Brain-Inspired Gaussian Weighted Normalization

Mariangeline Merced Perez[1], Sophia Salamanca[2], Nicolas Echevarrieta Catalan[3], Alicia Bilbao Martinez[3], Xu Pan[3], Odelia Schwartz[3], Vanessa Aguiar-Pulido[3]

[1]Department of Computer and Electrical Engineering, Ana G. Méndez University Gurabo, PR, USA

[2] School of engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

[3] Department of Computer Science, University of Miami, Coral Gables, FL, USA

Introduction

Normalization plays a crucial role in the visual system for image recognition, specifically in the primary visual cortex. In the brain, the visual cortex neurons react depending on the strength of the input of visual features such as color, motion, and orientation. With normalization, the brain adjusts the response of individual neurons based on the activity levels of surrounding neurons, ensuring that the overall response is proportional to the input strength [1].

Weighted divisive normalization is a method that simulates the normalization that is present in the brain. Here the weights become smaller the farther away they are from the center pixel. This means that as we move farther away from the center pixel, the influence of the surrounding pixels decreases[2]. By applying different gaussians to this normalization, it is possible to modify the importance of neighboring pixels based on the distance of the surrounding information (Fig. 1A) without using the information of the center pixel (Fig. 1B).

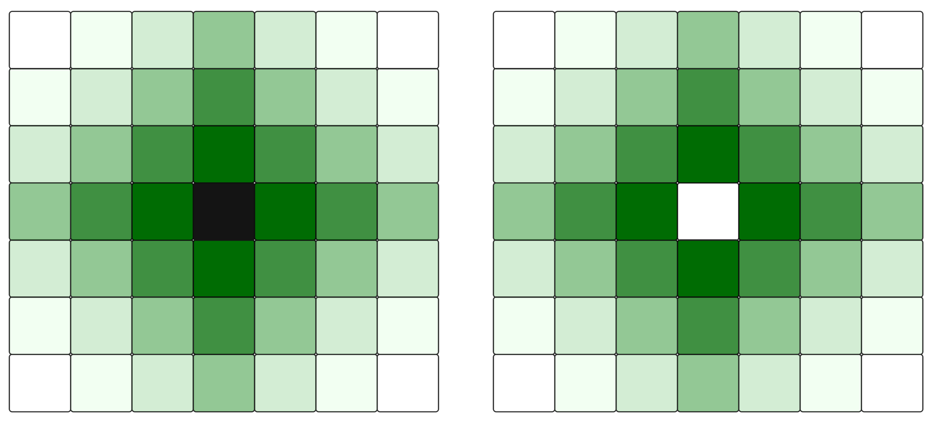

Figure 1 | **A. Gaussian Normalization** | **B. Proposed approach**

Methods

- **Data:**

- Imagenette2 Dataset[3]:

This dataset is a subset of 10 labeled classes from ImageNet containing a total of 2700 Images. The classes are: bench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute.

In this work, we tested the proposed approach on these 2700 images, in which 1350 belong to the cassette player, and the remaining 1350 are distributed between the rest of the classes.

- **Proposed Approach:**

To perform this, first, a filter is created based on the implemented gaussian. This filter is then normalized using divisive normalization. By doing so, it ensures that the central pixel is not considered, allowing us to obtain the proposed gaussian weighted normalization. After the image has been filtered, an image with better contrast is obtained with the use of histogram equalization. Finally, this is used as input to a CNN to obtain a classification. An overview of the proposed method is shown in Fig. 2.

