Two Deep Learning Approaches for Fraud Detection in Receipt

Vincent Barra[2], Alexandre Fabre[1],
Pascal Lafourcade[2], and Kergann Le Cornec[2]

[1] Coffreo, France
[2] LIMOS, Université Clermont-Auvergne, France

Abstract. This paper is a response to the **ICPR** Fraud Detection Contest. Our aim is to detect image manipulation using a training corpus of 600 jpeg files, 470 genuin and 130 manipulated. Our approach is to combine **deep-learning** with **fraud detection**'s techniques to achieve more than 90 % good guesses.

Keywords: ICPR · Deep-Learning · Fraud Detection

1 Introduction

Digital uses is now the norm in every organisation, from healthcare to any small company, each and every document is numerised. However, numerisation comes with digital manipulation and an easy access to image retouching tools simplifies their falsification. As a result, image fraud detection has become a strategic challenge for any company. The goal of ICPR Fraud Detection Contest is to detect tampered regions in receipts. We applied two deep-learning network on pre-processed images. We give in table 1 the result of the combined pre-proccessed methods fed to two network. In this paper we describe our methods used in fraud detection.

In table 1, RGB is the genuine image (Red,Green,Blue), GrayScale is the image transformed in grey. The other methods are explained in Section III.

Our contribution presented by this paper is to feed pre-processed image with Image Manipulation Techniques to a deep neural network. The network will try to classify images using informations given by these techniques. Usualy convolution layers within a deep neural network will find those informations on their own. However, we did not have a big enough dataset, we can see on table 1 that RGB barly achieved more than 60 % good guesses. We decided to help the convolutions layers by giving them pre-processed informations. We choose these methods because they are widely used in fraud detection. They also keep the same structure as the original image, the information of a pixel stays at the same place so they can be combined with each other and with the greyscale image. Moreover, they give different informations which can be usefull for the network.

Table 1. Accuracy of the different methods on AlexNet and ResNet.

Network	Combined Methods	Accuracy
AlexNet	RGB	62%
AlexNet	ELA+PCA+LBP	65%
AlexNet	ELA+Wavelet+LBP	74%
AlexNet	ELA+Wavelet+GrayScale	76%
AlexNet	ELA+Wavelet+GrayScale+Fraud Creation	85%
ResNet152	RGB	63%
ResNet152	ELA+PCA+LBP	65%
ResNet152	ELA+LBP+Wavelet	75%
ResNet152	ELA+Wavelet+GrayScale	80%
ResNet152	ELA+Wavelet+GrayScale+Fraud Creation	91%

Outline: The paper is organized as follows. Section II describs the dataset provided by this challenge, and data augmentation techniques used to get enough data to be able to use deep learning methods. Section III explains all the preprocessing methods we used to find the best information in order to detect image manipulation. Finally, section IV presents the deep-learning methods used, AlexNet and Resnet.

2 Datasets

ICPR Fraud Detection Challenge provided the following three datasets:

- 100 frauded receipts, where the manipulation is localized.
- 470 genuine receipts.
- 30 frauded receipt where the manipulation is not localized.

As 600 images is not enough for deep-learning techniques, we used data augmentation's techniques.

2.1 Data-Augmentation

We first decided to divide the image into small 227×227 frames, that size is optimize with our network implementation (perfect for the size of the convolution layers). We were then compeled to use the first dataset (where the manipulation is localized), in order to certify which frame was manipulated or was genuine. Then we overlap each frame with a choosen step (we choose 20) not to miss anything and to have more data. Each time we arrive to a fraudulous section, we lower the step (to 5) to create more manipulated section. Finally, we flip twice and rotate 3 times (90,180,270) all the manipulated frames.

We now have $200,000$ frames of size 227×227 with 50 % manipulated and 50 % genuine.

2.2 Fraud Creation

After achieving more that 80% good guesses with combining Error Level Analysis, Wavelet and GrayScale 1, we created frauds by randomly changing parts of the 227×227 frames. We replace them with part of a randomly choosen image. We chose not to do more than 90% of the frames and not less than 5% in order to clearly see the frauded part. We arrive to 1 million 227×227 frame, 500.000 tampered and 500.000 genuine. We thought that with more image, the network will better generalise the tampered part. However, as we only apply copy-move manipulation, it might generalize that type of image manipulation too much and not learn the other type of fraud manipulation.

