches are greedy in computation time and memory, and using them with high spectral resolution images may be intractable.

2.2. MSFA texture features

To consider spatio-spectral correlation, some studies directly process raw images [5, 6]. When a descriptor suitably analyzes a raw image, it can achieve similar or even better classification performances than from a demosaiced one because demosaicing generates artifacts that may alter the texturMSFA Texture featue representation. In [6], texture features are directly computed from raw images, which avoids the demosaicing step and provides discriminant features. Specifically, the method analyzes a raw image with respect to the MSFA basic pattern and its band arrangement to build an LBP-based texture descriptor. From the same idea, we propose a new CNN architecture that is adapted to raw images.

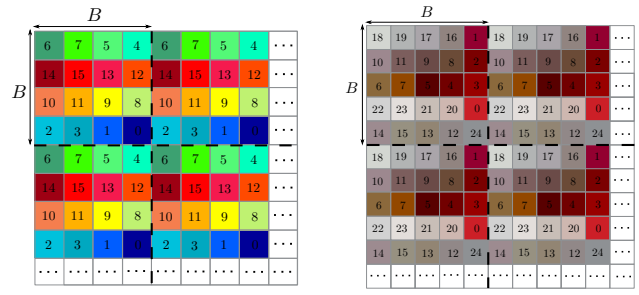

Figure 1: Considered MSFAs: IMEC VIS 4×4 ($\lambda^b \in \{469\,nm, \ \ldots, \ 633\,nm\}$, $b \in [\![0, 15]\!]$) (left) and NIR 5×5 ($\lambda^b \in \{678\,nm, \ \ldots, \ 960\,nm\}$, $b \in [\![0, 24]\!]$) (right).

The MSFAs used in this paper are defined by repetition of a $B \times B$ basic pattern that samples B^2 different bands. No consensus exists regarding the size of the basic pattern, and finding a trade-off between spatial and spectral samplings is a challenging open problem [22] that is beyond the scope of this paper. Therefore, we follow the MSFA arrangements in the VIS domain ($B = 4$) and NIR domain ($B = 5$) of two snapshot cameras manufactured by IMEC [23] (see Fig. 1).

2.3. Raw texture features based on CNN

Our CNN architecture called MSFA-Net directly extracts texture features from raw square patches of size $X \times X$ pixels, where $X = m \cdot B$ is a multiple of the MSFA basic pattern width. MSFA-Net is composed of three convolutional blocks, followed by an average pooling layer and two fully-connected layers (see Fig. 2). The first convolutional layer is of utmost importance because it guides the feature extraction according to the MSFA basic pattern. It uses 128 convolutional kernels $\{H_n\}_{n=0}^{127}$ of size $B \times B$ and depth 1, with a stride of B pixels along both spatial dimensions and without padding. B-pixel stride ensures that each kernel coefficient is always associated to the same MSFA band for all convolutions. This first layer learns spatio-spectral interactions among channel values in each raw patch part that matches the basic MSFA pattern. The convolution between a raw patch P^{raw} and a kernel H_n, $n \in [\![0, 127]\!]$, is defined at each pixel $(x, y) \in [\![0, m-1]\!]^2$ as:

$$O_n(x, y) = \sum_{i=0}^{B-1} \sum_{j=0}^{B-1} H_n(i, j) \cdot P^{\text{raw}}(B \cdot x + i, B \cdot y + j). \quad (1)$$

The resulting 128 feature maps $\{O_n\}_{n=0}^{127}$ of size $m \times m$ are fed into the second convolutional block that uses 256 kernels of size 3×3 with both a stride and zero-padding of one pixel, such that its input and output feature maps have the same size. The last convolutional block uses 384 kernels of size 3×3 with one pixel stride and no padding. Feature maps of the last convolutional layer are usually vectorized using a flattening layer before being fed into fully-connected layers. Following [18], we introduce a global pooling layer to average feature maps channel-wise so that the provided 384-dimensional feature vector is more robust against noise and

