A comprehensive analysis of Local Binary Convolutional Neural Network for fast face recognition in surveillance video

Carolina Toledo Ferraz and José Hiroki Saito
UNIFACCAMP
Campo Limpo Paulista, São Paulo
caroltoledoferraz@gmail.com,saito@cc.faccamp.br

ABSTRACT

Recent research in convolutional neural network (CNN) has provided a variety of new architectures for deep learning. One interesting new architecture is the local binary convolutional neural network (LBCNN), which has shown to provide significant reduction in the number of parameters to be learned at training. In this paper, we study the influence of network parameters in the scenario of face recognition, comparing LBCNN against other famous networks available in the literature in terms of sensibility and processing time. In our study, we also propose a pre-processing step on images to increase the accuracy of the model, besides investigating its behaviour with noisy images. Our experiments are carried on the Chokepoint dataset, whose face subimages were collected from video frames under real-world surveillance conditions, including variations in terms of illumination, sharpness, pose, and misalignment due to automatic face detection. The conclusion is that by using the Laplacian step and a reduced amount of LBC modules, it is possible to train LBCNN more quickly and with improved accuracy. In addition, it was found that LBCNN is very sensitive to noise and better results can be achieved when noisy images are inserted in the training set.

CCS CONCEPTS

• **Computing methodologies → Object recognition**; *Object identification*; Visual content-based indexing and retrieval;

KEYWORDS

Local Binary Pattern, Convolutional neural network, Face recognition

ACM Reference Format:

Carolina Toledo Ferraz and José Hiroki Saito. 2018. A comprehensive analysis of Local Binary Convolutional Neural Network for fast face recognition in surveillance video. In *Brazilian Symposium on Multimedia and the Web (WebMedia '18), October 16–19, 2018, Salvador-BA, Brazil.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3243082.3267444

WebMedia '18, October 16–19, 2018, Salvador-BA, Brazil

ACM ISBN 978-1-4503-5867-5/18/10.
https://doi.org/10.1145/3243082.3267444

1 INTRODUCTION

Face recognition has been used in many applications like security, forensic and commercial applications [4, 15, 19]. In video, face images are captured over multiple frames in different conditions, such as illumination, motion blur, shadowing, expressions, occlusions and poor quality CCTV video recordings [29]. Due to the fact that the number of surveillance cameras installed in public places is increasing, it is important to build robust video-based face recognition systems [3].

A lot of methods are used for face recognition. Among them there are descriptors that use the approach of local features [11, 24]. Local binary pattern (LBP) [17] is one of them and became famous in classifying texture in digital images and was used in many applications including face recognition [1, 14]. Nowadays modern approaches use local descriptors along with convolutional neural networks showing best performance among existing binary descriptors [27].

Convolutional neural networks (CNN) are a category of neural networks that have proven to be very effective to analyzing visual imagery [26]. Over the years a lot of architectures have been developed to solve different types of problems: Alexnet [16], VGG [23], Inception [25] and Resnet [12]. Some recent works that use CNN for face recognition are: CASIA WebFace [30], FaceNet [22] and VGGFace2 [5]. The main problem of these architectures is that training one network from scratch is computationally very expensive and it is necessary GPU-based machines [6]. To address these drawbacks, some binary versions of CNNs have been presented in the literature [8, 20] that aproximate the dense real-valued weights with binary weights [13]. Rastegari et al. [20] proposed an efficient approximation to standard convolutional neural networks that uses mostly bitwise operations to approximate convolutions. The authors showed that the method require a significantly less memory and reduced the convolutional operations. Juefei-Xu et al. [13] proposed a hybrid combination of fixed and learned weights introducing a local binary convolutional layer (LBC). The authors state that CNNs with LBC layers, called local binary convolutional networks (LBCNN) have lower model complexity and are less prone to over-fitting. The methodology achieved best performance than regular CNNs in four datasets: MNIST, SVHN, Cifar-10 and Imagenet, but the authors have not tested on face datasets.

