Confidence-based Local Feature Selection for Material Classification

Xu Sixiang, Muselet Damien, Trémeau Alain
Univ Lyon, UJM-Saint-Etienne, CNRS,
Institut Optique Graduate School
Laboratoire Hubert Curien UMR 5516
F-42023, SAINT-ETIENNE, France
sixiang.xu,damien.muselet, alain.tremeau@univ-st-etienne.fr

Laganière Robert
School of Electrical Engineering
and Computer Science
University of Ottawa
Ottawa, Canada
laganier@eecs.uottawa.ca

***Abstract*—With the rise of Convolutional Neural Network (CNN) in the recent years, image classification has shown outstanding performances in the computer vision field. Many well-known state of the art's CNN architectures such as the ResNet family are applying a Global Average Pooling (GAP) to reduce the number of parameters of the fully connected layers. Most of the time, this pooling operation helps to prevent overfitting but we claim that it has a serious weakness for specific images where small details are crucial to predict their category, such as material images. In this case, the details are lost in the global average, providing non accurate global features. In this paper, we propose to select the most important local features before applying the GAP. In this aim, we add a branch in the classification network that predicts the confidence the network should have in each local feature vector. The less confident features are filtered out before applying the GAP. Experimental results on three datasets show that our approach outperforms recent alternatives in terms of classification accuracy and output probability calibration.**

***Index Terms*—Network confidence, Global Average Pooling, Probability calibration, Material Classification.**

I. Introduction

Image classification consists in predicting a single class for each input image. Today, the most successful approaches rely on automatic extraction of local features with deep neural networks followed by a Global Average Pooling (GAP) layer that merges all the local features into a single global feature vector [1]. Then, a fully connected layer predicts the image class from this global feature vector. With this classical approach, each local feature vector equally contributes to the final decision through the averaging operation. Consequently, when large areas of the images are ambiguous or when useful information is mainly provided by fine image details, averaging all the local features could lead to bad predictions.

This phenomenon is exemplified here in the context of material or texture classification. Indeed, as illustrated in Fig. 1, large parts of an image can be ambiguous when it comes to identify the material of the pictured object which can lead to bad predictions. Contrastively, some other areas are very informative and should be emphasized. On the left column of Fig. 1 some small parts of the images are masked making the class prediction very difficult. When one has access to these details (right column), class prediction becomes much easier.

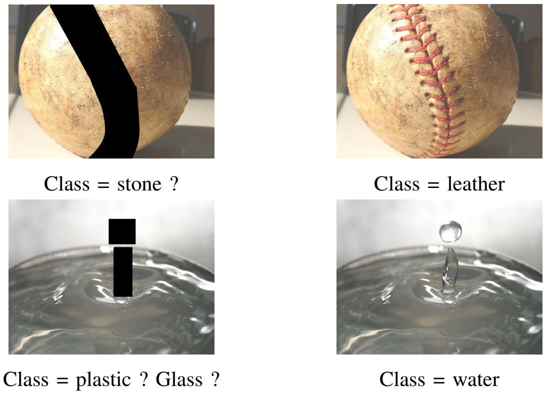
Class = stone ?
Class = leather
Class = plastic ? Glass ?
Class = water

Fig. 1. Images from the Flicker Material Dataset [2], showing that, sometimes, some details are essential to predict the correct class while large areas are ambiguous.

In this paper, we propose a method to automatically select the most informative local feature vectors before applying the GAP layer. The objective is to ensure that the most relevant features contribute to the final decision while the less informative ones are ignored. We hypothesize that the usefulness of each local feature vector is related to the confidence of the network when predicting the image class from this feature vector. Thus, we train a two-branch network to output local predictions as well as associated confidences. These predicted confidences are used to filter out the local feature vectors having lower confidence predictions before averaging all local features into a global feature vector (see Fig. 2).

Our contributions are multiple:

- we address the problem of Global Average Pooling in the context of material classification by weighting local features;
- we adapt a very recent and successful approach, designed for global failure prediction [3], to local feature confi-

This work was supported by the French government administered by the French National Research Agency (ANR) under the "Investissement d'Avenir" (ANR-11-IDEX-0007) program with the reference LABEX MANUTECH-SISE (ANR-10-LABX-0075) of Université de Lyon and the reference EUR MANUTECH SLEIGHT (ANR-17-EURE-0026) and by the Région Auvergne-Rhône-Alpes.

