Multimedia Tools and Applications (2019) 78: 8183–8196
https://doi.org/10.1007/s11042-018-6737-3

Detection of double JPEG compression using modified DenseNet model

Ximei Zeng[1,2] · Guorui Feng[1,2] · Xinpeng Zhang[1,2]

Received: 26 July 2018 / Revised: 14 September 2018 / Accepted: 27 September 2018 /
Published online: 8 October 2018

Abstract
With the increasing tendency of the tempering of JPEG images, development of methods detecting image forgery is of great importance. In many cases, JPEG image forgery is usually accompanied with double JPEG compression, leaving no visual traces. In this paper, a modified version of DenseNet (densely connected convolutional networks) is proposed to accomplish the detection task of primary JPEG compression among double compressed images. A special filtering layer in the front of the network contains typically selected filtering kernels that can help the network following to discriminating the images more easily. As shown in the results, the network has achieved great improvement compared to the-state-of-the-art method especially on the classification accuracy among images with lower quality factors.

Keywords Double JPEG compression · DenseNet · Filtering layer · F-LDA · Residual noises

1 Introduction

As one of the most popular media, JPEG (Joint photographic experts group) images are easily accessible and thus are liable to be altered or manipulated with various basic operations such as image resizing, filtering, splicing, noise addition, contrast enhancement, rotation, double compression and so on [25, 26]. With no visual traces left, such images are often not clear in processing history which could be detrimental in some specific situations [2].

✉ Guorui Feng
fgr2082@aliyun.com

[1] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China

[2] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

Therefore, it is of great significance to verify the trustworthy of a given digital image. A variety of image forensics measures have been developed on the purpose of detecting the traces of these operations.

Among these operations, double compression is a hotspot because it mainly has its place in two different situations. One is that JPEG images often result from image forgery. Operations like image splicing tends to generate a forged image by copying a local block of the source image to the target one, and a JPEG compression is needed if one of the two images is in JPEG format. Consequently, a trace of JPEG compression is exhibited. The other is that in several JPEG steganographic algorithms [27], the images are recompressed during embedding after decompressed into the spatial domain. A double compression will take place if the quantization matrix used in embedding differs from that used in the original compression. This steganalysis usually uses the similar idea with the double compression detection [17].

For JPEG double compression detection, researchers have proposed several methods based on different algorithm theory. In [19], a noisy mixing model which is studied in blind component analysis was formulated to interpret the shifted double JPEG compression problem. To adapt the noisy environment, aside from the independent component analysis (ICA), the asymmetry of independent value map (IVM) was learned to obtain the independency's normalized criteria. The features exploited were then fed to a support vector machine (SVM) classifier. Besides, in [4], differences between magnitude of JPEG coefficient 2-D array of a given image and the shifted versions along various directions are used to strengthen artifacts of double JPEG compression. Transition probability matrices are obtained with a thresholding technique applied to Markov random process. In another study, the neighboring joint density features and marginal density features on the discrete cosine transform (DCT) coefficients are extracted to help the learning classifiers to detect the clue of manipulation [16]. Moreover, the variation trend of the DCT coefficients when a JPEG image is recompressed with the same quantization matrix again and again was studied in [8]. The differences of the trends among uncompressed, single-compressed and double compressed images were then used to discriminate the images undergoing unknown times of compression with a novel random perturbation strategy. Similarly, feature vector formed by histograms of low-frequency DCT coefficients is used to accomplish a classification using SVM in [18]. Other methods exploiting features from DCT coefficients are proposed in [15, 22, 24].

In this paper, a special filtering layer and blocks of the DenseNet (densely connected convolutional networks) are comprised to form a new neural network trained to check the compression history of a given image. In particular, a network based on features from both spatial domain and frequency domain is proposed to perform image forgery detection on RGB images. The network is trained to differentiate images undergone uncompressed, single-compressed, and double-compressed. If the image is classified as double-compressed, the maximum likelihood of the quality factor (QF) of the primary JPEG compression is detected.

The rest of the paper is organized as follows. A brief overview of CNNs and DenseNet is given in Section 2. In Section 3, we describe the proposed model at full length, as well as the process of filtering kernel selection in the first layer. Experimental results and performance comparisons are presented in Section 4. Conclusions are given in Section 5.

2 Related work

In recent years, neural networks especially convolutional neural networks (CNN), have shown ability of extracting and learning complex features and their representations from images. CNN was first introduced in 1990 [14], but did not catch much attention until 2012 when it was used in the image classification on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [5, 12]. It benefits from fewer connections and parameters to train and reduced computational cost due to the pooling layer, compared with standard feedforward neural networks.