In this paper, we investigate the application of the Local Binary Convolutional Neural Networks [13] for face recognition. We conducted three studies: 1) we study the influence of network parameters in order to accelerate the training processing; 2) we propose a pre-processing step on images to increase the accuracy of the model; 3) we investigate the network behaviour with noisy images. We used the Chokepoint Dataset that are images taken from video frames and the localization of the faces has some problems like: illumination variations, pose, sharpness and misalignment due to automatic face detection. We compared the results to three other networks: Baseline described in [13], Alexnet and Resnet.

This paper is structured in more four sections, besides this introduction. Section 2 presents the Local Binary Convolutional neural networks which is the inspiration of this work. The methodology is discussed in Section 3, and the experiments are shown in Section 4. Finally, Section 5 concludes our work with the final remarks.

2 LOCAL BINARY CONVOLUTIONAL NEURAL NETWORKS (LBCNN)

A new research field is emerging, using binary weights in CNNs to reduce the network complexity [8, 9, 13, 20].

Juefei-Xu et al. [13] have recently developed a network that was inspired by the famous texture descriptor LBP and presents a good approximation of a standard learnable convolutional layer. The idea of the method is to use LBC layers in convolutional neural networks as follows: 1) the first layer is a set of fixed sparsed pre-defined binary convolutional layer, followed by a non-linear activation function layer (sigmoid or ReLU), 2) the last layer is a set of learnable $1x1$ convolutional weights. The level of sparsity is defined by the user that indicates the percentage of non-zero value weights of binary convolutional layer. Then the first layer is initialized through Bernoulli distribution with 0, 1 and -1 randomly using the sparsity. Assume that LBC has m pre-defined non-learnable binary filters and p 1×1 convolution filters. The input image is filtered by m binary filters resulting in m difference maps. These maps are changed to m bit maps by a non-linear activation layer. Lastly, the m bit maps are linearly combined using the p learnable 1×1 weights to approximate the traditional convolutional layer. The weights in first binary convolutional layer are fixed while the weights in the second convolutional layer are learnable. As the method has sparsity in weights and the fixed binary values, it becomes less representative compared to the typical convolutional layer. To achieve the similar performance with a traditional convolutional layer at each LBC, it needs a great number of local binary filters (512 in [13]) and 1×1 learnable weight.

These networks were tested on datasets like: MNIST, Cifar-10 and SVHN and were not tested on faces datasets. Therefore the objective of this paper is to study LBCNN for face recognition, analyzing its parameters and the behavior of the network with the insertion of noisy images, and to propose a pre-processing step on images to increase the accuracy of the model.

3 METHODOLOGY

In this paper, we analyse the behaviour of the network described in the Section 2 in the face recognition scenario. We modify some parameters of the network in order to decrease the computational processing time maintaining the same accuracy and compare to Alexnet, Resnet 34 and the Baseline described in [13]. After that, we applied a pre-processing step on images in order to increase the accuracy of the network result and analyze the behavior of the methodology with noisy images.

The parameters involved in the network are: filter size, number of LBCNN modules (we call as depth), LBC filters (we call as numWeights), number of intermediate channels (we call as numChannels), sparsity (we call as sparsity) and hidden units in fully connected layer (we call as full). The filter size was defined as 3×3.

3.1 Dataset

We consider classification task on Chokepoint Dataset [29]. The cropped face images were extracted from three cameras placed above several portals and have variations in terms of illumination conditions, pose, sharpness, as well as misalignment due to automatic face localization/detection. The dataset consists of 48 video sequences and 64.204 face images. The faces provided by the authors of the database have 96×96 pixels in gray color, however for our tests the images were resized to 32×32 pixels. The images were divided in two groups: training and test. We used 20044 faces for training and 4595 for test.

Some sample images of the dataset are illustrated in the Figure 1.

Figure 1: Samples of Chockepoint Dataset

3.2 Experimental Settings

We took random images and calculated the mean and standard deviation for them. The mean value found was subtracted for each pixel for all images in the train and test set. In addition each image was divided by the standard deviation. The values used in this experiment were: a) mean: 0.328 and b) standard deviation: 0.136.

We run all our experiments using 50 epochs. All the implementation used in this paper was based on Torch [7]. We run the experiments on CPU i7, 16GB RAM and GeForce GTX TITAN XP.