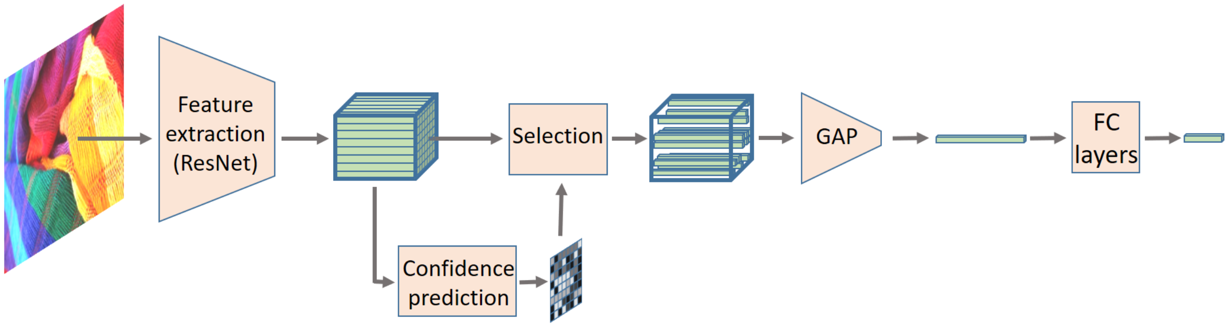

Fig. 2. The workflow of the proposed approach. See text for details.

dence prediction;

- we improve the calibration of the output probabilities for material classification;
- we provide both quantitative and qualitative results on three datasets.

II. Related works

This section reviews the works related to material classification, Global Weighted Average Pooling, confidence prediction and probability calibration.

A. Material classification

In the early of 2000's, material and texture classification tasks have benefited from the success of the global descriptors such as Bags-of-Words [4] and Fisher Vectors [4] that are based on an orderless aggregation of local features. With the recent advances of deep neural networks in image classification, new solutions appeared to automatically extract bags of deep features from images. They are obtained by computing the Fisher vectors of deep local features [5], by using RBF neurons in an end-to-end trainable solution [6] or by designing residual encoders such as VLAD and Fisher Vectors in an end-to-end learning framework [7], [8]. These orderless aggregation of local features can also be the input of a second multi-layer neural network [9], or combine with global features [10]. A recent study has also proposed to aggregate the class activation of each local patch in the image in a global feature vector [11]. Finally, classical deep networks have also been used for material classification with good results by using transfer learning [12].

B. Global Weighted Average Pooling

In order to cope with the drawback of the Global Average Pooling (GAP), Qiu [13] has proposed to weight the contribution of each local feature vector and to compute a Global Weighted Average Pooling (GWAP). The main problem of this solution is that it increases the number of trainable parameters without adding any supervision, increasing the risk of overfitting. Indeed, the weights in [13] are learned by back-propagating the gradient of the classification loss. On the contrary, our solution consists in supervising the weight learning with a confidence map, as detailed in the next section. We show in the experimental section, that our approach outperforms the GWAP proposed by Qiu.

C. Confidence prediction and probability calibration

An intuitive way to assess the confidence of a network prediction would be to check the output probability distribution over the classes. Unfortunately, it has been shown that the softmax probabilities of deep neural networks are not well-calibrated and modern models are clearly overconfident [14]. This is checked in the experimental section, where we show that the classical maximum probability or entropy do not allow to select the important features.

Guo et al. have proposed to calibrate the output probabilities of deep neural networks by modifying the classical softmax function with a temperature scaling [14]. Increasing the temperature parameter softens the softmax, leading to lower confident prediction, without modifying the model accuracy. This approach is also tested in our experimental section.

Gal and Ghahramani approximate a Bayesian neural network by using Drop-Out at inference time in order to get a distribution over the outputs [15]. The uncertainty of the network is deduced from the variations of the probabilities for the same sample with different Drop-Out. This approach is interesting but requires to feed several times the network with the same sample at test time. We also provide results for this approach on our data in the last section.

De Vries and Taylor propose a smart approach to predict the confidence of a neural network [16]. Their idea consists in letting the network partially access to ground truth information during training. The level of knowledge it is asking for, is related to its uncertainty. Through this approach looks very simple, it is not easy to optimize in practice as mentioned by the authors.

