Proceedings of the Fourth International Conference on Communication and Electronics Systems (ICCES 2019)
IEEE Conference Record # 45898; IEEE Xplore ISBN: 978-1-7281-1261-9

Neural Differential Equations for Face Recognition

Gadicherla Sameer
Department of Computer Science
PES University
Bengaluru, India
sameergadicherla56@gmail.com

Gogineni Manasa
Department of Computer Science
PES University
Bengaluru, India
manasapriya17@gmail.com

***Abstract*—Recently, Convolutional Neural Networks(CNNs) have been very efficient in many computer vision areas, especially in facial recognition where most CNN models showed near-perfect performance on standard benchmarks.One of the main problems faced by these CNN architectures today is that they need a large amount of processing power and memory to deliver their exceptional performance. This paper aims on comparing and contrasting these CNNs with a new advancement in deep learning called neural differential equations, where the whole architecture is thought of as a differential equation to be solved. This paper aims to show that this method can achieve comparable performance to a residual network with much less parameters. As a part of this project we will show that the models trained over the ODENet, that is neural net using the ordinary differential equations, takes very less memory and time to train when compared to that of CNNs. As to run these models on the benchmark datasets, these models have been run on the standard face datasets named Faces95 and Faces96 respectively.**

Keywords- Face Recognition, Activation Function, Minimum Barrier Detection (MBD), Ordinary Differential Equation network (ODENet,ODE), Scaled Exponential Linear Unit (SELU,selu)

I. INTRODUCTION

Face recognition is a task of identifying or recognising a person from an image or a video frame. This involves a lot of training to be undergone for any model to do the prediction. One such methods is the Convolutional neural network(CNN). Recent advances in these networks have made the tasks of tracking[1],recognition[2] and detection[3] pretty much very easy to solve.Deep models have been so influential in this field that almost all the architectures designed have achieved the near perfect results. One of the main challenges faced by CNNs today is that the requirement of heavy processing power hinders the implementation of CNN based solutions on less powerful but ubiquitous devices like embedded systems and mobile phones without diminished performance with wellknown benchmarks[4].

To overcome the existing problem, we have chosen the unique approach from [5] where the whole network being trained is visualised as a differential equation. This neural differential equation is implemented as an ODEBlock in our paper which is a custom keras layer consisting of the differential equation along with weights and the dopri5 method for solving the differnetial equations. This method mainly takes out the very big process of deciding the architecture for training. Any architecture defined with this one block in between will try to learn with shared weights among all the layers which further form the differential equation.

The most interesting part of this network is that it supports supervised learning very effectively. As said in [5], they applied it over the task of classification of written digits by MNIST. The aim in this paper was to show that the ODESolve method written can achieve a comparable performance with the standard residual networks[6]with very less number of parameters. As a result it did perform very well and reduced the parameters from 0.6M in residual network to 0.22M in ODESolve method.Thereby reducing the processing power requirements and the time taken to train the model.

The popularly known datasets Faces95 and Faces96 were used here by pre-processing with Minimum Barrier Detection(MBD) for saliency detection. Upon this we have used the SELU activation function. The motivation behind using this was to avoid overfitting in learning due to less number of parameters being trained using the ODENetwork and the MBD for saliency detection. So as to allow for many levels of abstract representation of the output feeded as input to a neuron, SELU activation function was written, created the Self-Normalizing Neural Networks[7] via inducing the properties for self-normalisation in Feed Forward neural networks(FNNs)[8].

II. RELATED WORK

Neural Ordinary Differential Equations[5] paper introduces a new concept in deep neural network models. It uses a black-box differential equation solver to compute the output. It is a continuous-depth model as it does not use discrete hidden layers. It shows back-propagating through ODE solvers without changing its internal operations. They used this black-box ODE solver for developing new models for time-series modeling, density estimation and supervised learning. These models are evaluated based on computation speed and accuracy. They derived an instantaneous version of the change of variables formula and developed continuous time normalizing flows, which can scale to large layer sizes.

Residual Attention Network[9] is a Convolutional Neural Network with attention mechanism. It incorporates state-of-art feed-forward network architecture. Within each attention module bottom-up top-down structure is used. This architecture is tested on CIFAR-10, CIFAR-100 and ImageNet data sets. It also shows that this network architecture is not affected by noisy channels. This Residual Attention Network outperformed state-of-art image classification methods.

Face Recognition: EigenFace, Elastic Matching and Neural Nets[10] is a relative study between the three face recognition algorithms. EigenFace algorithm works well only when there is very less lighting variations. Elastic Matching algorithm is more adaptable to lighting variations, face positions and expression variations.