3 Pre-Processing

3.1 Error Level Analysis

Error Level Analysis (ELA) is based on JPEG compression. The latter involves removing information that it determines unnoticeable to the human eye. This is done with the discrete cosine transform (DCT) algorithm. Each time an image is saved in the JPEG format, a certain amount of information is lost and can never be retrieved. This loss of information is the error level. After compression, on the same image, similar surfaces, textures, patterns, and so on should also have similar error levels. If the error level differs significantly between different parts of the same image, this suggests that parts of it have been edited.

To find the tampered parts, we apply JPEG compression with the quality loss that we want, we empirically choose 90% and we calculate the difference of the first image and the compressed one. In the figure 1 we applied ELA on a manipulated receipt, with a quality loss of 90 %.

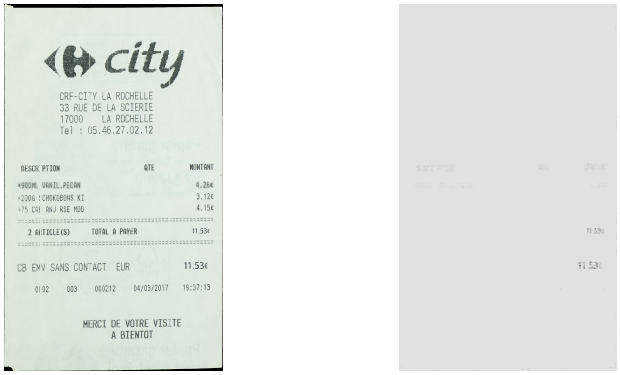

Fig. 1. left: A manipulated receipt, right: ELA on the receipt (inverse coloring for better visibility).

We can see that ELA found two sure manipulated places (two black visible places) and he is right. However, he missed two other tampered frame on this image and on other more difficult images he does not find anything.

We decided that the compression difference could still be a valuable information for our network. We choose a quality compression of 90% because more would have been to much information loss.

3.2 Discrete Wavelet Transform

We used the first invented Discrete Wavelet Transform (DWT), Haar wavelets. It was invented by Hungarian mathematician Alfréd Haar.

The key issues in DWT and inverse DWT are signal decomposition and reconstruction, respectively. The basic idea behind decomposition and reconstruction is low-pass and high-pass filtering with the use of down sampling and up sampling respectively. The result of wavelet decomposition is a hierarchically organized decompositions. One can choose the level of decomposition j based on a desired cutoff frequency. Wavelet functions can be thought of as a bandpass filter bank, the wavelet transform becomes a decomposition of a signal with this filter bank. Since the wavelets are bandpass, we require the notion of a lowpass scaling function that is the sum of all wavelets above a certain scale j in order to fully represent the signal

The Haar Wavelet Transform is the simplest of all wavelet transforms. Low frequency wavelet coefficients are generated by averaging two pixel values and high frequency coefficients are generated by taking half of the difference of the same two pixels. The four bands images obtained are approximate band (LL), Vertical Band (LH), Horizontal band (HL), and diagonal detail band (HH) as shown in figure 2. The approximation band consists of low frequency wavelet coefficients, which contain significant part of the spatial domain image. The other bands, also called detail bands, consist of high frequency coefficients, which contain the edge details of the spatial domain image (cf figure 2)

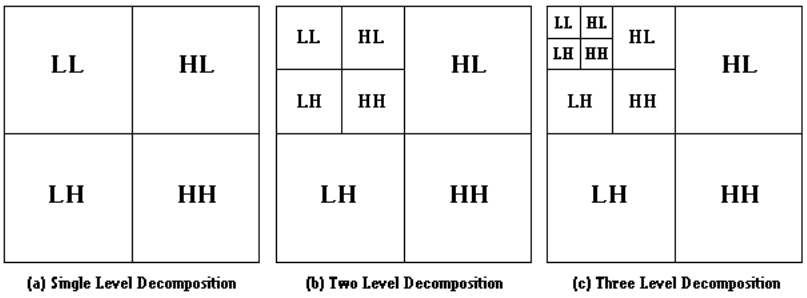

Fig. 2. Haar Wavelet[6].

