Proceedings of the Fifth International Conference on Inventive Computation Technologies (ICICT-2020)
IEEE Xplore Part Number:CFP20F70-ART; ISBN:978-1-7281-4685-0

Analysis of Deep Learning Architecture for Non-Uniformly Illuminated Images

Uma Subbiah
Department of Computer Science and Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
cb.en.u4cse16159@cb.students.amrita.edu

Padmavathi S
Department of Computer Science and Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
s_padmavathi@cb.amrita.edu

Abstract— **The use of deep learning to hone image processing techniques has become increasingly popular. Following the success of Convolutional Neural Networks (CNNs) for image classification, they have been tested for various applications. By training CNNs on a dataset with ground truth (light) images and the corresponding darkened version of the images, neural networks can be used for enhancement. This must account for the non-uniform illumination seen in night-time images. A novel method of training a neural network to enhance non-uniformly illuminated images is proposed. Further, the visualization of convolutional features extracted at each layer of the neural network is discussed, to understand which parts of an image helps the neural network identify the object, thereby enhancing its recognition power. The potential application of this system lies in detecting animals in the non-uniformly lit surveillance video, useful to settlements near forest regions, where wild animals pose a threat to the living areas.**

Keywords— Deep Learning, Convolutional Neural Networks, non-uniform illumination, surveillance systems, image processing, image enhancement, feature extraction.

I. Introduction

Neural Networks have been extensively used in image processing in the last few years. The application of neural networks to the segmentation of images [1] and image enhancement with the Low-Light Convolutional Neural Network (LLCNN) [2] have reformed traditional image processing techniques. While the enhancement and segmentation of well-lit and even uniformly dim images have been extensively documented in the literature, the segmentation and subsequent object detection of non-uniformly lit images with low contrast are still under research. This motivates the need for a method to effectively enhance images of non-uniform illumination. An augmented dataset that can aid and improve this process is discussed. Furthermore, a means of addressing this problem by dissecting the deep learning architecture, to extract convolutional features from each layer of the neural network is presented. Activation maximization can be performed to visualize the process of feature extraction. A novel method of identifying the distinguishing feature of an image is also proposed. This differentiating feature can be input to the deep learning architecture which is used for non-uniformly lit images, thereby enhancing the identification power of the neural network.

This paper has been organized as: Section 2 contains a literature review of research results in this domain and similar ongoing work. Section 3 discusses the design of the system proposed. Section 4 contains the proposed method to enhance the recognition power of the architecture by giving important features as input. The results and a brief discussion are found in Section 5. Section 6 discusses the conclusions and future scope.

II. Literature Survey

A. Improving image enhancement using deep learning architecture

A major motivation to enhance low-light images is to improve the performance of other deep learning architectures like the SegNet [1] on the image. One successful image-enhancing architecture is the Low-Light Convolutional Neural Network (LLCNN) [2]. A deep convolutional network can be employed to overcome the problems that arise due to the low-contrast seen in images, using a combination of deep learning neural networks with a retina based method, as opposed to a histogram-based method [3]. The LL-RefineNet proposed in [4] is a novel method of dealing with contrast and noise in-camera images from low light scenes. In [5], a Gaussian process is used for feature retrieval. Gaussian process regression is used in conjunction with a CNN to introduce feature enhancement functions into the distributions. A Gaussian process applies a Gaussian distribution to the varying intensities of the image at different points. Image restoration was significantly improved by Digital Image Inpainting [6], using information gained from regions around the damaged portion of the image. Similarly, an efficient method of image enhancement (described in [7]) determines the states of the pixel to be enhanced and the surrounding pixels using reinforcement learning techniques. Parallel to image enhancement, deep learning architecture can be used to perform denoising by converting the input into pixel vectors. This architecture is referred to as a Vectorized Convolutional Neural Network (VCNN), detailed in [8]