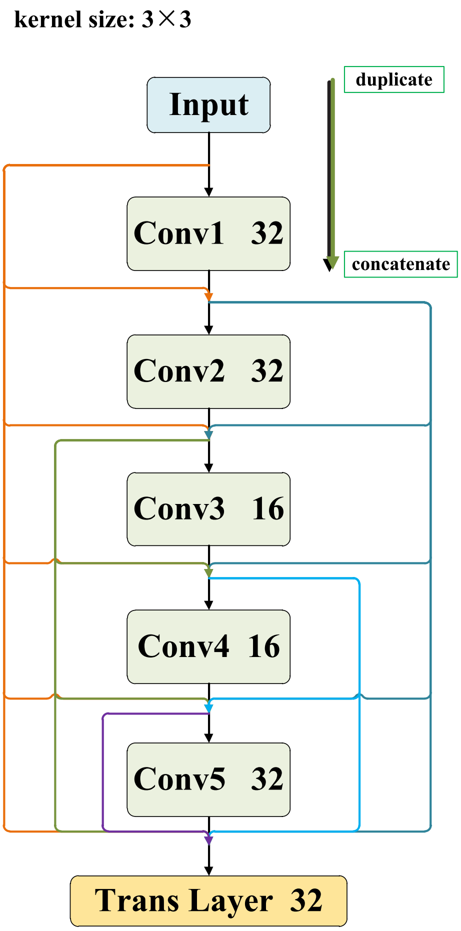
In the wake of the wide utilization of CNNs, they have been the major methods for visual object recognition, and many improved versions of CNN have been springing up. Among them, the residual deep learning framework introduced in [7] was designed to address the degradation problem: as the number of the layers increases, instead of steadily improving or remaining stable, accuracy degrades rapidly and training error rises with the depth of the network increasing [6, 9, 13, 20, 21]. A deep residual learning framework explicitly let the layers fit a residual mapping by simply performing an identity mapping with a shortcut connection which skips two convolutional layers. With the identity mapping, it is easier for the solver to find the perturbation when the optimal function is closer to an identity mapping rather than a zero mapping. Thus, the residual networks can ease the optimization by providing faster convergence. In fact, not only the residual network, but also several other researches have started to realize a similar improvement on CNN: to create short connections from early layers to later ones [9, 13, 21].

In [10], the researchers simply distilled the idea above, and applied it to a greater part of the network. They connects all layers each other to ensure the maximum usage of the output information of every layer involved. As shown in Fig. 1, each layer takes in additional inputs from the early layers, and meanwhile delivers its feature maps to all later layers. Instead of summation, concatenation is used at the joint points, so that each layer's input contains all the output information of the preceding layers, and thus much fewer feature maps are needed. The problem of gradient-vanishing is also alleviated for the implicit deep supervision that each layer has the direct original input signal as well as the gradients of loss function. All of these advantages will help a lot in avoiding overfitting problems.

More concretely, in one dense block, there are several convolutional layers including one transition layer. The duplicated outputs of the previous layers are directly concatenated to every input of the latter layers. That is to say, the number of the feature maps of each layer in a dense block has a trend of increase. In the present paper, the number of feature maps of each layer in a dense block in the proposed network is: 32, 32, 16, 16, 32, and 32. In addition, the input layer of the dense block has 32 feature maps, making a total of 160 feature maps concatenated together at the input of the last transition layer. The sizes of the convolutional kernels in the convolutional layers defined in the proposed network are all 3×3. The detailed structure of a dense block and the parameters set can be seen in Fig. 1.

In [1], the authors applied a simply CNN based network to accomplish detection of double JPEG compression. Besides, a multi-branch CNN network with raw DCT coefficients as input was proposed in [15]. Moreover, a well-designed pre-processing step concatenating the histograms of different selective sub-bands with different numbers of bins was applied before the deep convolutional neural network [24]. However, to our best knowledge, few works have applied the DenseNet in the field of double JPEG compression detection. Considering the training time and accuracy performance, four dense blocks are used in the proposed model.

Fig. 1 Layout of a dense block with all layers connected



3 CNN-based network structure

3.1 Network architecture

To accomplish the task of detecting the existence primary JPEG compression and its quality factor, a mainly spatial domain-based network with and without a frequency-based part is defined, respectively. The spatial domain-based network is composed of a cascade of several convolutional layers, pooling layers, fully-connected layers, and dense blocks. As for the frequency-based part, it appears in form of a branch and joins the spatial domain-based part at the second fully-connected layer. Figure 2 shows the architecture of the proposed networks.

The first layer of the spatial domain part is a special filtering layer containing 12 specifically defined filtering kernels selecting using Fisher Linear Discriminant Analysis (F-LDA). The details will be disclosed in the next section. A convolutional layer with 32 kernels of size 3 × 3 follows the first layer and the resolution of the feature maps the output is then reduced to 32 × 32 with a kernel of size 2 × 2 and a stride of two. After that, four dense blocks each containing five convolutional layers and one transition layer. In addition, as the