Let's explain Haar-wavelet using the following vector:

$$x_{n,i} = \{70\ 56\ 61\ 49\} \tag{1}$$

The transformation replaces the sequence with its pair wise average $x_{n-1,i}$ and difference $d_{n-1,i}$ defines follows:

$$x_{n-1,i} = \frac{x_{n,2i} + x_{n,2i+1}}{2} \tag{2}$$

$$d_{n-1,i} = \frac{x_{n,2i} - x_{n,2i+1}}{2} \tag{3}$$

We form a new sequence having length equal of the original sequence by concataning the two sequences x_{n-1} and d_{n-1} as follow:

$$\{x_{n-1,i}, d - 1_{n,i}\} = \{63\ 55\ 7\ 6\} \tag{4}$$

We extend the one-dimensional Haar-wavelet transform in two dimensions.

$$\begin{bmatrix} 70 & 56 & 61 & 49 \\ 52 & 46 & 39 & 43 \\ 63 & 45 & 46 & 54 \\ 53 & 39 & 40 & 44 \end{bmatrix}$$

Fig. 3. Example on a matrix

We apply one dimensional Haar-wavelet in each row and each column.

$$\begin{bmatrix} 63 & 55 & 7 & 6 \\ 49 & 41 & 3 & -2 \\ 54 & 50 & 9 & -4 \\ 46 & 42 & 7 & -2 \end{bmatrix} \qquad \begin{bmatrix} 56 & 48 & 5 & 2 \\ 50 & 46 & 8 & -3 \\ 7 & 7 & 2 & 4 \\ 4 & 4 & 1 & -1 \end{bmatrix}$$

Fig. 4. One dimensional Haar-Wavelet in left: each row, right: each column

This matrix is the result of 2D-dimensional Haar-wavelet for the first level. We can apply successively this transform to obtain more level.

The advantage of wavelet compression is that in contrast to JPEG, wavelet does not divide image into blocks but analyze the whole image.

3.3 Local Binary Patterns

Local Binary Patters (LBP) was first described in 1994 [5], it has since been found to be a powerful feature for texture classification. The idea is to threshold the neighborhood of each pixel and consider the result as a binary number. Let's

$$\begin{pmatrix} 15 & 200 & 115 \\ 27 & \mathbf{70} & 24 \\ 213 & 5 & 60 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 1 & \mathbf{x} & 1 \\ 0 & 1 & 1 \end{pmatrix} \mathcal{P} \begin{pmatrix} 2^0 & 2^1 & 2^2 \\ 2^3 & \mathbf{x} & 2^4 \\ 2^5 & 2^6 & 2^7 \end{pmatrix} \longrightarrow \begin{pmatrix} 15 & 200 & 115 \\ 27 & \mathbf{217} & 24 \\ 213 & 5 & 60 \end{pmatrix}$$

Fig. 5. We first compare the reference pixel to each of his neighboor, if the neighboor is greater we replace it by a 0 and if it is lower or equal we replace it by a 1. We use $\mathcal{P}$ for the ponderation operator, to transform the matrix into a binary number, $(11011001)_2 = 217$

explain on a grayscale (3×3) matrix, given in figure 5. We take a reference center pixel (here value 70) and its neighbourhood:

LBP is a feature extrator used in image manipulation detection, if you want a more detail explaining please refer to Image Inconsistency Detection Using Local Binary Pattern [3].

Principal component analysis PCA: It is used to reduce the dimensions of a dataset, it reduces the data down stripping away unnecessary parts.

Let's say we have a set of points, we can represent into some space. We could deconstruct the set into eigenvectors and eigenvalues. The eigenvector is a direction, and the eigenvalue is a number, telling you how much variance there is in the data in that direction. We then remove the vector with the lowest variance and rearange the data with the other vector, to gain one dimension.

We can use PCA for images, if we consider a 100×100 image as a very long 1D vector (10000) by concatenating image pixels column by column. The length of that vector is the dimensionality of our vector space. With PCA we can remove some information, keeping the important ones.

A more detail approache is explained in the article DWT-PCA (EVD) based copy-move image forgery detection [8]

4 Deep Learning

4.1 AlexNet

AlexNet was created in 2012 to win the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge). We used it because it is a small network and as we have few training datas, we thought it was better to use a small network. We now explain quickly each layers used on this network, for a more detail approch, refer to the article ImageNet Classification with Deep Convolutional Neural Networks (2012) [4]

Convolution Layers: we can see a convolution layer (conv in figure 6) as a filter, let's see on an example. The primary purpose of convolution is to extract features from the input image. Let's see how convolution works between a small matrix (5×5) and a filter (3×3):