4 EXPERIMENTS

The experiments were divided into three parts: 1) analysis of the LBCNN parameters in order to reduce the training time of the network (subsection 4.1); 2) pre-processing of the input images of the network in order to increase the accuracy (subsection 4.2); 3) analysis of the network behavior by inserting noise images into the training and test set (subsection 4.3).

4.1 Changing parameters

In the first experiment, we fixed the following parameters: numChannels: 16, depth: 75 and full: 512 and varied the numWeights and sparsity. Table 1 shows the best Top 1 accuracy for the test set changing the sparsity and numWeights parameters. We can note that for face recognition dataset the best result found was with numWeights 512 with sparsity 0.1.

Table 1: Classification accuracy with different numWeights parameter

Sparsity	0.1	0.5	0.9
numWeights: 32	99.13%	99.23%	99.39%
numWeights 64	99.00%	99.28%	99.30%
numWeights 128	99.34%	99.47%	99.37%
numWeights 512	**99.56%**	99.45%	99.47%

Using the best result found in Table 1, we show in Figure 2 the average loss for training and validation set. From this plot we can see that the model has comparable performance on both train and validation datasets.

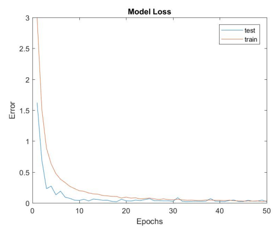

Figure 2: Plot of Model Loss on Training and Validation Datasets

In the second experiment we fixed the following parameters: numChannels: 16, full: 512 and NumWeights: 512 and decreased the depth parameter (which indicates the amount of LBCNN modules in the network). The more LBCNN modules, the slower the processing. Table 2 shows the best Top 1 accuracy for the test set changing the sparsity and depth parameters. We can note that the best accuracy is superior to the accuracy found in the Table 1 along with a shorter processing time (caused by the decreasing of the depth parameter) as can be seen in the Table 3.

Table 2: Classification accuracy with different depth parameter

Sparsity	0.1	0.5	0.9
depth 50	99.56%	99.58%	99.43%
depth 40	99.47%	99.28%	99.50%
depth 30	99.30%	99.65%	99.50%
depth 20	99.58%	99.45%	99.63%
depth 10	99.47%	99.56%	**99.67%**
depth 5	99.30%	99.63%	99.58%
depth 3	99.32%	99.45%	99.50%
depth 1	99.54%	97.80%	97.91%

Table 3 shows the processing time (in seconds) of each epoch in the training set and test set using the best configurations found in the Tables 1 and 2. We can notice a decay of 7 times in processing time. Therefore for the next experiments, we used the depth parameter set by 10.

Table 3: Processing time (in seconds)

Configurations	Training Set	Test Set
Table 1	164	14
Table 2	**23**	**2**

Since the depth parameter is responsible by the number of LBC modules in the network, we can observe that decreasing the depth number did not decrease the accuracy (Tables 1 and 2). We also notice that in the last line of Table 2 the accuracy was very close to the accuracy of Table 1, but with only one LBC module. The processing time (showed in Table 3) decrease the processing time by 7 times.

4.2 Pre-processing images

Pre-processing images in the CNN methodology has already been studied both to improve the accuracy of the model [18] and to enhance the quality of degraded images [31]. In this experiment we pre-processed the images using two different functions to improve the accuracy found in Tables 1 and 2.

We tested two functions separately: Sigmoid and Laplacian functions. Sigmoid function is useful for increasing the images contrast [28] and Laplacian function can be used to localize edges [2].

We fixed the following parameters: numChannels 16, full 512 and numWeights 512. The parameters used for the Laplacian function were: kernel size 3×3 and $sigma = 0.25$. Table 4 shows the test set accuracy using Laplacian and Sigmoid functions. We can note that the accuracy increased by 0.15% and 0.04% in relation to the accuracy found in Tables 1 and 2 respectively. The processing time was calculated for the best configuration found in Table 4 and there are no increase in the time compared to the second line of the Table 3.