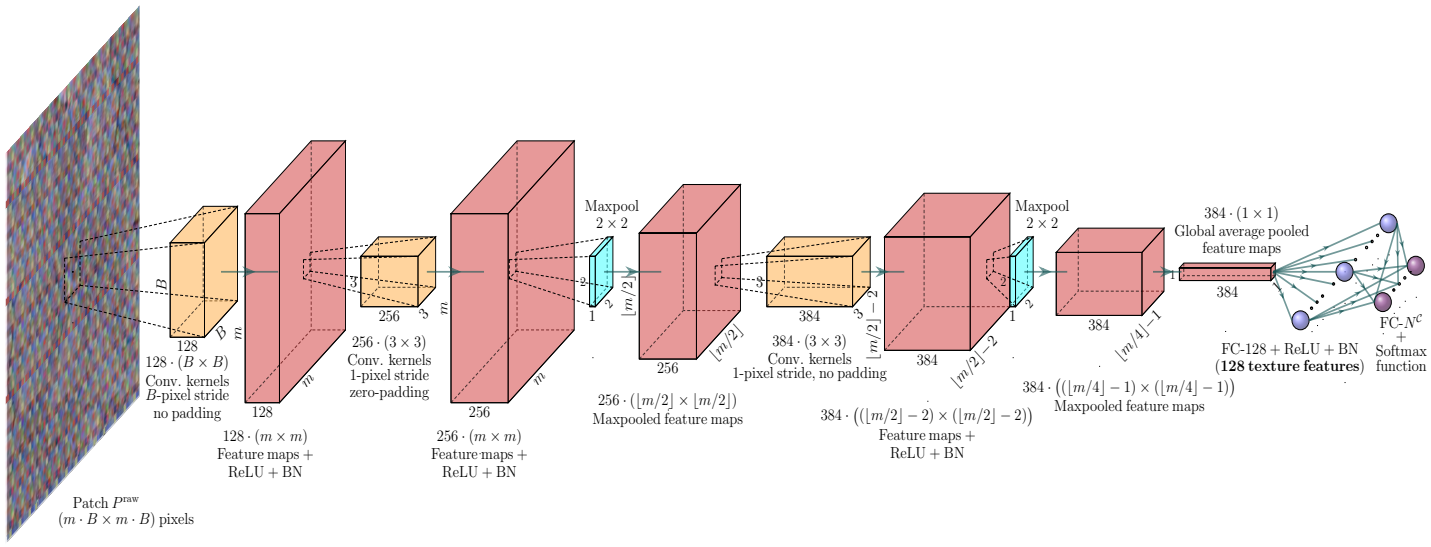

Figure 2: MSFA-Net architecture. ReLU: rectified linear unit activation, BN: batch normalization, FC: fully-connected layer. Filter depths (e.g., 1 for first layer and 128 for second) are not shown for sake of clarity.

spatial translations. To introduce non-linearity and reduce the feature size, a fully-connected layer provides the final 128-dimensional texture feature vector that is fed into the softmax layer.

3. Texture classification experiment

In this first experiment, we use a dataset of multispectral texture images called HyTexiLa [24]. It contains $N^C = 112$ reflectance images, each of which can be regarded as a distinctive class. An image $\mathbf{R}^{(K)} = \{R^k\}_{k=0}^{K-1}$ has $K = 186$ channels of size 1024×1024 pixels, and each channel R^k is associated to a spectral band of central wavelength $\lambda^k \in [405.37\,\text{nm},\ 995.83\,\text{nm}]$.

3.1. Patch extraction

Each K-channel reflectance image is first transformed into a B^2-channel one ($B^2 \in \{16, 25\}$) by selecting the channels whose associated wavelengths are the closest to the spectral sensitivity function centers of IMEC snapshot cameras. To obtain raw patches for feature extraction, we simulate raw images that would be acquired by these cameras by spatio-spectrally sub-sampling the fully-defined B^2-channel images according to the 4×4 or 5×5 MSFA. To compare with the classical strategy, we also demosaic these MSFA images using PPID as one of the state-of-the-art multispectral demosaicing methods [9]. We discard 10 pixels on image borders where estimation is inaccurate.

Each image is then split into non-overlapping square patches of width $X = m \cdot B$, some of which are picked for training and the others used for testing. To evaluate how patch size affects the descriptors, we extract patches of sizes 200×200, 124×124, and 64×64 pixels (for $B = 4$), or 200×200, 125×125, and 65×65 pixels ($B = 5$). We furthermore perform data augmentation to introduce some texture variations and to ensure that enough patches are available to train the network. We consider seven transformations, namely Gaussian noise ($\mu = 0$, $\sigma = 4.4$), 180° rotation, horizontal flip, vertical flip, random resized crop, grid distortion, and elastic transform. Only the latter two are used with the smallest patches ($X \in \{64, 65\}$). The number N of training and test patches is then $N \approx 37.8 \cdot 10^3$ for $X \in \{64, 65\}$, $N \approx 28.7 \cdot 10^3$ for $X \in \{124, 125\}$, and $N \approx 11.2 \cdot 10^3$ for $X = 200$.

3.2. Feature extraction

We consider several state-of-the-art descriptors, either deep learning-based or hand-crafted ones, to compare their performance to ours. We use a shallower version of SegNet model [14], called SegNet-Basic, that we adapt to image classification by considering the sole encoder part with an extra flattening layer to vectorize feature maps. All convolutional kernels are of size 3×3 instead of 7×7 to capture small details (and reduce the number of hyperparameters to learn), and we add two fully-connected layers for non-linearity and dimension reduction, which provides a 512-dimensional feature vector. We also consider the S-CNN deep learning model [20] that is composed of three convolutional and two fully-connected layers, whose first one provides a 1024-dimensional feature vector. At last, we consider feature extraction by deep residual learning [16]. The 18-layer architecture (ResNet18) is composed of a starting convolutional block and 8 residual blocks, followed by a global average pooling layer that provides a 512-dimensional feature vector. Each residual block can learn identity mapping thanks to a skip connection of (at least) two convolutional layers. In all models (see details in Table 1), the texture feature vector is finally fed into a fully-connected layer that uses the softmax function to provide the N^C-dimensional probability vector. Note that oppositely to MSFA-Net (see Eq. (1)), the first

Table 1
CNN architectures used for feature extraction from fully-defined images. The detailed architecture of ResNet18 is available at `https://paperswithcode.com/model/resnet`. See caption and colors of Fig. 2. Blue boxes indicate the layers that provide the final texture features we consider for classification.

