Performance Evaluation of Deep Learning Approaches for Blurred Image Processing

A PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

MASTER OF TECHNOLOGY

IN

Information Technology

Submitted by:

Mohit Tanwar

(23/ITY/21)

Under the supervision of

Dr. Seba Susan

DEPT. OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042
MAY 2025

CANDIDATE’S DECLARATION

I, **Mohit Tanwar**, Roll No. **2K23/ITY/21**, a student of **M. Tech (Information Technology)**, hereby declare that the Project Dissertation titled "**Performance Evaluation of Deep Learning Approaches for Blurred Image Processing**”, which I submitted to the Department of Information Technology at Delhi Technological University, is original and has not been plagiarized, either in part or in whole, without proper citation. Furthermore, no degree, certificate, associateship, fellowship, or any comparable title or honor has ever been granted based on this work.

Date: **Mohit Tanwar**

Place: New Delhi 2K23/ITY/21

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Performance Evaluation of Deep Learning Approaches for Blurred Image Processing**" submitted by **Mohit Tanwar**, Roll No. **2K23/ITY/21**, from the Department of Information Technology at Delhi Technological University, Delhi, fulfills the requirements for the partial completion of the Master of Technology degree. This dissertation is a record of the project work completed by the student under my supervision. To the best of my knowledge, this study has not been submitted for a degree or diploma at this university or any other institution.

Place: New Delhi	**Dr. Seba Susan**
Date:	SUPERVISOR

ABSTRACT

It has often been stated that an image is worth more than a thousand words. The details in a picture contain both high-level and low-level elements that help to distinguish across image classes and aid in categorization. Image blurring is an undesired phenomenon caused by the image capture method or equipment. In this research, we compare the feature extraction and classification performance of five modern deep pre-trained models to increasingly blurred pictures of handwritten digits from the MNIST handwritten digits dataset. The photos are gradually blurred, first using a Gaussian blur of Sigma=5, then with a Gaussian blur of Sigma=8. Sigma is also known as standard deviation; the higher the Sigma, the greater the degree of blur. The deep pre-trained models under consideration are VGG-16, DenseNet-121, Xception, ShuffleNet, and SqueezeNet, which were pre-trained on the ImageNet dataset and then shallow-tuned on the blurred pictures. DenseNet-121 had the greatest accuracy across deep learning models, at 98.77% for Sigma=5 and 98.62% for an enhanced blur of Sigma=8. With the exception of ShuffleNet, all other model accuracies decreased dramatically when the degree of blur rose (Sigma from 5 to 8). Comparisons with machine learning models such as Support Vector Machine (SVM), Convolutional Neural Network (CNN), and logistic regression show that logistic regression outperforms other machine learning models, despite having lower accuracies than the majority of deep learning models. We find that DenseNet-121, followed by ShuffleNet, are among the strongest modern models for classifying properly under progressive blur.

We conducted experiments to determine the best deblurring technique capable of sharpening images from blurred observations using various deblurring techniques, followed by computer vision image processing applications on the GoPro dataset, which consists of pairings of a realistic fuzzy image and an accompanying ground truth sharp image obtained by a high-speed camera. The photos are deblurred using any deblurring method, and PSNR and SSIM values are compared between the blurred and deblurred outputs. The deblurred output is then processed by a computer vision program, which improves the features and sharpens the objects and edges. After incorporating computer vision applications, the parameters SSIM and PSNR are assessed again. To begin, the techniques are used to compute the values and to construct ways for a dependable combination of deblurring methods, which ultimately helps to generate a crisp edged and less distorted image from a hazy image with PSNR 29.84dB and SSIM value 0.70. This study suggests that deblurred approaches used in combination can function more efficiently and reliably than individual methods, resulting in more accurate and less distorted pictures that give greater insights into images and aid in properly identifying objects in photos.

Keywords:

Blurred images; Gaussian Blur; Sigma; Deep learning; Machine learning; MNIST dataset; Blurred images; Wiener Filter; Lucy-Richardson for deblurring; Computer Vision; GoPro dataset; Histogram Equalization; Mean of image pixels

ACKNOWLEDGEMENT

I would like to thank my project guide, Dr. Seba Susan, for her invaluable assistance and expertise in developing this research. I respectfully express my gratitude to Dr. Dinesh Kumar Vishwakarma (Head of the Department of Information Technology) and other IT faculty members for their invaluable assistance and time whenever needed. I would want to thank all of my DTU friends who were always there to support me as I worked on this thesis. Special appreciation to the Almighty Lord for providing me life and the strength to persevere in my effort. Last but not least, I want to thank my family for believing in and supporting me.

Mohit Tanwar
2K23/ITY/21
M.Tech (Information Technology)
E-mail: tanmohit1310@gmail.com
Department of Information Technology
Delhi Technological Technology

TABLE OF CONTENTS

CANDIDATE'S DECLARATION **2**

CERTIFICATE **3**

ABSTRACT **4**

ACKNOWLEDGEMENT **6**

LIST OF FIGURES **9**

LIST OF TABLES **11**

1. INTRODUCTION **12**

2. LITERATURE REVIEW **17**

3. METHODOLOGY **26**

(a) Adopted Methodology for Blurred Images: 26

(b) Adopted Methodology for Deblurred Image 27

4. EXPERIMENTS AND RESULTS **29**

4.1.BLURRED IMAGES 29

4.2.K-FOLD CROSS-VALIDATION 30

4.3.VGG16 (Visual Geometry Group-16) 30

4.4.DENSENET 32

4.5.SHUFFLENET 33

4.6.INCEPTION 34

4.7.XCEPTION 35

4.8.DEBLURRED IMAGES 36

4.9.WIENER FILTER 38

4.10. LUCY-RICHARDSON DEBLURRING 38

4.11. PROPOSED METHOD..........39

5. RESEARCH GAPS AND CHALLENGES..........42

6. CONCLUSION..........43

REFERENCES..........44

PUBLICATION DETAILS..........52

PUBLICATION CERTIFICATE..........53

ACCEPTANCE LETTER..........54

LIST OF FIGURES

S.No.	Name	Caption	Page No.
1.	Figure 2.1	(a) An example MNIST image of the digit “7,” (b) A collection of 100 samples from the MNIST training dataset	17
2.	Figure 2.2	EMNIST Dataset for English Alphabets	18
3.	Figure 2.3	Diagram for data processing using DL	18
4.	Figure 2.4	Accuracy of Testing and Validation	19
5.	Figure 2.5	Sigmoid Activation Function	20
6.	Figure 2.6	Convolutional Neural Network (CNN)	21
7.	Figure 2.7	(a) Training set accuracy curve; (b) Training set loss curve.	21-22
8.	Figure 2.8	Artificial Neural Network	22
9.	Figure 2.9	Recognition result using adopted techniques	23
10.	Figure 4.1	Original image and blurred image using Gaussian blur (σ = 5) on MNIST dataset	29
11.	Figure 4.2	Original image and blurred image using Gaussian blur (σ = 8) on MNIST dataset	29
12.	Figure 4.3	Workflow of the adopted methodology for classifying blurred images	30
12.	Figure 4.4	Adopted VGG16 Neural Network Architecture	31
13.	Figure 4.5	Adopted Layer-wise Detailed Architecture of VGG16	32
14.	Figure 4.6	Adapted DenseNet Architecture	33
15.	Figure 4.7	ShuffleNet Units (a) Bottleneck unit using depthwise convolution; (b) Standard ShuffleNet unit with pointwise group convolution and channel shuffle; (c) ShuffleNet unit configured with stride = 2	34

16.	Figure 4.8	Adapted Structure of the Inception v3 Model	35
17.	Figure 4.9	Adopted Architecture of the Xception Deep CNN Model	36
18.	Figure 4.10	(a) Blurred image from the GoPro dataset, (b) Corresponding deblurred image after applying the Wiener filter	37
19.	Figure 4.11	(a) Blurred image from the GoPro dataset, (b) Corresponding deblurred image after applying the Lucy-Richardson deblurring algorithm.	37
20.	Figure 4.12	a) Blurred image from the GoPro dataset, (b)Corresponding deblurred image obtained by applying the Lucy-Richardson deblurring algorithm followed by histogram equalization.	37
21.	Figure 4.13	Workflow of the Wiener filter applied to produce a deblurred image.	38
22.	Figure 4.14	Workflow of the methodology adopted to get the deblurred images.	39
23.	Figure 4.15	Workflow of the Proposed Method (1st Approach – Mean of Pixels of Deblurred Images from Two Different Methods)	40
24.	Figure 4.16	Workflow of the Proposed Method (2nd Approach – One Deblurring Method Followed by Another)	40

LIST OF TABLES

S.No.	Name	Caption	Page No.
1.	Table 4.1	Accuracy of Different Deep Learning Models on the MNIST Dataset	40
2.	Table 4.2	Performance Comparison of Deblurring Approaches on the GoPro Dataset with and without Histogram Equalization	41

1. INTRODUCTION

Computer vision and machine learning researchers widely use the Modified National Institute of Standards and Technology (MNIST) dataset. It has grayscale pictures of handwritten numbers from 0 to 9, each in the shape of a 28-by-28-pixel square (Baldominos et al., 2019). 60,000 training images and 10,000 testing images are included in this dataset. It is a reliable way to compare image classification models, especially neural networks (MNIST Database, 2025). MNIST is still a fundamental tool for creating and testing algorithms in supervised learning tasks because it is easy to use and has a standard format (Rosalina et al., 2020).

Researchers and professionals often use the MNIST dataset to learn more about and improve image recognition methods (Zhang et al., 2021). There are two parts to it: one is for training models, and the other is for testing how well they work.

a) The training set is a group of 60,000 labelled images, each showing a digit. These pictures teach machine learning models how to recognize patterns and features that go with each digit.

b) The testing set comprises 10,000 labelled images not used in the training process. Testing the model's performance on data it has not seen before is a way to check its accuracy and generalization.

The MNIST dataset has greatly aided the development and assessment of numerous machine learning and deep learning methods. Researchers have used it to train neural networks, CNNs (Convolutional Neural Networks), and other image analysis algorithms. Before using new models on increasingly complex image recognition tasks, it is frequently used as a preliminary benchmark for testing and validation. Even though MNIST offers a helpful starting point, real-world image classification problems frequently involve more complexity and variability. Notwithstanding its drawbacks, the dataset is still a helpful teaching tool and a starting point for solving more complicated computer vision issues.

Graphic noise and features are lowered by the familiar image processing technique known as Gaussian blur. Using a Gaussian kernel to convolve the image adds a weighted average to the surrounding pixels. Rapid changes in intensity are generally smoothed out by this averaging, giving the image a softer appearance. An image is convolved with a Gaussian kernel to apply the Gaussian blur function. This is a basic explanation of the working:

a.) **Gaussian Kernel Generation**: In image processing, smoothing tasks frequently employ a Gaussian kernel, a two-dimensional matrix that simulates the Gaussian distribution. The amount of blurring applied to the image depends on the kernel's size; larger kernels average over a larger neighbourhood of pixels, creating a more substantial blurring effect.

b.) **Convolution**: By centering the kernel over the target pixel and its surrounding neighbors, Every pixel in the image is applied with the Gaussian kernel. The new value of the central pixel is determined by multiplying each value in the kernel by the corresponding pixel intensity below it and then adding the results. This method successfully smoothes the image by averaging pixel values and giving those closer to the center more weight.

c.) **Normalization:** Normalizing the resultant pixel values after applying the convolution is frequently required. This process ensures that the original image's brightness and contrast remain intact. In order to prevent the filtered image from inadvertently becoming brighter or darker, each pixel value is usually normalized by dividing it by the sum of all the kernel weights. (Campbell, 2018).

d.) **Output Image:** Following convolution, a blurred version of the original image is created, with fine details and sharp edges that are softer by the Gaussian distribution.

