JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Convolutional Neural Network Based Blur Images Detection

Ashwini Burkule[1], Shardul Badwanay[2], Rohit Pharate[3], Komal Wanzare[4], Chandrashekar Patil[5]
Department of E&TC, SKNCOE, SPPU, Pune

Abstract— **Digital photographs are becoming increasingly common because of the proliferation of digital cameras, which increases the need for automated image quality evaluation. Several techniques are considered beneficial for detecting blurriness; however, in this system, we present and test a novel technique that uses a convolutional neural network to identify whether or not a picture is blurry. The suggested system employs. This system utilized the CERTH dataset for evaluation. The proposed system shows promising results. This study proposes a method for blur image classification using a convolutional neural network (CNN) algorithm. Fully connected and max-pooling layers follow convolutional layers in the CNN design. The findings demonstrate that the suggested technique accurately differentiates between blurry and non-blurry photos. The performance of the proposed method is assessed using a dataset of blurry and non-blurry photographs. The proposed method can be helpful in various applications that require automatic image blur detection, such as image restoration, surveillance, and medical imaging.**

Keywords— ***Blur detection, CNN, Deep learning, Image enhancement*.**

I. Introduction

Today, enormous identification databases like Adhaar, the electoral commission, etc., need to be cleaned up. Millions of face photos and other pertinent information may be found in these databases. These face photographs may be subjected to many forms of analytics, such as de-duplication (detecting duplicates using face images) and facial recognition (for social purposes), among others. Only after evaluating the quality of these database photographs against several criteria can the analytics mentioned above function effectively.

These parameters evaluate blur, deterioration, suitable dimensions and resolutions, among other things. Therefore, evaluating picture quality is a crucial first step in cleaning image databases. The assessment of image quality (IQA) is a persistent problem in computer vision. A challenge in IQA is the No-Reference Image Quality Assessment, which evaluates an image without any reference [1]. Much research has been done to address the subjective and perceptual nature of picture evaluation. Evaluating an image's quality is an important problem since it has several uses, such as preprocessing for image calculations and assessing the quality of digital camera pictures.

We must separate the hazy pictures from the remainder of our scenario. A blurred picture is distinct from a distorted or degraded image; it is essential to note. We first used statistical methods to extract differences from photos. However, owing to convolutional neural networks' incapacity to differentiate between a clear and fuzzy image, we moved on to deep neural networks after their invention since they are very successful with images and much more intuitive for categorization. They may be used to any vision-based categorization issue and are very resilient. Additionally, CNN [2] does away with the traditional hand-crafted feature format. The ability to immediately upload raw photos to the network is a bonus.

Many home users have amassed increasing digital images as digital cameras have become more widely used. Additionally, the photographers hired to capture rituals like weddings acknowledge that up to 40% of the photographs are of poor quality to be offered to clients [1]. Blur is one of the leading causes of quality deterioration. Therefore, automated blur detection determines whether or not a particular photograph is blurry and how to blur it is highly desirable to aid these users in discarding the photos.

In computer vision, blur detection is a crucial and fascinating task. Choosing effective characteristics to distinguish between distorted and clean picture parts is a crucial aspect of blur detection. There are other ways to solve this issue, but mostuse the two-step method to separate the clear and blurred zones. A picture's components are first carefully created using a range of empirical data in gradient. The distorted and clear sections are then distinguished using a binary classifier. This study focuses on the Laplacian variance and CNN, two essential methods for recognizing fuzzy pictures.The strictness of traditional edge detection techniques like the Laplacian of Gaussian (LOG) is a benefit. They require further post-processing, however. Edges and unplanned changes in image fragmentation may be found using differential filters or even the LOG, according to El-Sayed and Sennari1. The pixels crucial for providing the critical characteristics of the edges are processed after the masks have been moved around the picture. Due to the injection of noise in the face of mask motion around the picture, this method of edge identification may be inaccurate. CNN has a simple operation, making it effective at detecting edges and trustworthy at reducing noise impacts.

Blur detection using convolutional neural networks (CNNs) involves training a model to classify images as blurry or sharp. Here are the general steps for implementing a CNN for blur detection:

- Data Collection: Collect a dataset of blurry and sharp images.
- Data Preprocessing: Preprocess the data by resizing the images to a fixed size, normalizing the pixel values, and augmenting the data to increase the size of the dataset.
- Model Architecture: Design a CNN architecture that takes in an image as input and outputs a binary classification (blurry or sharp).
- Training: Train the CNN using the preprocessed data. Use an optimizer like Adam or SGD and a loss function like binary cross-entropy.
- Testing: Evaluate the trained model on a separate test set to assess its accuracy in detecting blur.
- Fine-tuning: Fine-tune the model by adjusting hyperparameters, changing the architecture, or using transfer learning to improve performance.
- Deployment: Deploy the trained model to detect blur in new images.