SegNet-Basic encoder	S-CNN
$64 \cdot (3 \times 3)$ Conv. kernels 1-pixel stride, zero-padding	$96 \cdot (6 \times 6)$ Conv. kernels 2-pixel stride, zero-padding
BN + ReLU	BN + ReLU
Maxpool 2×2	Maxpool 2×2
$128 \cdot (3 \times 3)$ Conv. kernels 1-pixel stride, zero-padding	$256 \cdot (3 \times 3)$ Conv. kernels 2-pixel stride, zero-padding
BN + ReLU	BN + ReLU
Maxpool 2×2	Maxpool 2×2
$256 \cdot (3 \times 3)$ Conv. kernels 1-pixel stride, zero-padding	$512 \cdot (3 \times 3)$ Conv. kernels 1-pixel stride, zero-padding
BN + ReLU	ReLU
Maxpool 2×2	Flatten
$512 \cdot (3 \times 3)$ Conv. kernels 1-pixel stride, zero-padding	FC-1024 + ReLU
BN + ReLU	Dropout(0.5)
Maxpool 2×2	-
Flatten	-
FC-1024 + ReLU	-
FC-512+ ReLU	-
FC-N^C + Softmax	FC-N^C + Softmax

convolutional layer of these models applies kernels $\mathbf{H}_n^{(B^2)}$ of depth B^2 and size $w \times w$ to the input B^2-channel patch $\mathbf{P}^{(B^2)}$ at each pixel $(x, y) \in [\![0, (m \cdot B) - 1]\!]^2$ in a classical way (with 1-pixel stride, right and bottom zero-padding):

$$O_n(x, y) = \sum_{b=0}^{B^2-1} \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} H_n^b(i, j) \cdot P^b(x + i, y + j). \quad (2)$$

Note that $w = 3$ for SegNet-Basic, $w = 6$ for S-CNN, and $w = 7$ for ResNet18.

As hand-crafted features, we compute histograms of LBP operators that have proven to be powerful for texture extraction, namely the marginal LBP (as baseline) [6], the local angular patterns (LAP) [11], the LBP-LCC descriptor [25] that is a fusion of LBP features extracted from the pseudo-panchromatic image and the local color contrast (LCC) patterns, and the M-LBP descriptor that extracts features from raw patches [6].

3.3. CNN Training

To train and validate CNN-based features, 95% of the training patches are used for learning and the remainder for validation. The CNNs are then trained using the learning patches as follows.

Initialization: As we aim to train MSFA-Net from scratch, we must choose an appropriate way to initialize the weights of its convolutional and fully-connected layers. Initializing the weights with too small values may slow down the learning process and lead to vanishing gradients, and larger values may cause exploding gradients. Since all compared CNNs use the ReLU as activation function, we follow He et al.'s weight initialization [26]. Specifically, we use the He-uniform variant that draws the weights from a uniform distribution instead of a normal one because it slightly provides better results in our case.

Training: We use the stochastic gradient descent (SGD) weight optimizer for all models. The loss function to be minimized is the multi-class log-loss, and the optimization is performed for 40 epochs because no performance increase is observed afterwards. The batch size is set to 128. For SegNet-Basic, we use a fixed learning rate $\epsilon = 10^{-2}$, no weight decay, and momentum of 0.9 [14]. For S-CNN, we follow a similar procedure to that described in [20] with minor changes. We use $\epsilon = 0.005$ instead of 0.05 because it provides better convergence on HyTexiLa. Weight decay and momentum values are kept to $5 \cdot 10^{-4}$ and 0.9. For ResNet18, we use $\epsilon = 0.1$ with a weight decay of 10^{-4} and momentum of 0.9 [16]. We train MSFA-Net with the same learning parameters as those of S-CNN.

3.4. Classification results and discussion

Table 2 shows the classification results obtained for each feature with 1-nearest neighbor (NN) classifier coupled with the Euclidean distance.

Among hand-crafted features, M-LBP outperforms the other LBP-based descriptors because it considers spatio-spectral correlation in the raw image and avoids the demosaicing step, whose estimation potentially affects texture representation. Though marginal LBP does not consider inter-channel correlation, it provides better results than LAP. Globally, the performance of all descriptors increases with respect to the patch size, especially marginal LBP and LAP that are sensitive to the number of patch pixels.

Texture features extracted by deep learning from demosaiced patches outperform hand-crafted ones. SegNet-Basic, S-CNN, and ResNet18 encoders are little affected by the patch size and outperform M-LBP-based features with IMEC 5×5 in most cases. As noticed in [6], the performances of M-LBP, LAP, LBP-LCC, and marginal LBP are better with IMEC 4×4 than with IMEC 5×5, though the latter contains richer spectral information. This suggests that LBP-based features perform better on HyTexiLa in the VIS domain than the NIR domain. We observe the opposite behavior with the three state-of-the-art deep learning-based descriptors. CNNs may learn more spectral characteristics with IMEC 5×5 that samples more channels than IMEC 4×4, regardless of their spectral domains.