B. *Extraction of Convolutional Features from a CNN*

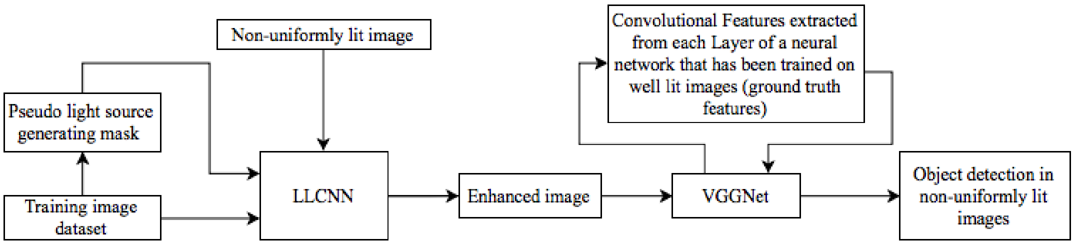

Fig. 1. Block diagram of the proposed architecture

A variant of the method found in [9] to extract the intermediate output of the convolutional neural network is adopted here. This bestows us with a deeper understanding of the way the CNN sees the input image and the regions deemed important for classification by the neural network. Activation maximization as described in the same source shows the relation between the feature extracted and the input image. Many features may be extracted from a single image, or similar features may be extracted for widely differing images. Overcoming this ambiguity is a guaranteed way to enhance the performance of neural networks. Extraction of features and feeding them as input to a neural network has been successfully used in a number of applications such as Indian sign language recognition, as described in [10].

An effective method of manipulating the kernel values is seen in [11] which also visualizes the important regions identified using a heat map. [12] analyses the output of each layer of an InceptionV3 model trained on a floral dataset. The inference is insightful and provides a possible explanation to the increase in complexity observed as the depth of the layer in the neural network increases. Activation heat maps indicate that the neural network is extracting appropriate regions of the image in order to identify the flower in the image.

A block-wise inference of the convolutional visualization of features and the possible relation between the kernels and the input can help in the diagnosis of weakness in a particular deep learning architecture [13]. The extraction and visualization of the intermediate output and subsequent feeding of important feature information using kernel difference is based on a skeletal adaptation of Chollet's implementation seen in [14].

III. DESIGN

A block diagram of the method proposed is seen in Fig. 1. The training image dataset (described in Section IV. A) is fed to the enhancement of deep learning architecture. Further, the training images are subject to a masking process that applies a mask of randomly generated zones of brightness and darkness. This creates a pseudo or false impression of light sources present in the training images, creating images of non-uniform illumination. The training image dataset is augmented with these images and used to train the deep learning architecture - a Low-Light Convolutional Neural Network (LLCNN). The main process begins when a new non-uniformly lit image is input to the pre-trained LLCNN. The output of this stage is a light enhanced version of the dark input image.

This enhanced image output is given to a VGGNet with 16 layers. The VGGNet is a deep learning architecture that uses fixed-size kernels to detect and identify objects in a given image. The output of each layer of this architecture is obtained and analyzed to gain an insight into the process used by a neural network for object recognition. These convolutional features extracted from the VGG-16, that has been trained on well-lit images, are treated as the ground truth features (i.e.) features to be searched for in the test image. These features are fed back into the VGGNet to be used when it encounters a dimly/non-uniformly lit image. Thus the VGG-16 is equipped with the knowledge of important features and is capable of satisfactorily detecting objects in images of non-uniform illumination

IV. EXPERIMENT

A. *Datasets Used*

The two main datasets used are the ExDark Image dataset [15] to test the performance of our proposed training method and the Google Image Scraped dataset [16] to experiment with various masking methods to create an augmented training dataset and subsequently train the deep learning architecture. The former is a dataset of 7363 low-light images with light sources of varying sizes. The latter is a custom dataset of images scraped from Google, with images belonging to 4 categories: Art & Culture, Architecture, Food & Drinks, Travel & Adventure.

