Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 152 (2019) 74–83

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Pervasive Computing Advances and Applications - PerCAA 2019

XNORNet and Minimum Barrier Detection for Efficient Face Recognition.

Rahul R Bharadwaj[a], Sagar Belavadi[a], Sameer Gadicherla[a], Vinay A[a], S Natarajan[a], K N Balasubramanya Murthy[a]

[a]*PES University, 100 Feet Ring Road, BSK III Stage, Bangalore -560085*

Abstract

Deep Convolutional Neural Networks (CNNs) have been extremely effective in many areas of computer vision, particularly in the domain of facial recognition where many CNN models have delivered a near perfect performance on long-standing benchmarks. A major challenge being faced by CNN architectures today is their need for a large amount of memory and processing power to deliver their exceptional performance. This hampers their deployment on less powerful devices like smart-phones whose ubiquity is unmatched. This paper critically analyzes a CNN based network architecture wherein convolutions are approximated by XNOR operations and bit-counts which are primarily binary operations hence making the network much faster and more memory efficient than standard CNN architectures. The aim of this paper is to assess the performance of the aforementioned architecture for the task of facial recognition with Minimum Barrier Salient Object Detection (MBD) as a preprocessing step and to examine the affect self-normalizing properties induced by the Scaled Exponential Linear Unit (SELU) activation function on the convergence of the network and benchmark it against the convergence performance of the same network incorporating the widely used Rectified Linear Unit (RELU) as its activation function. Two widely known datasets, Faces95 and Faces96, pre-processed using Minimum Barrier Detection (MBD) for saliency detection were used benchmarking the performance of the proposed networks. It was observed that the network using SELU converged faster than RELU for the Faces95 dataset and both converged at similar rates for Faces96 dataset. The network with SELU gave better accuracy than RELU for both the datasets.

Keywords: Face Recognition; XNORNet; Minimum Barrier Saliency Object Detection;

1. Introduction

Recent developments in deep learning and Convolutional Neural Networks(CNNs) has resulted in appreciable performance improvement for solving a number of computer vision problems, like object tracking [1], recognition [2] and detection [3]. Deep models have been outstanding in the field of face recognition where many models perform very well, scoring highly on long-standing and well known benchmarks [4]. On the other hand, CNNs perform well only on high performance GPUs which provide a huge amount of memory bandwidth [5] and high computational performance of billions of MACs(Multiply Accumulate operations) per second. This requirement has become one of the crucial challenges which hinders the implementation of CNN based solutions on less powerful but pervasive devices like smart-phones and embedded systems without performance degradation.

10.1016/j.procs.2019.05.029

1.1. Motivation behind choosing XNORNet

The current era of computing is vested in hand held devices. This has led to an increased demand for more secure and faster but efficient security solutions. Thus face recognition has gained an increased popularity due to it's high accuracy and ease of automation. This has resulted in an increased research for exploring accurate yet efficient neural network models like XNORNet. XNORNet is an efficient CNN based architecture [7]. Typically, in standard CNNs architectures, convolution is comprised of addition, subtraction and multiplication. In XNORNet, convolution is approximated with XNOR and bit-counts which are primarily binary operations. Due to this, thirty two times lesser memory is used and convolution operations are performed fifty eight times faster which brings about a tremendous improvement in performance. Therefore, this improvement removes the requirement of high-performance GPUs to implement this network making it a very attractive alternative to standard CNNs.

This paper explores the convergence performance of XNORNet with the recently introduced activation function, Scaled Exponential Rectified Linear Unit(SELU) and compares it with convergence performance using the well known Rectified Exponential Linear Unit(RELU) activation function. Although architectures like CNN and Recurrent Neural Networks (RNN) have been successful with deep learning, the same success has been uncommon with the standard feed-forward neural networks(FNNs). FNNs which perform well cannot make use of many levels of abstract representations because of their shallow nature.

1.2. Motivation behind exploring SELU as an activation function

Activation function defines the output upon being applied to the input of a neuron (or a node) in a neural network. A good activation function can generate effective yet non-linear and non-overfit mappings from inputs to outputs.

To enable many high levels of abstract representations, the SELU activation function was introduced which creates Self-Normalizing Neural Networks(SNNs) by inducing self-normalizing properties in FNNs. The property of convergence has resulted in training of deeper networks, with robust learning [6] and strong regularization.