Our proposed approach is ranked first five times among the six tested cases, followed by ResNet18 that performs best once. This confirms that features provided by MSFA-Net are more discriminant despite their small size. The accuracy reached by MSFA-Net is close to that of RSDOM descriptor (98.5%) applied to higher-dimensional (204×204 pixels $\times$ 186 channels) and original (undemosaiced) fully-defined HyTexiLa patches [12]. MSFA-Net requires to learn much

Table 2
Classification accuracy (%) on HyTexiLa database for 1-NN classifier with features extracted from either raw or demosaiced patches. The best result in each column is shown as bold and second best as italics. Superscript * refers to IMEC 4×4 and † to IMEC 5×5.

Input patches	Feature	size	IMEC $4\times4^*$ 200×200	IMEC $4\times4^*$ 124×124	IMEC $4\times4^*$ 64×64	IMEC $5\times5^\dagger$ 200×200	IMEC $5\times5^\dagger$ 125×125	IMEC $5\times5^\dagger$ 65×65
MSFA	MSFA-Net	$128^{*,\dagger}$	**99.5**	**98.3**	**98.7**	**99.0**	**98.4**	95.1
	M-LBP	$4096^*/6400^\dagger$	*97.2*	*96.9*	*94.6*	96.9	95.4	90.4
Demosaiced	SegNet-Basic	$512^{*,\dagger}$	86.2	83.5	86.4	97.1	96.9	*96.6*
	S-CNN	$1024^{*,\dagger}$	82.5	83.6	81.1	94.4	97.4	93.4
	ResNet18	$512^{*,\dagger}$	95.5	88.1	81.7	*98.5*	*97.8*	**97.4**
	LAP	$256^{*,\dagger}$	80.0	69.1	41.3	68.4	65.6	36.1
	LBP-LCC	$512^{*,\dagger}$	87.0	83.8	69.6	70.9	71.4	53.7
	Marginal LBP	$4096^*/6400^\dagger$	81.5	76.4	45.9	77.2	71.6	51.2

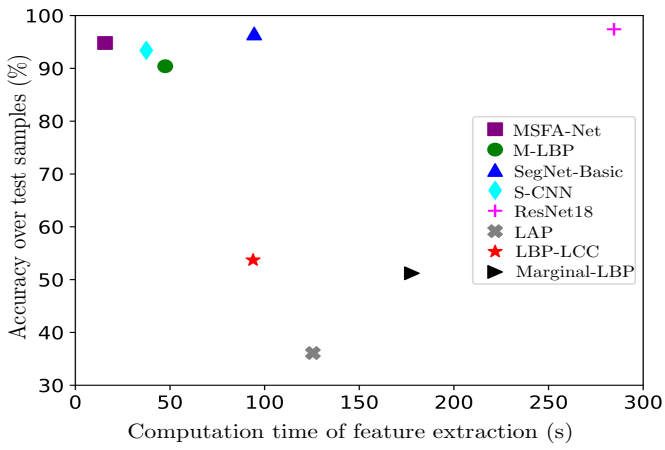

Figure 3: Classification accuracy obtained by 1-NN classifier vs. computation time of feature extraction from the $(0.95\cdot N \approx 35.9\cdot10^3)$ 65×65 learning patches (IMEC 5×5) of HyTexiLa database. Nor demosaicing nor CNN training computation times are considered.

fewer hyperparameters than the other three CNNs, e.g., about nine times fewer than ResNet18 and eight times fewer than SegNet-Basic and S-CNN for the smallest patches. All in all, it provides better or comparable performances than other approaches at much reduced computation costs (see Fig. 3). Table 2 also shows that MSFA-Net performance is less affected by patch size than hand-crafted descriptors, which suggests that it can learn texture feature maps from small patches to perform segmentation tasks.

From the MSFA-Net architecture of Fig. 2, we finally assess variant models with 2 or 4 convolutional layers, built by keeping the first one and feature size untouched. Table 3 shows that the 3-layer model provides the best trade-off between classification accuracy and computation time.

4. Crop/weed detection and identification

In this section, we evaluate the contribution of raw-based feature extraction to image segmentation. Specifically, we are interested in both problems of crop/weed detection and identification.

Table 3
1-NN classification accuracy (%) of HyTexiLa raw patches ($X = 64..200$) and computation time (s) achieved by MSFA-Net in the same case as in Fig. 3 (IMEC 5×5, $X = 65$), but with different numbers of layers.