The GoPro dataset, which includes 3,214 blurry images with a resolution of 1280 x 720 pixels (HD), is frequently used for image deblurring research. It is separated into 1,111 images for testing and 2,103 images for training. The dataset is intended to aid in creating and assessing algorithms intended to eliminate blur from pictures and videos. A GoPro Hero4 camera was used to take the pictures at 240 frames per second. In order to simulate realistic motion blur, each blurred image is created by averaging a series of consecutive sharp frames captured at high frame rates. Thirty-three scenes— indoor and outdoor—are included in the dataset, which is divided into 22 training and 11 testing sequences. To accurately evaluate deblurring performance, each scene folder includes pairs of blurred images and their corresponding sharp counterparts. This dataset is valuable training data for generative models and is essential to computer vision, especially in image and video restoration tasks.

A signal is converted from the spatial (or time) domain into the frequency domain by the Fourier Transform, *eq. 1.1* (Campbell, 2018). Tasks such as compression, deblurring, and filtering are made easier by this method in image processing. Operations like noise reduction and image enhancement are made easier and more effective by representing images in the frequency domain.

$$\boldsymbol{F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y).e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}} \tag{1.1}$$

Where:

u, v = Frequency components

$f(x, y)$ = Image in the spatial domain

$F(u, v)$ = Image in the frequency domain

The IFT (Inverse Fourier Transform), *eq. 1.2* transforms data from the frequency domain back into the spatial or time domain. It is used in image processing to restore the original image following frequency-domain adjustments or filters. This process recovers images following deblurring and the application of frequency filters, such as low-pass and high-pass filters.

$$(\boldsymbol{Fx}, \boldsymbol{y}) = \int\int_{-\infty}^{\infty} \boldsymbol{F}(\boldsymbol{u}, \boldsymbol{v}).\, \boldsymbol{e}^{\boldsymbol{j2\pi(ux+vy)}} \boldsymbol{du}\ \boldsymbol{dv} \qquad (1.2)$$

Where:

(Fx, y) = Image in the spatial domain

$F(u, v)$ = Image in the frequency domain

j = Imaginary unit

A well-known signal and image processing method, the Wiener filter is frequently used for image deblurring and noise reduction. It reduces the mean square error between the actual and estimated signals. By reducing the effects of noise and blur, the Wiener filter seeks to restore the original signal under the assumption that both the signal and the noise are stochastic processes. To carry out image deblurring, this method needs a blurred image, the blur kernel or Point Spread Function (PSF), and an estimated value for the noise power.

The Wiener filter is easy to use, quick, and efficient, especially when working with linear degradation models and Gaussian noise. However, it generally lacks the robustness and efficacy of contemporary deep learning-based methods, and its performance deteriorates when dealing with complex or non-linear motion blur. The Wiener filter (Garma, R. J. D., 2015), *eq. 1.3* is explained as follows in the frequency domain:

$$\boldsymbol{H_w}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{H*(u,v)}}{|\boldsymbol{H(u,v)}|^2 + \frac{\boldsymbol{S_n(u,v)}}{\boldsymbol{S_f(u,v)}}} \qquad (1.3)$$

Where:

H(u,v) is the Fourier transform of the blur kernel, also called the Point Spread Function (PSF).

H*(u,v) represents the complex conjugate of H(u,v).

S_n(u,v) stands for the power spectral density of the noise in the image.

S_f(u,v) is the power spectral density of the original, unblurred image.

H_w(u,v) refers to the Wiener filter expressed in the frequency domain.

A popular iterative method for recovering images damaged by PSF or blur kernel, is the Lucy-Richardson deblurring algorithm (Lee & Qiu, 2016). It is frequently used on datasets like GoPro and works exceptionally well for repairing astronomical observation images, out-of-focus blur, and motion blur. Sharp images are recovered by continuously improving its estimations using the observed blurred image and the known PSF. Based on Bayesian statistics, it assumes that the blur kernel is known or can be precisely estimated. Before processing, normalization of the input image to a [0,1] scale is needed, and the number of iterations can be changed, usually between 10 and 50. Furthermore, the technique can be adjusted to consider image noise, increasing its resilience for practical, real-world applications. Its iterative process improves the deblurring result gradually, making it stable in challenging blur scenarios and low light noise.

Histogram equalization *eq. 1.4* is a popular method for improving image contrast in computer vision and image processing (Survey Instrument Services, 2020). It redistributes an image's intensity values to make features more visible, particularly when low contrast is caused by bad lighting. The histogram, which counts the number of pixels at each gray level, is calculated first. A probability distribution is then created by normalizing this histogram. An improved image is then created by remapping the pixel values using the cumulative distribution function (CDF), which is calculated. In an 8-bit grayscale image, this technique distributes pixel intensities more evenly over the entire range of possible values, usually from 0 to 255.

$$\boldsymbol{s_k = T(r_{k)} = (L - 1)\,.\sum_{i=0}^{k} P(r_i)} \tag{1.4}$$

Where:

s_k= new pixel value after equalization

L = total number of gray levels

$\sum_{i=0}^{k} P(r_i)$ = Cumulative distribution function

This study suggested a novel strategy combining several deblurring techniques to take advantage of their complementary strengths to overcome the drawbacks noted in the current deblurring methods (Hemmatian, Hajizadeh, & Nazari, 2025). In the first method, two sets of deblurred images are created by independently applying the Wiener filter and the Lucy-Richardson algorithm, each following its steps. A final set of processed images is then produced by combining these deblurred outputs by averaging the pixel values from the two techniques. The quality of these combined images is assessed using SSIM (Structural Similarity Index Measure) and PSNR (Peak Signal-to-Noise Ratio) metrics. Histogram equalization is then applied to the final images to improve image quality by sharpening the edges, which makes it easier to classify objects and analyse features. PSNR and SSIM values are recalculated by comparing the improved images with their respective sharp references following the application of histogram equalization, and the outcomes are recorded for future analysis.

In the second method, the blurred images are first subjected to a deblurring technique, which yields a first set of deblurred outputs. In order to recover more details and improve image quality, these photos are subsequently processed once more using a different deblurring method. Performance metrics are calculated after this two-step deblurring procedure to assess the method's efficacy. Histogram equalization is then used to enhance visual information and improve edge definition. Ultimately, the quality of the processed images is re-evaluated, and the outcomes are noted for thorough examination.

2. LITERATURE REVIEW

Baldominos et al. (2020) used the MNIST and EMNIST datasets to study the state of handwritten character recognition. MNIST, a commonly used benchmark in the field, has seen a notable increase in the use of convolutional neural networks (CNNs) (Mohsenzadegan & Kyamakya, 2020). Their study provided insights into the effects of preprocessing techniques by classifying approaches into those that rely exclusively on the original dataset and those that use data augmentation. Since CNNs are the most popular models, much research has been done on them because of their capacity to recognize intricate details in pictures. Notably, recent developments have produced error rates on MNIST test sets that are less than 1%, which begs whether the dataset continues to be difficult for contemporary algorithms to handle.

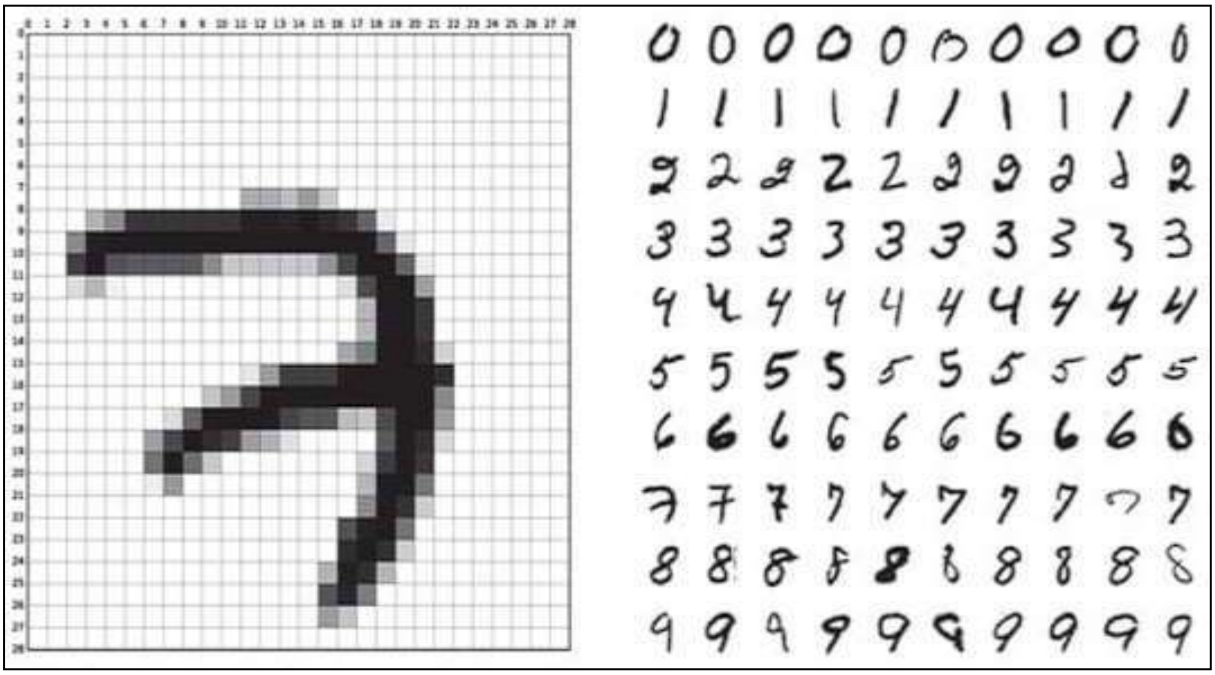


Figure 2.1: (a) An example MNIST image of the digit “7,” (b) A collection of 100 samples from the MNIST training dataset

In 2017, EMNIST was released which represents a change in research focus toward more varied character recognition tasks by expanding the original MNIST dataset to include handwritten letters and digits. The results emphasize the distinct difficulties presented by EMNIST, highlighting its intricacy in contrast to MNIST. In summarizing earlier findings, this study challenges the continued applicability of MNIST as a benchmark while reaffirming the superiority of convolutional neural networks (CNNs) on MNIST. The launch of EMNIST highlights the need for bigger and more diverse datasets. To push handwritten character recognition beyond MNIST's limits, future studies must tackle these new issues.

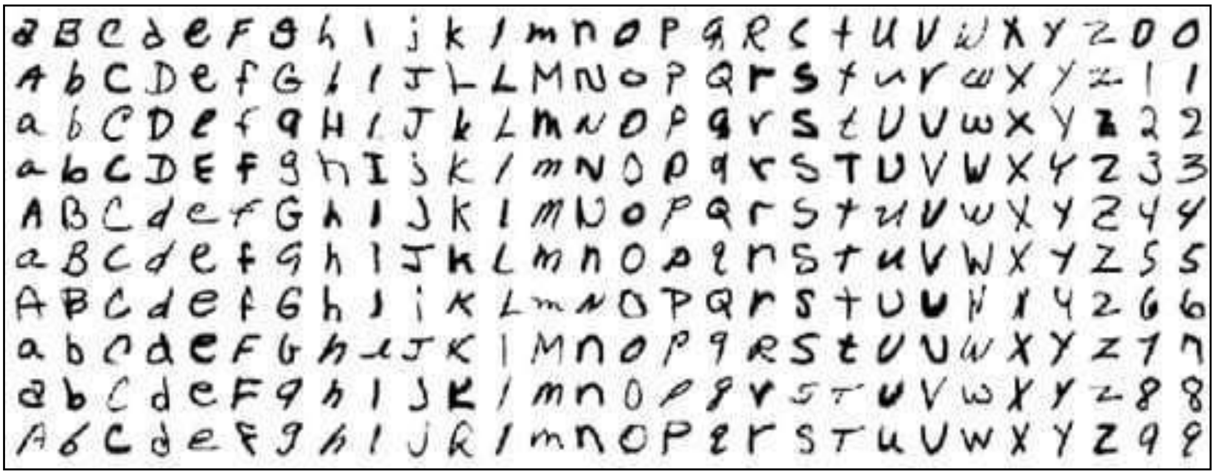


Figure 2.2: EMNIST Dataset for English Alphabets

A deep CNN model suggested by Shao et al. (2020) intended for quick conjunction to improve the recognition accuracy of the MNIST handwritten digit database. The model's multi-layered architecture consists of three convolutional layers dedicated to feature extraction and two fully connected layers responsible for classification. By adjusting important parameters batch size, activation functions, learning rate, seed size, and batch normalization, recognition performance was enhanced. With a high accuracy of 99.82% for classification on the training set and 99.40% on the test set was attained, this method proved to be a reliable MNIST classifier.