Convolutional Neural Networks[11] for Image Classification paper showed image classification using Convolutional Neural Networks. It used traffic signals dataset and classified it.

Face recognition/detection by probabilistic decision-based neural network[12] paper divides face recognition into three parts: Face detection, position of eyes and face recognizer.

Deep Face Recognition[13] paper does face recognition in photos with single face or with multiple faces. It uses very large data sets that are available like LFW, YTF, FaceBook, etc.

III. PROPOSED WORK

A. Overview

Every image from the dataset was pre-processed using two steps. First step was resizing the image into smaller scale, second is "Minimum Barrier Detection (MBD) Transform". MBD Transform is well known for salient object detection. After the MBD Transform is done the image is passed to the neural network with the ODEBlock and trained over the datasets.

B. Minimum Barrier Detection

The aim of salient object detection[14] is to create a prominence map which highlights the face(object) and overcomes the background[15] in the scene. This transform is being widely used in research due to its vast use in computer vision applications. This method is prominently used in mobile devices as they require high quality salient maps using lesser space and is fast in computing it. For this method we used FastMBD Transform Algorithm[16]. Let m be a pixel value in Inverse Raster Scan or Raster Scan. The following distance map can be used to minimize the cost of path m using each n in the neighbourhood of m:

$$DM(m) \leftarrow min\{DM(m), P(I(n).\langle n, m\rangle)\} \quad (1)$$

In this equation,

(1).β_P is used in Image Distance Transform, which is a cost function that is defined as:

$$\beta_P(\pi) = max_{i=0}^{k} P(\pi(i)) - min_{i=0}^{k} P(\pi(i)) \quad (2)$$

where P is the value of a Pixel.

(2).$I(n)$ denotes the assigned path to a pixel n and $\langle n, m\rangle$ is an edge joining pixel m to pixel n. Thus $I(n).\langle n, m\rangle$ is a path for m that appends edge $\langle n, m\rangle$ to $I(n)$ Therefore, the cost function becomes:

$$\beta_P(I_n(m)) = max\{H(n), P(m)\} - min\{L(n), P(m)\} \quad (3)$$

where $H(n)$ and $L(n)$ are the highest and lowest pixel values in $I(n)$ respectively. The value of H and L is updated accordingly when the path assignment changes in each iteration in the algorithm.

To perform the MBD transform following algorithm was used, it is based on [16].

Result: A list DM which is the distance map
Input: Number of Iterations : $iter$ for the Image $Img = (i, V)$;
Algo:$H, L \leftarrow Img, Img$;
set $DM(x)$ to 0 or to inf;
for *m from* 0 *to iter* **do**
  if *m is even* **then**
    InvRasterScan(Img,H,L,DM);
  else
    RasterScan(Img,H,L,DM);
  end
end

Algorithm 1: MBD transform

Result: A list DM which is the distance map
Input: Img, H, L, DM from MBD transform algorithm.; number of rows and columns are set to $rows$ and $cols$ in image Img;
for *m from* $rows - 2$ *to 1* **do**
  for *n from* $cols - 2$ *to 1* **do**
    imx,dist = Img[m][n],DM[m][n];
    high1,high2 = H[m+1][n],H[m][n+1];
    low1,low2 = L[m+1][n],L[m][n+1];
    u1 = max(high1,imx) - min(low1,imx);
    u2 = max(high2,imx) - min(low2,imx);
    if *u1<dist and u1<=u2* **then**
      DM[m][n] = u1;
      H[m][n] = max(high1,imx);
      L[m][n] = min(low1,imx);
    else
      DM[m][n] = u2;
      H[m][n] = max(high2,imx);
      L[m][n] = min(low2,imx);
    end
  end
end

Algorithm 2: Inverse Raster Scan

A sample of how salient object is detected using MBD is shown in Fig 1.

Fig. 1. An example of highlighting salient objects in an image

using MBD transform.

C. ODENet

ODENet refers to Ordinary DIfferential Equation Network. In normal neural networks, the previous state vector V_n produces the new state vector V_{n+1} using the formula $V_{n+1} = F(V_n)$. Here function F is $F(x) = \sigma(\sum_i \theta_i x_i)$, where θ is a vector of parameters and σ is an activation function. Whereas in Deep Residual Learning[5], the new state vector is calculated using $V_{n+1} = F(V_n) + V_n$. Using Euler's Method in Residual Networks: Consider a constant $\triangle t \in R$, now the new state in neural network becomes,

$$V_{t+1} = F(V_t) + V_t = \frac{\triangle t}{\triangle t} F(V_t) + V_t = \triangle t G(V_t) + V_t \quad (4)$$

where $G(V_t) = \frac{F(V_t)}{\triangle t}$. If we consider every step of the neural network to be doing Euler's method, then our system can be modeled as

$$\frac{dV(t)}{dt} = G(V(t), t, \theta) \quad (5)$$

Here G is dependent on t and θ. The output of network at some time t_1 is $V(t_1)$. Hence, if we know G, we can use ODE Solvers to evaluate the neural network.