Model	IMEC 4×4 200	IMEC 4×4 124	IMEC 4×4 64	IMEC 5×5 200	IMEC 5×5 125	IMEC 5×5 65	Comp. time
2-layer	94.7	98.4	98.3	98.4	97.8	93.7	13.8
3-layer	99.5	98.3	98.7	99.0	98.4	95.1	15.7
4-layer	99.5	98.5	99.1	99.0	98.8	95.9	31.4

4.1. Image database

We own a database of 96 multispectral images of crop (beet, wheat, and bean) and weed species (thistle, goosefoot, and datura) (see Fig. 4(a,b)) that have been acquired by the Chambre d'Agriculture (CA) de la Somme, France, in early April 2019 using IMEC Snapscan camera [1]. From the 141-channel radiance images acquired in outdoor conditions, reflectance is estimated in order to extract illumination-invariant spectral signatures [27]. To perform classification based on raw patches, we design a specific 5×5 basic MSFA pattern (CA 5×5) inspired by IMEC 5×5. We first select $B^2 = 25$ channels from the available 141 ones using a sequential forward selection (SFS) approach. The spectral bands associated to the selected channels are arranged in CA 5×5 according to their central wavelengths in the same band arrangement as IMEC 5×5. Note that after SFS, eight out of the 25 channels are associated to NIR spectral bands. We then simulate images that would be acquired by a snapshot camera equipped with CA 5×5 MSFA. Finally, we demosaic these images using PPID method to obtain fully-defined images for the sake of comparison.

4.2. Feature extraction and crop/weed classification

Our segmentation approach is a supervised pixel classification. First, vegetation is distinguished pixel-wise from background using the normalized difference vegetation index [28]. At each vegetation pixel, we consider a centered neighborhood as a patch whose size is a small multiple of $B = 5$ and in which at least 88% of the pixels represent

Table 4
Crop/weed identification accuracy (%) of LGBM classifier. Weed species: Th(istle), Go(osefoot), Da(tura). $\overline{\text{Acc.}}$ is the overall weighted accuracy score.

Feature	Beet	Th	Go	Da	$\overline{\text{Acc.}}$	Wheat	Th	Go	Da	$\overline{\text{Acc.}}$	Bean	Th	Go	Da	$\overline{\text{Acc.}}$
MSFA-Net	82.6	75.5	*70.2*	*82.5*	**77.4**	99.2	**58.1**	51.6	*85.7*	63.4	**87.7**	*53.0*	48.9	*79.8*	*62.4*
M-LBP	62.5	70.1	39.0	60.9	55.3	90.3	36.7	34.2	65.9	45.2	59.8	**53.4**	30.8	54.4	42.3
SegNet-Basic	83.2	69.2	**70.6**	*83.9*	*77.2*	**99.4**	*50.6*	52.9	**86.0**	63.2	81.1	47.7	47.6	**80.2**	59.9
BlobNet	*90.4*	**81.7**	29.0	82.9	66.2	*99.3*	50.0	*57.2*	83.6	*65.1*	*83.5*	49.2	*52.1*	79.3	**62.8**
cNet	**90.9**	*80.9*	25.4	**84.4**	65.6	**99.4**	46.8	**58.6**	81.9	**65.2**	80.7	45.8	**52.4**	79.4	62.1
$\mathbf{R}^{(25)}$	71.3	56.1	51.3	76.1	64.3	98.3	40.2	49.5	77.0	57.9	61.3	34.2	43.8	71.9	51.2
$\bar{\mathbf{R}}^{(25)}$	80.5	63.9	60.0	75.6	69.2	98.2	43.9	44.7	77.0	55.6	58.6	47.3	41.0	73.3	50.5

Number of test patches for beet/weed (P1), wheat/weed (P2), and bean/weed (P3) detection or identification problem:
P1: Beet: 694,201 | Th: 509,547 | Go: 259,130 | Da: 203,273
P2: Wheat: 584,599 | Th: 342,925 | Go: 79,169 | Da: 215,540
P3: Bean: 166,761 | Th: 381,717 | Go: 63,260 | Da:203,767

Table 5
Recall (Re) and precision (Pr) results (%) of weed detection by LGBM classifier. The best result in each column is shown as bold and second best as italics.

Feature	Beet vs. weed		Wheat vs. weed		Bean vs. weed	
	Re	Pr	Re	Pr	Re	Pr
MSFA-Net	**97.2**	89.7	**92.6**	*99.4*	**92.6**	**96.8**
M-LBP	90.3	83.0	73.1	93.5	77.4	93.8
SegNet-Basic	*96.3*	92.3	91.0	**99.5**	*90.2*	**96.8**
BlobNet	95.1	*93.1*	*91.2*	**99.5**	88.0	94.3
cNet	94.8	**93.2**	90.1	**99.5**	84.7	*94.7*
$\mathbf{R}^{(25)}$	92.8	86.1	88.0	98.7	76.2	91.7
$\bar{\mathbf{R}}^{(25)}$	92.9	90.2	90.8	98.6	81.1	90.6

vegetation (to avoid feature extraction on zeroed-out background pixels). For each case (beet vs. weeds, wheat vs. weeds, and bean vs. weeds), a specific number of learning and test patches are extracted. We extract patches of size 25×25 pixels, or 20×20 for the wheat vs. weeds case to have enough samples to characterize thin leaves of wheat. For crop/weed detection, we randomly extract $N_1 \approx 180 \cdot 10^3$ patches from learning images, half for crop and half for weed class. As we merge thistle, goosefoot, and datura patches to build a single weed class, we extract $(N_1/2)/3$ patches for each of them. For crop/weed identification, we extract $N_2 \approx 50 \cdot 10^3$ patches per class. The number of test patches for both problems are displayed below Table 4. We extract texture features from these patches, then the central pixel of each test patch is classified as crop or weed (detection problem), or as one of the four (one crop and three weeds) vegetation classes (identification problem).