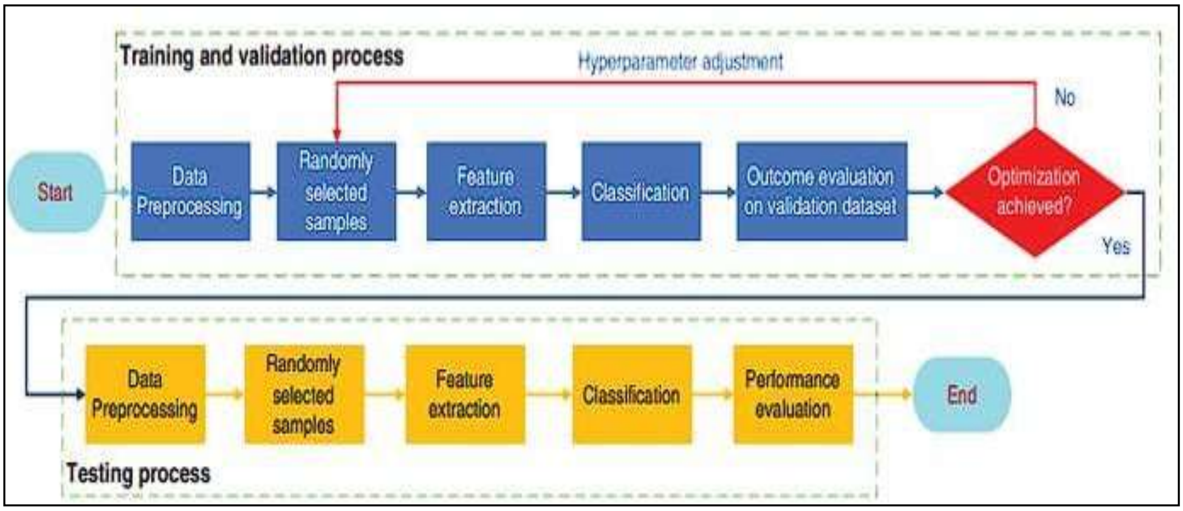


Figure 2.3: Diagram for data processing using DL

The authors presented an independent damage model that is thorough and considers a number of uncertainties missed in other models, defying popular beliefs regarding adversarial robustness. According to their analysis, several previously noted flaws severely limit the MNIST-C dataset's efficacy.

The EMNIST (Extended MNIST) dataset is an enhanced version of the original NIST dataset, designed to support more complex handwritten character recognition tasks, was first presented by Cohen et al. (2017). Like MNIST, EMNIST uses a pre-processing technique that preserves the original dataset's structure and properties while increasing the number of classes to include handwritten letters and numbers. This design makes its adoption in more general handwritten character recognition tasks easier, which guarantees compatibility with current classifiers and systems created for MNIST.

Palvanov et al. (2021) highlight the challenge of recognizing handwritten digits persists, despite significant advancements made by techniques of machine learning in addressing this problem, many systems still need a lot of training and data processing time. Even well-trained models frequently encounter challenges in real-time applications where speed is crucial. Additionally, memory usage is a significant consideration because models with lower memory requirements typically perform faster than those with higher computational demands. By comparing four models in terms of training and evaluation time, memory usage, and other crucial performance metrics, the researchers hoped to determine the most effective technique for real-time handwritten digit recognition.

Image processing and deep learning are major research topics worldwide, with various applications in industries like robotics, medicine, security, and surveillance, claim Garg et al. (2022). Deep learning allows for more efficient analysis of complex data types, such as text, audio, and images, by utilizing hierarchical learning through multiple levels of abstraction and representation. Many handwritten digit samples in the MNIST dataset have not been thoroughly investigated regarding training, testing, and validation. By combining several convolutional layers, ReLU activation functions, and clustering techniques, Garg et al. (2022) created a strong model, an accuracy of 98.45% was achieved on the MNIST dataset. In order to enhance the model's generalizability and overall performance, it was also assessed on several other datasets.

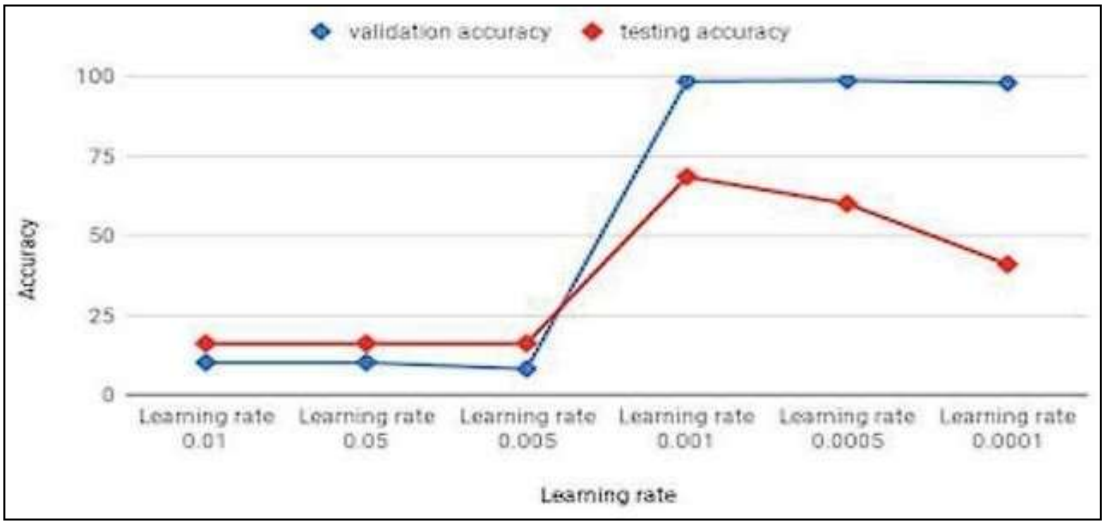


Figure 2.4: Accuracy of Testing and Validation

Kadam et al. (2021) demonstrated how CNN can classify images. The MNIST and Fashion-MNIST datasets were used in their study to assess the performance of different CNN architectures. They specifically looked at five distinct network topologies with differing fully connected layers, filter sizes, and convolutional layer counts. The Fashion-MNIST dataset presented a bigger classification challenge due to its complexity, but the models achieved over 99% accuracy on the MNIST dataset.

Considerable progress has been made in image recognition, especially with the use of convolutional neural networks (CNNs), according to Wang et al. (2019). The LeNet-5 architecture, which effectively classified handwritten digits from the MNIST dataset, was the primary focus of their investigation. According to experimental results, putting a Dropout layer before the output layer and optimizing the activation function and learning rate increased training accuracy, decreased sensitivity to initial parameter settings, and accelerated convergence.

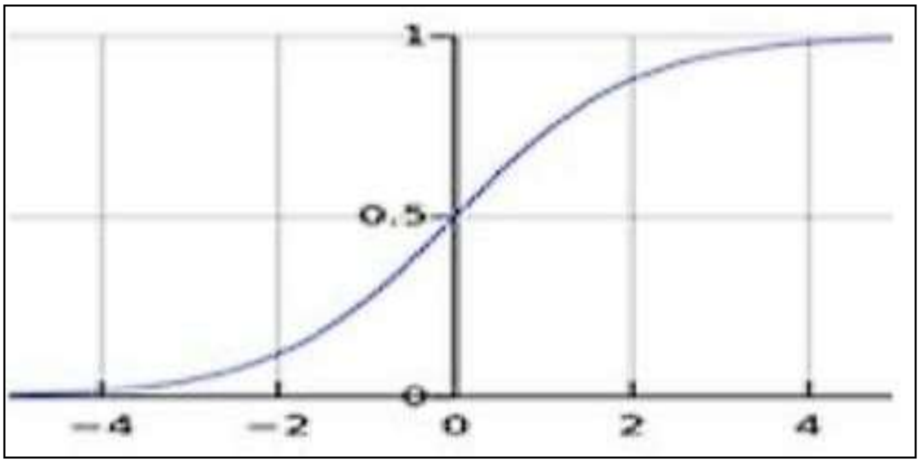


Figure 2.5: Sigmoid Activation Function

Using a deep convolutional neural network (CNN), Jain et al. (2020) demonstrated an efficient handwritten English alphabet and digit recognition technique. The CNN model's performance was assessed in their study using the standardized MNIST dataset, which consists of handwritten letters and numbers. Multiple experiments were conducted to assess the model's performance across different scenarios, and different CNN parameters were methodically changed. Various optimization techniques were used in the main configurations, such as convex and non-convex optimization methods, error function optimization, activation function selection, and experimentation with multiple hidden layers.

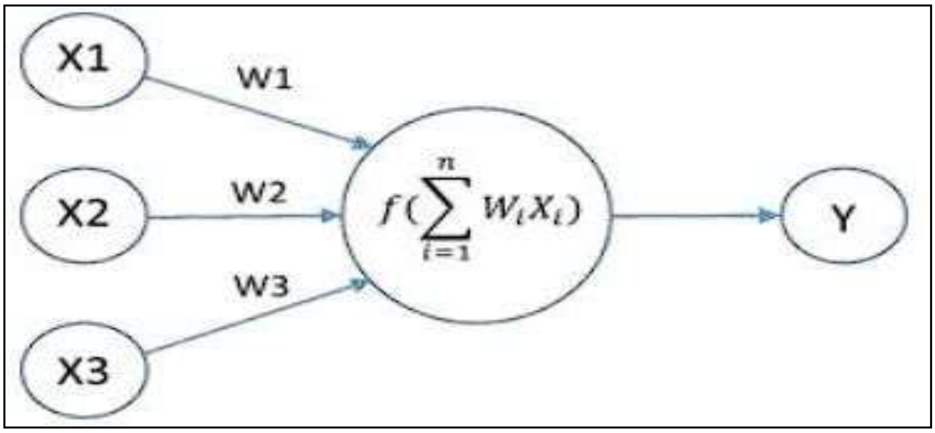
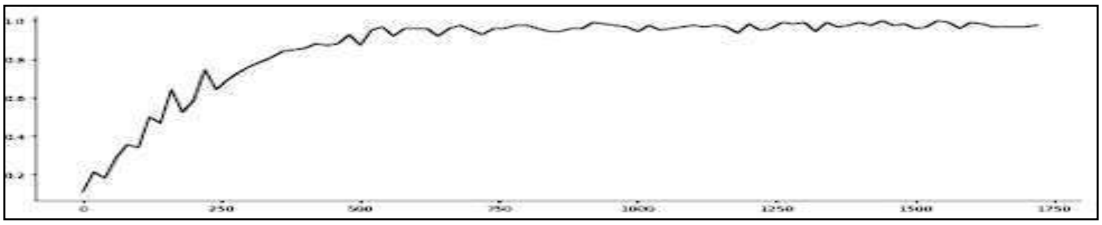


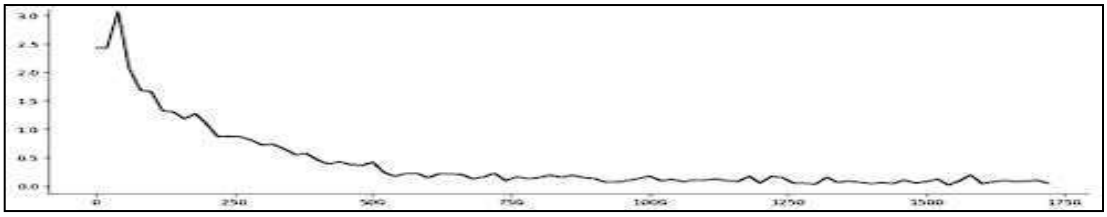
Figure 2.6: Convolutional Neural Network (CNN)

LIRA is a neural receptive field method for image classification proposed by Kussul et al. (2018). Three layers comprise the LIRA model: an output layer, a connection layer, and a sensor layer. Trainable links connect the connection layer to the output layer, while fixed random links connect the sensor layer to the connection layer. The training is efficient, requiring no multiplication or floating-point operations. Two datasets were used to evaluate the model. The first was MNIST, which contains 10,000 images for testing and 60,000 for training in handwritten numbers. The microstructure fitting problem, which involved inserting a pin into a hole, was represented by 441 images in the second dataset. This dataset was split into training and testing subsets using a random split. Reported error rates are among the lowest at 0.7%, 0.63%, and 0.42%, respectively, despite many MNIST classification results.