In this paper we explore if the SELU activation function has a similar positive impact on the performance of the CNN based XNORNet architecture as it did for SNNs and to this end, benchmark its performance against that of RELU activation function on the same XNORNet architecture.

The publicly available and well known datasets Faces95 and Faces96 have been used for experimentation. These datasets were pre-processed using Minimum Barrier Detection(MBD) for saliency detection.

2. Literary Survey

Due to it's widespread applications, from security (as a password for handheld devices like iPhone X, as a replacement for ATM PIN in countries like China etc.) to healthcare and marketing, Face recognition has become a very important technique. Over the years, many new techniques have been invented to make face recognition faster and more efficient. We now look at the various significant methods of face recognition.

2.1. Ridgelet Transforms(RT)

Ridgelet transform was used for Face recognition in [14]. First, the training and testing images are subjected to a combination (in sequence) of radon and wavelet transforms. To perform recognition, Euclidean distance is calculated between testing and train feature vectors. The color model YCbCr is used for normalizing the images by using segmentation. Then transform is applied onto images. The Ridgelet transform is equivalent to applying a 1d wavelet transform to the projections of Radon transform:

$$CRT_f(a,b,\theta) = \int_{\mathbb{R}^2} \psi_{a,b}(t) R_f(\theta,t)dt \tag{1}$$

where CRT is the Continuous Ridgelet Transform defined as:

$$CRT_f(a,b,\theta) = \int_{\mathbb{R}^2} \psi_{a,b,\theta}(\mathbf{x}) f(\mathbf{x})dx \tag{2}$$

2.2. Local Binary Pattern Histograms(LBP)

LBP was introduced in [12]. As the paper describes, the area corresponding to a face in an image is first partitioned into small regions which are used to obtain LBP Histograms. A single and a spatially enhanced feature histogram is obtained by concatenating the LBP histograms. Using nearest neighbour classifier and a dissimilarity measure like

Additive Chi square the recognition is performed. The spatially enhanced histogram H for a labeled image $f_l(x, y)$ is defined in the paper is:

$$H_{i,j} = \sigma_{x,y} I\{f_l(x, y) = i\} I\{(x, y) \in R_j\} \tag{3}$$

where $i = 0, ..., n - 1$ and $j = 0, ..., m - 1$. $R_0, R_1, ..., R_{m-1}$ represent the regions in the histogram.

2.3. Linear Discriminative Analysis(LDA)

Face recognition using LDA was done in [13]. It is a generalization of a method used prominently in machine learning and pattern recognition called Fisher's linear discriminant.The basic idea behind LDA is to find a linear combination of features which can characterize the required classes. The resulting combination is usually used for dimensionality reduction of the feature space before classification, but, can also be utilized as a linear classifier.

3. Approach

3.1. Overview

Each image from the dataset was first resized to a smaller scale and then pre-processed using Minimum Barrier Detection (MBD) transform [7]. MBD transform is used to detect salient objects. The pre-processed image is then passed into the neural Network (network architecture is shown later), containing XNOR convolution and Dense layer [6], where the networks activation function can be either SELU (Scaled Exponential Linear Unit) or RELU (Rectified Exponential Linear Unit). The last step is classification, to get the appropriate class label for a given image from the testing dataset.

3.2. Block Diagram

The approach explained above has been depicted in a pictorial fashion in Fig 1

3.3. Minimum Barrier Salient Object Detection

The objective of salient object detection[10] of outstanding objects is to calculate a prominence map that accentuates outstanding objects and masks the background [9] in an image. This transform has been well appreciated in research for being very useful in various computer vision applications from object detection to many video and image processing applications. Applications in hand-held devices demand high quality salience maps from an expedient outbound object detection method, at more computationally efficient, lesser space and power using methods, thus making this method desirable for such situations. We specifically used the FastMBD transform algorithm, specified in [8], which uses a Raster Scan and a complementary Inverse Raster Scan. Let x be a pixel in a raster scan or inverse raster scan. Then the cost of path at x can be minimized by using each y in the corresponding neighborhood of x by using the following distance map :

$$D(x) \leftarrow min\{D(x), I(P(y).\langle y, x\rangle)\} \tag{4}$$

In this distance map equation,

(1). β_I is a cost function used in recent times Image Distance Transform, defined as:

$$\beta_I(\pi) = max_{i=0}^{k} I(\pi(i)) - min_{i=0}^{k} I(\pi(i)) \tag{5}$$

where $I(.)$ is a pixel value.