B. Improving image enhancement using deep learning architecture by dataset augmentation

The results of the use of a Low Light Convolutional Neural Network (LLCNN) [2] are seen in Fig. 2. An LLCNN is typically a deep learning architecture that is trained on a dataset consisting of well-lit images and their corresponding darkened versions. The neural network is trained to extract the former from the latter. With sufficient training examples, it has been observed that this architecture is successfully able to extract an enhanced image from a low-lit image, in cases where the dim illumination is uniform. In Fig. 2., it is observed that the LLCNN gives satisfactory results for uniformly dark images (row 1), while it fails to perform well on images with varying brightness and contrast (row 2 and 3). In order to enable the neural network to overcome this drawback, brightness jitter and contrast jitter were added, followed by Gaussian noise to the training images. Jitter refers to the random replacement of pixels in an image. By training our neural network on these images, it is able to learn to deal

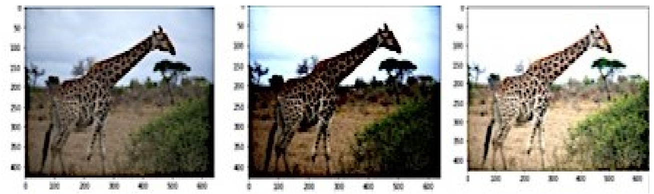

Fig. 3. Sample of the jitter added to the training data - from left to right: original image, with contrast jitter and with brightness jitter

with varying light intensities and contrast areas. After adding jitter to the training images (Fig. 3), a moderate spike in the performance of the algorithm was noted, noticeable enough to conclude that the training dataset has a huge impact on the outcome of the algorithm.

While jitter may increase the enhancement power of the LLCNN, the problem arises in dealing with images that have light sources. In order to overcome this, pseudo-light sources were introduced in the training image dataset, experimenting

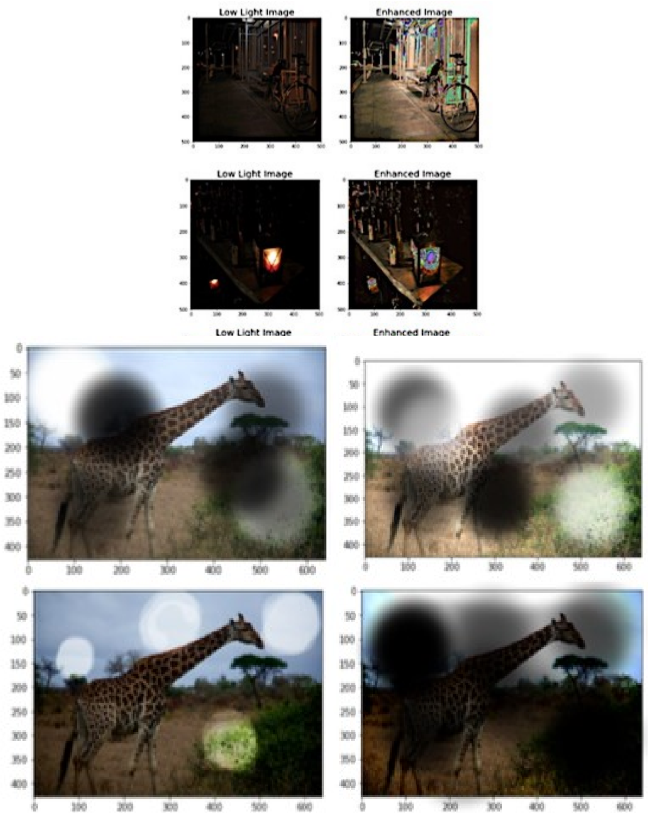

Fig. 4. Visualization of the modifications made to the training images.

with masks of varying sizes and shapes applied to the images. By applying masks with round and ovular areas of darkness

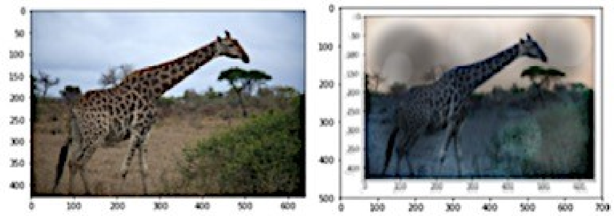

Fig. 5. The resultant image, after adding all the modified images

and subsequent brightness (Fig. 4), a significant decrease in the uniformity of illumination for the training input was observed, for the same ground truth image.