Table 4: Classification accuracy using Laplacian and Sigmoid function on pre-processing

Sparsity	Laplacian	Sigmoid
depth 10/sparsity 0.1	99.52%	99.34%
depth 10/sparsity 0.5	99.45%	99.54%
depth 10/sparsity 0.9	**99.71%**	99.47%
depth 5/sparsity 0.1	99.06%	99.30%
depth 5/sparsity 0.5	99.60%	99.32%
depth 5/sparsity 0.9	99.43%	99.54%

Table 5 shows the accuracy of the LBCNN, Baseline, Resnet 34 and Alexnet using the best configurations found in Table 4. To ensure a fair comparison and to quantify the exact empirical difference between LBCNN approach and a traditional CNN, Juefei-Xu et al. [13] implemented the exact same architecture for both the networks, albeit with sparse, binary and fixed weights in LBCNN and dense learnable weights for CNN. We also use the same pre-processing used in the LBCNN.

Table 5: Classification accuracy with pre-processing step

LBCNN	Baseline	Alexnet	Resnet
99.71%	99.13 %	91.66%	95.54%

Analyzing the results we can see that the best result was obtained using the Laplacian function that overcame the result found without pre-processing. How Laplacian function returns the edges of the images, this result suggests that perhaps deep neural networks can be designed with less convolutional layers or without convolutional layers.

4.3 Noisy images

Some works have tried to understand how the images quality affects deep neural networks [10, 21]. Our objective in this experiment is to analyze the efficiency of the network inserting noise in the images. For LBCNN we used the following parameters: depth=10, full=512, NumWeights=512 and sparsity=0.1. We compared the LBCNN with other 4 methodologies: Baseline, Alexnet and Resnet 34.

In the first experiment we applyed a Gaussian noisy with mean 0 and standard deviation 1 and multiply the function by a noise factor of 0.5. We used this images only in the test set. The results are shown in Table 6.

Table 6: Classification accuracy using noisy images only in test set

LBCNN	Baseline	Resnet	Alexnet
54.15%	60.97%	**93.56%**	83.13%

In the second experiment we inserted the noisy images in the training. We calculated the classification accuracy in the test set and show the results in Table 7. Analyzing the results we can see that in the first test, LBCNN lost to all compared CNNs. In the second test, where the noisy images were inserted in the training set, the LBCNN had an accuracy higher than the other architectures. We can conclude that the LBCNN is very sensitive to noise and we can get better results when the noisy images are inserted in the training set.

Table 7: Classification accuracy using noisy images in the training set

LBCNN	Baseline	Resnet	Alexnet
97.34%	93.52%	95.65%	90.40%

5 CONCLUSIONS

Along with this study we conclude that despite the images have low resolution and the faces have different conditions, LBCNN had a good performance in terms of the processing time and accuracy and can be considered a promising network for several applications in the field of computer vision.

ACKNOWLEDGMENTS

The authors would like to thank NVidia for GPU donation and Capes for financial support.

REFERENCES

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. 2004. Face Recognition with Local Binary Patterns. In *Computer Vision - ECCV 2004*, Tomás Pajdla and Jiří Matas (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 469–481.

[2] Reshma Suresh Babu, B. Radhakrishnan, and L. Padma Suresh. 2016. Detection and extraction of roads from satellite images based on Laplacian of Gaussian operator. In *2016 International Conference on Emerging Technological Trends (ICETT)*. 1–7. https://doi.org/10.1109/ICETT.2016.7873747

[3] Saman Bashbaghi, Eric Granger, Robert Sabourin, and Mostafa Parchami. 2018. Deep Learning Architectures for Face Recognition in Video Surveillance. *CoRR* abs/1802.09990 (2018). arXiv:1802.09990 http://arxiv.org/abs/1802.09990

[4] Moulay Smail Bouzakraoui, Abdelalim Sadiq, and Nourddine Enneya. 2017. A Customer Emotion Recognition Through Facial Expression Using POEM Descriptor and SVM Classifier. In *Proceedings of the 2Nd International Conference on Big Data, Cloud and Applications (BDCA'17)*. ACM, New York, NY, USA, Article 80, 6 pages. https://doi.org/10.1145/3090354.3090436

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. 2017. VGGFace2: A dataset for recognising faces across pose and age. *CoRR* abs/1710.08092 (2017). arXiv:1710.08092 http://arxiv.org/abs/1710.08092

[6] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu. 2017. Quantized CNN: A Unified Approach to Accelerate and Compress Convolutional Networks. *IEEE Transactions on Neural Networks and Learning Systems* (2017), 1–14. https://doi.org/10.1109/TNNLS.2017.2774288

[7] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*.