(2). $P(y)$ represents a path assigned to a pixel y and $\langle y, x\rangle$ is an edge from pixel x to pixel y.
Thus $P(y).\langle y, x\rangle$ is a path for x that appends edge $\langle y, x\rangle$ to $P(y)$ Hence, the actual cost function is defined as:

$$\beta_I(P_y(x)) = max\{U(y), I(x)\} - min\{L(y), I(x)\} \tag{6}$$

where $U(y)$ is the highest pixel value and $L(y)$ is the lowest pixel value in $P(y)$. When path assignment changes in each iteration in the algorithm, the value of U and L are accordingly updated. The following algorithm (Algorithm 1) was used to perform MBD transform, it is based on [8].

A sample of how MBD transform works to detect a salient object is shown in Fig 2.

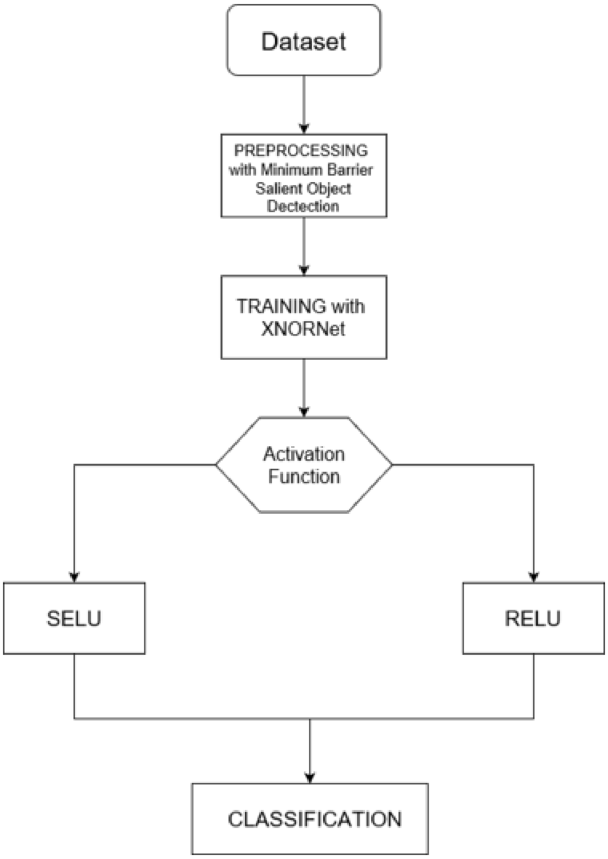

Fig. 1. Face recognition pipeline used in this paper.

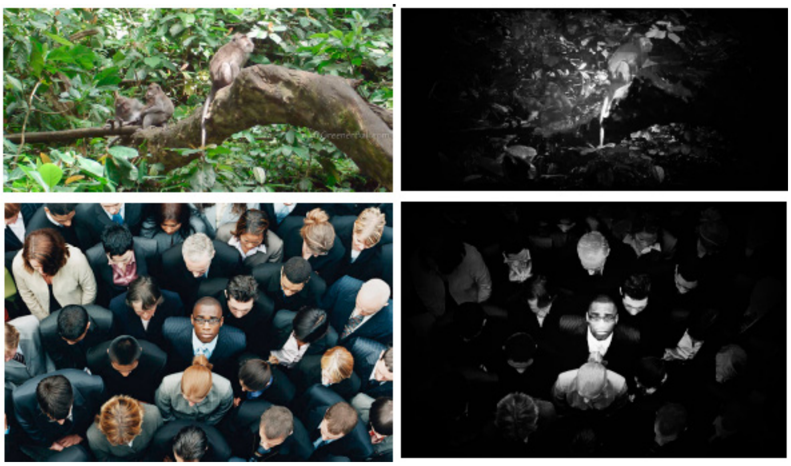

Fig. 2. A sample of MBD transform on an image highlighting salient objects.