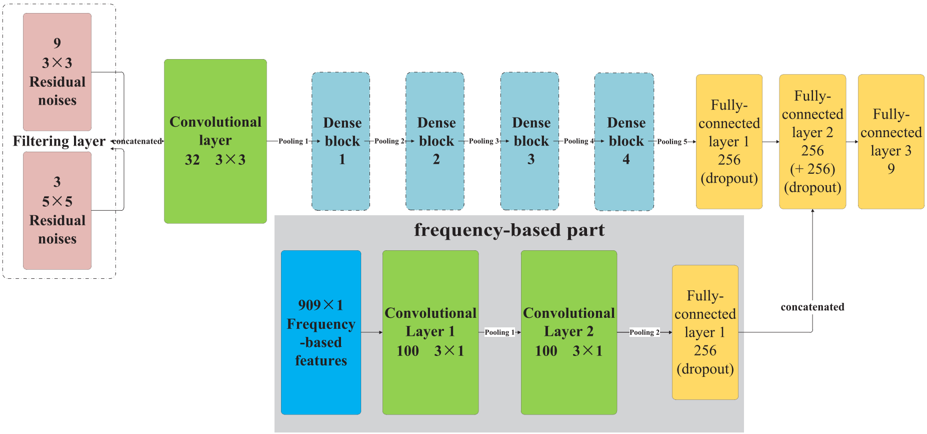


Fig. 2 Framework of the proposed neural network

last part of each dense block, an average pooling layer downsamples the output of them with a 2×2 sized kernel and a stride of two. Thus, the resolution is reduced to 2×2 after the fifth average pooling layer. The output is then reshaped to a column vector of length 128 and followed with three fully-connected layers. The first two fully-connected layers contain 256 neurons while the last one contains nine neurons since it is the classification results output from the network. To prevent overfitting, dropout technique is used in the first two fully-connected layers so that half of the neurons are randomly dropped during training. The same network is applied to eight different datasets with QF2 varying from 60 to 95 with step 5.

The frequency-based part of the network keeps in accordance with that in the research [1]. The input of it is a 909×1 sized vector extracted from every image based on the first nine spatial frequencies in the Zigzag scan order (the direct component is omitted) and the corresponding histogram of their DCT coefficients following the idea described in [23] and [1]. The next two layers are both convolutional layers with 100 kernels of size 3×1, each followed by a max-pooling layer with a stride of two. After them, there is a fully-connected layer containing 256 neurons. The output of it joins together into the output of the first fully-connected layer of the spatial domain-based part, making a second fully-connected layer containing 512 neurons with a dropout technique. During the validation experiments, a purely spatial domain-based network and a multi-domain-based one are carried out to complete the classification task respectively and the results are listed in two tables. We will then give a simple introduction to every kind of layer in the network.

Convolutional layer: a conventional convolutional layer contains two main operations: convolution and non-linearity. The convolution operation extracts larger quantity of features through the theory of receptive filed and shared weights. It aggregates the response of both local and global areas of the input as particular feature representations called feature maps. In the proposed network, every convolutional layer in the spatial domain-based part follows a batch normalization (BN) layer and a Rectified Linear Unit (ReLU) technique sequentially (Fig. 3). Batch normalization is a technique introduced by [11], which can accelerate the convergence rate of training process by standardizing the distribution of the inputs of each layer so as to reduce the 'internal covariate shift' (that is, the phenomenon that the distribution of each layer's input changes during training). Normalization is applied to each training mini-batch as a part of the model architecture. With batch normalization, higher learning

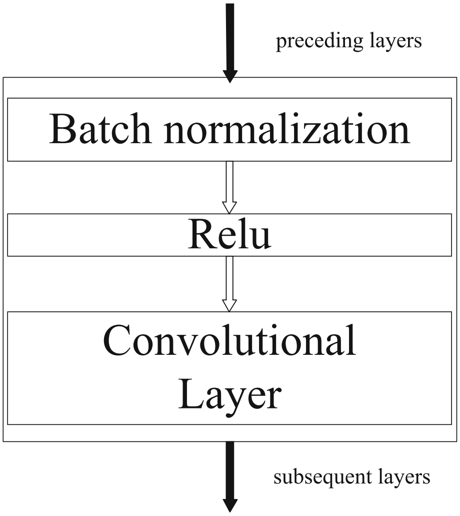


Fig. 3 Extended configuration of a convolutional layer

rate can be used to train the model and we no longer need to take much care of parameter initialization. In addition, the activation ReLU non-linearity, i.e., $f(x) = max(0, x)$, also allows the model train faster than other activation functions [3].

Pooling layer: to relieve the computational complex and probability of over-fitting, after obtaining feature maps from the convolutional layers, the pooling layer is used to discard irrelevant information while preserving important one over a local region of the feature maps. There are mainly two pooling methods: average pooling and max pooling. The outputs of neighboring neurons in a local region of the feature maps are summarized as the average and maximum value and then propagated to the next layer as higher-level feature representations.

Fully-connected layer: several various-sized fully-connected layers comprise a classification module. A dropout technique is used for the two previous fully-connected layers by randomly setting the neurons to zero for the alleviation of over-fitting problem. For the last fully connected layer, a softmax activation function is applied to generate a distribution over all nine class labels to accomplish the classification.

