Multi-layer Feature Fusion and Selection from Convolutional Neural Networks for Texture Classification

Hajer Fradi¹, Anis Fradi²,³ and Jean-Luc Dugelay¹

¹Digital Security Department, EURECOM, Sophia Antipolis, France
²University of Clermont Auvergne, CNRS LIMOS, Clermont-Ferrand, France
³University of Monastir, Faculty of Sciences of Monastir, Tunisia

Keywords: Texture Classification, Feature Extraction, Feature Selection, Discriminative Features, Convolutional Layers.

Abstract: Deep feature representation in Convolutional Neural Networks (CNN) can act as a set of feature extractors. However, since CNN architectures embed different representations at different abstraction levels, it is not trivial to choose the most relevant layers for a given classification task. For instance, for texture classification, low-level patterns and fine details from intermediate layers could be more relevant than high-level semantic information from top layers (commonly used for generic classification). In this paper, we address this problem by aggregating CNN activations from different convolutional layers and encoding them into a single feature vector after applying a pooling operation. The proposed approach also involves a feature selection step. This process is favorable for the classification accuracy since the influence of irrelevant features is minimized and the final dimension is reduced. The extracted and selected features from multiple layers can be further manageable by a classifier. The proposed approach is evaluated on three challenging datasets, and the results demonstrate the effectiveness of selecting and fusing multi-layer features for texture classification problem. Furthermore, by means of comparisons to other existing methods, we demonstrate that the proposed approach outperforms the state-of-the-art methods with a significant margin.

1 INTRODUCTION

Deep networks have promoted the research in many computer vision applications, specifically a great success has recently been achieved in the field of image classification using deep learning models. In this context, extensive study has been conducted to address the problem of large-scale and generic classification that spans a large number of classes and images (such as the case of ImageNet Large Scale Visual Recognition Challenge) (Qu et al., 2016). Particularly, the problem of texture classification has been a long-standing research topic due to both its significant role in understanding the texture recognition process and its importance in a wide range of applications including medical imaging, document analysis, object recognition, fingerprint recognition, and material classification (Liu et al., 2019).

Even though texture classification is similar to other classification problems, it presents some distinct challenges since it has to deal with potential intra-class variations such as noise, scale, illumination, rotation, and translation. Due to these variations, the texture of the same objects appears visually different. Likewise, for different texture patterns, the difference might be subtle and fine. All these factors besides the large number of texture classes make the problem of texture classification challenging (Bu et al., 2019; Almakady et al., 2020; Alkhatib and Hafiane, 2019).

One of the key aspects of texture analysis is the features extraction step which aims at building a powerful texture representation that is useful for the classification task. Texture, by definition, is a visual cue that provides useful information to recognize regions and objects of interest. It refers to the appearance, the structure and the arrangement or the spatial organization of a set of basic elements or primitives within the image. Since the extraction of powerful features to encode the underlying texture structure is of great interest to the success of the classification task, many research works in this field focus on the texture representation (Lin and Maji, 2015; Liu et al., 2019).

This topic has been extensively studied through different texture features such as filter bank texton (Varma and Zisserman, 2005), Local Binary Pattern (LBP) (Wang et al., 2017; Alkhatib and Hafiane,
First, a novel approach for extracting and fusing features from multiple layers into a joint feature representation after applying a pooling operation is proposed. This feature representation is adopted to incorporate the full capability of the network in the feature learning process instead of truncating the CNN pipeline at one specific layer (usually chosen empirically). Second, since the fused features do not perform equally well and some of them are more relevant than others, the proposed approach incorporates a feature selection step in order to enhance the discriminative power of the feature representation. By alleviating the effect of high dimensional feature vector and by discarding the impact of irrelevant features, the overall classification rate is significantly improved using three challenging datasets for texture classification. Third, by means of comparisons to existing methods, the effectiveness of our proposed approach is proven. The obtained results outperform the current state-of-the-art methods, which proves the generalization ability and the representative capacity of the learned and selected features from multiple layers.

The remainder of the paper is organized as follows: Related work to texture classification using deep learning models is presented in Section 2. In Section 3, our proposed approach based on feature fusion and selection from multiple convolutional layers is presented. The proposed approach is evaluated using three challenging datasets, commonly used for texture classification and experimental results are analyzed in Section 4. Finally, we conclude and present some potential future works in Section 5.