II. Literature Survey

The blur detection method described by Renting Liu, Zhaorong Li, and Jiaya Jia [3] includes a framework for analyzing and categorizing different blurs without deblurring. They employed several blur features created from an image's colours and spectrum and gradient information, and they made intense training & classification decisions using these feature parameters. The blur is divided into two categories: out-of-focus blur, often brought on by lenses that are out-of-focus from the subject in a picture, or motion-blurred areas brought on by the subject's motion.

A unique method for identifying motion blur and out-of-focus blur was suggested by Beomseok Kim, Hyeongseok Son, Seong-Jin Park, Sunghyun Cho, and Seungyong Lee [4]. They suggested combining high-level contextual information with low-level structure information using a deep encoder-decoder network with lengthy residual skip links. This study performs better than other cutting-edge techniques. However, this study could not consider complicated scenarios of picture blurring since it used a small dataset.

A method to identify global motion blur from a video source was put forward by Karl S. Ni, Zachary Z. Sun, and Nadya T. Bliss [5]. The method has two ways of operating: first, it generates a blur metric from any single frame or picture; second, it adds temporal information by correlating data from neighbouring frames in a video stream. This approach has the benefit of blur detection from still pictures in that it employs neighbouring reference frames from movies. Additionally, it is simple to comprehend and produces excellent blur detection results.

Bing Li and Zhen Huan Zhan covered the topic of recovering a damaged picture using linear and inverse filter approaches [6]. The criteria employed in recovering blurred pictures are motion-blurred distance and motion-blurred direction. The scientists employed mean square error to assess the quality of recovered and blurred photos.

A spatially invariant kernel-based blur detection technique that uses blurred-edge profiles was covered by Taeg Sang Cho [7]. The study suggested a hardware and software solution for blur detection and removal. The software-based techniques use phase information and blur kernel estimation using blurred line profiles. The hardware approach, however, calls for using a camera that enhances local motion estimates via calculations.

Dong Gong and Jie Yang [8] suggested a pixel-wise linear motion blur model for heterogeneous motion blur. The approach the authors suggested uses a fully-convolutional deep neural network to estimate a dense motion flow map. They employed a dataset that includes both real-world and artificial pictures. The study's findings for actual photos with diverse motion blur are encouraging.

A method proposed by Shuang Zhang and Ada Zhen [9] involves passing a pair of input pictures through denoising and deblurring encoders before merging and passing them through a deblurring decoder. From the pair of pictures, the two encoders extract complementary information. The information is combined simultaneously via the merging. However, their approach requires pairings of hazy photos, which is a disadvantage.

For photos with non-uniform motion blur, Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce [10] suggested a unique deblurring method based on convolutional neural networks. This deep learning technique predicts the motion blur's patch-level probabilistic distribution. A non-uniform deblurring model removes the motion blur after ensuring motion smoothness using a Markov random field.

Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen [11] presented a unique approach to removing motion blur based on the high sparsity of the motion blur kernel in the curvelet system and that of the image in the framelet system. The technique differs from current methods because it does not require previous kernel knowledge. The algorithm was thoroughly evaluated on synthetic and real-world photos, and the results were encouraging compared to earlier methods.

The blind image deconvolution method by R. Fergus [12] tries to predict the blur filter and latent unblurred pictures. It is an issue that is poorly presented. Although several techniques for picture deblurring have recently been described, most exclusively address spatially invariant blur, in which the same PSF blurs every pixel of the input image.

Zhang and Bergholm [13] devised the Gaussian Difference Signature, which functions similarly to the first-order derivative of Gaussian, to assess the diffuseness caused by out-of-focus objects. These methods assume that a Gaussian blur filter simulates the Point Spread Function (PSF). They are useless for detecting ongoing non-Gaussian blur.

L. Bar [14] suggested using user interaction or the blur kernel assumption to address the partial blur issue. If the PSF can be successfully rebuilt for all of these approaches, the kind of blur may also be determined using the PSF's structure. Even with strong assumptions on the image and kernel structures, blind deconvolution often performs poorly in reality. It struggles to handle our slightly blurred photos. Furthermore, a visually good deconvolution result does not always indicate accurate PSF estimation. Blind deconvolution is unsuitable for general blur detection due to these issues, mainly when dealing with photos in an extensive database. Auto-segmentation of Low Depth of Field (DoF) images is another sort of blur analysis. By clearly focusing on an Object of Interest, the photographer's aim is abstracted by the low depth of field method (OOI).