3.2 Special filtering layer

To find out the most appropriate filtering kernels in the first layer of the spatial domain-based network for the classification task, the F-LDA method is used to rank the ability of residual noise defined for image classes distinguishing. F-LDA is a method used in machine learning to find out a linear combination of features which are able to separate several classes from each other. It projects the high dimensional pattern samples over the optimum discriminant vector space so as to extract useful information for classification and reduce the dimensions of feature subspaces. To ensure the best separability of the samples of different classes, the projection makes sure that the pattern samples have the maximal between-class scatter but the minimal within-class scatter. Mathematically, the between-class scatter can be defined as:

$$\mathbf{S}_{be} = \sum_{i=1}^{c} N_i(\mathbf{u}_i - \mathbf{u})(\mathbf{u}_i - \mathbf{u})^T, \quad \mathbf{u} = \frac{1}{N}\sum \mathbf{x} \tag{1}$$

where N is the number of samples in the dataset, c is the number of classes, $\mathbf{x}$ is the input training feature and $\mathbf{u}_i$ is the mean vector of every class. The within-class scatter can be defined as:

$$\mathbf{S}_{in} = \sum_{i=1}^{c} \mathbf{S}_i, \quad \mathbf{S}_i = \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{u}_i)(\mathbf{x} - \mathbf{u}_i)^T \tag{2}$$

where $\mathbf{S}_i$ is the covariance matrix of every class and C_i is the set of each class. Thus, the object function can be described as:

$$J(\mathbf{W}) = \frac{\|\mathbf{W}^T \mathbf{S}_{be} \mathbf{W}\|}{\|\mathbf{W}^T \mathbf{S}_{in} \mathbf{W}\|} \tag{3}$$

where $\mathbf{W}$ is the projection matrix. The projection matrix can also be referred to $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k]$, and $\mathbf{w}_i$ satisfies the following formula:

$$\mathbf{S}_{in}^{-1} \mathbf{S}_{be} \mathbf{w}_i = \lambda_i \mathbf{w}_i \tag{4}$$

indicating that $\mathbf{w}_i$ is the eigenvectors of the matrix $\mathbf{S}_{in}^{-1}\mathbf{S}_{be}$. In turn, the corresponding eigenvectors to the first k greatest eigenvalues obtained after applying matrix eigenvalue decomposition to the matrix $\mathbf{S}_{in}^{-1}\mathbf{S}_{be}$ can be chosen to form the projection matrix, and the value of k is no more than $C - 1$.

In the present paper, we rank every residual noise's ability to get the class separation for each class through F-LDA and find out the residuals that are good at separation as many classes as possible. First of all, we defined seven basic filters (BF) among which four are of size 3 × 3 and the other three are 5 × 5 (Fig. 4). To expand the diversity of RN, taking the 3 × 3 filter named BF3 as an example, we generate more filters using the formula below:

$$R_N = \alpha \times R_A \pm \beta \times R_B \tag{5}$$

where $\alpha, \beta \in \{0, 0.1, \ldots, 1\}$, and $R_A, R_B \in \{BF3_1, BF3_2, BF3_3, BF3_4\}$.

Altogether 144 unique BF3 are obtained after eliminating the repeated ones. Next, 1000 images for each class are randomly chosen after cropping the origin images of resolution

0	**0**	**0**
0.5	**-1**	**0.5**
0	**0**	**0**

0	**0.5**	**0**
0	**-1**	**0**
0	**0.5**	**0**

0.5	**0**	**0**
0	**-1**	**0**
0	**0**	**0.5**

0	**0**	**0.5**
0	**-1**	**0**
0.5	**0**	**0**

-1	**2**	**-2**	**2**	**-1**
2	**-6**	**8**	**-6**	**2**
-2	**8**	**-12**	**8**	**-2**
2	**-6**	**8**	**-6**	**2**
-1	**2**	**-2**	**2**	**-1**

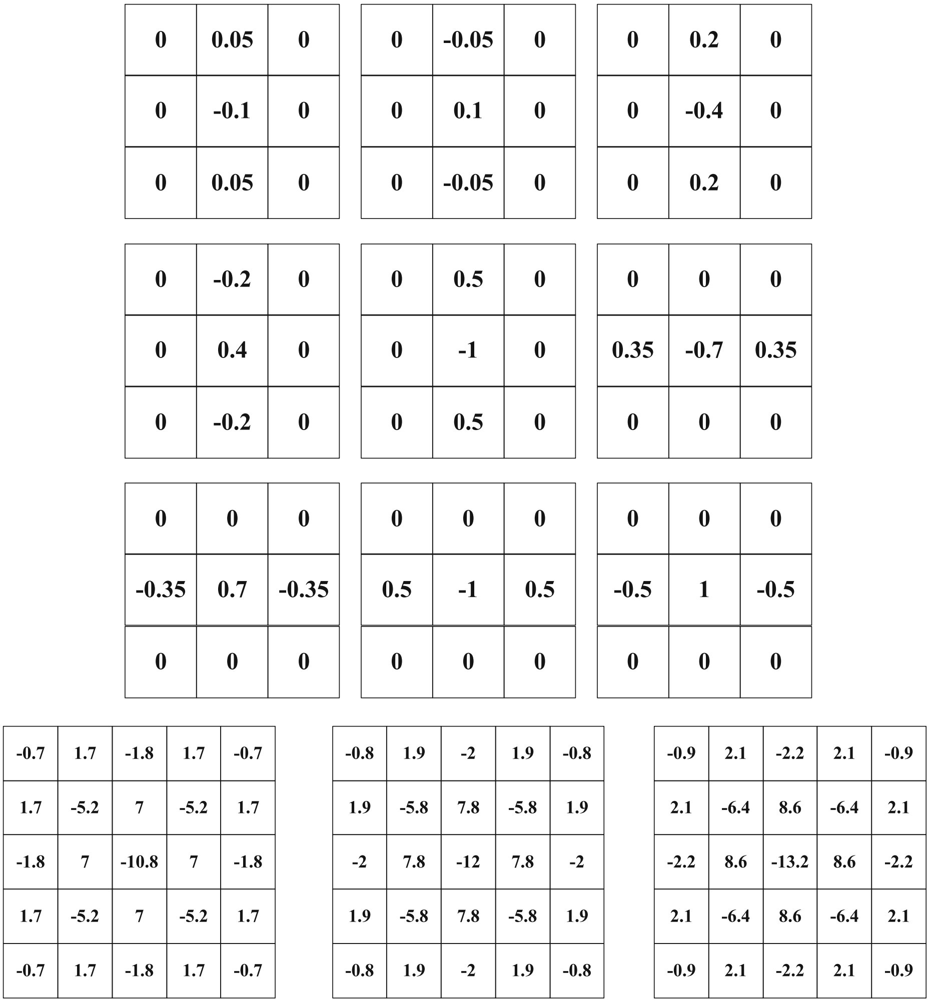
1	**1**	**-2**	**1**	**1**
1	**-4**	**6**	**-4**	**1**
-2	**6**	**-12**	**6**	**-2**
1	**-4**	**6**	**-4**	**1**
1	**1**	**-2**	**1**	**1**