3.4. XNOR-Net

XNORNet [6] is a type of CNN where both the dense layers and convolutional layers have binary weights and inputs are also binary values. One major drawback of CNN is the amount of memory and time taken for the convolution operation which is due to the large number of weights and inputs [11]. In XNORNet, efficiency in implementing

```
Result: The distance map $D$ which is a list
Input:Image $I = (i, V)$ and Number of Iterations :
  $iters$;
$L, U \leftarrow I, I$;
set $D(x)$ to inf or to 0;
for $k$ from 0 to $iters$ do
    if $k$ is odd then
        RasterScan($I$,$L$,$U$,$D$);
    else
        InvRasterScan($I$,$L$,$U$,$D$);
    end
end
```

Algorithm 1: MBD transform

```
Result: The distance map $D$ which is a list
Input: $I, L, U, D$ from MBD transform algorithm.;
set $rows$ and $cols$ to number of rows and columns in
  image I;
for $x$ from $rows - 2$ to $1$ do
    for $y$ from $cols - 2$ to $1$ do
        imX,dist = I[x][y],D[x][y];
        h1 = U[x+1][y];
        h2 = U[x][y+1];
        l1 = L[x+1][y];
        l2 = L[x][y+1];
        b1 = max(u1,ix) - min(l1,ix);
        b2 = max(u2,ix) - min(l2,ix);
        if b1<dist and b1<=b2 then
            D[x][y] = b1;
            U[x][y] = max(h1,ix);
            L[x][y] = min(l1,ix);
        else
            D[x][y] = b2;
            U[x][y] = max(h2,ix);
            L[x][y] = min(l2,ix);
        end
    end
end
```

Algorithm 2: Inverse Raster Scan

the convolution operation is achieved through the use of binary inputs and binary weights. Going a step further, XNORNets have binary operands in all convolutions thus leading to well estimated convolutions and bit-counting operations.

Even though XNORNets use binary weights, it generates results which are comparable with CNNs for the same application. As indicated in [6], the results are an accurate approximation of CNNs while offering 58x computation saving and 32x memory saving.

This enables real-time testing in hand-held devices efficiently, even on CPUs, thus allowing for creation of powerful application in these devices which have limited resources. The number of layers in the L-layer CNN used is represented by $\langle I, W, * \rangle$ where $\mathbf{I}$ is the input tensor $\mathbf{I} \in \mathbb{R}^{c \times w_{in} \times h_{in}}$ (where $w_{in} \ll w$ is weights, $h_{in} \ll h$ is heights and c is channels) and $\mathbf{W}$ is the weight filter defined as $\mathbf{W} \in \mathbb{R}^{c \times w \times h}$. Before defining the binary dot product and binary convolution for

XNORNets, it's necessary to define binary dot products for Binary-Weight-Networks. **Binary-Weight-Network** is a type of Binary CNN (like XNORNet) which, unlike XNORNet, has only elements of W as binary tensors.

3.4.1. Binary weights Estimation:

Consider two vectors **W,B**$\in \mathbb{R}^n$ where $n = w \times h \times c$. The following optimization problem needs to be solved in order to find an estimation for $\mathbf{W}$ (where $\mathbf{W} \approx \alpha\mathbf{B}$).

$$\begin{aligned} J(\mathbf{B},\alpha) &= ||\mathbf{W}-\alpha\mathbf{B}||^2 \\ \alpha^*,\mathbf{B}^* &= \underset{\alpha,\mathbf{B}}{\operatorname{argmin}}\, J(\mathbf{B},\alpha) \end{aligned} \tag{7}$$

In **XNORNets** the elements are binary tensors of both I and W. So both weights and inputs have to be converted to binary in order to perform convolution operations. A typical convolution operation consists of a repeated moving of a kernel over an image (shifting) and performing dot products. To approximate convolution it is crucial to define dot products in terms of binary operations (since the network deals with binary weights). Thus the doc product between vectors in $\mathbb{R}^n$ will be approximated to a dot product of two vectors in $\{+1,-1\}^n$ which will approximate convolution.