Gope et al. (2020) state that handwritten numeral recognition is still a major pattern recognition challenge. This task is essential for many applications, including postal code processing, online banking verification, and digital form submissions, where users enter handwritten numbers using gadgets like smartphones and scanners. Hand-scanned images from different NIST archives are included in the MNIST dataset, which is a subset of the NIST database, was used in their investigation. The suggested method uses Support Vector Machines, Decision Trees, Naive Bayes, K-Nearest Neighbors, Random Forests, and Multilayer Perceptrons machine learning algorithms, to recognize handwritten digits offline using Python. Improving the precision and dependability of handwritten digit recognition systems was the primary goal.



(a)



(b)

Figure 2.7 (a) Training set accuracy curve; (b) Training set loss curve.

A significant amount of digital data entry is necessary because many registration forms in the inspection and testing sector are filled out by Chen Zhong et al. (2021). They addressed this by proposing a handwriting recognition system that uses a convolutional neural network (CNN) in conjunction with the Harris edge detection method is used to accurately and automatically recognize handwritten data. The Harris edge detection technique helped to locate critical landmarks in the labelled images with handwritten numbers. On the test set, their experiments showed a 98.14% recognition accuracy. This method presents a viable technical solution for processing handwritten data intelligently and effectively.

Gong et al. (2023) created a CNN model to meet the growing need for handwritten digit recognition. The architecture has an input layer, two convolutional layers (each with 5×5 kernels), two pooling layers (each with 2×2 kernels), a fully connected layer, and an output layer. After many trainings and iterations, the model demonstrated strong performance in handwritten digit detection, achieving 100% training accuracy and 99.25% test accuracy.

According to Beohar et al. (2022), handwritten digit recognition is still tricky in application development. Computer vision relies heavily on digital recognition, and because deep learning methods can replicate human thought processes, they have been used to address this issue. CNN and ANN are involved to analyse the feature extraction and classification phases. The models were trained on the MNIST dataset with the ADAM optimizer and cross-entropy loss to improve performance.

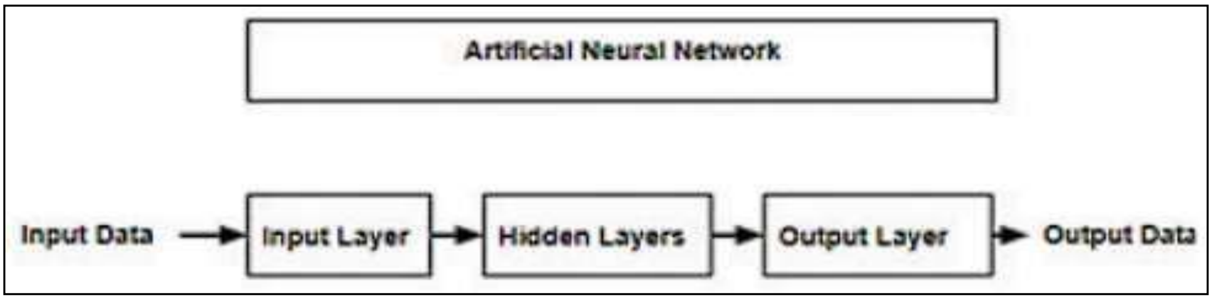


Figure 2.8: Artificial Neural Network

The largest manual digital dataset for language and data modelling in the world, MNISTMIX, was made available by Weiwei Jiang [15]. Manual digit recognition investigations can easily incorporate MNIST-MIX, which uses the same data format as MNIST. Because MNIST-MIX includes numbers from ten different languages, it is a more complicated dataset, and more careful model building is required to address the classification uncertainty.

Despite significant advancements in machine learning, achieving real-time speed while maintaining high performance remains a major challenge, according to Dong-yuan Ge et al. For MNIST handwritten digit detection, a convolutional neural network comprising first layer having 32 images and second layer having 64 images was developed. The first layer contains 2048 neurons and second with 784 neurons, the two layers are accumulated to make connecting layer. As there are 10-digit labels, required ten neurons in the output layer. For the study, TensorFlow was cast-off to create the network. After 500 training iterations, the model achieved 95.7% accuracy on the test set and 96.1% on the training set.

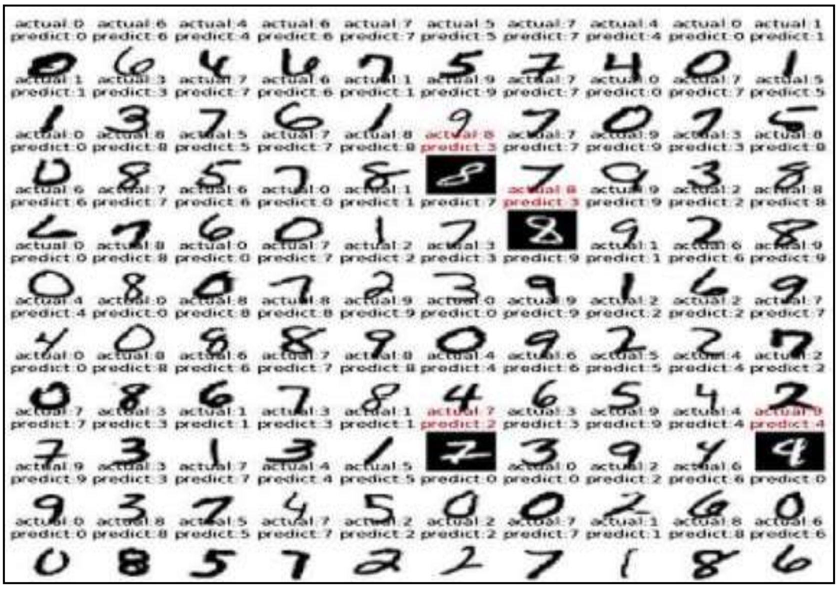


Figure 2.9: Recognition result using adopted techniques

The problem of SISR (Single Image Super Resolution) combined with motion deblurring, where old-style super-resolution techniques fall short due to nonuniform motion blur in LR images, is addressed by Zhang et al. The GFN (Gated Fusion Network) proposed by authors, a dual-branch CNN that merges super-resolution and deblurring within a single end-to-end system. A notable advancement is the fusion at the feature level, which retains motion characteristics from deblurring and spatial details from super-resolution, leading to better results.

When trained on a self-generated low-resolution GOPRO dataset (LR-GOPRO), GFN outperforms state-of-the-art techniques like EDSR and Deep Deblur in terms of PSNR, SSIM, and processing speed. This leads to up to 116× faster inference and 78% fewer parameters.

A novel framework for solving the well-known single-image heterogeneous motion blur removal problem is presented by Gong et al. A detailed motion flow map is predicted from a blurred image using an FCN, with emphasis on the cause of the blur rather than its effect, in contrast to current deblurring techniques that mainly rely on hand-specified priors or iterative kernel estimation. This method's key finding shows that using a pixel-by-pixel motion flow map avoids complicated assumptions and helps the network focus on motion that causes blur. In order to train the FCN end-to-end, the authors synthesize a method for capturing a range of realistic motion flows, including translation and rotation, within three-dimensional space. The learned model then uses non-blind deconvolution to eliminate noise from fine images and predict motion flow maps.

The first system to remove motion blur without knowing the blur, using conditional generative adversarial networks, Deblur GAN, was presented by Kupyn et al. Other deblurring methods rely on CNN designs or kernel methods that use loss functions such as MSE. Deblur GAN restores clear, detailed images from a single blurred input using WGAN-GP and perceptual loss. Utilizing a patch-based critic and residual blocks, the architecture is a lightweight generator. Deblur GAN's design enables it to be five times faster than Deep Deblur without sacrificing perceptual quality. Faster convergence and improved generalization are achieved by training the generator to predict a residual image, which is then added to the blurred input. In dynamic scenes affected by factors such as camera shake, object motion, depth variation, and occlusion, Nah et al. proposed an innovative approach to blind deblurring. Traditional methods, which often depend on blur kernel estimation or simplified assumptions, struggle with real-world motion blur due to its non-uniform and spatially varying nature. The researchers overcame these constraints by creating a multi-scale CNN architecture that instantly produces sharp images without relying on explicit blur kernel estimation. The design is akin to a coarse-to-fine processing pipeline: a Gaussian pyramid is used to process a blurred input image at various resolutions, and a multi-scale loss function is used to supervise the outputs at each scale. This permits convergence and the preservation of fine details while guaranteeing consistent sharpness and stability across scales.

Sun et al. (2015) introduced a deep learning method to remove non-uniform motion blur from a single image, often caused by camera shake or moving objects. A CNN is used to predict blur patterns in small image patches, in contrast to previous methods that depend on hand-engineered features or uniform kernel assumptions. The researchers used 73 discretized motion kernels to create over 1.4 million fuzzy picture patches on images from the PASCAL VOC dataset in order to train the CNN. Prediction detail is increased fivefold by expanding the motion kernel set, with no extra training required, and the CNN model is robust despite its modest size. Our approach outperforms previous methods, including those based on spectral analysis or manual regression, Motion kernel estimation precision (MSE = 7.83) and deblurring quality (PSNR = 24.81 dB) were achieved. According to quantitative experiments conducted on both synthetic and real-world datasets. Improved spatially variable blur handling and reduced artifacts are visible upon visual inspection, particularly in highly complex scenarios involving layered motion.

3. METHODOLOGY

In this explanation, we discuss each stage of the study which covers preparing the data, looking at images, training the model, reviewing the results and writing the final report.

(a) Adopted Methodology for Blurred Images:

3.1. Data Preparation

(a) The original MNIST dataset of gray-scale handwritten digits is made available for this research.

(b) Blur filters will only be applied to a certain subset of MNIST images. With this subset, we make a new dataset for conducting experiments.

3.2. Image Blurring

(a) Image processing techniques in this project are implemented using Python's OpenCV library.

(b) Two types of blurs are presented here, namely, Gaussian, motion and defocus, along with other ways of blurring.

(c) The parameters of the kernel size and standard deviation are adjusted to choose how blurry the images will be.

(d) Selected blur methods are used on the selected MNIST images to obtain a fresh set of blurred images. This section outlines the step-by-step process followed in the study, beginning with data preparation and progressing through image processing, model training, performance evaluation, and documentation of findings.

3.3. Dataset Augmentation

a) The original dataset from MNIST is combined with the set of blurry images. The dataset is labeled properly for images that are easy to see as well as for images that are hard to read.

b) The data is randomly reordered so that training the model is more impartial.

3.4. Model Adaptation

a) Because it is capable of image recognition, a convolutional neural network (CNN) is preferred for this purpose.

b) The model accepts gray-scale images that are 28x28 pixels in size, also including photos of reduced clarity.

c) The training setup is complete, with the optimizer, evaluation metrics and loss function.

3.5. Model Training

a) The augmented data is divided into two groups, one for training and one for testing.

b) The model is created by using the training set and tested using the testing set. Performance assessment can be done by following the training and validation accuracy.