0	**0**	**-1**	**0**	**0**
0	**-2**	**6**	**-2**	**0**
-1	**6**	**-12**	**6**	**-1**
0	**-2**	**6**	**-2**	**0**
0	**0**	**-1**	**0**	**0**

Fig. 4 Definitions of seven basic filters

512 × 384 to 48 subimages sized 64 × 64. For each image, we extracted 400 blocks of size 3 × 3 in the center. As for the RGB images have three channels, each block is an average result of three blocks extracted from three channels. In all, a training sample of 400 × 1000 × 144 = 57600000 residual noises are obtained for each class. After that, a feature matrix of size 9000 × 400 is generated with each BF. Thus, altogether 144 feature matrixes are generated.

Using 400 features of each image among the nine classes, we calculate the linear discriminant coefficients through F-LDA. The linear discriminant coefficients are used for classification. We calculate the linear scores with the coefficients and finally, a posterior probability of each one of the 9000 images are obtained. To see the ability to distinguish among the nine classes of images, we calculate the average posterior probabilities of the 1000 images in one class so that for each BF, there is a 9 × 1 vector demonstrating its capacity of classification. We rank the 144 posterior probabilities of each class and find out the



0	0.05	0
0	-0.1	0
0	0.05	0

0	-0.05	0
0	0.1	0
0	-0.05	0

0	0.2	0
0	-0.4	0
0	0.2	0

0	-0.2	0
0	0.4	0
0	-0.2	0

0	0.5	0
0	-1	0
0	0.5	0

0	0	0
0.35	-0.7	0.35
0	0	0

0	0	0
-0.35	0.7	-0.35
0	0	0

0	0	0
0.5	-1	0.5
0	0	0

0	0	0
-0.5	1	-0.5
0	0	0

-0.7	1.7	-1.8	1.7	-0.7
1.7	-5.2	7	-5.2	1.7
-1.8	7	-10.8	7	-1.8
1.7	-5.2	7	-5.2	1.7
-0.7	1.7	-1.8	1.7	-0.7

-0.8	1.9	-2	1.9	-0.8
1.9	-5.8	7.8	-5.8	1.9
-2	7.8	-12	7.8	-2
1.9	-5.8	7.8	-5.8	1.9
-0.8	1.9	-2	1.9	-0.8

-0.9	2.1	-2.2	2.1	-0.9
2.1	-6.4	8.6	-6.4	2.1
-2.2	8.6	-13.2	8.6	-2.2
2.1	-6.4	8.6	-6.4	2.1
-0.9	2.1	-2.2	2.1	-0.9

Fig. 5 Details about 12 selected basic filters

same BFs shared by all nine classes in the top 50 ones ranked in each class. Nine BF3 come out during the selection. The same operations are done to three basic BF5. We obtain 440 unique BF, and in the end, three of them are chosen. In Fig. 5, we depict the details about 12 selected basic filters. Furthermore, In Fig. 6, a simple workflow of the process of BF selection is also provided.