2 RELATED WORK

Deep learning models have recently shown good performances in different applications, essentially for image classification by means of CNNs. Particularly, different architectures have been proposed for texture classification (Liu et al., 2019). The existing methods based on FC layers mainly include FC-CNN (Cimpoi et al., 2014), which is the top output from the last FC layer of VGG-16 architecture. Given the limitations of FC features (discussed in the introduction), major attempts in this field instead made use of convolutional features. For instance, in (Cimpoi et al., 2015; Cimpoi et al., 2016), the authors introduced FV-CNN descriptor which is obtained using Fisher Vector to aggregate the convolutional features extracted from VGG-16. This descriptor is suitable for texture classification since it is orderless pooled. FV-CNN substantially improved the state-of-the-art in texture and materials by obtaining better performances compared to FC-CNN descriptor, but it is not trained in an
end-to-end manner.
Lin et al. in (Lin and Maji, 2015; Lin et al., 2015) proposed the bilinear CNN (B-CNN), which consists in feeding the convolutional features to bilinear pooling. It multiplies the outputs of two feature extractors using output product at each location of the image. This model is close to Fisher vectors but has the advantage of the bilinear form that simplifies gradient computation. It also allows an end-to-end training of both networks using image labels only. Both previous related works based on FV-CNN and B-CNN made use of pre-trained VGG models as the base network, but they further apply different encoding techniques which are FV and bilinear encoding. These descriptors are computed by the respective encoding technique from the last convolutional layer of the pre-trained model, which shows better performances than the results from the penultimate fully connected layer.

Zhang et al. in (Zhang et al., 2017) proposed a Deep Texture Encoding Network (Deep-TEN) with a novel Encoding Layer integrated on top of convolutional layers, which ports the entire dictionary learning and encoding pipeline into a single model. Different from other methods built from distinct components such as pre-trained CNN features, Deep-TEN provides an end-to-end learning framework, where the inherent visual vocabularies are directly learned from the loss function. Following the same scheme, Locally-transferred Fisher Vectors (LVF) which involve a multi-layer neural network model, have been proposed in (Song et al., 2017). It contains locally connected layers to transform the input FV descriptors with filters of locally shared weights. Finally, in (Bu et al., 2019), convolutional layer activations are employed to constitute a powerful descriptor for texture classification under an end-to-end learning framework. Different from the previous works, a locality-aware coding layer is designed with the locality constraint, where the dictionary and the encoding representation are learned simultaneously.

3 FEATURE EXTRACTION FOR TEXTURE CLASSIFICATION

Deep learning has reformed machine learning field and has brought a revolution to the computer vision community in recent years. Particularly, good performance has been achieved in image classification by means of CNNs. In this context, different architectures have been proposed so far, composed of many layers organized in a hierarchical way, where each layer adds certain abstraction level to the overall feature representation (Nanni et al., 2017).

Deep feature representations in these networks can act as a set of feature extractors which have the potential of being representative and generic enough with the increasing depth of layers. They encode from bottom to top layers, low-level to more semantic features. Thus, it is not trivial to choose the most relevant layers for a given classification problem. For example, in generic classification tasks, features from top layers are commonly used because they capture semantic information which is more relevant for this kind of applications. However, for texture classification, low-level patterns and fine details from bottom or intermediate layers could be more important than high-level semantic information.

For the aforementioned reasons, instead of empirically choosing one specific layer for the classification task, we first attempt to learn different visual representations by fusing convolutional features from multiple layers for complementary aspect, see section 3.1. Then, we apply a feature selection process to encode multi-layer features into a single feature vector by finding low-dimensional representation that enhances the overall classification performance, see section 3.2. An overview of the proposed approach based on fusion and selection steps from multiple layers is shown in Fig. 1.