Very recent approaches have proposed to train networks to predict the confidence along with the class prediction [3], [17]. The method that gave us inspiration for our problem is the one proposed by Corbiere et al [3]. In this work, the ground truth confidence is defined as the True Class Probability (TCP), i.e. the probability returned by the network for the ground truth class (which can be different from the predicted class) of the given input image. This is detailed in the next section. This concept is very similar to the work of Yoo and Kweon [17] who train a branch of their network to predict the loss of each prediction. This value is clearly related to the confidence but the weakness of this loss prediction is that it is not normalized, compared to the TCP that is in the range $[0, 1]$.

It is worth mentioning that these previous works about confidence prediction have been proposed in other contexts than ours, namely failure prediction [3] and active learning [17]. Furthermore, these approaches were applied to whole images while the aim of our work is to predict local confidences in order to select the best local features inside one image.

III. Our approach

A. Deep neural network with Global Average Pooling

Let denote a training sample as $(\mathbf{I}, \mathrm{y})$ where $\mathbf{I} \in \mathbb{R}^{W\times H\times 3}$ is an RGB image and $\mathrm{y} \in \mathcal{Y} = \{1, ..., K\}$ is its ground truth category. Recent deep networks such as the ResNet series can be decomposed into three parts: the feature extractor f_{conv} constituted by convolutional layers, the Global Average Pooling (GAP) f_{avg} that discards any spatial information and a fully-connected (FC) layer f_{FC} followed by a Softmax function returning the predicted distribution $\hat{\mathbf{p}}$ of the probabilities over all the classes:

$$\hat{\mathbf{p}}(\mathbf{I}) = Softmax(f_{FC}(f_{avg}(f_{conv}(\mathbf{I})))). \tag{1}$$

Note that $\hat{\mathbf{p}}(\mathbf{I})$ is a K-dimensional vector $\hat{\mathbf{p}} = [\hat{p_1}, \hat{p_2}, ..., \hat{p_K}]$.

While training the network, the parameters are updated in order to minimize the cross-entropy loss (over a batch of images) $\mathrm{L}_{ce}(\mathbf{p}, \hat{\mathbf{p}})$ between the ground truth hot-vector $\mathbf{p}$ (deduced from y) and the predicted $\hat{\mathbf{p}}$.

Since the FC layer and the GAP layer are linear transforms, they can be switched in the process so that the FC layers are applied before the GAP. The predicted probabilities are then:

$$\hat{\mathbf{p}}(\mathbf{I}) = Softmax(f_{avg}(f_{FC}(f_{conv}(\mathbf{I})))). \tag{2}$$

Obviously, in this case, the fully-connected layer is applied individually to each local feature vector returned by f_{conv}, in the form of 1x1 convolutions, as shown in the left workflow of figure 4. This formulation is interesting for our approach since it represents individual processing of each feature vector.

This architecture has shown very good results in many classification applications, but it might not be the optimal solution for material image classification. Indeed, as discussed in the introduction and confirmed in many successful orderless aggregation solutions proposed for this task; large areas of material images can be ambiguous about the class of the considered image, while some details appear to be very discriminative. A simple average of all the local features into a global vector can lead to useful information lost.

This is illustrated with the two images from Fig. 3, where we propose to have a look at the map prediction provided by the network without applying the GAP:

$$\widehat{\mathbf{p_{map}}}(\mathbf{I}) = Softmax(f_{FC}(f_{conv}(\mathbf{I}))), \tag{3}$$

where each local feature vector $\mathbf{v_i}$ is associated with one local prediction at the i^{th} location in the map:

$$\hat{\mathbf{p}}(\mathbf{v_i}) = \widehat{\mathbf{p_{map_i}}}(\mathbf{I}). \tag{4}$$

The first column of this figure shows two images with their ground truth category. The second column shows, for each local feature vector $\mathbf{v_i}$, the category $\hat{y}(\mathbf{v_i})$ that locally gets the maximum score as well as its score $\hat{p_y}(\mathbf{v_i})$:

$$\hat{y}(\mathbf{v_i}) = \underset{k\in\mathcal{Y}}{\mathrm{argmax}}\, \hat{p_k}(\mathbf{v_i}), \tag{5}$$

$$\hat{p_y}(\mathbf{v_i}) = \max_{k\in\mathcal{Y}} \hat{p_k}(\mathbf{v_i}). \tag{6}$$

The color legend for the category is shown at the bottom of the figure and the lightness of each color is related to its score $\hat{p_y}(\mathbf{v_i})$, i.e. dark colors mean that the associated probability is low whereas lighter colors represent high probabilities.