4 Experiments

4.1 Experimental setup

We use the UCID database for the experimental tests. For the sufficiency of the data, we crop each of the 1338 images (size: 512 × 384) into 48 blocks sized 64 × 64 and thus 64224 images are made. Among them, 90% images (57798) are used for training and the others (6426) for testing. For the first and second JPEG compression, quality factors (QF) are set from 60 to 95 with a step of five. Moreover, when $QF1 = QF2$, the second compression is skipped considering the class would fall in the single-compressed class. Therefore, there are seven double-compressed classes and another two classes, uncompressed and single-compressed, for a network to be trained to classify. Eight networks are trained on eight groups of datasets varying with QF2 to detect the primary JPEG compression. In addition, based on the DCT coefficients, features of frequency-domain are extracted. In detail, for each 8 × 8 block, DCT coefficients are extracted from a given $N \times N$ patch. We select the first nine spatial frequencies in the zigzag scan order (eliminating the DC component). We then form a 101 × 1 vector based on the occurrences of the absolute values of quantized DCT values varying from -50 to 50. Thus, a vector containing 909 elements (101 vectors × 9 columns) is taken as the input of the frequency-based part of network. After that, another eight networks with the extra frequency-domain features are trained as before on the same eight groups of datasets.

We implement the proposed model using the Tensorflow 1.5 deep learning framework, Nvidia 1080 Ti GPU with 16GB RAM. The network is optimized using Momentum Optimizer, with momentum=0.9. The learning rate is set as 0.1 due to the existence of BN layers. We set the batch size for training and testing to 64 images.

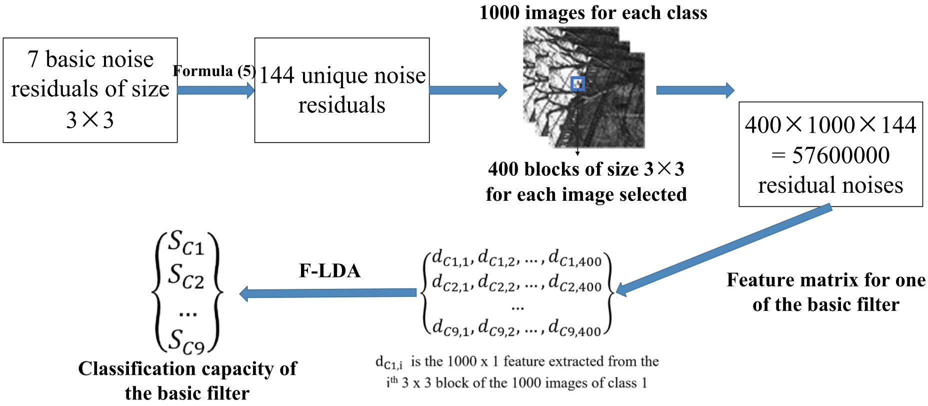

1000 images for each class
7 basic noise residuals of size 3×3
Formula (5)
144 unique noise residuals
400 blocks of size 3×3 for each image selected
400×1000×144 = 57600000 residual noises
Feature matrix for one of the basic filter
F-LDA
S_{C1}, S_{C2}, …, S_{C9}
$d_{C1,1}, d_{C1,2}, \dots, d_{C1,400}$
$d_{C2,1}, d_{C2,2}, \dots, d_{C2,400}$
…
$d_{C9,1}, d_{C9,2}, \dots, d_{C9,400}$
Classification capacity of the basic filter
$d_{C1,i}$ is the 1000 x 1 feature extracted from the i^{th} 3 x 3 block of the 1000 images of class 1

Fig. 6 A simple workflow of the process of BF selection

Table 1 Classification results over eight groups of datasets using the spatial domain-based network

		QF2								
	%	60	65	70	75	80	85	90	95	AVG
	Uncompressed	97.8	97.6	97.8	97.5	97.7	97.9	97.6	97.8	97.7
		−2.1	−2.2	−1.6	−2.2	−2.0	−2.0	−2.1	−1.8	−2.0
	Single compressed	83.2	85.6	86.2	86.4	88.3	91.2	91.7	92.4	88.2
		+23.3	+15.5	+14.5	+7.5	+4.0	−4.3	−6.4	−6.2	+6.1
QF1	60	—	74.3	78.3	81.2	83.6	86.9	89.3	91.1	83.5
		—	+34.0	−8.7	−10.6	+3.2	−4.3	+9.0	+8.4	+4.4
	65	73.2	—	77.2	79.2	80.8	83.4	86.1	86.7	80.9
		+49.7	—	+30.2	+0.9	+27.6	+16.2	+19.2	+9.6	+21.9
	70	74.6	76.7	—	80.4	81.5	82.3	83.5	83.8	80.4
		+32.3	+41.1	—	+24.9	+16.9	+27.2	+17.4	+1.9	+23.1
	75	85.6	85.2	83.9	—	89.4	90.2	93.3	94.2	88.8
		+22.3	+29.1	+42.4	—	+14.8	+18.6	+19.4	+15.7	+23.2
	80	87.5	84.7	85.1	86.2	—	90.5	93.2	94.2	88.8
		+7.9	+13.3	+27.1	+39.5		+1.4	+12.2	+9.0	+15.8
	85	89.8	88.8	90.5	91.6	93.7	—	95.3	95.6	92.2
		+26.2	+41.9	+11.3	+9.0	+14.3	—	+4.5	+3.0	+15.8
	90	90.0	91.3	93.2	93.5	93.8	94.2	—	95.0	93.0
		+16.0	+15.8	+16.1	+18.9	+3.9	−1.4	—	−4.1	+9.3
	95	90.8	90.4	86.2	91.3	94.7	95.3	95.6	—	92.0
		+20.6	+19.1	+46.7	+17.9	+5.1	+2.1	+1.4	—	+16.1