3.1 Multi-layer Convolutional Feature Fusion

Given a CNN model \( C \) of \( L \) convolutional layers \( \{L_1, L_2, \ldots, L_L\} \), an input image \( I \) is forward propagated through the network \( C \), which results different numbers of feature maps at each layer. The generated feature maps are denoted by \( \mathcal{M}_{L_1}, \mathcal{M}_{L_2}, \ldots, \mathcal{M}_{L_L} \in \mathbb{R}^{n_i \times m_i \times c_i} \), where \( n_i \times m_i \) is the dimension of channels and \( c_i \) refers to the number of filters in \( L_i \) convolutional layer. These feature maps from different layers have different dimensions, that are usually high. Since high dimensional feature vector increases the computation time of classification, pooling layers that follow convolutional layers are preferred to reduce the dimensionality. Practically, from each layer \( L_i \), the feature vector \( \mathcal{F}_i \) is extracted by applying average or maximum pooling operation on every channel of feature maps \( \mathcal{M}_i \) to get a single value. \( \mathcal{F}_i \) is defined for both cases (average and maximum pooling) as:

\[
\mathcal{F}_i(k) = \frac{1}{n_i \times m_i} \sum_{p=1}^{n_i} \sum_{q=1}^{m_i} \mathcal{M}_i(p, q, k), k = 1, \ldots, c_i \quad (1)
\]

\[
\mathcal{F}_i(k) = \max \mathcal{M}_i(:, :, k), k = 1, \ldots, c_i \quad (2)
\]

From each convolutional layer \( L_i \), the resulting feature vector \( \mathcal{F}_i \) is of size \( c_i \). As a result, a set of feature vectors \( \{\mathcal{F}_1, \ldots, \mathcal{F}_L\} \) is obtained to encode the input image \( I \). These features from different layers are

576
Our approach is applicable to various convolutional neural network architectures. Note that by increasing the depth of layers, the network usually gets better feature representation, but its complexity is increased. For this reason, we choose to experiment Inception-v3 architecture (Szegedy et al., 2016) as a good compromise between the depth of layers and the complexity of network (23.8M parameters). For more details, Inception architecture was initially introduced by Szegedy et al. in (Szegedy et al., 2015). It is essentially based on using variable filter sizes to capture different sizes of visual patterns. Specifically, inception architecture includes 1 x 1 convolutions which are placed before 3 x 3 and 5 x 5 convolutions to act as dimension reduction modules, which enables increasing the depth of CNN without increasing the computational complexity. Following the same strategy, the representation size of Inception-v3 (Szegedy et al., 2016) is slightly decreased from inputs to outputs by balancing the number of filters per layer and the depth of the network without much loss in the representation power.

Inception-v3 is a 48-layers deep convolutional architecture that first consists of few regular convolutional and max-pooling layers. The network follows by three blocks of Inception layers separated by two grid reduction modules. After that, the output of the last Inception block is aggregated via global average-pooling followed by a FC layer. Based on this architecture features can be extracted from multiple layers, precisely we choose them from ‘mixed0’ to ‘mixed10’ layers, as shown in Fig. 1. Obviously, other CNN architectures can be readily employed such as Inception-v4 and Inception-ResNet-v2 (Szegedy et al., 2017), but their computational cost is higher (55.8M). For shallower architectures, VGG-19 (Simonyan and Zisserman, 2015) can be selected.

3.2 Multi-layer Convolutional Feature Selection

Instead of using raw features from multiple layers to encode texture information, we propose to learn a discriminant subspace of the concatenated feature vector, where samples of different texture patterns are optimally separated. For instance, using Inception-v3 for convolutional feature extractors and after applying average or maximum pooling operation, the final representation from multiple layers is of size 10048. By directly classifying such relatively high-dimensional feature vector, apart the fact that the computation time increases, another problem that might incur is that the feature vector generally contains some components irrelevant to texture. Thus, the use of the whole feature vector without any feature selection process could lead to unsatisfactory classification performance (Fradi et al., 2018).

For these reasons, after stacking features from multiple layers and before feeding them into a classifier, we propose the combination of Principle Compo...
Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to find a low dimensional discriminative subspace in which same-texture-pattern samples are projected close to each other while different texture-pattern samples are projected further apart. This process is favorable for the later texture classification step since the influence of irrelevant feature components is minimized. It is important to mention that this combination of PCA followed by LDA has been mainly used in face recognition domain and is commonly referred as “Fisherface”. But, to the best of our knowledge, such feature selection method is applied for the first time on multi-layer convolutional features in order to enhance their discriminative power.