Below each illustration, we mention the three most probable categories provided by the whole network, including the GAP. In this classical case, each image gets a single global probability vector and these three mentioned categories are the ones that get the highest probabilities. We can see that, for both examples, the most probable category is not the ground truth one, leading to a bad classification for these images. We can also notice that most of the local predictions are associated with very light colors, showing that the network is overconfident in most of the cases, even for non-correct predictions.

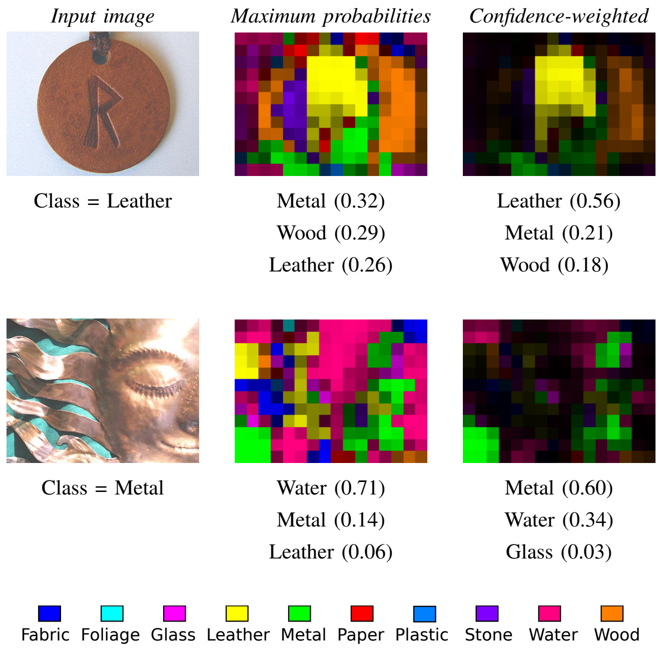

Fig. 3. Local decision maps for two different images. The two right columns show the categories and scores of the locally maximum probabilities before (second column) and after (third column) weighting them with the corresponding local confidences.

As illustrated in the last column of Fig. 3, our aim is to select the most important local feature vectors, and remove

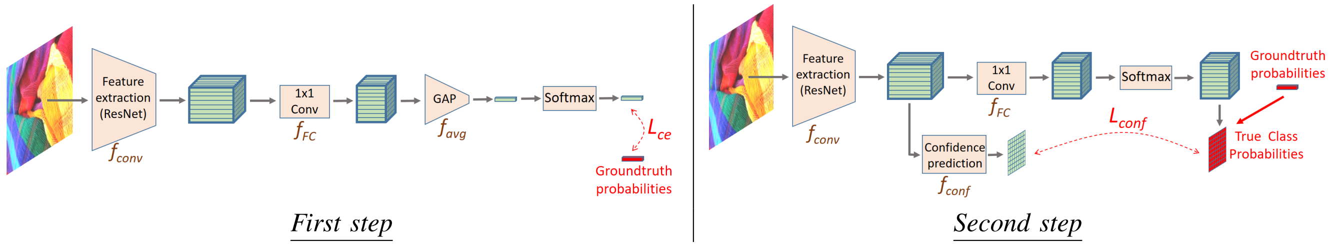

Fig. 4. The two successive training steps. See text for details.

the least ones, before applying the GAP in the network. We propose to relate the ”importance” of each local feature vector to its associated class prediction confidence.

B. Predicting local confidences

In order to select the most important local feature vectors, we propose to train a branch of our network to predict the confidence of the category prediction related to each local feature vector. In this aim, we take inspiration from [3] that deals with failure prediction in image classification task. In this paper, the authors tried to find out images potentially misclassified by estimating the True Class Probability (TCP) along with the category prediction. We propose to adapt this approach in order to predict local TCP that help us to select the most confident local feature vectors.

As defined in [3], the TCP is the predicted probability of the ground truth category y of the considered image:

$$TCP(\mathbf{v_i} \in \mathbf{I}) = \widehat{p}_{k=y}(\mathbf{I}) \tag{7}$$

Given a local feature vector, a high TCP means that this vector leads to a prediction that gives a high probability to the correct class, which means that we should trust it. On contrary, if the TCP of a local feature vector is low, it means that it predicts a low probability to the correct class and, so, should not be considered in the final global decision.