3.4.2. Binary Dot Product

Consider $\mathbf{X},\mathbf{W} \in \mathbb{R}^n$ to be two vectors in $\mathbb{R}^n$. Approximating a dot product between them in **H,B** for $\mathbf{H},\mathbf{B} \in \{+1,-1\}^n$ should be done such that

$$\mathbf{X}^T\mathbf{W} \approx \beta\mathbf{H}^T\alpha\mathbf{B} \tag{8}$$

where $\alpha,\beta \in \mathbb{R}^+$. Thus the following optimization needs to be solved first:

$$\alpha^*,\mathbf{B}^*,\beta^*,\mathbf{H}_* = \underset{\alpha,\mathbf{B},\beta,\mathbf{H}}{\operatorname{argmin}} ||\mathbf{X}\odot\mathbf{W}-\beta\alpha\mathbf{H}\odot\mathbf{B}|| \tag{9}$$

where $\odot$ is the product done element-wise. Now consider vectors $\mathbf{Y}\in\mathbb{R}^n$, $\mathbf{C}\in\{+1,-1\}^n$ and $\gamma \in \mathbb{R}^+$ such that

$$\mathbf{Y}_i = \mathbf{X}_i\mathbf{W}_i \tag{10}$$

$$\mathbf{C}_i = \mathbf{H}_i\mathbf{B}_i \tag{11}$$

$$\gamma = \beta\alpha \tag{12}$$

Thus, using equations (9),(10) and (11), equation (8) is written as:

$$\gamma^*,\mathbf{C}^* = \underset{\gamma,\mathbf{C}}{\operatorname{argmin}} ||\mathbf{Y}-\gamma\mathbf{C}|| \tag{13}$$

The optimal solutions are obtained by using equation (7) as follows:

$$\mathbf{C}^* = signof(\mathbf{X})\odot signof(\mathbf{W}) = \mathbf{H}^*\odot\mathbf{B}^* \tag{14}$$

As $|\mathbf{X}_i|$, $|\mathbf{W}_i|$ are independent, equation (10) is reduced to:

$$\mathbf{E}[|\mathbf{Y}_i|] = \mathbf{E}[|\mathbf{X}_i|]\mathbf{E}[|\mathbf{W}_i|] \tag{15}$$

Thus we obtain:

$$\gamma^* = \frac{\sigma|\mathbf{Y}_i|}{n} = \frac{\sigma|\mathbf{Y}_i||\mathbf{W}_i|}{n} = \beta^*\alpha^* \tag{16}$$

3.4.3. Binary Convolution

The binary convolution between **I** and **W** is defined as:

$$\mathbf{I} * \mathbf{W} \approx (signof(\mathbf{I} \circledast signof(\mathbf{W}))\odot\mathbf{K}\alpha)) \tag{17}$$

where $\circledast$ is the convolution operator. The scaling factors of all the sub-tensors of **I** are contained in **K**. **K=A*k** where $\forall_{ij}\mathbf{k}_{ij} = \frac{1}{w \times h}$. Note that $*$ is the convolution operator, used to convolve **A** with $\mathbf{k} \in \mathbb{R}^{w \times h}$ is the filter in 2D dimensions.

3.4.4. *Training approach used in XNORNet*

Training a neural network with XNOR convolution and dense layers is similar to training any Artificial Neural Network. We start with inputs as pre-processed data, then pass it through the neural network's layers.In experiment and results section, the exact parameters used will be stated. The exact neural network used is:

- 1-XnorConv2D,1-BatchNormalization,1-Activation
- 1-XnorConv2D,1-MaxPooling2D,1-BatchNormalization,1-Activation
- 1-XnorConv2D,1-BatchNormalization,1-Activation
- 1-XnorConv2D,1-MaxPooling2D,1-BatchNormalization,1-Activation
- 1-Flatten,1-XnorDense,1-BatchNormalization,1-Activation
- 1-XnorDense,1-BatchNormalization

3.5. *Activation Functions*

In neural networks, the activation function is a function that given the inputs to the neuron will determine the output of the neuron. Activation function introduce non-linearities in neural networks which enable them to learn any arbitrary pattern in the data. We explore two activation functions in this paper:

3.5.1. *Rectified Linear Unit (RELU)*

Rectified Linear Unit (RELU) is an activation function, which was first introduced by Hahnloser et al.[16] in the year 2000. In 2011,[17] it was shown to endow superior training of deep neural networks compared to the other activation functions widely used at the time. This was due to lowered likelihood of the gradient to disappear as when $x > 0$, the gradient has a constant value in contrast with functions like sigmoids whose gradient becomes increasingly small as x increases. Another benefit of RELU is that if $x \leq 0$, the output becomes 0. This leads to a sparser representation of the network.