Linear Discriminant Analysis is an efficient approach of dimensionality reduction widely used in various classification problems. It basically aims at finding an optimized projection $W_{opt}$ that projects $D$ dimensional data vectors $U$ into a $d$ dimensional space by: $V = W_{opt}U$, in which the intra-class scatter ($S_W$) is minimized while the inter-class scatter ($S_B$) is maximized. $W_{opt}$ is obtained according to the objective function:

$$W_{opt} = \arg \max_w \frac{W^T S_B W}{W^T S_W W} = [w_1, ..., w_g] \quad (3)$$

There are at most $N - 1$ non-zero generalized eigenvalues ($N$ is the number of classes), thus $g$ is upper-bounded by $N - 1$. Since $S_W$ is often singular, it is common to first apply PCA (Jolliffe, 2002) to reduce the dimension of the original vector. Once the dimensionality reduction process of PCA followed by LDA is applied on the concatenated feature vector from multiple convolutional layers, texture classification is performed by adopting multi-class SVM following one-vs-one strategy.

4 EXPERIMENTAL RESULTS

4.1 Datasets and Experiments

The proposed approach is evaluated within three challenging datasets widely used for texture classification, namely Flickr Material Database (FMD) (Sharan et al., 2010), Describable Textures Dataset (DTD) (Cimpoi et al., 2014) and KTH-TIPS2 (Mallikarjuna et al., 2006). FMD is a well known dataset for texture classification that includes 10 different material categories with 100 images in each class. It focuses on identifying materials such as plastic, wood and glass. During experiments, half of images are randomly selected for training and the other half for testing. KTH-TIPS2 is a database of materials, and is an extension of KTH-TIPS database where TIPS refers to Textures under varying Illumination, Pose and Scale. It contains images of 11 material classes, with 432 images for each class. The images in each class are divided into four samples of different scales (each sample contains 108 images). Following the standard protocol, one sample is used for training and the three remaining ones are used for testing.

DTD is another widely used dataset to assess texture classification algorithms, that contains 5640 images belonging to 47 texture classes, with 120 images for each class. This dataset is considered as the most challenging one since it has varying size of images and some of them are natural images. Following the evaluation protocol published with this dataset, 2/3 of the images are used for training and the remaining 1/3 for testing. In Fig. 2, we show some sample images of different texture classes from the three aforementioned datasets.

The proposed approach presented in Section 3 is evaluated for texture classification. This multi-classification problem is a 10-class, 47-class and 11-class for FMD, DTD and KTH-TIPS2 datasets, respectively. For tests, the proposed feature representation from multiple layers is identified as one of the classes by multi-class SVM classifier following one-vs-one strategy using linear kernel. Following the evaluation protocols published with the datasets, the experiments are repeated and the average top-1 identification accuracy is reported. Also, cross-validation is used to optimize PCA parameters within the training set is adopted.

The obtained results of the proposed approach are compared to the baseline method (Inception-v3) in order to demonstrate the representative and the discriminative power of the proposed feature representation. For feature selection, the proposed method is compared to two other methods: PCA and Neighbourhood Components Analysis (NCA) (Yang et al., 2012). Furthermore, extensive comparative study is conducted in order to highlight the effectiveness of the proposed approach regarding the state-of-the-art methods for texture classification, namely, FC-CNN (Cimpoi et al., 2014), FV-CNN (Cimpoi et al., 2015), B-CNN (Lin and Maji, 2015), Deep-TEN (Zhang et al., 2017), LFV-CNN (Song et al., 2017), and (Bu et al., 2019).

4.2 Results and Analysis

In this section, we first intend to investigate the effect of different layers on the performance of convolutional features using Inception-v3 architecture. The classification accuracies obtained for each convolutional layer (from ‘mixed0’ to ‘mixed10’) on the three datasets are reported in Fig. 3.
First, we notice that average pooling significantly outperforms maximum pooling on the three datasets. For this reason average pooling operator is retained in the next experiments. Not surprisingly, it is shown from the obtained results that the assumption of last layers would give the best performance is not always true. In many cases, it is clear that intermediate layers perform better than the last layer. The best results are 82.8%, 72.52%, and 79.46% obtained from ‘mixed9’, ‘mixed7’, and ‘mixed5’ layers for FMD, DTD and KTH, respectively. Likewise for FC layers, the obtained classification accuracies are 81.80%, 70.03%, and 74.47% for the three datasets. By comparing these results to those shown in Fig. 3, in many cases results from convolutional layers exceed those obtained from FC layers. These observations comply with our main proposal, and justify that the automatic fusion and selection strategy from different layers adopted in this paper could be effective to improve the overall performance.