Obviously, at test time, the ground truth category is not available and therefore neither is the TCP. Thus, we propose to add a branch f_{conf} in our network whose aim is to predict the TCP of each local feature vector. As illustrated on the right of Fig. 4, the input of this branch is the feature map extracted from the image and its output is a predicted TCP map:

$$\widehat{\mathbf{TCP}}_{map}(\mathbf{I}) = f_{conf}(f_{conv}(\mathbf{I})) \tag{8}$$

The idea of this new branch is that the network is learning if some local features are rather ambiguous or not with respect to the category they predict. The details about the structure of f_{conf} are available in the next section.

Thus, if the network is able to automatically predict the TCP of each local feature vector, we can use these predictions as the confidence we should have in each vector and select the most confident ones before applying their average (see Fig. 2).

In order to illustrate the intuition of our idea, we show in the last column of Fig. 3 how the local probabilities are transformed when they are weighted by their corresponding predicted confidence (TCP). It is worth mentioning that this weighting scheme is just presented for illustration. In practice, the confidences are used to select the most confident local feature vectors, with a threshold, as detailed below.

TABLE I
THE TWO STEPS OF THE PROPOSED LEARNING SCHEME.

Step	Loss	Frozen parameters	Learned parameters
Step 1	L_{ce}	f_{conf}	$f_{conv} + f_{FC}$
Step 2	L_{conf}	$f_{conv} + f_{FC}$	f_{conf}

C. The training process

The whole training process is composed of two steps as shown in Fig 4. During the first step, the classification network is trained with the cross-entropy loss L_{ce}. After reaching convergence, the parameters of the trained network are frozen and the confidence prediction branch is trained. To this end, we feed the classification network with images and their ground truth category in order to evaluate their ground-truth TCP map $\mathbf{TCP}_{map}$ (confidence map). Then, we train the confidence prediction branch f_{conf} so that it is able to automatically predict the TCP map for each image by minimizing L_{conf}, the mean square error between the ground truth map $\mathbf{TCP}_{map}$ and the predicted one $\widehat{\mathbf{TCP}}_{map}$. A summary of the two training steps is provided in Table I.

IV. Experiments

In this section, we present the experimental results provided by our approach in a material classification task. The tests are conducted over three datasets and the results are compared with recent alternatives.

A. The datasets

Three classical material datasets are used for testing. The Flicker Material Dataset (FMD) [2] is a popular benchmark material dataset which contains 10 categories and 100 images per category (see Fig. 1 and 3 for image examples and class names). KTH-TIPS-2b [18] (called hereafter KTH) has 11 categories with 432 images for each category and the 4D-light dataset [19] is a light-field material dataset which consists of 12 categories with 100 images per category.

For FMD and 4D-Light, we run a 5-fold experiment by splitting the dataset into 5 non-overlapping subsets. For each run, 4 subsets are used for training and 1 for testing. For KTH, following the experiments from [8], we randomly choose half

TABLE II
THE RESULTS OF THE TESTED APPROACHES ON THE THREE DATASETS. AVERAGES OVER 5 RUNS.

	FMD			KTH			4D-Light		
Approaches	ECE	NLL	Accuracy (%)	ECE	NLL	Accuracy (%)	ECE	NLL	Accuracy (%)
Baseline	0.080	0.517	83.2	0.060	0.54	82.1	0.074	0.537	83.1
Baseline (90% Training)	0.087	0.543	83.1	0.064	0.55	81.9	0.061	0.535	83.0
Temperature (90% Train./10% Val.)	0.071	0.529	83.1	0.120	1.20	81.9	**0.049**	0.532	83.0
Entropy	0.070	0.510	83.1	0.060	0.54	82.1	0.073	0.537	83.1
MaxProb	0.079	0.517	83.2	0.060	0.54	82.1	0.074	0.537	83.1
MCDropout	0.081	0.516	83.0	0.060	0.54	82.2	0.073	0.537	83.0
GWAP	0.067	0.525	83.3	0.063	0.55	81.7	0.063	0.529	84.0
Confidence prediction (Our)	**0.061**	**0.470**	**84.8**	**0.058**	**0.52**	**83.1**	0.058	**0.527**	**84.8**

of the images for training (216 per category) and half for testing. The results are also averaged over 5 runs.