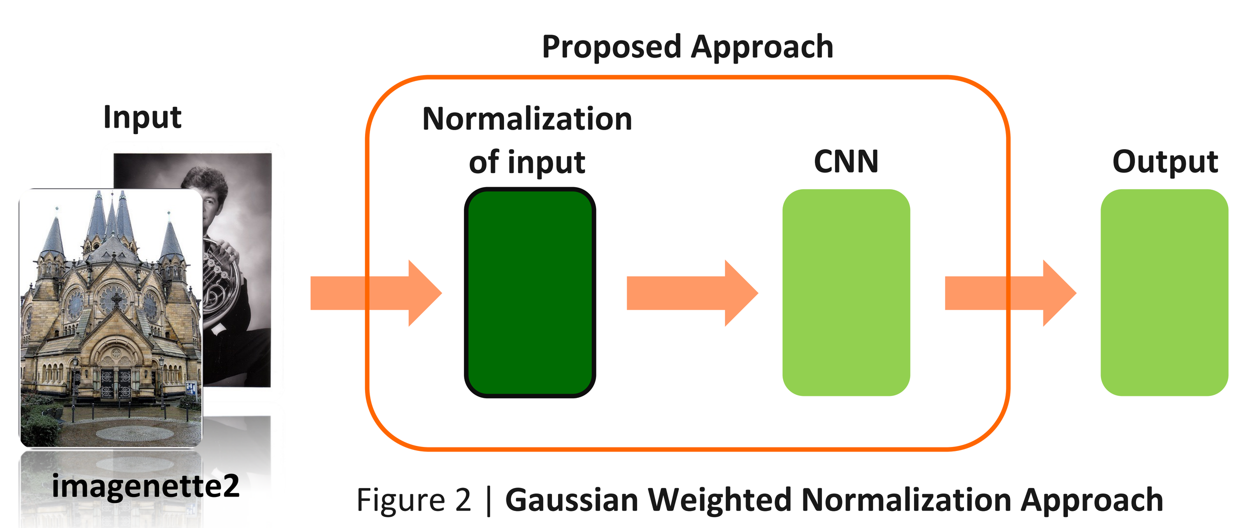

Figure 2 | **Gaussian Weighted Normalization Approach**

- Divisive Normalization:

Divisive normalization is a type of normalization that occurs in the brain which adjusts the response of a neuron to the overall level of activity of its surrounding neurons. This process ensures that the neurons' response is scaled. [1]

$$R_j = \gamma \frac{D_j{}^n}{\sigma^n + \sum_k D_k^n}$$

- Gaussian weighted normalization:

The formulas used to obtain the proposed normalization are the following:

$$FWHM = 2\sqrt{2\ln 2} \times \sigma$$

- Square Gaussian:

$$\exp(-4\ \log 2 \times \tfrac{(x-x_0)^2+(y-y_0)^2}{FWHM^2})$$

- Isotropic Gaussian:

$$\exp(-\left(\frac{(x-x_0)^2+(y-y_0)^2}{2 \times FWHM^2}\right))$$

- Histogram equalization:

This method redistributes the pixel intensity value of an image, to cover the entire range of possible values. This leads to a higher contrasted image, making it easier to see details and enhancing visual quality (Fig. 3).

- Convolutional Neural Networks As Deep Learning Classifier:

- The ResNet-18, ResNet-50, ResNet-101 and VGG19 architectures were employed as prime examples of CNNs.
- Binary classifier was implemented to separate between two classes, positive and negative.
- Transfer learning and fine tunning were used as the weights and the parameters of the pre-trained models were modified to adapt the new transferred weights and parameters.
- Binary cross entropy loss was implemented.

- Network optimization:

Parameters such as the kernel size (3, 5, 7), gaussian type (square, isotropic), optimizers (Adam, RMSprop) and learning rates (0.001, 0.00001) were used to test a variety of combinations to ultimately evaluate the model's performance on the validation set.

- Stratified 5-Fold Cross-Validation:

5-fold Cross-Validation is a method used to evaluate the performance of models. This method consists of separating the data into 5 folds and each model is evaluated 5 times using different folds as the validation set and the others as training set. Stratified sampling ensures that all classes are evenly distributed within a set.

Results

The highest scores for each kernel of each model tested is represented below (Table 1.). Additionally, the resulting effects of the histogram equalization are shown in Fig.3.