In Fig. 4, we report the classification results of our proposed approach on the three datasets. To justify the effectiveness of each component from the proposed approach, we first compare the obtained results to the results of the baseline Inception-v3 architecture (using Softmax). Also, to prove the usefulness of the feature selection process using PCA+LDA proposed in our approach, we compare the results to the raw fused convolutional features without performing feature selection (called as MLCF). Besides we substitute PCA+LDA by other feature selection methods, namely PCA and NCA. Note that in this figure, we use the abbreviation MLCF to refer to Multi-Layer Convolutional Features.

As depicted in the figure, the effectiveness of fusing results from multiple convolutional layers has been demonstrated by obtaining higher accuracies on the three datasets compared to the baseline method. Moreover, the proposed feature selection step on the fused convolutional features achieves the best performance compared to the results without feature selection and to the results of other feature selection methods (namely PCA and NCA). The overall enhancement of our proposed approach regarding the baseline method is of 3%, 3.9% and 5.55% for the three datasets, respectively. The main reason behind is that the proposed approach has a higher generalization capability to better capture features at different abstraction levels. Also, the discriminative power of features is enhanced by applying the feature selection step.

Finally, we list the classification accuracies of the state-of-the-art methods FC-CNN (Cimpoi et al., 2014), FV-CNN (Cimpoi et al., 2015), B-CNN (Lin and Maji, 2015), Deep-TEN (Zhang et al., 2017), LFV-CNN (Song et al., 2017), and (Bu et al., 2019) for texture classification on the same datasets in Table 1. We consider in this comparison only recent CNN-based methods since it has been demonstrated in a recently published survey (Liu et al., 2019) that they significantly outperform hand-crafted methods for texture classification.

<table>
<thead>
<tr>
<th>Method</th>
<th>FMD</th>
<th>DTD</th>
<th>KTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC-CNN (Cimpoi et al., 2014)</td>
<td>69.3</td>
<td>59.5</td>
<td>71.1</td>
</tr>
<tr>
<td>FV-CNN (Cimpoi et al., 2015)</td>
<td>80.8</td>
<td>73.6</td>
<td>77.9</td>
</tr>
<tr>
<td>B-CNN (Lin and Maji, 2015)</td>
<td>81.6</td>
<td>72.9</td>
<td>77.9</td>
</tr>
<tr>
<td>Deep-TEN (Zhang et al., 2017)</td>
<td>80.2</td>
<td>-</td>
<td>82.0</td>
</tr>
<tr>
<td>LFV-CNN (Song et al., 2017)</td>
<td>82.1</td>
<td>73.8</td>
<td>82.6</td>
</tr>
<tr>
<td>(Bu et al., 2019)</td>
<td>82.4</td>
<td>71.1</td>
<td>76.9</td>
</tr>
<tr>
<td>Ours</td>
<td>84.8</td>
<td>74.41</td>
<td>81.17</td>
</tr>
</tbody>
</table>

From these comparisons, our proposed approach has been experimentally validated showing more accurate results against the recent state-of-the-art meth-
In this paper, we presented our proposed approach for texture classification based on encoding feature vectors from multiple convolutional layers and fusing them into a joint feature representation after applying a feature selection step. From the obtained results using Inception-v3 as baseline architecture, it has been demonstrated that the classification accuracy is significantly improved on three challenging datasets. Furthermore, the results demonstrate the relevance of the discriminant feature selection process. Also, by means of comparisons to the state-of-the-art methods better results are obtained. As perspectives, the obtained results can be further enhanced using more performing networks. Also, other applications such as fine-grained classification and image retrieval could be investigated using our proposed multi-layer convolutional feature fusion and selection.

5 CONCLUSION

Figure 4: Comparisons of the proposed approach to the baseline architecture (Inception-v3 using Softmax), the original fused multi-layer convolutional features (MLCF), and other feature selection methods (PCA and NCA) on the three experimented datasets in terms of classification accuracy. MLCF stands for Multi-Layer Convolutional Features.

Figure 3: Results of different convolutional layers indexed from 1 (refers to ‘mixed0’) to 11 (refers to ‘mixed10’) from Inception-v3 model on FMD, DTD and KTH-TIPS2 datasets in terms of classification accuracy. Comparisons between maximum and average pooling operators are shown as well.

REPRESENTATIONS