B. Tested approaches

Our method is compared with several recent and classical approaches. The **baseline** is a classical network without weighting scheme as illustrated in the left of Fig. 4 (first step).

Since output probability calibration is one aim of our framework, we propose to compare our results with the **temperature scaling** solution [14]. As recommended by the authors, this approach required a validation set to fix the temperature. Thus, for this method, about 10% of the images for each category are randomly extracted from the training set to constitute the validation set. The test set is the same for all the approaches for fair comparison. The baseline is also tested on this reduced training set (mentioned as ”90% Training” in the Tables) for information.

The entropy of the predicted class probabilities could be seen as a confidence score and is used in some recent papers [20], [21]. Indeed, a peaky prediction vector (low entropy) means that the network is confident in its prediction while a flat probability vector (high entropy) shows that the network is hesitating between the different classes. In our experiment, we propose to compare our method with a local selection based on the entropy of each local classification prediction. In the experiment, we have chosen the threshold that performs best for this approach. We have also tested the maximum probability (**MaxProb**) as a confidence measure.

The Monte-Carlo Dropout approach [15] is denoted **MCDropout** in the Tables.

The Global Weighted Average Pooling (**GWAP**) is similar to our approach, except that it predicts a score map without any additive supervision than the classification loss [13]. In this case, the architecture is similar to our proposed solution with two branches that are simultaneously trained with a single cross-entropy loss L_{ce}.

Finally, we are also presenting the results of state-of-the-art approaches on the respective used material datasets. Even if the architectures are different, the results inform us about the best current results provided on these datasets.

C. Experimental settings

For all the tested models, the network backbone is ResNet-50 [1] pretrained on the ImageNet dataset [22].

Our confidence prediction block f_{conf} is composed of 3 successive 3x3 convolutional layers with respectively 384 kernels with ReLu, 192 kernels with ReLu and 1 kernel with a Sigmoid. The input of this block is the concatenation of the feature maps from the two last convolutional blocks of the backbone.

As previously explained, the aim of our solution is to filter out the least confident local feature vectors before applying the GAP. One threshold has to be fixed in order to decide which vectors should be discarded. For all our experiments, we have chosen to remove the feature vectors whose associated predicted confidence is lower than 0.2. This threshold is fixed for all the runs and all the datasets.

As recommended, for all the approaches, channel-wise normalization is applied (zero mean and unit variance) as a pre-processing. For data augmentation, all images are resized to 384x384. 8% to 100% of the area of each image is randomly cropped, transformed with a random aspect ratio between $\frac{3}{4}$ and $\frac{4}{3}$ of the original aspect ratio, and resized to 352x352. Additionally, a 50% chance horizontal and vertical flip is applied. At test time, we just use the images with their original sizes.

We use Adagrad as optimization algorithm with a mini-batch size of 8. The learning rate starts from 0.01 at step 1 and from 0.001 at step 2 and is divided by 10 each time the training loss meets a plateau.

D. Results

The first results are presented in Table II, where three criteria are provided: the classification accuracy, the Expected Calibration Error (ECE) [23] and the Negative Log Likelihood (NLL) [14]. ECE and NLL are both measuring the degree of miscalibration of the output probabilities. They are low for well calibrated probabilities.

In this Table, we can see that the temperature scaling overall actually improves the output calibration over the baseline with the same settings, while preserving the accuracy. Indeed, the single aim of this approach is to calibrate the output probabilities of the network without modifying the classification accuracy of the baseline, since the probability ranking is not modified by this scaling. Nevertheless, for the KTH dataset, we can see that the scaling does not improve the calibration. We think that it is due to the high diversity within each category of this dataset, that makes difficult to estimate the temperature

TABLE III
COMPARISON OF THE CLASSIFICATION ACCURACY (%) WITH THE STATE-OF-THE-ART SOLUTIONS ON THE THREE DATASETS.