3.6. Performance Evaluation

a) Robustness is checked by measuring the model’s success in recognizing both clear and blurred images.

b) To assess the model’s performance with blurred digits, a confusion matrix is used.

3.7. Analytical Review

a) The model’s generalization is checked by comparing its results on clear and blurred photos.

b) The team looks for and documents any decrease in image quality because of blurring.

3.8. Documentation

a) The key findings identified in the performance evaluation are clearly explained.

b) The way blurry images influence the performance of the model is explained and ideas for enhancements in ongoing work are considered.

(b) Adopted Methodology for Deblurred Image

3.9. This project focuses on preparing the GoPro Dataset

a) We add the GoPro dataset to the program, as it consists of real blurry images.

b) The training data and testing data are already separated for you. The blurred scene images are found in 22 folders in the training set and they are also present in the testing set, divided among 11 folders.

3.10. Method Development for Deblurring

a) The research team selects an appropriate technique to remove blurriness from the photos.

b) All the required information is supplied to allow for an easy application of the method.

c) You choose the correct image size to match with your chosen deblurring method.

3.11. How Well Does Deblurring Work

a) To check image quality, the PSNR is determined by comparing the sharp original with the deblurred version of the image.

b) This method also uses the Structural Similarity Index Measure (SSIM), with a score between 0 and 1 and higher scores are better.

3.12. Using the Computer Vision Technique

a) After deblurring, computer vision is used to further keep away distortions and sharpen the pictures.

b) After utilizing the computer vision approach, the PSNR and SSIM values are worked out again to compare the quality of the original image and the image after the method was used.

3.13. Final Documentation

a) Observations from the evaluation are captured in detail as part of the recording the analysis results.

b) Different deblurring approaches are discussed and key areas to further develop these techniques are mentioned.

4. EXPERIMENTS AND RESULTS

4.1. BLURRED IMAGES

Images that appear blurry are widely used in fields such as data privacy—such as hiding faces or delicate objects—or for creative and artistic purposes. With digital image processing, blurring is carried out by changing the numbers in pixels with averaging or using math filters which makes the picture less detailed and slightly out of focus. Many photographers, engineers and designers rely on this method to help clean up visual noise or better highlight chosen objects.

For this study, grayscale images of handwritten digits from the MNIST dataset are blurred. This approach lets us investigate how image quality affects learning machine models used in tasks that identify car models. When blur is applied, it helps to see if the models can still work well. In this situation, blurring is achieved with Gaussian filters by choosing the sigma parameter. Rising sigma causes the image to look smoother.

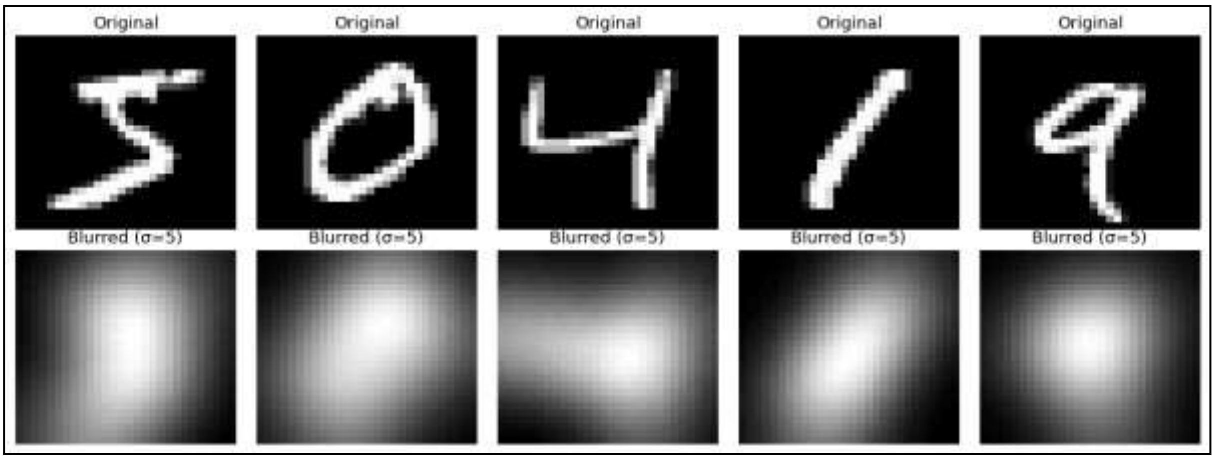


Figure 4.1: Original image and blurred image using Gaussian blur (σ = 5) on MNIST dataset

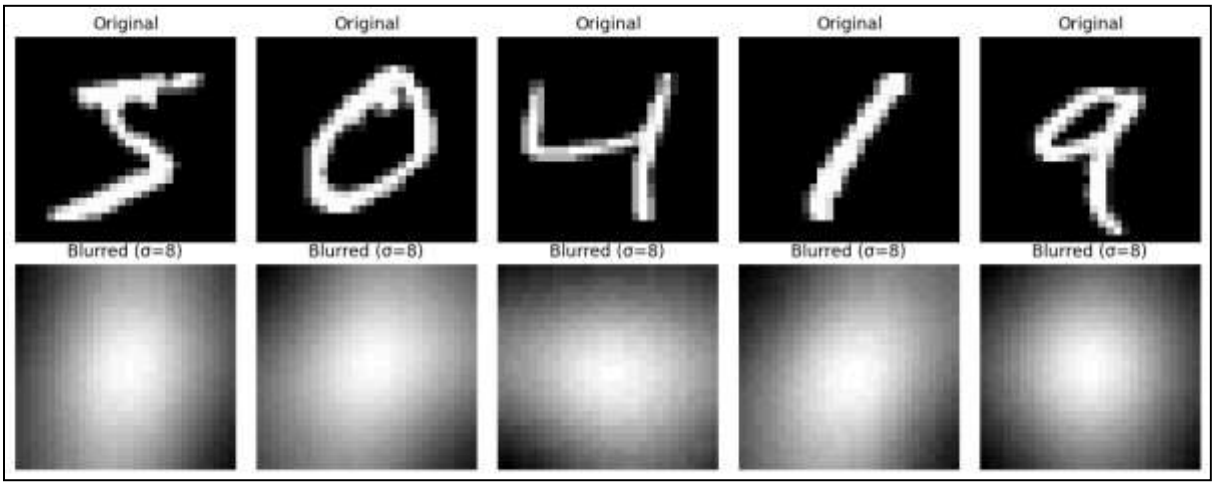


Figure 4.2: Original image and blurred image using Gaussian blur (σ = 8) on MNIST dataset

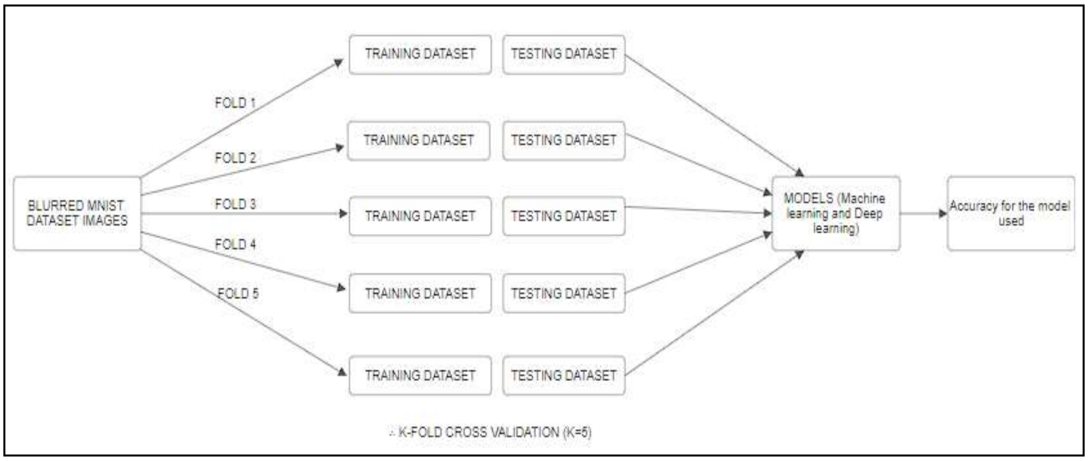

TRAINING DATASET
TESTING DATASET
FOLD 1
FOLD 2
FOLD 3
FOLD 4
FOLD 5
BLURRED MNIST DATASET IMAGES
MODELS (Machine learning and Deep learning)
Accuracy for the model used
K-FOLD CROSS VALIDATION (K=5)

Figure 4.3: Workflow of the adopted methodology for classifying blurred images

4.2. K-FOLD CROSS-VALIDATION

Many people use K-fold cross-validation to check how accurate machine learning models are. It helps to estimate the future success of a model when we divide the whole dataset into several parts commonly known as "folds" (VBT Academie, n.d.).

There are a variety of steps needed to complete the process.

(a) To begin, the whole dataset is cut into K equal or almost equal pieces called folds. As a result, every fold is tested exactly once and also included in the training set K times (Elvanco, 2024).

(b) It Will Run the Algorithm K times. Every time, the model is trained with K-1 folds and validated with the one, unique fold (Artificial Intelligence, n.d.).

(c) At each next iteration, the model trains on the training part and is checked with the validation portion. Accuracy, precision, recall or F1-score are all performance metrics calculated using the model’s performance on the validation set.

(d) After each of the K iterations, a list of K performance scores is obtained. Experts usually take the average of these scores to give a single, more trustworthy figure for the model’s behavior.

(e) If many models or hyperparameter settings are tried, cross-validation makes it possible to pick the one that performs best. Following the selection of the best model, it can be assessed again by using a separate, unrepeated test set.

The method allows a balanced evaluation of a model’s accuracy while minimizing the danger of overfitting when the data set is not large.

4.3. VGG16 (Visual Geometry Group-16)

The Visual Geometry Group (VGG) at the University of Oxford developed the VGG16 convolutional neural network which is now quite well known. Simonyan and Zisserman (2014) first presented it and people like it because it is straightforward and excellent at classifying images.

(a) The Study of Architecture: VGG16 stands for the medical network's 16 weight layers: 13 convolutional and 3 fully connected layers. Sequential and uniform architecture is achieved by relying on small 3×3 convolutional filters throughout the neural network. In addition,

(b) Convolutional Layers should be used: All convolutional layers operate on 3×3 filters that have a stride of 1 and 'same' padding, so the image size does not change after convolution. Later in the model, the number of filters goes up from 64 to 512 (Manzoor et al., 2023). With this, the network is able to find and use more complex details in the images.

(c) From Layer 3 on, I use max pooling layers.: The subsequent section involves applying 2×2 max-pooling layers where each layer considers 2×2 patches and skips every other element (with stride 2). By lowering the number of channels in feature maps with pooling, we lessen the amount of computation needed and reduce overfitting (Nguyen et al., 2020).

(d) Full-Connected Layers: There is a single sequence concluding with three connected (dense) layers. There are 4,096 neurons in each of the first two layers, but the last one has 1,000 neurons. Its structure was designed to match the 1,000 classes in ImageNet, the data VGG16 was trained with (Kandel & Castelli, 2020).

(e) Activation Function: Immediately after every convolutional and fully connected layer, ReLU (Rectified Linear Unit) activation occurs. Such activation in the hidden layer helps the network process more difficult data and makes the training converge more quickly.

(f) The fifth layer, called the output layer.: This last layer applies softmax to produce a list of 1,000 probabilities, helping to choose a classification choice (Zahin, 2023).