Automatic OOI extraction techniques suggested by C. Kim [15] are inadequate for our blur detection since they only function on low DoF input photos with out-of-focus backgrounds. Low DoF images are found by computing a low DoF indicator, which is the ratio of the wavelet coefficients in high-frequency of the core areas of the whole picture. This approach implies that out-of-focus pixels surround the centre of low DoF photos with focussed objects. Additionally, this approach is inappropriate for our all-purpose blur detection.

For usage in a mobile phone, Razligh and Kehtarnavaz [16] presented a deblurring picture algorithm. This deblurring technique corrects low-exposure photos by taking into account the brightness and contrast of the blurred input images. Contrarily, Levin's approach from 2006 [17] employs a technique based on inferred blur kernel. The blur and non-blur layers are separated from the picture by an energy function constructed using this kernel. According to Elder and Zucker [18], just the amount of the blur is assessed; no technique has been developed to identify the region of the picture as being blurred or not.

III. Proposed System

The block diagram of the proposed system is shown in Fig. 1.

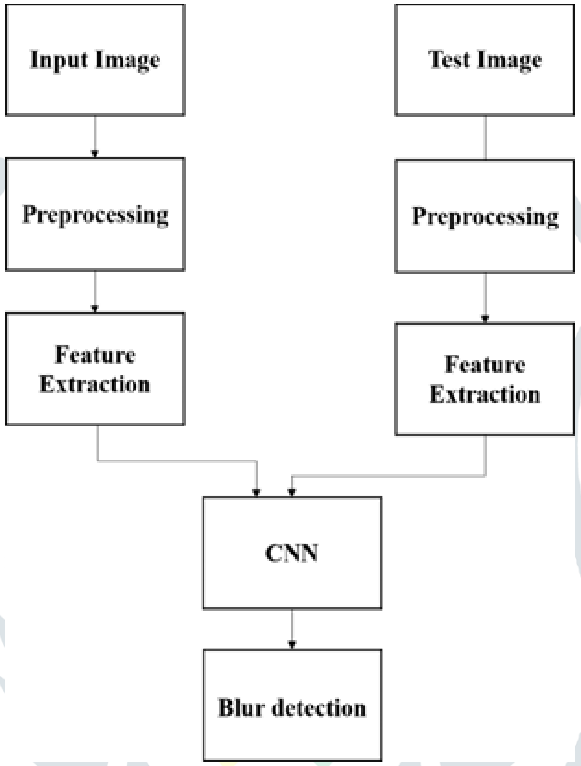


Fig. 1. Block Diagram of the proposed system

A. Dataset

The suggested system uses the CERTH dataset. The CERTH image blur dataset comprises 2450 digital pictures, of which 1850 are unaltered photos taken using different camera types under varying lighting circumstances. The remaining 600 are photos that have been digitally blurred. They were produced by applying various Gaussian, motion, and circular averaging filters on 60 undistorted photos that were chosen at random. The dataset will be preprocessed to be ready for classification purposes. The preprocessing techniques include resizing, ROI selection, etc. operations. The model will first be trained with the Laplacian method and the convolutional neural network. The test samples will be tested with both algorithms and compare their performance using sensitivity, specificity, and accuracy parameters.

The dataset consists of three classes, Artificially-Blurred, Naturally-Blurred, and Undistorted. Sample images of CERTH blur detection are shown in Fig.2.

Fig. 2. Sample images of the CERTH dataset

The dataset distribution is shown in Table I.

Table I: Dataset Distribution

Dataset label	Total images	Training images	Testing images
Artificially-Blurred	150	120	30
Naturally-Blurred	220	176	44
Undistorted	630	504	126

B. *Training using CNN*

The architecture of the CNN algorithm is shown in Fig. 3.