RELU Activation Function[18] is given by

$$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

3.5.2. *Scaled Exponential Linear Unit (SELU)*

Scaled Exponential Linear Unit (SELU)[19], first introduced in 2017, is an activation function that has self-normalizing properties in self-normalizing neural networks (SNNs). Despite the presence of perturbations and noise, activations with values near unit variance and zero mean when propagated through multiple network layers will converge towards unit variance and zero mean. Also, for activations not close to unit variance, exploding or vanishing gradient is not possible as there is an upper and lower bound on output variance. The SELU Activation Function[19] is given by

$$f(x) = \lambda \begin{cases} \alpha(e^x - 1), & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

with $\alpha = 1.67326$ and $\lambda = 1.05070$

4. Experimental Setup

For our experimentation, we have used publicly available datasets Faces95 and Faces96 to enable comparison with existing work and further research.

4.1. *Faces95 Dataset*

Faces95 [20] is a dataset containing 20 images of each of the faces (and some part of upper body) of 72 males and females. Each of the 1440 images have a resolution of $180x200$. There's strong head scale variation, minor head tilts and minor variations in facial expressions. Though red curtain is used as a background, variations in background were caused by the shadows of the individual. Since pictures were taken under artificial lighting the illumination of an individual's face changes in their image.

4.2. *Faces96 Dataset*

Faces96 [21] is a dataset containing 20 images of each of the faces (and some part of upper body) of 152 males and females. Each of the 3040 images have a resolution of 196*x*196. Similar to Faces95, the images of any individual have variations in head scale, minor head tilts, minor variations in facial expressions and changes in a face's illumination. Unlike Faces95, these pictures have glossy background posters leading to increased noise and complexity in illumination and lighting.

4.3. *System specification on which experimentation is done:*

- CPU: Intel core i5 7th generation.
- GPU: NVIDIA GeForce 940MX.

5. EXPERIMENTS AND RESULTS

Approach used:

- Images pre-processed with MBD were split into 20% for testing and 80% for training purpose.
- The neural network (described in Subsection 3.4 of XNORNet) is then trained for 50 epochs (batch size as 10) and evaluated over testing data.
- The appropriate accuracy for Faces95 and Faces96 are shown in Table 1.The results are shown as an accuracy plot in Fig 3 and Fig 4.The accuracy values in table 2 are obtained from [15].

Table 1. Accuracies obtained for Faces95 and Faces96 with SELU and RELU

Dataset Used	SELU	RELU
Faces96	94.03%	66.23%
Faces95	70.83%	62.96%

Table 2. Comparision of accuracies obtained for Faces96 with LDA,LBP, Ridgelet Transforms and XNORNet+SELU