4.2 Spatial domain-based network results

The performance of the spatial domain-based network is shown in Table 1. Both of the accuracy and the True Positive Rate (TPR) are shown in the table. As the results tested on 6426 images shown, the comprehensive accuracy rate is up to 88.6%. Among them, it is obvious that the performance is much better in the top-right area of the table because it is easier to detect the existence of the primary JPEG compression as the traces of it could easily remain when $QF2 > QF1$. Note that we only use features extracted from the spatial domain and the results are far better than the results in [1] using the spatial domain-based CNN and are close to the results using the multi-domain-based CNN in it. The positive and negative differences between the results of the network we proposed and the spatial domain-based network in [1] are also listed in Table 1.

4.3 Multi-domain-based network results

The results of the multi-domain-based network are shown in Table 2. As the results tested on 6426 images shown, the comprehensive accuracy rate is up to 95.0%. Generally, significant improvement can be spotted in all of the eight networks. Similarly, the positive and negative differences between the results of the network we proposed and the multi-domain-based network in [1] are also listed in different colors.

Table 2 Classification results over eight groups of datasets using the multi-domain-based network

		QF2								
	%	60	65	70	75	80	85	90	95	AVG
	Uncompressed	98.6	98.3	98.5	98.6	98.6	98.5	98.7	98.9	98.6
		−1.3	−1.7	−1.5	−1.3	−1.3	−1.5	−1.3	−0.8	−1.3
	Single compressed	94.4	95.2	96.3	96.9	97.1	96.7	96.9	97.3	96.4
		+11.1	+11.9	+8.2	+6.6	+2.2	−0.4	+0.4	−2.1	+4.8
QF1	60	—	91.6	91.8	93.3	94.0	95.3	95.1	95.6	93.8
		—	+2.4	−6.5	−5.4	−3.6	−3.7	−2.9	−3.6	−3.4
	65	88.7	—	90.0	92.2	92.5	93.0	94.2	93.8	92.1
		+20.8	—	+3.6	−3.6	−2.1	−5.0	−5.2	−5.3	+0.5
	70	88.0	88.3	—	91.4	92.5	93.7	94.4	94.6	91.8
		+4.2	+12.3	—	+4.5	−6.0	−4.5	−3.7	−3.9	+0.4
	75	89.6	89.0	89.5	—	93.4	95.0	95.6	95.8	92.6
		+3.2	+8.1	+16.1	—	−1.8	−2.4	−2.6	−3.3	+2.5
	80	89.9	90.4	92.1	93.1	—	95.8	96.4	96.4	93.4
		+12.7	+2.4	+8.0	+4.7		−2.4	−3.0	−2.7	+2.7
	85	93.4	94.1	94.5	94.6	96.3	—	97.7	98.1	95.5
		+17.0	+18.1	+13.0	+7.0	+4.4	—	−0.7	−1.2	+8.2
	90	97.2	97.3	97.8	98.2	98.1	98.2	—	98.5	97.9
		+31.3	+20.6	+23.9	+15.6	+5.7	+2.7	—	−1.1	+14.1
	95	96.3	96.8	97.5	97.3	97.7	98.0	98.3	—	97.4
		+13.3	+26.5	+20.2	+21.3	+4.5	+2.1	+9.4	—	+13.9

From Fig. 7, we can see that the performances of eight proposed networks rise steadily with QF2 increasing from 60 to 95. The performance of the proposed network is well better than that of the network in [1] when QF2 is no more than 80, but only not as good as the multi-domain-based one in [1] after QF2 surpasses 85.