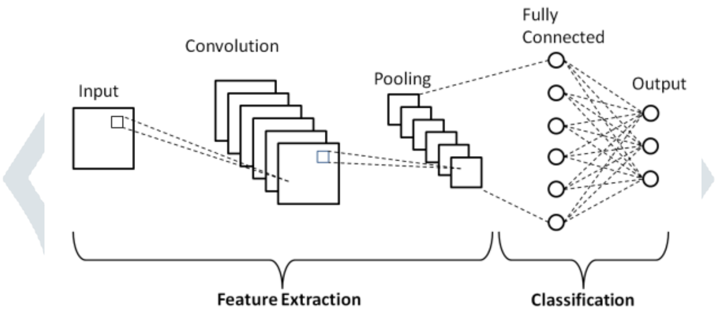


Fig. 3. Block Diagram of CNN algorithm

Convolutional neural networks (CNNs) contain groups of neurons connected by a standard set of parameters. They have several filtering layers, like most neural networks, and each one performs an affine transformation on the vector input before applying an element-wise non-linearity. The affine transformation may be applied to convolutional networks as a discrete convolution rather than an utterly generic matrix multiplication. Convolutional networks may scale to huge pictures due to their efficient computational design. Additionally, the equivariance of the translation is accounted for (in other words, if the image is shifted by one pixel to the right, the output of the convolution is also shifted one pixel to the right; the two representations vary equally with the translation). When dealing with images in convolutional networks, a pooling layer is often used to aggregate the outputs of multiple neighbouring filters into one. To summarise the activations of group units, these pooling layers may use any filter, including their maximum, mean, or any other. These pooling layers make the network more robust against subtle changes to the input.

- *Convolutional Layer*

The convolutional layer is the core component of a convolutional network, which controls most of the computational effort. Feature extraction from the input data, an image, is the primary goal of the convolution layer. Convolution uses tiny squares from the input picture to retain the spatial connection between pixels to learn the image's attributes. A group of teachable neurons is used to change the input images.

The output image is left with a feature map or activation map as a consequence of this process; this map is then used as input data by the subsequent convolutional layer.

- *ReLU Layer*

The outcome is a feature map or activation map in the output image, which is subsequently utilized as input information by the next convolutional layer.

- *Pooling Layer*

Each activation map is less dimensional, but the pooling layer retains the mostcritical data. Using the provided photos, several non-overlapping rectangles are produced. Similar to previous sliding window approaches, it applies a statistical function to the contents of its window rather than weights that can be taught. The most well-liked pooling method, called max pooling, uses the contents of the window and the max() function. Other types are sometimes used, such as mean pooling.

- *Flattening Layer*

Using a convolutional neural network, high-resolution data is effectively resolved into representations of things. To "understand" the results and eventually offer a classification result, it is reasonable to think of the fully connected layer as adding a traditional classifier to the network's information-rich output. The output of the convolutional neural network's dimensions must be flattened to link this fully connected layer to the network. We should have a pooled feature map when the first two steps are finished.

- *Fully Connected Layer*

The FCL divides the input picture into categories based on the training dataset using these properties. The Softmax activation function classifier receives features from the final pooling layer or FCL. The FCL's output probabilities add up to one. Softmax is used as the activation strategy to guarantee this. The Softmax function may transform each real-valued score into a vector of summable values between 0 and 1.

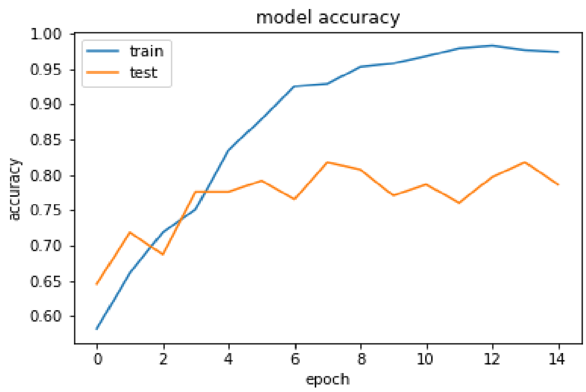
IV. RESULTS

In the proposed system, the CNN algorithm is used to classify the image into Artificially-Blurred, Naturally-Blurred, and Undistorted. The model summary of the CNN algorithm for the proposed system is shown in Fig.4.

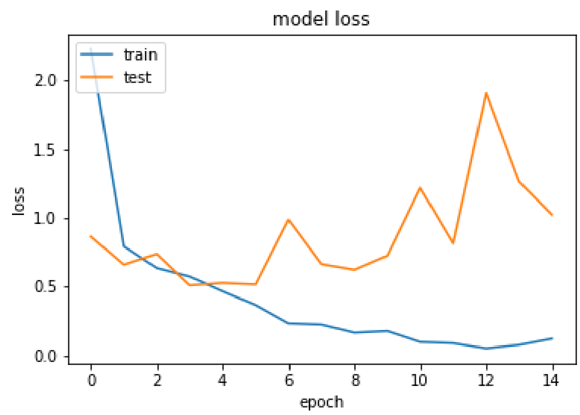
Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 223, 223, 256)	3328
activation_3 (Activation)	(None, 223, 223, 256)	0
max_pooling2d_2 (MaxPooling 2D	(None, 111, 111, 256)	0
conv2d_3 (Conv2D)	(None, 110, 110, 128)	131200
activation_4 (Activation)	(None, 110, 110, 128)	0
max_pooling2d_3 (MaxPooling 2D)	(None, 55, 55, 128)	0
flatten_1 (Flatten)	(None, 387200)	0
dense_2 (Dense)	(None, 64)	24780864
activation_5 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 3)	195