Dataset	LDA	LBP	RT	XNORNet+SELU
Faces96	78.34%	84.14%	81.64%	**94.03%**

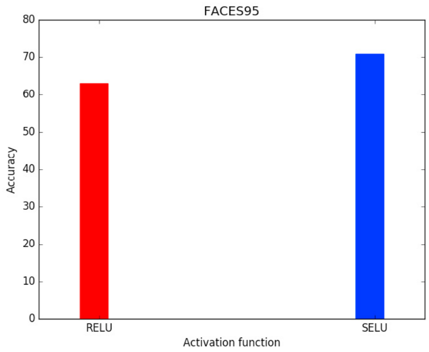

Fig. 3. Accuracy Plot for Faces95 Dataset

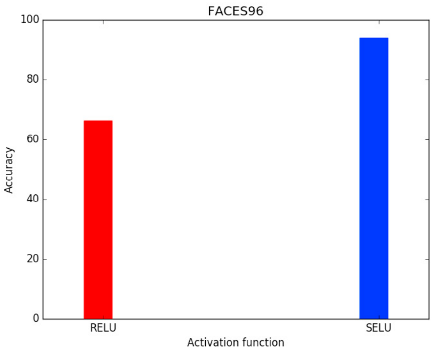

Fig. 4. Accuracy Plot for Faces96 Dataset

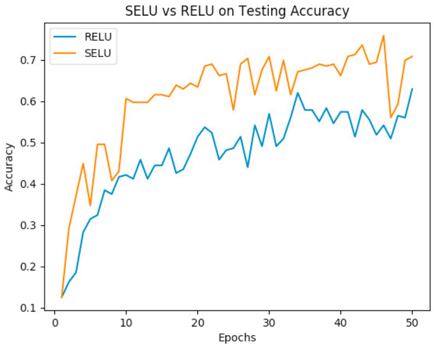

Fig. 5. Faces95 : SELU vs RELU on Testing Accuracy

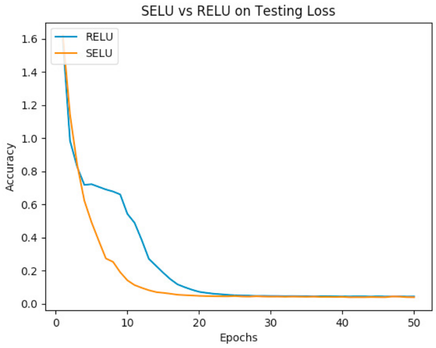

Fig. 6. Faces95 : SELU vs RELU on Testing Loss

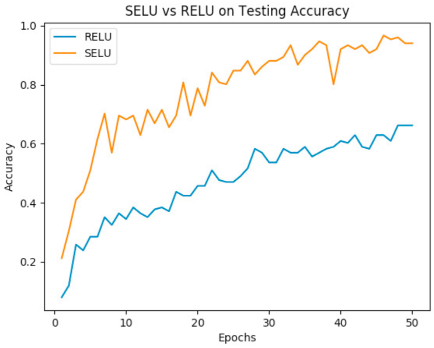

Fig. 7. Faces96 : SELU vs RELU on Testing Accuracy

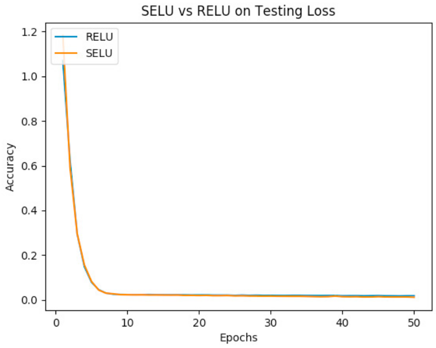

Fig. 8. Faces96 : SELU vs RELU on Testing Loss

6. Conclusions

This paper examined the performance of XNORNet Neural Network in consolidation with two activation functions: SELU,RELU and pre-processing transform Minimum Barrier Salient Object Detection on Faces95 and Faces96 image datasets. It is observed that Faces96, XNORNet+SELU with MBD shows very high accuracies as seen in Table 2. This can be attributed to the effectiveness of MBD in highlighting salient objects. Contrasting MBD is LDA which shows better accuracy for Faces95 than for Faces96. This is because it works best when the number of samples are low and the number of variables are large.

While LBP has the following disadvantages: (1) Computational complexity in terms of time and space increases when the size of the features increases exponentially with the number of neighbours.(2) It is not rotation invariant.(3) Limited structural information is captured. Only pixel difference is used, magnitude information ignored. LBP thus gives lesser accuracy in Faces96 due to significant lighting and expression variation. Finally, we conclude that Ridgelet Transform is not very effective as the images have lot of curved edges.Hence it proves to be less effective than our proposed method. In graphs regarding testing loss (Fig 6 and Fig 8), we observed that ReLU's loss saturates to a final value at a much slower rate than that of SeLU. From Table 1) better testing accuracy is shown in SeLU. This is because ReLU leads to Bias shift. Due to a positive mean activation, in subsequent layers of the network, a positive bias is observed from ReLU. A slower learning rate is thus observed even though they are less computationally expensive. Thus to reduce activation bias, activation functions like ReLU are preferred. Upon observing results, for datasets with copious background noise, like Faces96, our proposed method of MBD with SeLU is an ideal combination. While for image datasets with decrease in lighting, like Faces95, didn't yield satisfactory results.