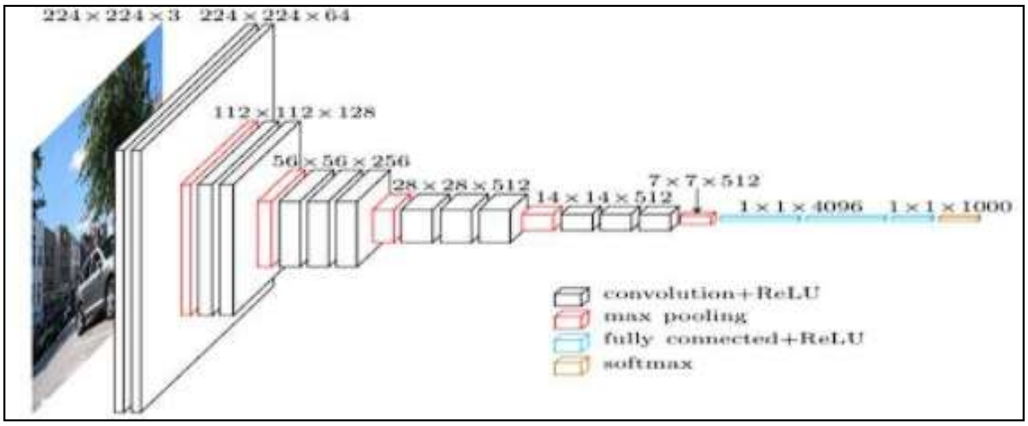


Figure 4.4: Adopted VGG16 Neural Network Architecture

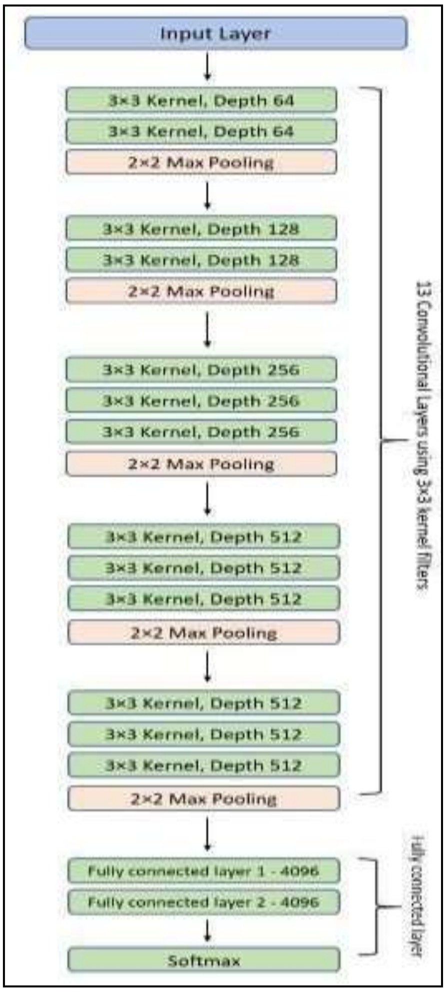


Figure 4.5 Adopted Layer-wise Detailed Architecture of VGG16

4.4.DENSENET

The Densely Connected Convolutional Network, abbreviated DenseNet, was recently introduced by Huang et al. (2017). The model works to raise the performance and efficiency of convolutional neural networks (CNNs) by making layers more strongly linked. DenseNet is based on the idea of using features again and improving how gradients move across the model.

(a) A lot of connections exist between the brain and the body. Its dense connectivity pattern is one of the most significant attractions of DenseNet. The data from every layer earlier in the network goes into a layer which then passes its results to the layers that come after it. By having access to earlier learned features, each layer can use a wider and richer set of features in the final stage (Huang et al., 2017).

(b) Good Use of Parameters: Using services from the past layers, DenseNet eliminates much of the redundancy during learning. With this strategy, the model gets equally accurate results while needing much smaller numbers of parameters than other deep nets, like ResNet. Because of this, DenseNet combines high power with a small size.

(c) It helps address the Vanishing Gradient Problem. With lots of connections, gradients can travel straight backwards through the network. As a result, gradient disappearance, an issue in very deep networks, is reduced, making training more stable and efficient.

(d) **Better moving of information through the network**: This model permits earlier features to stay involved in the deeper layer calculations. Because of this design, patterns at every level of complexity become easier to discover in the data.

(e) Streamlined and organized: What stands out about DenseNet is its ability to achieve solid results using a model that's not too big. Being compact, it can be run on systems with little computer power.

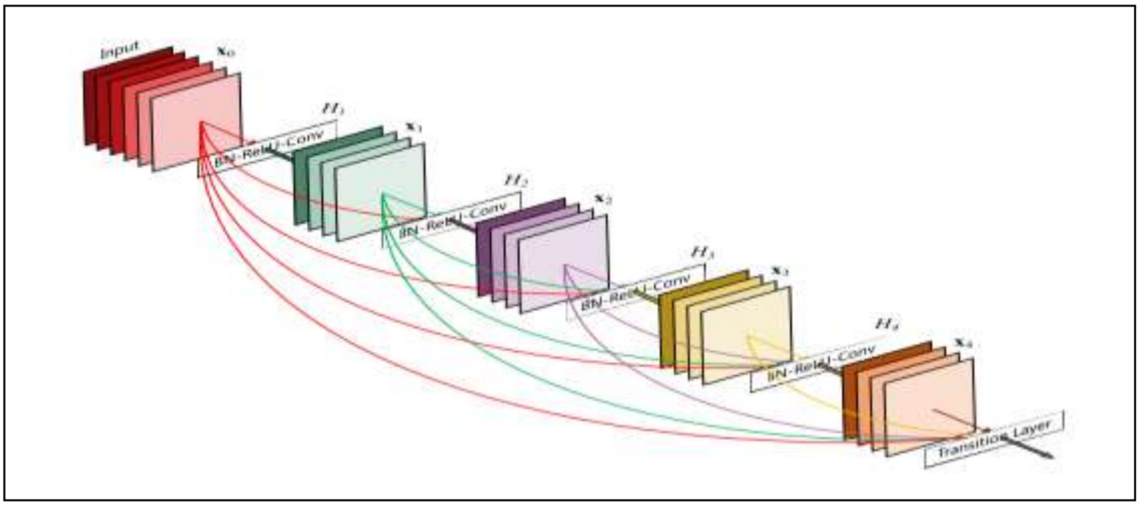


Figure 4.6 Adapted DenseNet Architecture

4.5. SHUFFLENET

ShuffleNet is a CNN developed especially for mobile and low-power systems. The approach was proposed by Zhang et al. (2018) to fix the challenge of reaching high accuracy with very little computational power. ShuffleNet applies grouped convolutions and a new channel shuffle trick to both boost efficiency and maintain great performance.

(a) Grouped Convolutions: Using grouped convolutions lets us lower both the cost of computations and the number of parameters needed. Here, the input data is arranged into several parts, each has a convolution layer applied separately. Earlier systems like ResNeXt influenced this method which provides a combination of size and predictive strength (Zhang et al., 2018).

(b) **Channel Swapping:** Here, the biggest problem is that information can only flow so much between the channel groups. This issue is solved by ShuffleNet by including a channel shuffle operation. Following grouped convolutions, the channels in the feature maps are switched so that channels from each group are mixed together. When information is shuffled, every group in the following layer benefits from a wide range of details and shares them more easily with others, according to Karri et al. (2021).

(c) Lighter and better results: Created to run efficiently, ShuffleNet is perfect for situations that require limited computing, including smartphones and others. The study shows that it gets high accuracy with significantly fewer FLOPs than other lighter models including MobileNet. Thanks to its small size, Neural9 is useful for carrying out computer vision tasks in real time on limited systems.

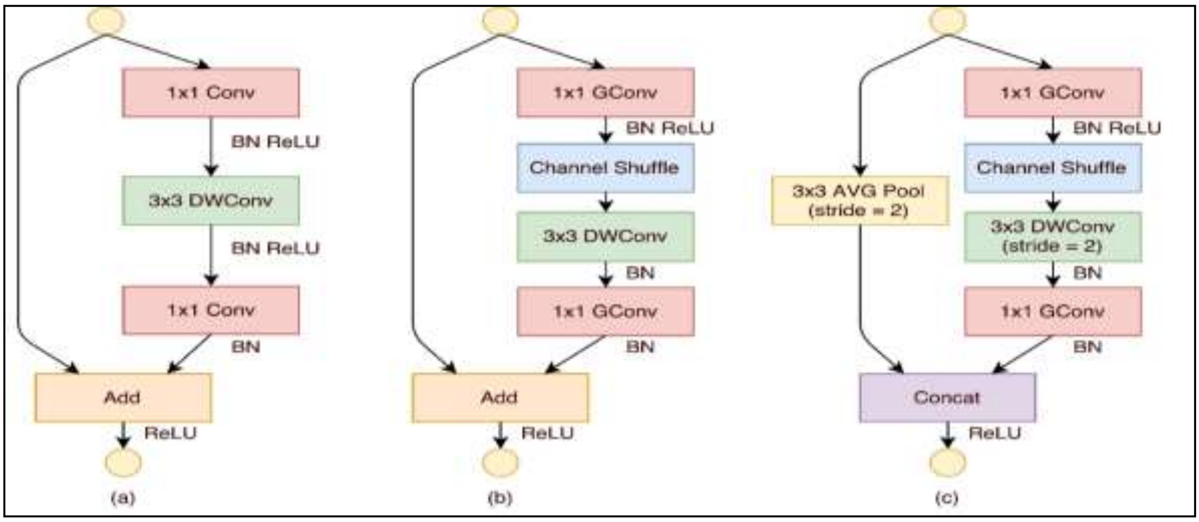


Figure 4.7: ShuffleNet Units (a) Bottleneck unit using depthwise convolution; (b) Standard ShuffleNet unit with pointwise group convolution and channel shuffle; (c) ShuffleNet unit configured with stride = 2

4.6. INCEPTION

Szegedy et.al named the Inception architecture (also known as GoogLeNet) in their paper Going Deeper with Convolutions (2015). Google developed the network to tackle the problem of improving how deep and broad the network is, while also keeping the computing costs low. Ever since it was first developed, Inception has gone through different versions: Inception-v1, v2, v3, v4 and Inception-ResNet, each one refining some aspect (Wulveryck, 2017).

(a) Multi-Scale Processing: Being capable of dealing with spatial information over different scales is a characteristic of the Inception design. Each Inception module applies filters that range from 1×1 to 5×5 in size all at the same time to help it process the input. Because of this technique, the network is able to notice both detailed and large elements in the input image.

(b) **Dimensionality Reduction:** Efficiency improvement in Inception is achieved by using small size filters to decrease the number of parameters and then conducting complex convolutions (Ji et al., 2019). They reduce the input channels and do not reduce performance.

(c) Parallel feature extraction means several channels are used for separate feature extraction. Filters in all modules of Inception operate side by side, each with a different size. By combining the filters, the model can collect multiple features from the same data at once. By using parallel design, the network becomes more supportive of the detail in visual data.

(d) Efficiency: Combining filters and dimensionality reduction methods allows the Inception architecture to achieve strong accuracy that uses little computation. This reason makes neural networks helpful for processing a large number of images at one time.

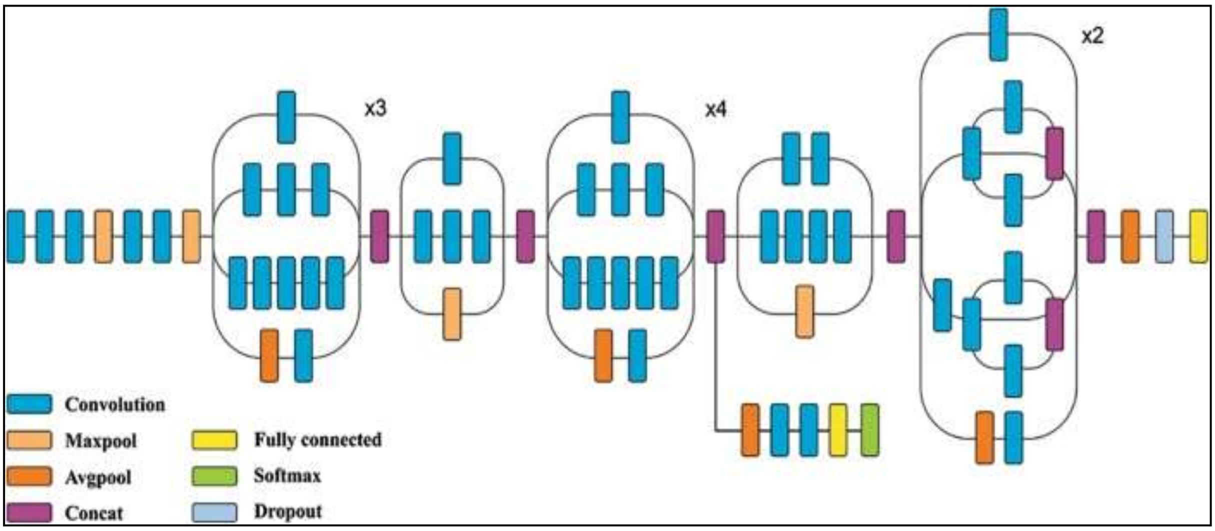


Figure 4.8: Adapted Structure of the Inception v3 Model

4.7.XCEPTION

Xception got its name, Extreme Inception and was designed by Chollet in a paper named Xception: Deep Learning with Depthwise Separable Convolutions (2017). The Inception model is improved by substituting regular convolutions with depthwise separable convolutions, helping it work more efficiently and better.