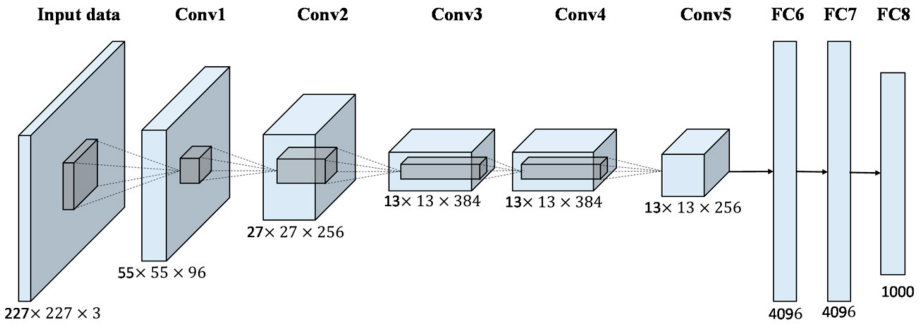

Fig. 6. AlexNet Architecture[4].

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 5 & \mathbf{6} & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 & 24 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{30} & 35 & 40 \\ 55 & 60 & 65 \\ 80 & 85 & 90 \end{pmatrix}.$$

Fig. 7. Convolution example, $*$ represents the convolution product .

We apply this filter to the pixel 6, position (2,2), calculation is done as follow: $0 \times 1 + 1 \times 0 + 2 \times 1 + 5 \times 0 + 6 \times 1 + 7 \times 0 + 10 \times 1 + 11 \times 0 + 12 \times 1 = 30$, it will be the final value in the matrix.

A convolution layer is an application of different filters, which is learned and gets better at finding features usefull for image manipulation detection.

Max Pooling: A pooling layer is also apply after each convolution. Pooling allows us to reduce the dimension of feature vector without losing essential informations. In the 8 is depicted the easiest type of pooling, Max-Pooling. It just consists to take the maximum of a neighboorhood (here 4). We get an image 2×2 from an image 4×4.

Dense Layers: Dense Layers or Fully Connected (FC in figure 6) is a neural network (multi-layer perceptron), fully connected from one layer to an other.

FineTuning: The idea of finetuning is to retrieve the weigth of the well known image classification network, AlexNet. The weigth we took (free on internet) are image classification weigth, trained on ImageNet on 10000 classes. We just adapted the last fully connected (fc8) to classify into two classes, frauded and genuine. The first few layers of AlexNet are convolutions layers, it detects edges, corners, lines, etc. We thought it could be useful in manipulation detections too, so we kept the weight of the convolutions layers (at first all of them and then only the first three) and we trained on the fully connected layers.

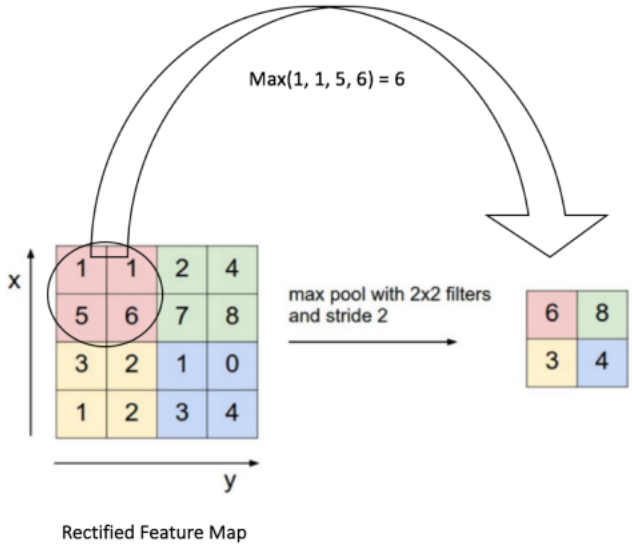

Fig. 8. MaxPooling.
[7]

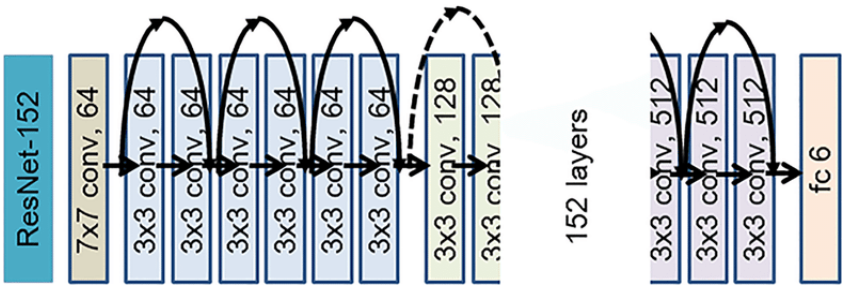

Fig. 9. ResNet152 Architecture, the dotted shortcuts increase dimensions [1].