In this experiment, we consider MSFA-Net, M-LBP, and SegNet-Basic encoder that is suited to segmentation and has shown better performances than S-CNN in texture classification (see Sec. 3.4). To make it possible to learn from very small patches, we omit the two maxpooling layers of MSFA-Net. For SegNet-Basic encoder, we omit the second and third maxpooling layers. We additionally consider two CNNs that follow a patch-based classification approach for crop/weed detection using RGB-NIR images. The model (here called BlobNet) proposed by Milioto et al. [29] is composed of three convolutional layers (two of 5×5 kernels and one of 3×3 kernels), followed by two 2×2 maxpooling layers, and two fully-connected layers that provide a 512-dimensional feature vector. The model called cNet [30] is composed of two convolutional layers (of 5×5 kernels), two 2×2 maxpooling layers, and two fully-connected layers that provide a 192-dimensional feature vector.

We also consider reflectance spectra as features. Each demosaiced learning or test patch (see Sec. 4.1) is represented by a B^2-dimensional reflectance vector $\mathbf{R}^{(B^2)}$, whose b-th component is the average reflectance value over a small square window (of 25 pixels) centered at the middle of the considered patch to reduce noise influence. Furthermore, to make reflectance signatures robust against shading and specular reflection, we normalize each spectrum as $\bar{\mathbf{R}}^{(B^2)}$ so that its energy sums up to 1. As $B^2 = 25$ here, $\mathbf{R}^{(25)}$ and $\bar{\mathbf{R}}^{(25)}$ represent the 25-dimensional unnormalized and normalized reflectance features extracted from demosaiced patches.

To get results that are comparable with a previous study on weed detection [27], we use the supervised gradient boosting classifier LightGBM [31]. To reduce sparse pixel misclassification after prediction, we assume that reflectance does (almost) not change across locally close surface elements of a scene. Plausibly, these elements belong to the same material, hence to the same class. Each prediction associated to a test vegetation pixel is then filtered using a majority voting rule, and its final class label is the most frequent one over its 9×9 neighborhood.

4.3. Segmentation results and discussion

Tables 4 and 5 show the crop/weed detection precision/recall and identification accuracies obtained by LGBM classifier, respectively. In both problems, M-LBP descriptor does not reach high performances with small patches because there are not enough pixels to efficiently capture the spatio-spectral band interactions. Reflectance features (that neither take these interactions into account) even perform better than M-LBP in most cases. Table 5 shows that deep features outperform handcrafted ones for crop/weed