Depthwise Separable Convolutions:

While traditional convolutions are used in other networks, Xception replaces them with depthwise separable convolutions, made of these two steps:

(a) In depthwise convolution, one convolutional filter reads through each input channel and identifies important features inside that channel independently from others.

After depthwise convolution, pointwise convolutions combine the different channels and mix the features between them.

(b) **Brilliantly Nesting:** You could say that Xception replaces all the Inception modules with depthwise separable convolutions. Because of this, the design becomes simpler and the program works faster.

(c) **Linear Stack:** Unlike Inception, where there are many branches, Xception stacks the different layers only one after the other. Because it is simple, the network is simpler to grasp and set up.

(d) Better ability to describe and represent things. The difference between extracting features locally and across channels boosts Xception's ability to represent information. It offers more accurate results than Inception-v3 with roughly the same number of parameters which means it can do more with less computation.

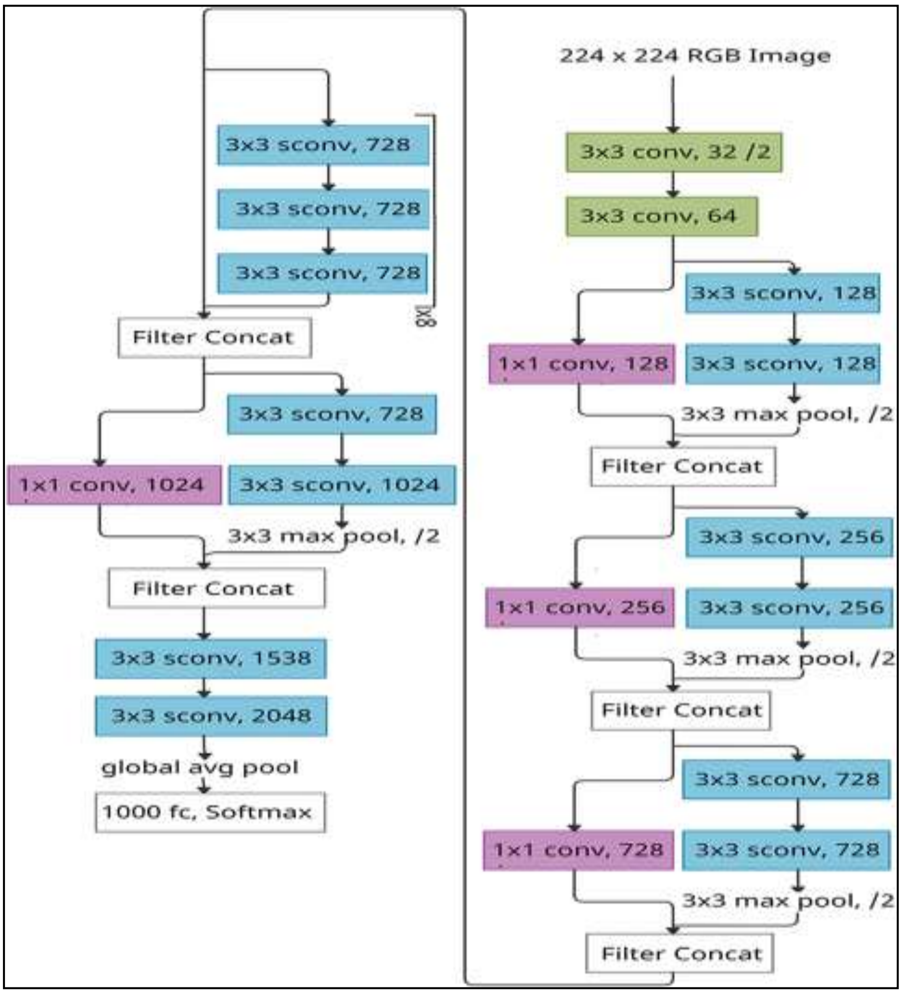
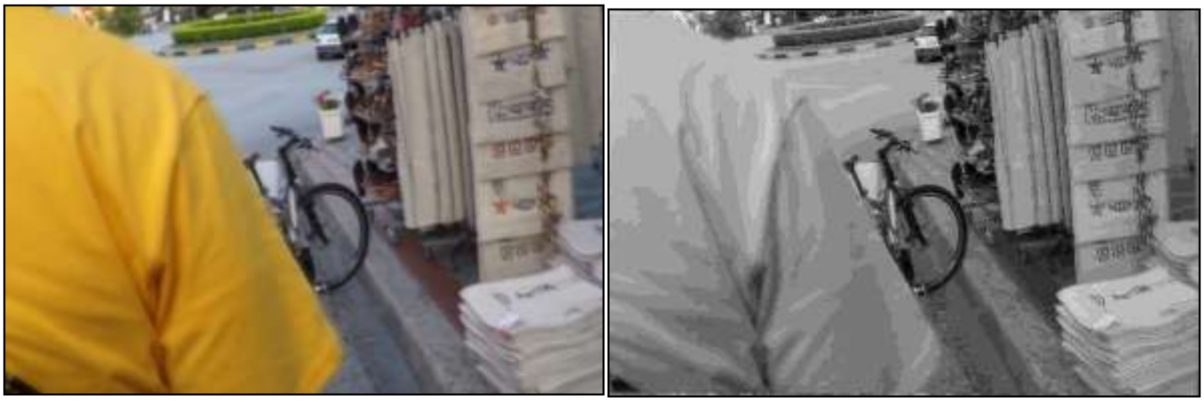


Figure 4.9: Adopted Architecture of the Xception Deep CNN Model

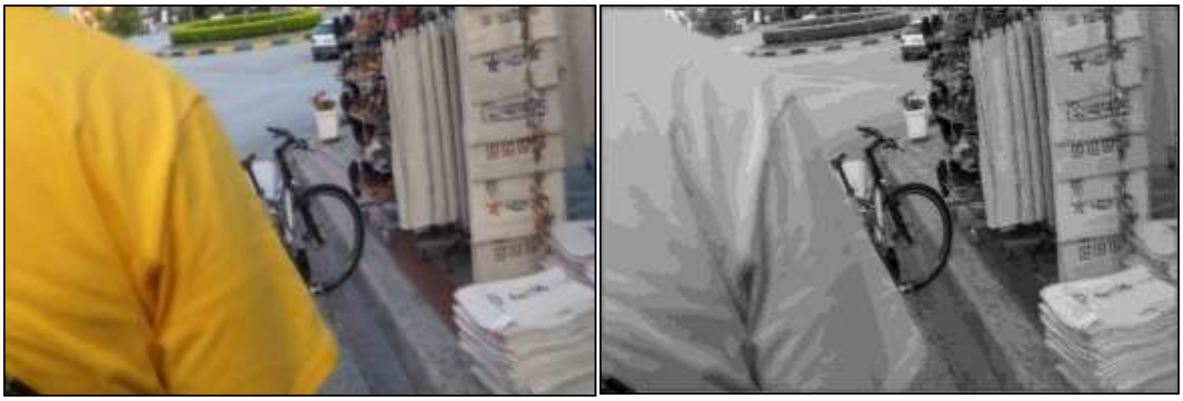
4.8. DEBLURRED IMAGES

To deblur means to turn a blurry image into a sharper and clearer one because of wrinkles, movements or poor settings. Several ways and tools are available for deblurring and how well they work relies on how clear the estimated or known blur is. Sharpness in an image is restores through the process known as deblurring.



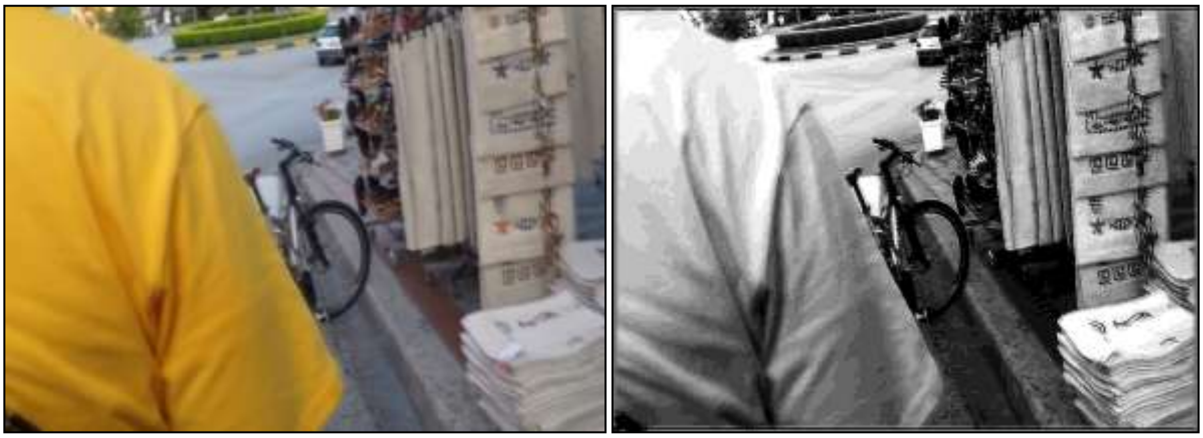
a) (b)

Figure 4.10: (a) Blurred image from the GoPro dataset, (b) Corresponding deblurred image after applying the Wiener filter.



a) (b)

Figure 4.11: (a) Blurred image from the GoPro dataset, (b) Corresponding deblurred image after applying the Lucy-Richardson deblurring algorithm.



(a) (b)

Figure 4.12: (a) Blurred image from the GoPro dataset, (b)Corresponding deblurred image obtained by applying the Lucy-Richardson deblurring algorithm followed by histogram equalization.

4.9. WIENER FILTER

Many times, in signal processing, the Wiener filter is applied as a linear and time-invariant method to recover a clean signal from a noisy measurement. The main objective is to reduce the difference in MSE between what is predicted and what is actually observed. Frequently, image processing engineers use the Wiener filter to handle problems related to noise and image degradation.

The filter does this by trying to find the most balanced transfer function between the degradation from noise and the blur. This transfer function is used in the frequency domain by the Wiener filter which derives it by using the power spectral densities of the original signal and the noise (Insight Software Consortium, n.d.). This filtering process is linear and do not change, so the output is a linear combination of the input over time.

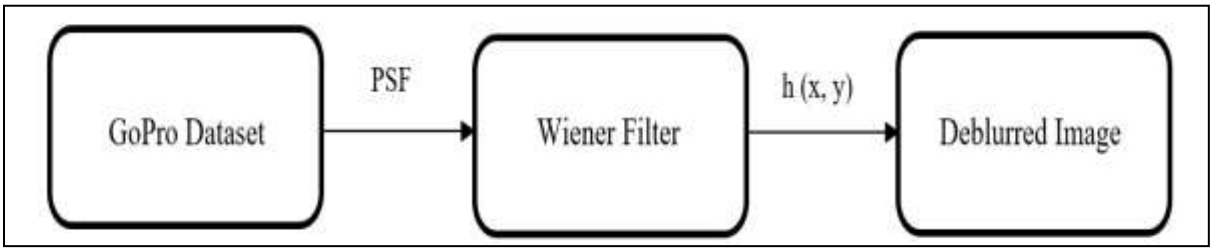


Figure 4.13: Workflow of the Wiener filter applied to produce a deblurred image.