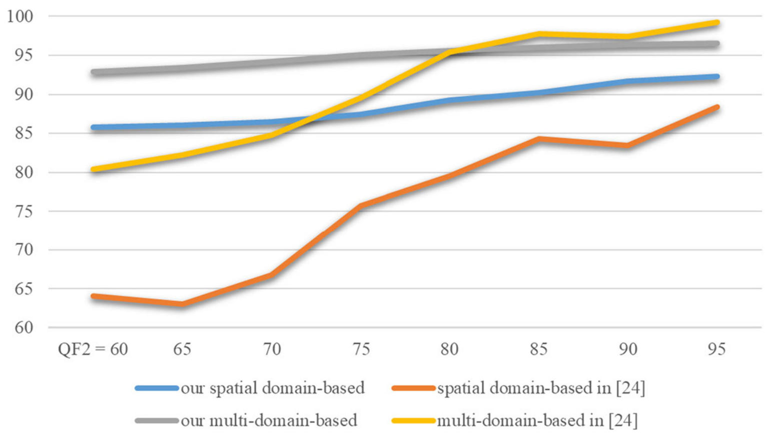


Fig. 7 Comparison of the classification results of four networks

5 Conclusions

To meet the challenge of double JPEG compression detection, the present paper has proposed a modified version of DenseNet with a special filtering layer applied previously in the front of the network and a frequency-based part exploiting DCT features from the images. F-LDA method is used to select the most appropriate filtering kernels among a large amount of predefined residual noises based on the posterior probability obtained. The DenseNet was applied to achieve incremental training accuracy, efficient reutilization of the training features and persistent feature propagation. Results show that the networks perfectly accomplish the task of double JPEG compression detection, especially making great progress when QF2 is less than 80 comparing with the other works.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grants (U1536109, U1636206, 61525203, 61373151, 61472235).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Amerini I, Uricchio T, Ballan L, Caldelli R (2017) Localization of JPEG double compression through multi-domain convolutional neural networks. In: IEEE Conference on computer vision and pattern recognition workshops on media forensics
2. Amerini I, Uricchio T, Caldelli R (2017) Tracing images back to their social network of origin: a CNN-based approach. In: IEEE Workshop on information forensics and security, pp 1–6
3. Bayar B, Stamm MC (2016) A deep learning approach to universal image manipulation detection using a new convolutional layer. In: The 4th ACM workshop on information hiding and multimedia security, pp 5–10
4. Chen C, Shi Y-Q, Su W (2008) A machine learning based scheme for double JPEG compression detection. In: IEEE International conference on pattern recognition, pp 1–4
5. Girshick R, Donahue J, Darrell T, Malik J, Berkeley U (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE conference on computer vision and pattern recognition, pp 580–587
6. He K, Sun J (2015) Convolutional neural networks at constrained time cost. In: IEEE conference on computer vision and pattern recognition, pp 5353–5360
7. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
8. Huang F, Huang J, Shi Y-Q (2010) Detecting double JPEG compression with the same quantization matrix. IEEE Trans Inf Forens Secur 5(4):848–856
9. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: European conference on computer vision, pp 646–661
10. Huang G, Liu Z, Maaten LVD, Weiberger KQ (2017) Densely connected convolutional networks. In: IEEE Conference on computer vision and pattern recognition
11. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456
12. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: International conference on neural information processing systems, pp 1097–1105
13. Larsson G, Maire M, Shakhnarovich G (2016) FractalNet: ultra-deep neural networks without residuals, arXiv:1605.07648
14. LeCun Y, Boser BE, Denker JS et al (1990) Handwritten digit recognition with a back-propagation network. Adv Neural Inf Process Syst, 396–404

15. Li B, Luo H, Zhang H, Tan S, Ji Z (2017) A multi-branch convolutional neural network for detecting double JPEG compression, arXiv:1710.05477
16. Liu Q, Sung A-H, Qiao M (2011) A method to detect JPEG-based double compression. Int Symp Neural Netw LNCS 6676:466–476
17. Luo X, Song X, Li X et al (2016) Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes. Multimed Tools Appl 75(21):13557–13583
18. Pevny T, Fridrich J (2008) Detection of double-compression in JPEG images for applications in steganography. IEEE Trans Inf Forens Secur 3(2):247–258
19. Qu Z, Luo W, Huang J (2008) A convolutive mixing model for shifted double JPEG compression with application to passive image authentication. In: IEEE International conference on acoustics
20. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks, arXiv:1505.00387
21. Srivastava RK, Greff K, Schmidhuber J (2015) Training very deep networks. Adv Neural Inf Process Syst, 2377–2385
22. Verma V, Agarmal N, Khanna N (2018) DCT-domain deep convolutional neural networks for multiple JPEG compression classification. Signal Process-Image Commun 67:22–23
23. Wang Q, Zhang R (2016) Double JPEG compression forensics based on a convolutional neural network. EURASIP J Inf Secur 2016:23
24. Yang P, Ni R, Zhao Y (2018) Double JPEG compression detection by exploring the correlations in DCT domain, arXiv:1806.01571v1
25. Yao H, Song S, Qin C, Tang Z, Liu X (2017) Detection of double-compressed H.264/AVC video incorporating the features of the string of data bits and skip. Macroblocks 9(12):313
26. Yao H, Cao F, Tang Z, Wang J, Qiao T (2018) Expose noise level inconsistency incorporating the inhomogeneity scoring strategy. Multimed Tools Appl 77(14):18139–18161
27. Zhang Y, Qin C, Zhang W, Liu F, Luo X (2018) On the fault-tolerant performance for a class of robust image steganography. Signal Process 146:99–111

Ximei Zeng received B.S. degree in Physical and Electrical Engineering Institute of Anqing Normal University and continues her M.S. in School of Communication and Information Engineering in Shanghai University.

Guorui Feng received the B.S. and M.S. degree in computational mathematic from Jilin University, China, in 1998 and 2001 respectively. He received Ph.D. degree in electronic engineering from Shanghai Jiaotong University, China, 2005. From January 2006 to December 2006, he was an assistant professor in East China Normal University, China. During 2007, he was a research fellow in Nanyang Technological University, Singapore. Now he is with the school of communication and information engineering, Shanghai University, China. His current research interests include image processing, image analysis and computational intelligence.

Xinpeng Zhang received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He was with the State University of New York at Binghamton as a visiting scholar from January 2010 to January 2011, and Konstanz University as an experienced researcher sponsored by the Alexander von Humboldt Foundation from March 2011 to May 2012. His research interests include multimedia security, image processing, and digital forensics. He has published more than 170 papers in these areas.