Following the improved performance of our algorithm using this technique of masking, all of the masked images were added together, to obtain a single training image per ground truth image. The final ground truth along with the input training image after masking and blurring is shown in Fig. 5.

Despite the significant increase in the performance of the neural network on non-uniformly lit images, the architecture now failed to give satisfactory results on uniformly bright images. So, in the inclusion of this algorithm in a larger system, the first step would be to take a histogram of the image before the input to the neural net. If the histogram shows the image is sufficiently bright, this LLCNN phase of the system can be bypassed.

C. Extraction and Visualization of Features from the Convolutional Layers

In order to obtain a deeper insight to the enhancement of images using convolutional neural networks, we've analyzed the output of each layer of the convolutional layers of a VGG-16 [13], which is a sufficiently complex architecture, with a number of layers. The output of the convolutional layers (the convolutional features) has been analyzed. For instance, for the input image (top left corner), the convolved output at each layer of the neural network is seen in Fig. 6. For the feature extraction, five categories of animal images viz., bear, elephant, leopard, cheetah and tiger have been considered.

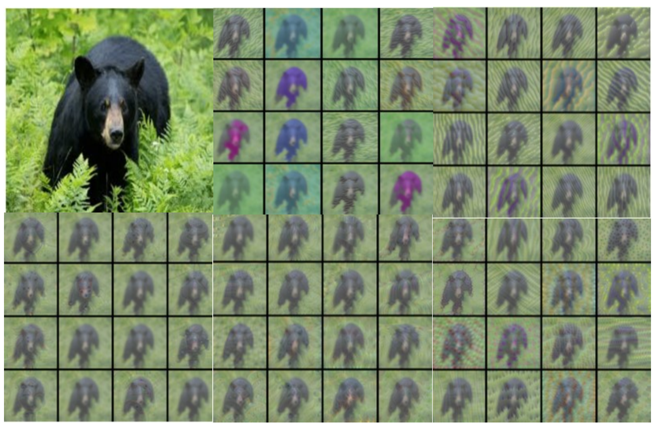

Fig. 6. Visualization of the output of various layers of the deep learning architecture for the input image (seen top left corner). The top row of images is the output of layers 1 and 2. The bottom row is the output of layers 3-5.

Fig. 7. Heatmap of the important convolutional features used to identify the object in the image

From the convolved output of each layer, it is observed that the most important features at that layer are being extracted and given more importance. For a more thorough understanding, the most important feature was extracted and placed as a heatmap over the original image. In order to find which layer extracts the most distinguishing feature, the method described in the following section is used.

D. Kernel Difference Maximization Method

The main factor in the extraction of features is the kernel, which is essentially a matrix of weights convolved over the input image. Since the weights determine the importance given to a feature, if the kernel values are extracted from the convolutional layers, the values that enhance the feature extraction in non-uniformly lit images can be analyzed. In order to identify the differentiating kernel value, a novel method of kernel difference maximization is detailed below.

Keeping the neural network architecture constant, for varying input images, the matrix of kernel values is retrieved at each layer. The layer at which the difference between the two matrices is maximum is the distinguishing layer of the VGG-16. This is justified by the fact that the kernel values matrix is essentially a matrix of the importance values given to the image by each layer. The layer which gives different importance to a particular region (feature) is the distinguishing layer.

Thus having identified the distinguishing layer of the architecture using kernel values, the features extracted by these layers are labeled as the most important features determined by the neural network. Using these features, the deep learning architecture is able to classify images of these categories with over 90% confidence for every category considered. These are further plotted as a heatmap on top of the original image as seen in Fig. 7, a method adapted from [14]. A bear's muzzle, a leopard/cheetah's spotted pattern, a tiger's stripes and an elephant's trunk, tusks and ears are seen to be the most distinguishing features of the image, as one may intuitively guess. The feature extracted seems to depend on the view of the animal (head-on/lateral) and the level of occlusion of the animal in the image. Nevertheless, the confidence of the neural network did not fall below 90%.