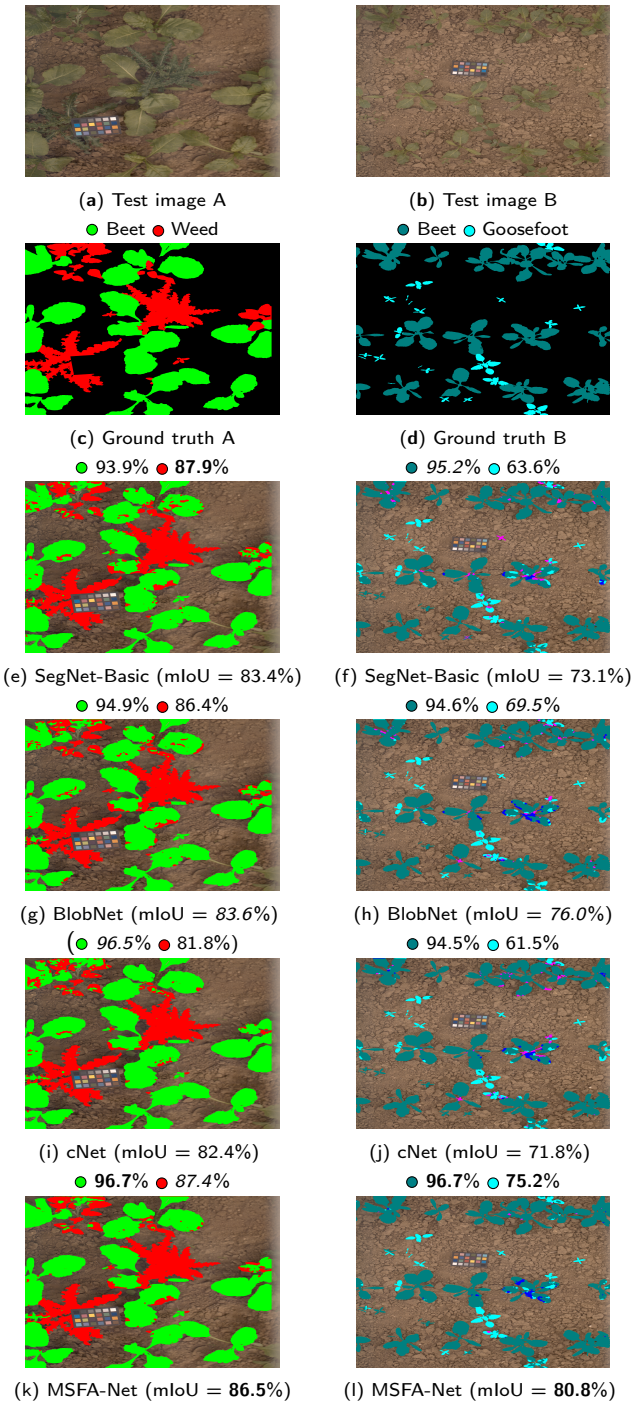

Figure 4: Segmentation results (mean intersection over union (mIoU) and per-class accuracy scores) obtained by SegNet-Basic, BlobNet, cNet, and MSFA-Net-based texture features. (a, b): RGB renderings of two multispectral test images, (c, d): ground truths, (e, g, i, and k): beet/weed detection results, (f, h, j, and l): beet/goosefoot identification results. Bold values show the best results between SegNet-Basic, BlobNet, cNet, and MSFA-Net. Magenta and blue colors in (f, h, j, and l) are beet or goosefoot pixels misclassified either as datura or thistle, respectively.

detection. Moreover, our MSFA-Net provides the best recall for the three detection problems and the best precision for bean/weed detection. It also provides a precision (99.4%) that is comparable to that of the other deep features (99.5%) for wheat/weed detection. Table 4 shows that for crop/weed identification, cNet is ranked first five times, SegNet-Basic four times, MSFA-Net twice, and BlobNet once among the 12 classes. Because test pixel classes are highly skewed, we also evaluate the overall classification performance using the accuracy score weighted by the per-class number of test patches. According to this criterion (see gray columns in Table 4), MSFA-Net outperforms the other descriptors in the beet/weed identification problem because it has more success in recognizing goosefoot class. It also provides comparable weighted accuracy scores with the best ones in the wheat/weed and bean/weed identification problems.

These experiments show that the performance of outdoor crop/weed recognition systems based on an analysis of spectral signatures [27] can be improved by deep texture features.

For illustration purposes, Fig. 4 displays the segmentation results obtained by the considered deep learning approaches on two of our images for the beet/weed detection and identification problems. It shows comparable weed detection performances between SegNet-Basic and MSFA-Net that outperforms cNet and BlobNet. It also shows that for beet/weed identification, MSFA-Net has more success in recognizing beet and goosefoot leaves.

5. Conclusion

This work presents an approach for multispectral texture feature extraction from raw patches thanks to a CNN architecture called MSFA-Net. Its first layer learns spatio-spectral interactions among channel values that match the basic MSFA pattern. This approach avoids the demosaicing step that can be greedy in computation requirements and may alter the texture representations. It requires learning much fewer hyperparameters than state-of-the-art CNNs. Extensive experiments on image classification and crop/weed segmentation show that MSFA-Net globally outperforms other tested approaches at much reduced computation costs.

However, deploying multispectral cameras in outdoor crop fields faces several challenges, such as illumination variations, shadows brought by leaves, and wind that makes plant leaves move. Future work will focus on designing another MSFA-Net whose discriminant power is robust against these perturbations.

Acknowledgments

We thank the Region Hauts-de-France, the Chambre d'Agriculture de la Somme, and SCV-IrDIVE.

References

[1] J. Pichette, W. Charle, A. Lambrechts, Fast and compact internal scanning CMOS-based hyperspectral camera: the Snapscan, in: Proceedings of the SPIE Electronic Imaging Annual Symposium: Photonic Instrumentation Engineering IV, Vol. 10110, San Francisco, CA,

USA, 2017, pp. 1–10. doi:10.1117/12.2253614.

[2] N. Genser, J. Seiler, A. Kaup, Camera array for multi-spectral imaging, IEEE Transactions on Image Processing 29 (2020) 9234–9249. doi:10.1109/TIP.2020.3024738.

[3] V. Rathi, P. Goyal, Generic multispectral demosaicking based on directional interpolation, IEEE Access 10 (2022) 64715–64728. doi:10.1109/ACCESS.2022.3182493.

[4] V. Kitanovski, J.-B. Thomas, J. Y. Hardeberg, Reflectance estimation from snapshot multispectral images captured under unknown illumination, in: Proceedings of the 29th Color and Imaging Conference, Society for Imaging Science and Technology, Online, 2021, pp. 264–269. doi:10.2352/issn.2169-2629.2021.29.264.

[5] W. Zhou, S. Gao, L. Zhang, X. Lou, Histogram of oriented gradients feature extraction from raw Bayer pattern images, IEEE Transactions on Circuits and Systems II: Express Briefs 67 (5) (2020) 946–950. doi:10.1109/TCSII.2020.2980557.

[6] S. Mihoubi, O. Losson, B. Mathon, L. Macaire, Spatio-spectral binary patterns based on multispectral filter arrays for texture classification, Journal of the Optical Society of America A 35 (9) (2018) 1532–1542. doi:10.1364/JOSAA.35.001532.

[7] A. Porebski, M. Alimoussa, N. Vandenbroucke, Comparison of color imaging vs. hyperspectral imaging for texture classification, Pattern Recognition Letters 161 (2022) 115–121. doi:https://doi.org/10.1016/j.patrec.2022.08.001.