4.2 ResNet

After we tested AlexNet, we tried to get better results using a more recent Network, Resnet152. The architecture is depicted in 9.

Architecture It is quite the same as AlexNet, one difference is the number of layers, we have 152 Layers instead of 8 for Alexnet. The other difference is that we use a shortcut connection every two layers and add it before the Relu layers. As depicted in figure 10, every two layers we add the prior result to the new one, we do that to preserve information across layers, not to forget about thing

we saw before. To learn more about this techique,you can read Deep Residual Learning for Image Recognition [2]

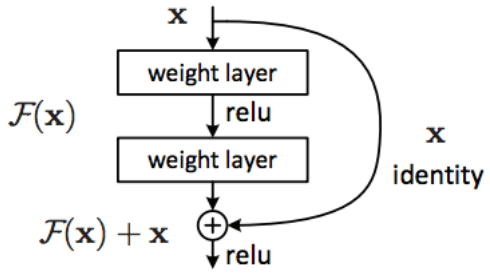

Fig. 10. Residual Learning: a building block [2].

4.3 Training

AlexNet was made for RGB image, thus the $(227 * 227 * 3)$ as input data in figure 6 (3 dimensions). As RGB did not gave good enough result 1 we decided to feed the network with our methods as input.

Both networks were trained in the same way, with a exponential decay learning, starting to 0.1, changing every 500 step with a batchsize of 100. We optimized them with an usual AdamOptimizer. The idea was to combined methods (which could be combined) and to train and test them to see which one achieved the best results. All learning were done with a graphic card Titan X Pascal. About 24 hours were needed to fully trained our best network on one (227×227) million images and 12 hours without Fraud Creation, on 200.000 images.

5 Results

The accuracy is simply calculated with a basic test on 90 frauded images and 100 genuin images, as we had about 30 test images, we flipped them twice. The test were done many times and the mean of the results is written in table 1

Our idea was to seperate the test image into small 227×227 frames, and to test every frame. If the network detects one of the small frame to be tampered, the global image is tampered. That only works if the algorithm is perfect (100 % accuracy), we decided to add a threshold (t) to the probability of a window not to be manipulated.

$$\Pr(Frauded) > Pr(Genuine) + t$$

The network now detects a manipulated frame only if he is sure of it, if the probability of fraud is more than the probability of non fraud $+t$.

However t is not the same for every methods and network, it strongly depends on the training. We can empiricaly say that for every test, we had better result with a strong threshold: $t > 0.7$. The better the network gets, the lower the threshold will be.

When a manipulation is detected in a frame, we flip and rotate the frame and test it again, just to be sure that the manipulation is truly there. All Result are given in table 1, we achived about 80 % combining ELA, Wavelet and Grayscale image, on 200.000 size 227×227 images and 90 % on 1 million frames. We can notice that ResNet152 performs better than AlexNet.

6 Conclusion

In this paper, we have presented techniques to deal with manipulation in receipt. The methods and networks we used are easily adaptable to manipulation detection in every types of image, document, bills, and could have a lot of everyday applications. Mixing imaging technique then deep learning, helped us to achieve more than 80% accuracy with a dataset of only 100 images. We can noticed that creating copy-move tampered images significantly improved our accuracy by achieving 90% accuracy, training is however longer.

To improve the generalisation of our model and get better results, we might have created different sort of image manipulation.

Aknowledgments

Ths research was funded by a "Ressourcement S3" project from Region Rhône-Alpes Auvergne, FEDER and Europe.

References

1. Han, S.S., Park, G.H., Lim, W., Chang, S.E.: Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network (2018)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015)
3. H.MahaleaMouad, M.H.AlibPravin, L.YannawarcAshok, T.Gaikwadd: Image inconsistency detection using local binary pattern (lbp) (2017)
4. Krizhevsky, A., Sutskeve, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Network (2012)
5. Ojala, T., Pietikäinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition. pp. 582–585 (1994)
6. Thakkar, F., Kher, D.R.K., Modi, C., Kher, H.: Selecting most favorable basis function for compressing natural and medical images using wavelet transform coding (2009)

7. ujjwalkarn: An intuitive explanation of convolutional neural networks (2016)
8. Zimba, M., Xingming, S.: Dwt-pca (evd) based copy-move image forgery detection (2011)