The experiment was repeated on night-time images, where it was observed that the neural network extracts (incorrectly) the

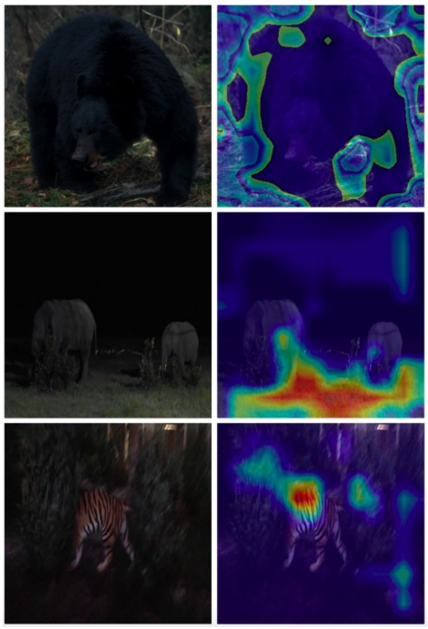

Fig. 8. Poor performance of the feature extraction process on low-light images.

background and the void's shape (Fig. 8), in order to attempt to predict the object in the image. The confidence of each prediction was reduced significantly to under 20% for all categories.

In order to overcome this shortcoming, the dark images were given to an LLCNN trained on the augmented dataset. Following this, the enhanced images were fed into the kernel difference calculator to determine the layer that extracts the most important feature. Then, these important features were plotted as a heatmap over the original image and the results

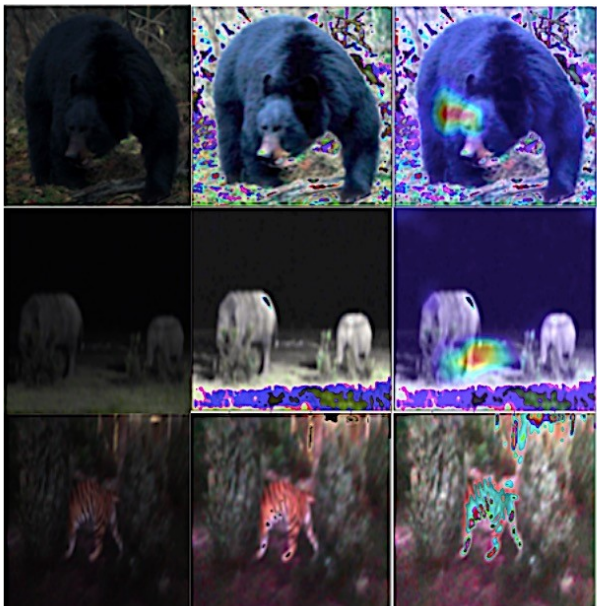

Fig. 9. Improved performance of the feature extraction process on enhanced low-light images

are seen in Fig. 9. The convolutional features plotted are used by the deep learning architecture and it is able to label the images with increased confidence (over 80% for all categories).

V. Results And Discussion

Augmentation of the training image dataset with masks producing pseudo-light sources to give an illusion of non-uniform illumination was performed and an increased performance was noted. In training the Low Light Convolutional Neural Network (LLCNN) with this dataset, a loss reduction (a reduction in the sum total of errors committed by the neural network on the input at each iteration) from 7362.3 to 201.3 was noted. The trained LLCNN is capable of enhancing images of varying levels of illumination – both uniform and non-uniform. In the classification of the images, the confidence levels of the neural network are over 90% for well-illuminated images of all categories considered. For the unenhanced dark images, the confidence interval did not exceed 20% for any of the categories involved. After enhancement with the trained LLCNN, given the most probable important layer for identification, the classification confidence of the enhanced dark images was greater than 80% for all categories. From this, it can be concluded that the kernel difference maximization method works satisfactorily when used in combination with an enhancement deep learning architecture like the LLCNN.