$$V(t_1) = ODE_S(V(t_0), G, t_0, t_1, \theta) \quad (6)$$

If this is substituted in Euler's Method we get something like the residual network state as above.

$$ODE_S(V(t_0), G, t_0, t_1, \theta) = V(t_0) + (t_1 - t_0) G(V(t_0), t_0, \theta) \quad (7)$$

D. Training approach used in ODENet

Train a neural network with ODEBlock as a layer with combination of few other layers, and thereby run as a normal artificial neural network.

The inputs start with preprocessed data, then it is forwarded through the neural network's layers.

In the layers, a function if applied on inputs and weights (inputs and weights are matrices) and then the activation function is applied(SELU). Further in experiment and results section, the exact parameters used will be shown.

The neural network used for ODENet is:

- 1-Conv2D,1-Activation(selu)
- 1-MaxPooling2D
- 1-Conv2D,1-Activation(selu)
- 1-MaxPooling2D
- 1-ODEBlock , 1-Flattening
- 1-DenseLayer , 1-Activation(softmax)

The exact neural network used for CNN for comaparision purpose is

- 1-Conv2D,1-MaxPooling2D,1-BatchNormalization,1-Activation
- 1-Conv2D,1-MaxPooling2D,1-BatchNormalization,1-Activation
- 1-Flatten,1-Dense,1-BatchNormalization,1-Activation
- 1-Dense,1-BatchNormalization

E. Activation Function - SeLU

SeLU is basically useful because it does normalization. Basically there are three types of normalization: input normalization, batch normalization and internal normalization. This function falls in the third category. This activation function needs negative and positive values for y to shift the mean.The gradients are used to adjust the variance. The formula for SeLU is

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

where λ = 1.05070 and α = 1.67326

This is similar to ReLU for positive values, except the λ. This is the reason for the S(caled). When λ is larger than 1, so is the gradient and the variance can be increased. Normalized outputs are very useful in stabilizing the training process. Additionally, SeLUs learn faster and better on their own compared to other activation functions, even when they are combined with batch normalization. The graph of SeLU looks like Fig. 2.

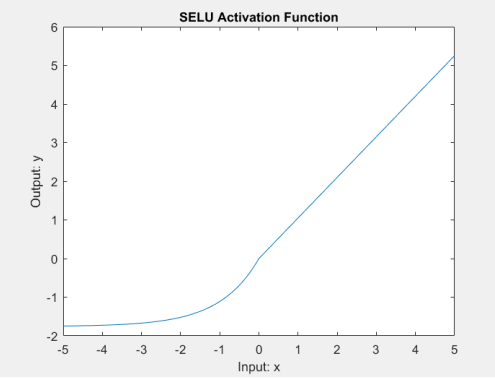

Fig. 2. Graph of SeLU activation function.

IV. EXPERIMENTAL SETUP

For the experimentation, the Faces95 and Faces96 publicly available datasets have been used to compare with current work and do further research.

A. Faces95 Dataset

The Faces95 [17] dataset contains 1440 images, 20 images each of 72 individuals both male and female. The resolution of the images is 180 by 200 pixels taken in portrait format. The subject moves towards the camera as the images are captured introducing head scale substantial variations between the images. Red curtains have been used in the background. Significant head scale variations and background changes were caused by the shadows as the subject moves towards the camera. Major changes in illumination of the face are seen due to the use of artificial lighting. Minor variations in the facial expressions of the subject and their head tilt, slant and turn were also seen.

Fig. 3. Samples from the Faces95 dataset.

B. Faces96 Dataset

The Faces96 [18] dataset contains 3040 images, 20 images each of 152 individuals. The resolution of the images is 196 by 196 pixels taken in square format. Just like the Face95 dataset, the subject moves towards the camera as the images are captured introducing similar head scale variations in images.Glossy posters have been used for the background increasing the complexity and noise. The hairstyle and age remain same as all images were captured in a single session. Major changes consist illumination of the face are seen due to the use of artificial lighting. Minor variations have the facial expressions of the subject and their head tilt, slant and turn.