Thus for real time applications, XNORNet in combination with MBD and SELU is an ideal model.

7. Discussion and Future Work

This paper explored a method to improve face recognition in time sensitive and accuracy demanding applications. Future scope for this involves taking multiple paths : (1).Researching more efficient neural network architectures. (2). Researching faster and efficient pre-processing techniques which can work well in tandem with the chosen neural network. The coupling between pre-processing and neural network invites a challenge to find methods which can work in harmony. Also, targeting resource constrained devices like mobiles etc. insists creating new efficient techniques or modifying existing techniques to make them efficient.

References

[1] L. Wang, W. Ouyang, X. Wang, and H. Lu. *STCT: Sequentially training convolutional networks for visual tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1373 1381, 2016

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. *"Imagenet classification with deep convolutional neural networks."* Advances in neural information processing systems. 2012.

[3] Girshick, R., Donahue, J., Darrell, T., Malik, J.: *Rich feature hierarchies for accurate object detection and semantic segmentation.* In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580587

[4] Grm, K., Struc, V., Artiges, A., Caron, M., Ekenel, H.K.: *Strengths and weaknesses of deep learning models for face recognition against image degradations.* IET Biometrics 7(1) (2018) 8189

[5] Hijazi S, Kumar R, Rowen C: *Using convolutional neural networks for image recognition.* Technical Report, 2015, available online: http://ip.cadence.com/uploads/901/cnn-wp-pdf

[6] Gnter Klambauer, Thomas Unterthiner, Andreas Mayr, Sepp Hochreiter *Self-Normalizing Neural Networks* http://cs-people.bu.edu/jmzhang/fastmbd/MBS_preprint.pdf

[7] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi *XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks.* https://arxiv.org/pdf/1603.05279.pdf

[8] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price and Radomr Mch. *"Minimum Barrier Salient Object Detection at 80 FPS."* http://cs-people.bu.edu/jmzhang/fastmbd/MBS_preprint.pdf

[9] Wangjiang Zhu, Shuang Liang, Yichen Wei and Jian Sun, *Saliency Optimization from Robust Background Detection*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[10] R. Achanta, S. Hemami, F. Estrada and S. Ssstrunk, *Frequency-tuned Salient Region Detection, IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR 2009), pp. 1597 - 1604, 2009

[11] Courbariaux, M., Bengio, Y.: *Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1."* CoRR (2016).

[12] T. Ahonen, A. Hadid, M. Peitikainen, Face recognition with local binary patterns. *In Proc. of European Conference of Computer Vision*, 2004.

[13] J Lu, K. N. Plataniotis, A. N. Venetsanopoulos, Face recognition using LDA-based algorithms, *IEEE Neural Networks Transaction*, 2003.

[14] Satyajit Kautkara, Rahulkumar Kocheb, Tushar Keskarc, Aniket Panded, Milind Ranee, Gary A. Atkinsona. *Face Recognition Based on Ridgelet Transforms*, ICEBT 2010.

[15] Faizan Ahmad1, Aaima Najam2 and Zeeshan Ahmed3. *Image-based Face Detection and Recognition: State of the Art*, IJCSI International Journal of Computer Science Issues, Volume: 9, November-2012.

[16] R Hahnloser, R. Sarpeshkar, M A Mahowald, R. J. Douglas, H.S. Seung *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature.* 405. pp. 947951 2000

[17] Xavier Glorot, Antoine Bordes and Yoshua Bengio *Deep sparse rectifier neural networks.* 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15 2011

[18] Nair, Vinod; Hinton, Geoffrey E. *"Rectified Linear Units Improve Restricted Boltzmann Machines"*, 27th International Conference on International Conference on Machine Learning, ICML'10, USA: Omnipress, pp. 807814, ISBN 9781605589077 2010

[19] Klambauer, Gnter; Unterthiner, Thomas; Mayr, Andreas; Hochreiter, Sepp *"Self-Normalizing Neural Networks"*. arXiv:1706.02515 2017

[20] D. L. Spacek, *Collection of facial images,* http://cswww.essex.ac.uk/ mv/allfaces/faces95.html.

[21] D. L. Spacek, *Collection of facial images,* http://cswww.essex.ac.uk/ mv/allfaces/faces96.html.