[8] Matthieu Courbariaux and Yoshua Bengio. 2016. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. *CoRR* abs/1602.02830 (2016). arXiv:1602.02830 http://arxiv.org/abs/1602.02830

[9] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. *CoRR* abs/1511.00363 (2015). arXiv:1511.00363 http://arxiv.org/abs/1511.00363

[10] S. Dodge and L. Karam. 2016. Understanding how image quality affects deep neural networks. In *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. https://doi.org/10.1109/QoMEX.2016.7498955

[11] E. F. Ersi and J. S. Zelek. 2006. Local Feature Matching For Face Recognition. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. 4–4. https://doi.org/10.1109/CRV.2006.48

[12] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[13] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. 2017. Local Binary Convolutional Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 4284–4293. https://doi.org/10.1109/CVPR.2017.456

[14] Idelette Laure Kambi Beli and Chunsheng Guo. 2017. Enhancing Face Identification Using Local Binary Patterns and K-Nearest Neighbors. *Journal of Imaging* 3, 3 (2017). https://doi.org/10.3390/jimaging3030037

[15] Christian Kraetzer, Andrey Makrushin, Tom Neubert, Mario Hildebrandt, and Jana Dittmann. 2017. Modeling Attacks on Photo-ID Documents and Applying Media Forensics for the Detection of Facial Morphing. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '17)*. ACM, New York, NY, USA, 21–32. https://doi.org/10.1145/3082031.3083244

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[17] T. Ojala, M. Pietikainen, and T. Maenpaa. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 7 (Jul 2002), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623

[18] K. K. Pal and K. S. Sudeep. 2016. Preprocessing for image classification by convolutional neural networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*. 1778–1781. https://doi.org/10.1109/RTEICT.2016.7808140

[19] Raghavendra Ramachandra and Christoph Busch. 2017. Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey. *ACM Comput. Surv.* 50, 1, Article 8 (March 2017), 37 pages. https://doi.org/10.1145/3038924

[20] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *ECCV*.

[21] S. S. Roy, M. Ahmed, and M. A. H. Akhand. 2017. Classification of massive noisy image using auto-encoders and convolutional neural network. In *2017 8th International Conference on Information Technology (ICIT)*. 971–979. https://doi.org/10.1109/ICITECH.2017.8079976

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR* abs/1503.03832 (2015). arXiv:1503.03832 http://arxiv.org/abs/1503.03832

[23] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)* (2015).

[24] Sima Soltanpour, Boubakeur Boufama, and Q.M. Jonathan Wu. 2017. A survey of local feature methods for 3D face recognition. *Pattern Recognition* 72 (2017), 391 – 406. https://doi.org/10.1016/j.patcog.2017.08.003

[25] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[26] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. 2016. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging* 35, 5 (May 2016), 1299–1312. https://doi.org/10.1109/TMI.2016.2535302

[27] Y. Tian, B. Fan, and F. Wu. 2017. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6128–6136. https://doi.org/10.1109/CVPR.2017.649

[28] S. Widyarto, S. R. B. Kassim, and W. K. Sari. 2017. 2D-sigmoid enhancement prior to segment MRI Glioma tumour: Pre image-processing. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. 1–5. https://doi.org/10.1109/EECSI.2017.8239103

[29] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. 2011. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, 81–88.

[30] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. 2014. Learning Face Representation from Scratch. *CoRR* abs/1411.7923 (2014). arXiv:1411.7923 http://arxiv.org/abs/1411.7923

[31] Jonghwa Yim and Kyung-Ah Sohn. 2017. Enhancing the Performance of Convolutional Neural Networks on Quality Degraded Datasets. *CoRR* abs/1710.06805 (2017). arXiv:1710.06805 http://arxiv.org/abs/1710.06805