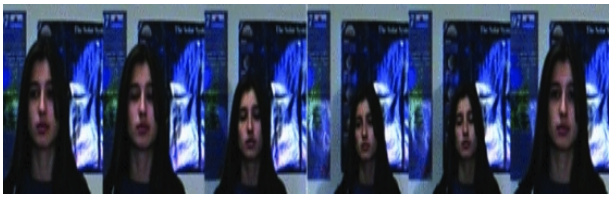

Fig. 4. Samples from the Faces96 dataset.

V. EXPERIMENT AND RESULT ANALYSIS

System description in which tests are conducted:

- CPU: Intel core i7 7th generation.
- GPU: NVIDIA GeForce 940MX.

Approach used(on CPU only) :

- Images were pre-processed with Minimum Barrier Detection transform and then the processed image was passed as input into the neural network.
- The pre-processed dataset was then split into 80% training and 20% testing images.
- The neural network (training algorithm and architecture described in Subsection 4 of Approach Section) is then used to train upon the training data and tested with the testing data. The network was trained with SeLU for 30 epochs, with 10 as batch size for ODENet and 50 epochs with same batch size for CNN.
- The results are shown as an accuracy table in table I.
- The size consumed by models are described in table II.
- The time consumed for training by models are shown in Fig.10
- The time consumed for loading the models are described in table III.

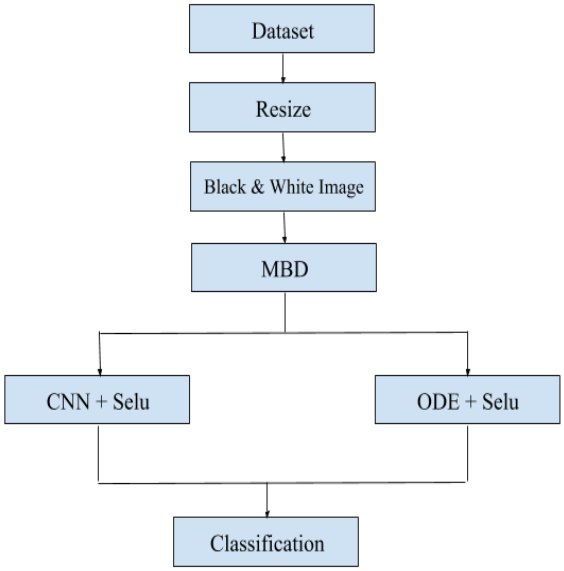

Fig. 5. Block Diagram of the experiment.

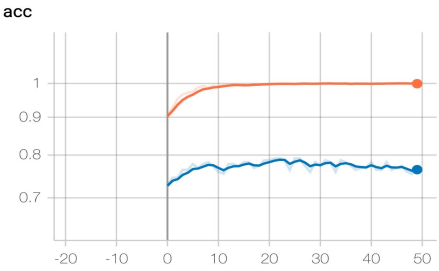

Fig. 6. Accuracy Vs Epochs for CNN over Faces95.

The accuracy for each epoch (50 epochs) for CNN over Faces95 is shown in Fig. 6.

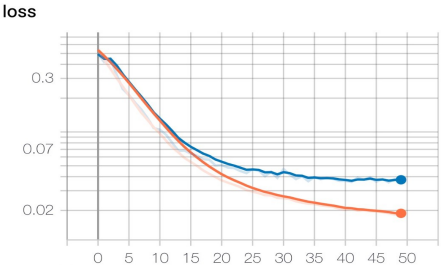

Fig. 7. Loss Vs Epochs for CNN over Faces95.

The loss for each epoch (50 epochs) for CNN over Faces95 is shown in Fig. 7. After training for 50 epochs, the loss of this model in terms or training and testing has reached very much near to zero, which says that the model has avoided overfitting.

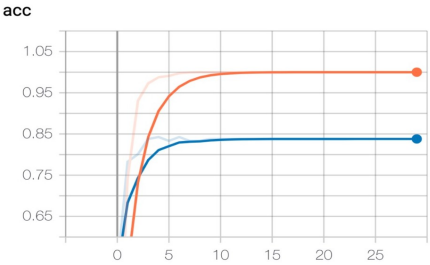

Fig. 8. Accuracy Vs Epochs for ODE over Faces95.

The accuracy for each epoch (30 epochs) for ODENet over Faces95 is shown in Fig. 8.

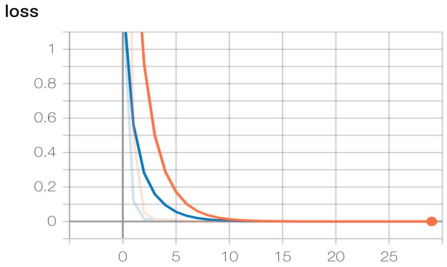

Fig. 9. Loss Vs Epochs for ODE over Faces95.