Approaches	FMD	KTH	4D-Light
LFV+FC-CNN [9]	83.5	**83.1**	-
Deep Ten [8]	80.2	82.0	84.1
FV-CNN [5]	82.4	81.1	82.6
B-CNN [24]	80.5	80.2	84.3
Confidence prediction (Our)	**84.8**	**83.1**	**84.8**

scaling on the validation set. Interestingly, the entropy-based approach also reduces the calibration error on FMD but does not improve the accuracy over the baseline. Overall, entropy- and maximum probability-based approaches have very small impacts on the results. The GWAP approach provides mixed results for the calibration quality and slightly improves the accuracy. We can notice that our approach clearly outperforms all the tested methods for the three criteria. Indeed, by discarding the least confident local feature vectors, our model is able to predict calibrated and accurate probabilities. It is worth mentioning that the architectures of our solution and GWAP are identical. This clearly shows that supervising the second branch with the True Class Probabilities is a good solution to predict accurate confidences and select the best local features.

Finally, we propose to compare the accuracy provided by our method with state-of-the-art solutions designed for material classification (see Table III). The reported results have been extracted from the published papers, when available. Despite the simplicity of our approach, we notice that it outperforms all the recent state-of-the-art solutions designed for material classification. These results confirm that it is very interesting to concentrate the category decision on specific areas of material images and that predicting the confidence of each local feature vector is a smart way to do that.

V. CONCLUSIONS

In this paper, we have proposed an original solution for material classification. Since material images present large ambiguous areas that do not help or even disturb the classification process, our idea consists in removing these parts from the feature maps before taking the average final decision. To this end, we propose to add a branch in the classical network in order to predict the confidence associated with each local feature vector. This branch is trained to predict the True Class Probability (TCP) during the learning step. This TCP can be seen as a confidence and allows us to filter out ambiguous or disturbing local feature vectors before applying the Global Average Pooling. Experimental results on three datasets show that our solution outperforms the alternatives and the classical models for both the accuracy of the network and the output probability calibration. In order to select the most confident feature vectors, a fixed threshold has been used in this paper. Future works will consist to train the network to predict this value.

REFERENCES

[1] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)

[2] Sharan, L., Rosenholtz, R., Adelson, E.H.: Accuracy and speed of material categorization in real-world images. Journal of Vision **14** (2014)

[3] Corbiere, C., Thome, N., Bar-Hen, A., Cord, M., Perez, P.: Addressing failure prediction by learning model confidence. In: Conference on Neural Information Processing Systems 2019 (NIPS 2019). (2019)

[4] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: In Workshop on Statistical Learning in Computer Vision, ECCV. (2004) 1–22

[5] Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). (2015) 3828–3836

[6] Passalis, N., Tefas, A.: Learning bag-of-features pooling for deep convolutional neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). (2017) 5766–5774

[7] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 5766–5774

[8] Zhang, H., Xue, J., Dana, K.: Deep ten: Texture encoding network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 708–717

[9] Song, Y., Zhang, F., Li, Q., Huang, H., O'Donnell, L.J., Cai, W.: Locally-transferred fisher vectors for texture classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 4912–4920

[10] Xue, J., Zhang, H., Dana, K.: Deep texture manifold for ground terrain recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 558–567

[11] Brendel, W., Bethge, M.: Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. arXiv preprint arXiv:1904.00760 (2019)

[12] Wieschollek, P., Lensch, H.: Transfer learning for material classification using convolutional networks. arXiv preprint arXiv:1609.06188 (2016)

[13] Qiu, S.: Global weighted average pooling bridges pixel-level localization and image-level classification. CoRR **abs/1809.08264** (2018)

[14] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org (2017) 1321–1330

[15] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: The International Conference on Machine Learning (ICML). (2016) 1050–1059

[16] DeVries, T., Taylor, G.W.: Learning confidence for out-of-distribution detection in neural networks. arXiv preprint arXiv:1802.04865 (2018)

[17] Yoo, D., Kweon, I.S.: Learning loss for active learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 93–102

[18] Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. Volume 2., IEEE (2005) 1597–1604

[19] Wang, T.C., Zhu, J.Y., Hiroaki, E., Chandraker, M., Efros, A.A., Ramamoorthi, R.: A 4d light-field dataset and cnn architectures for material recognition. In: European Conference on Computer Vision, Springer (2016) 121–138

[20] Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. arXiv preprint arXiv:1703.02910 (2017)

[21] Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 93–102

[22] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009)

[23] Pakdaman, N.M., F., C.G., Hauskrecht, M.: Obtaining well calibrated probabili-ties using bayesian binning. In: Proc. Conf. AAAI Artificial Intelligence. (2015) 2901–2907

[24] Lin, T.Y., Maji, S.: Visualizing and understanding deep texture representations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2791–2799