MODELS:	RESNET-18			RESNET-50			RESNET-101			VGG19		
KERNEL:	3	5	7	3	5	7	3	5	7	3	5	7
GAUSSIAN	Isotropic	Square	Square	Isotropic	Square	Square	**Square**	**Square**	**Square**	Square	Isotropic	Square
OPTIMIZER	RMSprop	RMSprop	Adam	RMSprop	RMSprop	RMSprop	**RMSprop**	**Adam**	**Adam**	RMSprop	RMSprop	RMSprop
LEARNING RATE	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	**0.00001**	**0.00001**	**0.00001**	0.00001	0.00001	0.00001
ACCURACY	0.9828	0.9837	0.9832	0.9902	0.9898	0.9907	**0.9921**	**0.9911**	**0.9911**	0.9758	0.9762	0.9772
F1	0.9830	0.9840	0.9834	0.9903	0.9899	0.9908	**0.9922**	**0.9913**	**0.9913**	0.9762	0.9766	0.9775
ROC/AUC	0.9827	0.9836	0.9832	0.9902	0.9897	0.9907	**0.9921**	**0.9911**	**0.9911**	0.9757	0.9762	0.9771

Table 1 | **Model 5-Cross Validation Results**

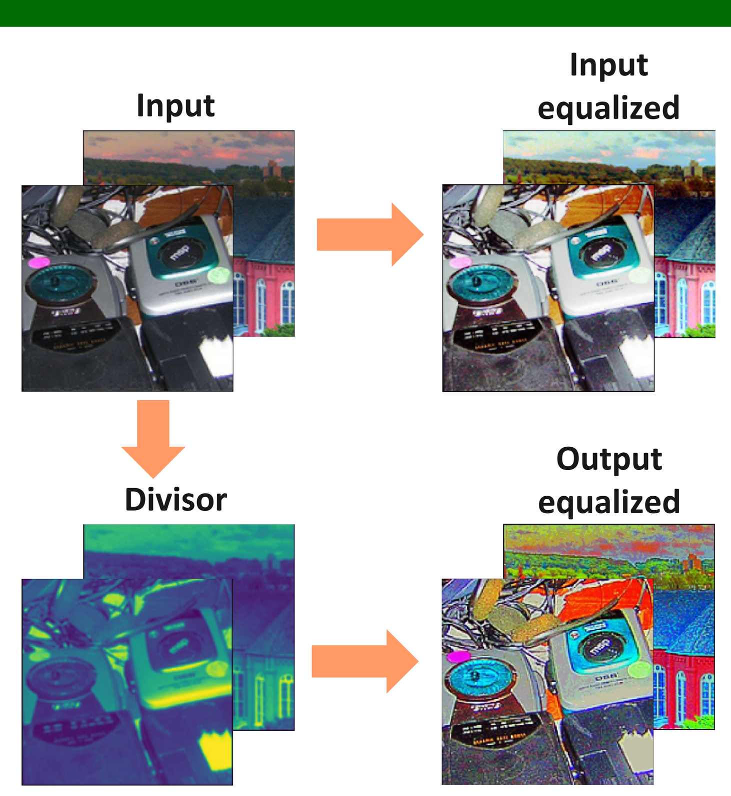

Figure 3 | **Equalization**

Conclusions

The proposed approach was able to achieve classification scores of 99%, performing slightly better than approaches that don't include the gaussian weighted normalization. The results show that all the ResNet models achieved the highest scores, outperforming the models based on VGG19. In addition, RMSprop was the overall best optimizer, employing a learning rate of 0.00001, yet the Adam optimizer is still performing proficiently. In the case of gaussians, the square gaussian seemed to perform slightly better than the isotropic gaussian for most of the scores. Finally, scores such as accuracy, F1 and ROC/AUC were compared and the model based on ResNet-101 obtained the highest scores for all tested kernel sizes.

Future Work

For the future of this project, it would be ideal to test a wider variety of gaussians, optimizers and models. In addition, we will integrate the proposed normalization in the blocks of the CNN classifiers. We will also try a recurrent approach, to emulate the recursive influence from one neuron to another until the interaction converges. Finally, we will design experiments to determine how the proposed approach compares to the way that the human visual cortex processes information.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1949972. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Carandini, M., Heeger, D. Normalization as a canonical neural computation. Nat Rev Neurosci 13, 51–62 (2012). https://doi.org/10.1038/nrn3136

2. Pan, X., Giraldo Sánchez, L., Kartal, E., Schwartz, O. Brain-inspired Weighted Normalization for CNN Image Classification. (2021). doi: https://doi.org/10.1101/2021.05.20.445029

3. jph00. (2020). imagenette [imagenette2]. GitHub. https://github.com/fastai/imagenette/issues?q=is%3Aissue+is%3Aclosed