The loss for each epoch (30 epochs) for ODENet over Faces95 is shown in Fig. 9. Even here the losses have reached near to zero. Loss for Faces96 have also been almost to zero.

* Orange Line - Training Parameters

** Blue Line - Testing Parameters

TABLE I
AVERAGE VALIDATION ACCURACY

	CNN	ODE
Faces95	77.5%	83.46%
Faces96	96.24%	98.79%

From TABLE I, we observe that both the CNN and the ODENet on Faces95 performed poor than that over Faces96 in terms of accuracy. But if we see in Faces96, ODENet performed approximately same as CNN but in Faces95 it performed better than CNN. And all the models have a test loss approximately close to zero.

TABLE II
SIZE OF THE MODEL

	CNN	ODE
Faces95	3.52 GB (3605MB)	29.3 MB
Faces96	3.03 GB (3103MB)	50 MB

In TABLE II, we can see that over Faces95, model on ODENet is 123 times less size of that using CNN. Similarly over Faces96, model on ODENet is 62 times less size of that using CNN.

TABLE III
TIME TAKEN TO LOAD THE MODEL (IN SEC)

	CNN	ODE
Faces95	40	9
Faces96	37	10

TABLE III clearly shows that over Faces95, ODENet loads four times faster and over Faces96 also around four times faster than that over CNN on both the datasets.

Fig. 10. Training Times for each Epoch.

Apart from having less model size, we can also see that ODENet has taken very less time to train as shown in Fig.10. CNN has taken 255 and 235 seconds to train per epoch on the given system specifications. Where are ODENet has taken 45 and 83 seconds for Faces95 and Faces96 respectively. By this we can say that we have reduced the training time per epoch by almost greater than five times.

VI. CONCLUSION AND FUTURE WORK

We can see that the ODENet has outperformed the CNNs in face recognition tasks completely in terms of performance. They have consumed 123 and 62 times lesser the memory on both the datasets Faces95 and Faces96 respectively and also very less training time as shown in results section than the CNNs. Both the architectures performed very well in the recognition tasks,but we know that the ODENet needed less epochs than CNNs to achieve a test loss less than ten percent of 1. The model loading times of the ODENet is one third of CNNs which says that ODENets can be extremely fast to even train and test on devices with lesser computational resources. Thereby we can say that neural differential equations will be a very good advancement in the field of recognition tasks mainly avoiding the task of designing a neural network architecture and giving outperforming results.

Applying these networks over time series is a very good application to be done. We will be using this approach over different tasks like stock prediction and also try to refine the architecture based on the application.Training these networks with different differential equation solvers would be a very good step for doing analysis and getting the most effective one will help us with better training. Training these kinds of networks on small devices like mobiles phones is what our end goal is and will be proceeding with that as the next step.

VII. ACKNOWLEDGMENT

We would like to thank Prof.Shetty Mamatha Gopal for guiding us to carry out this research. Department of Computer Science of PES University has always been very encouraging for us to work on interesting topics like this. Finally our parents and friends have always been motivating and supporting in carrying forward our research, and we are very grateful to them.

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] Chen, T. Q., Rubanova, Y., Bettencourt, J., Duvenaud, D. K. (2018). Neural ordinary differential equations. In Advances in Neural Information Processing Systems (pp. 6571-6583).

[6] He, Kaiming, et al. "Identity mappings in deep residual networks." European conference on computer vision. Springer, Cham, 2016.

[7] Klambauer, Gnter, et al. "Self-normalizing neural networks." Advances in neural information processing systems. 2017.

[8] Mirjalili, SeyedAli, Siti Zaiton Mohd Hashim, and Hossein Moradian Sardroudi. "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm." Applied Mathematics and Computation 218.22 (2012): 11125-11137.

[9] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., ... Tang, X. (2017). Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3156-3164).

[10] Zhang, J., Yan, Y., Lades, M. (1997). Face recognition: eigenface, elastic matching, and neural nets. Proceedings of the IEEE, 85(9), 1423-1435.

[11] Jmour, N., Zayen, S., Abdelkrim, A. (2018, March). Convolutional neural networks for image classification. In 2018 International Conference on Advanced Systems and Electric Technologies (IC-ASET) (pp. 397-402). IEEE.

[12] Lin, S. H., Kung, S. Y., Lin, L. J. (1997). Face recognition/detection by probabilistic decision-based neural network. IEEE transactions on neural networks, 8(1), 114-132.

[13] Parkhi, O. M., Vedaldi, A., Zisserman, A. (2015, September). Deep face recognition. In bmvc (Vol. 1, No. 3, p. 6).

[14] 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

[15] 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.

[16] 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

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

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