Fig. 4. Training parameter of CNN algorithm used in the proposed system

The proposed system shows a training accuracy of 97.37% and while validation accuracy of 78.65%. The execution time for the training is 11980 seconds.



(a)

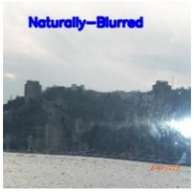


(b)

Fig. 5. Training progress graph of CNN algorithm (a) model accuracy (b) model loss

The qualitative analysis of the proposed system is shown in Table II. Table II shows that the proposed system accurately classifies the images into an artificial blur, natural blur and undistorted classes

Table II: Result of the proposed system

Image Type	Input image	Output image	Image Type	Input image	Output image
Artificially-Blurred			Naturally-Blurred		
					
					
					
Undistorted Input Images			Undistorted Output images		
					

.

V. CONCLUSION

Blur detection and segmentation techniques are used to remove the blur from the image source and extract the perfect quality of the image using the techniques that stay anticipated. This project uses the CNN algorithm to classify images into Artificially-Blurred—natural-Blurred and undistorted images. The proposed system achieved a training accuracy of 97.37% and a validation accuracy of 78.65% on the CERTH dataset.

In the future, the proposed system can be extended for the amount of blur in the image using a deep learning algorithm.

REFERENCES

[1] H. Talebi and P. Milanfar, "Nima: Neural image assessment," IEEE Transactions on Image Processing, vol. 27, no. 8, pp. 3998–4011, Aug 2018.

[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, May 2010, pp. 253–256.

[3] Renting Liu, Zhaorong Li & Jiaya Jia, "Image partial blur detection and classification" in CVPR 2008.

[4] Beomseok Kim, Hyeongseok Son, Seong-Jin Park, Sunghyun Cho, and Seungyong Lee, “Defocus and Motion Blur Detection with Deep Contextual Features” in Computer Graphics Forum 37 (7) (2018)

[5] Karl S. Ni, Zachary Z. Sun & Nadya T. Bliss, "Realtime global motion blur detection" in 9th IEEE International Conference on Image Processing, 2012

[6] Bing Li, ZhenHuan Zhan, "Research on Motion Blurred Image Restoration," in 5th International Congress on Image and Signal Processing, 2012.

[7] Taeg Sang Cho, "Motion Blur Removal from Photographs," in. MIT PhD dissertation (2010)

[8] Dong Gong, Jie Yang, "From Motion Blur to Motion Flow: a Deep Learning Solution for Removing Heterogeneous Motion Blur," CVPR, 2017

[9] Shuang Zhang, Ada Zhen, "Deep Motion Blur Removal Using Noisy/Blurry Image Pairs," Cornell University, 2019

[10] Jian Sun, Wenfei Cao, Zongben Xu, Jean Ponce, "Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal," CVPR, 2015

[11] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen, "Blind motion deblurring from a single image using sparse approximation," CVPR, 2009

[12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. ACM Trans. Graph., 25(3):787–794, 2006.

[13] W. Zhang and F. Bergholm. Multi-Scale Blur Estimation and Edge Type Classification for Scene Analysis. International Journal of Computer Vision, 24(3):219–250, 1997.

[14] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In ICCV, 2007.

[15] C. Kim. Segmenting a low-depth-of-field image using morphological filters and region merging. IEEE Transactions on Image Processing, 14(10):1503–1511, 2005.

[16] Q. R. Razligh, and N. Kehtarnavaz, "Image blur reduction for cellphone cameras via adaptive tonal correction," In International Conference on Image Processing, ICIP, pp. 113-116, 2007.

[17] A. Levin, "Blind motion deblurring using image statistics," In Neural Information Processing Systems, NIPS, pp. 841–848, 2006.

[18] J. H. Elder, and S. W. Zucker, "Local scale control for edge detection and blur estimation," Pattern Analysis and Machine Intelligence, PAMI, vol. 20, no.7, pp. 699–716, 1998