VI. Conclusions

Thus a novel method of object detection in non-uniformly illuminated images is proposed. An augmented dataset was generated by placing masks over the normal image dataset to create an illusion of non-uniform illumination and light sources in the training image dataset. It was observed that the enhancement capacity of the neural network increased significantly, after being trained with this augmented dataset. The kernel difference maximization method was introduced as a method to extract the most important features from images. Having extracted the important features and fed them back to the neural network, it is now capable of predicting labels for objects in images with higher confidence of over 80% for the five categories of images considered. Combining both of these stages (i.e.,) the image enhancement and the extraction of convolutional features, a heatmap of the 'important regions' was placed over the original image to gain an insight into the working of the convolutional neural network. Further, it was shown that this new architecture consisting of an LLCNN stage followed by an object detection neural network (here, a VGGNet with 16 layers) can be effectively used to detect animals in non-uniformly lit images, commonly seen in CCTV footage and surveillance cameras. This architecture will prove extremely useful in areas of settlement in and around forest regions, where vicious animals may encroach living spaces.

References

[1] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495, 2017. Available: 10.1109/tpami.2016.2644615.

[2] L. Tao, C. Zhu, G. Xiang, Y. Li, H. Jia and X. Xie, "LLCNN: A convolutional neural network for low-light image enhancement", *IEEE Visual Communications and Image Processing*, pp. 1-4, 2017.

[3] L. Shen, Z. Yue, F. Feng, Q. Chen, S. Liu, J. Ma, MSR-net: Low-light image enhancement using deep convolutional network, Nov. 2017, [online] Available: https://arxiv.org/abs/1711.02488

[4] L. Jiang, Y. Jing, S. Hu, B. Ge and W. Xiao, "Deep Refinement Network for Natural Low-Light Image Enhancement in Symmetric Pathways", *Symmetry*, vol. 10, no. 10, p. 491, 2018. Available: 10.3390/sym10100491.

[5] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding," *Procedings of the British Machine Vision Conference 2017*, 2017.

[6] S. Padmavathi, K. P. Soman, and R. Aarthi, "Image Restoration Using Knowledge from the Image," *Advances in Computing and Information Technology Advances in Intelligent Systems and Computing*, pp. 19–25, 2013.

[7] R. Jiang, et al. "DeepUrbanEvent: A System for Predicting Citywide Crowd Dynamics at Big Events." in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2019.

[8] S. Divya, S. Kiruthika, A. L. Nivin Anton, and S. Padmavathi, "Segmentation, Tracking and Feature Extraction for Indian Sign Language Recognition", *International Journal on Computational Science & Applications*, vol. 4, no. 2, pp. 57-72, 2014. Available: 10.5121/ijcsa.2014.4207.

[9] F. M. Graetz, "How to visualize convolutional features in 40 lines of code," *Medium*, 21-Jul-2019. [Online]. Available: https://towardsdatascience.com/how-to-visualize-convolutional-features-in-40-lines-of-code-70b7d87b0030

[10] K. D. M. Dixon, V. Sowmya, and K. P. Soman, "Effect of Denoising on Vectorized Convolutional Neural Network for Hyperspectral Image Classification," *Lecture Notes in Electrical Engineering Computational Signal Processing and Analysis*, pp. 305–313, 2018.

[11] D. Vasani, "Heatmaps and Convolutional Neural Networks Using Fast.ai," *Medium*, 28-Oct-2019. [Online]. Available: https://heartbeat.fritz.ai/heatmaps-and-convolutional-neural-networks-using-fast-ai-16d5b7d02a86

[12] A. Paliwal, "Understanding your Convolution network with Visualizations," *Medium*, 05-Oct-2018. [Online]. Available: https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b

[13] H. Rawlani, "Visual Interpretability for Convolutional Neural Networks," *Medium*, 09-Jan-2019. [Online]. Available: https://towardsdatascience.com/visual-interpretability-for-convolutional-neural-networks-2453856210ce

[14] F. Chollet, "Visualising What ConvNets Learn," in *Deep Learning with Python*, 1st ed., Manning Publications, 2017.

[15] Y. Loh and C. Chan, "Getting to know low-light images with the Exclusively Dark dataset", *Computer Vision and Image Understanding*, vol. 178, pp. 30-42, 2019. Available: 10.1016/j.cviu.2018.10.010.

[16] D. Dutta, "Google Scraped Image Dataset," *Kaggle*, 24-Sep-2018. [Online]. Available: https://www.kaggle.com/duttadebadri/image-classification/activity.