[8] J. Brauers, T. Aach, A color filter array based multispectral camera, in: 12. Workshop Farbbildverarbeitung, Illmenau, Germany, 2006, pp. 55–64.

[9] S. Mihoubi, O. Losson, B. Mathon, L. Macaire, Multispectral demosaicing using pseudo-panchromatic image, IEEE Transactions on Computational Imaging 3 (4) (2017) 982–995. doi:10.1109/TCI.2017.2691553.

[10] M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, Computer Vision Using Local Binary Patterns, Vol. 40, Springer London, 2011. doi:10.1007/978-0-85729-748-8_14.

[11] C. Cusano, P. Napoletano, R. Schettini, Local angular patterns for color texture classification, in: Proceedings of the 18th International Conference on Image Analysis and Processing (ICIAP 2015) Workshops, Vol. 9281 of Lecture Notes in Computer Science, Genoa, Italy, 2015, pp. 111–118. doi:10.1007/978-3-319-23222-5_14.

[12] R. J. Chu, N. Richard, H. Chatoux, C. Fernandez-Maloigne, J. Y. Hardeberg, Hyperspectral texture metrology based on joint probability of spectral and spatial distribution, IEEE Transactions on Image Processing 30 (2021) 4341–4356. doi:10.1109/TIP.2021.3071557.

[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 2015.
URL http://arxiv.org/abs/1409.1556

[14] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (12) (2017) 2481–2495. doi:10.1109/TPAMI.2016.2644615.

[15] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. I. Nieto, R. Siegwart, weedNet: Dense semantic weed classification using multispectral images and MAV for smart farming, IEEE Robotics and Automation Letters 3 (1) (2018) 588–595. doi:10.1109/LRA.2017.2774979.
URL https://github.com/inkyusa/weedNet

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A deep convolutional activation feature for generic visual recognition, in: Proceedings of the 31st International Conference on Machine Learning, Vol. 32, Beijing, China, 2014, pp. 647–655.
URL https://proceedings.mlr.press/v32/donahue14.html

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, 2016, pp. 2921–2929. doi:10.1109/CVPR.2016.319.

[19] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 2014, pp. 512–519. doi:10.1109/CVPRW.2014.131.

[20] V. Sharma, A. Diba, T. Tuytelaars, L. V. Gool, Hyperspectral CNN for image classification & band selection, with application to face recognition, Technical report KUL/ESAT/PSI/1604, KU Leuven, ESAT, Leuven, Belgium.

[21] H. Tang, Y. Li, X. Han, Q. Huang, , W. Xie, A spatial–spectral prototypical network for hyperspectral remote sensing image, IEEE Geoscience and Remote Sensing Letters 17 (1) (2019) 167–171. doi:10.1109/LGRS.2019.2916083.

[22] T. W. Sawyer, M. Taylor-Williams, R. Tao, R. Xia, C. Williams, S. E. Bohndiek, Opti-MSFA: a toolbox for generalized design and optimization of multispectral filter arrays, Optics Express 30 (5) (2022) 7591–7611. doi:10.1364/OE.446767.

[23] B. Geelen, N. Tack, A. Lambrechts, A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic, in: Proceedings of the SPIE: Advanced Fabrication Technologies for Micro/Nano Optics and Photonics VII, Vol. 8974, San Francisco, CA, USA, 2014, pp. 89740L–89740L–8. doi:10.1117/12.2037607.

[24] H. A. Khan, S. Mihoubi, B. Mathon, J.-B. Thomas, J. Y. Hardeberg, HyTexiLa: High resolution visible and near infrared hyperspectral texture images, Sensors 18 (7) (2018) 2045. doi:10.3390/s18072045.

[25] C. Cusano, P. Napoletano, R. Schettini, Combining local binary patterns and local color contrast for texture classification under varying illumination, Journal of the Optical Society of America A 31 (7) (2014) 1453–1461. doi:10.1364/JOSAA.31.001453.

[26] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1026–1034. doi:10.1109/ICCV.2015.123.

[27] A. Amziane, O. Losson, B. Mathon, A. Dumenil, L. Macaire, Reflectance estimation from multispectral linescan acquisitions under varying illumination–Application to outdoor weed identification, Sensors 21 (11) (2021) 3601. doi:10.3390/s21113601.

[28] P. S. Thenkabail, R. B. Smith, E. De Pauw, Evaluation of narrowband and broadband vegetation indices for determining optimal hyperspectral wavebands for agricultural crop characterization, Photogrammetric Engineering & Remote Sensing 68 (6) (2002) 607–621.

[29] A. Milioto, P. Lottes, C. Stachniss, Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs, in: Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 2229–2235. doi:10.1109/ICRA.2018.8460962.

[30] C. Potena, D. Nardi, A. Pretto, Fast and accurate crop and weed identification with summarized train sets for precision agriculture, in: Proceedings of the 14th International Conference on Intelligent Autonomous Systems, Vol. 531, Shangai, China, 2016, pp. 105–121. doi:10.1007/978-3-319-48036-7_9.

[31] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017), Vol. 30, Long Beach, CA, USA, 2017, pp. 3146–3154.