4.10. LUCY-RICHARDSON DEBLURRING

Many times, in image processing, the Lucy-Richardson algorithm is used to restore sharpness to images that are blurry. The technique performs best when the blur can be estimated by a known PSF. The algorithm works using maximum likelihood estimation and proves valuable in astronomical research, imaging X-rays in medical procedures and taking photographs. The estimate of the real image is improved by seeing how the blurred image matches up with the PSF-convolved copy.

Lucy-Richardson Deblurring Iterative Formula

$$\boldsymbol{f}^{(\boldsymbol{n}+\mathbf{1})} = \boldsymbol{f}^{(\boldsymbol{n})}.\left[\frac{\boldsymbol{g}}{\boldsymbol{f}^{(\boldsymbol{n})} * \boldsymbol{h}} * \boldsymbol{h}^{*}\right] \tag{4.1}$$

Where:

g = Blurred image

h = PSF (Point Spread Function)

$f^{(n)}$ = Estimated image at iteration n

* = Convolution

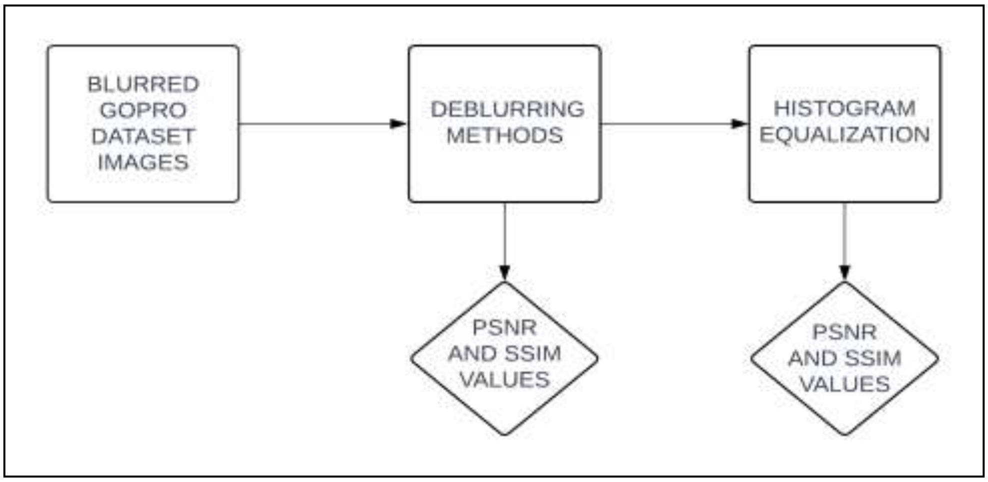


Figure 4.14: Workflow of the methodology adopted to get the deblurred images.

4.11. PROPOSED METHOD

To solve some of the shortcomings in present deblurring approaches, we suggest a method that combines several techniques to gain from their particular strengths and do a better job of restoring images.

First Approach:

First, the Wiener Filter and the Lucy-Richardson algorithm are separately used on the blurred images to produce two sets of deblured outputs. After that, the system merges the two results by taking the average of similar pixels in both pictures. The image quality is tested by comparing the PSNR and SSIM values of the combined results with sharp original images. Afterward, Histogram Equalization (Concessao, 2021) is used to make edges tighter and the image clearer so that classification and interpretation will be more accurate. The effectiveness of the combined method is analyzed by re-calculating the PSNR and SSIM values after the enhancement process.

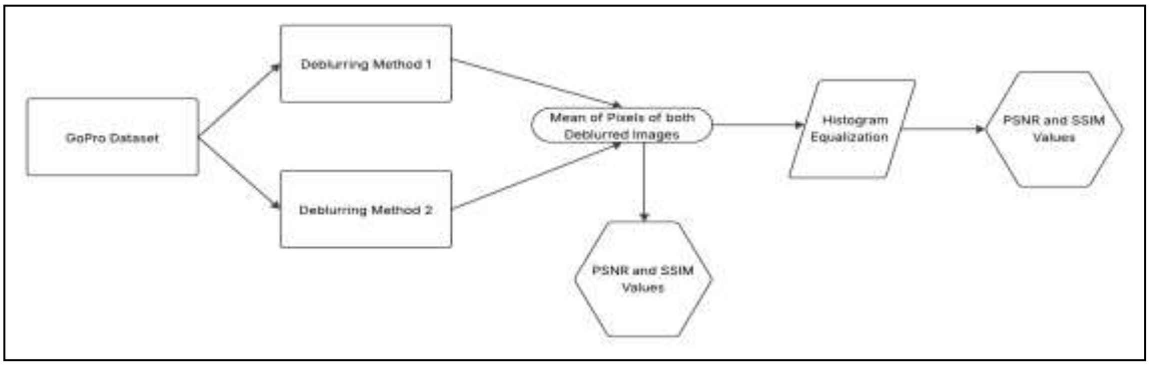


Figure 1.15: Workflow of the Proposed Method (1st Approach – Mean of Pixels of Deblurred Images from Two Different Methods)

Second Approach:

Here, first, the blurred image is processed using a single deblurring method. The improved image is then sent through a different algorithm which further sharpens it and brings back more data. After the two stages of deblurring, performance metrics are generated. Then, Histogram Equalization is used for a second time to boost clarity and sharpness and the results are assessed using PSNR and SSIM.

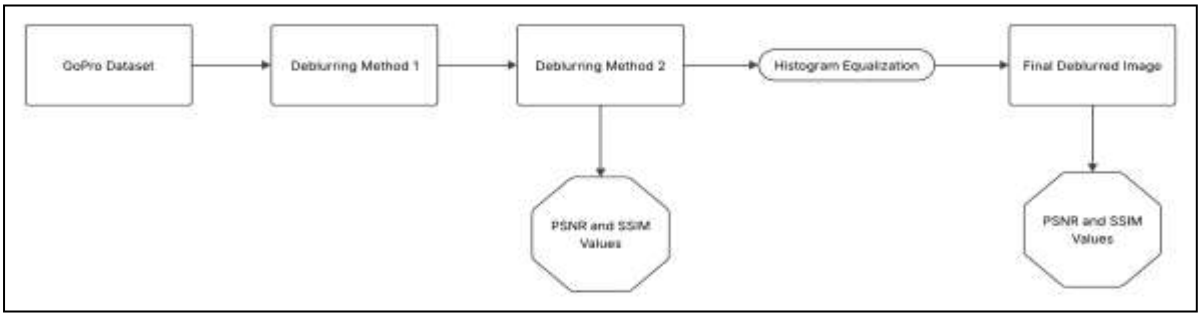


Figure 1.16: Workflow of the Proposed Method (2nd Approach – One Deblurring Method Followed by Another)

Table 4.1: Accuracy of Different Deep Learning Models on the MNIST Dataset

Method used	**Accuracy Without Blur**	**Accuracy With Gaussian Blur (Sigma=5)**	**Accuracy With Gaussian Blur (Sigma=8)**
VGG16	99.47000026702881%	95.50166726112366%	87.93166637420654%
DENSENET	98.968000411987304%	98.772001266479492%	98.62399934768677%
XCEPTION	98.8655%	95.97666621208191%	84.56166744232178%
Shuffle-net	97.869893188%	97.27013160705%	97.189997854232%
Inception	99.0862510681%	98.7442686843%	98.04856807708%

Table 4.2: Performance Comparison of Deblurring Approaches on the GoPro Dataset with and without Histogram Equalization

	WITHOUT ANY CV APPLICATION		APPLYING HISTOGRAM EQUALIZATION	
Method Used	**PSNR VALUE**	**SSIM VALUE**	**PSNR VALUE**	**SSIM VALUE**
Weiner Filter	28.1586363	0.61544090	29.390909	0.69576363
Lucy-Richardson	28.1572727	0.571159090	29.560	0.69131363
Wiener + Lucy-Richardson	28.29136363	0.6547352	29.845365	0.70353863
Wiener followed by Lucy-Richardson	28.15590909	0.56381363	29.2790909	0.68547045

5. RESEARCH GAPS AND CHALLENGES

5.1.Images Lacking Fine Detail

When an image is blurred, the details important for correctly recognizing digits might be lost. Because boundaries and patterns are not clear, the model cannot spot important elements needed for precise classifications.

5.2.Blurry Results in Edge Discovery

By using blurring, the borders between the digit and its background are smoothed which adds uncertainty to edge detection. So, it’s hard to distinguish how the digits look and as a result, the system may misclassify them.

5.3.Reduced Difference Between Colors

As an image becomes blurred, the contrast between each pixel becomes lower, so the numbers don’t stand out as strongly. When there are big differences in pixel intensity, models can have trouble telling digits apart, mainly when the images are very blurry.

5.4.Damage to Features Close to Home and Around the World

Blurry images will mix up the details in the foreground and in the background. At times, small details such as how the marks are curved or cross may not stand out, whereas the whole shape can look abnormal. This combined issue makes it hard for the model to pick out useful and stable patterns during training.

5.5.Greater Noise Sensitivity

When blur occurs, noise might become a problem for the model’s understanding. Noises added during blurring can hide important parts of the image which can weaken the model’s overall performance.

5.6.Problems with using Traditional Techniques

Using the usual image processing approaches on blurred images usually leads to worse results. Using blurry images makes it necessary to introduce new, smart recognition processes to get satisfactory performance.

6. CONCLUSION

Blurring the MNIST dataset brings a number of challenges that have a negative effect on the performance of models recognizing handwritten digits. Blurring hides small details, makes edges less distinct, lowers contrast and changes both the details in a single area and the whole image. As a result, it becomes hard for traditional methods to correctly handle the new images. Being sensitive to sound raises the challenge of detecting and identifying natural disasters. Yet, dealing with these obstacles improves model strength and makes recognition systems ready for actual situations where image clarity often changes.

The experiments clearly proved that deep learning accuracy is lowered by blurring images, since good image quality provides useful information for recognition. The application of hazy filters such as Gaussian kernels, smooths out essential parts of an image and makes important details harder to detect (Latif et al., 2022). This situation lowers the accuracy of what the algorithm can recognize.

Table 1 shows that DenseNet achieves higher accuracy when the images are heavily blurred (e.g., with Gaussian sigma of 8), leading us to conclude that it works better than other models. While sigma gets larger and there’s more distortion, DenseNet is still accurate in classifying images while other models see a bigger drop in performance. In another experiment with the GoPro data, several deblurring approaches were tested to restore the fine details removed by both motion blur and noise. There was no one method that could fully restore the image quality. Instead, applying several types of blending methods made the images clearer and more useful, as determined by PSNR and SSIM.

This research introduces an approach that combines multiple algorithms for removing blur with computer vision methods. By using this approach, edges in the image are improved, noise is reduced and important image metrics are boosted, with a PSNR of 29.28 dB and an SSIM of 0.685. Because of these improvements, objects can be found more accurately and information from hazy images can be retrieved better.

Generally, it appears that learning techniques designed for image distortion are essential for development. With models that include visual damage, we can still study valuable data in images that are visually lowered. Further improvements in both image reconstruction and analysis can be made in future research by merging deep learning deblurring with advanced computer vision tools.

REFERENCES

[1]Akmaljon Palvanov, Young Im Cho, “Comparisons of Deep Learning Algorithms for MNIST in Real-Time Environment”, International Journal of Fuzzy Logic and Intelligent Systems, vol. 18, no. 2, pp. 126–134, 2018.

[2]Artificial Intelligence +. (n.d.). *Overfitting vs underfitting in machine learning algorithms*. https://www.aiplusinfo.com/blog/overfitting-vs-underfitting-in-machine-learning-algorithms/

[3]Baldominos, A., Saez, Y., & Isasi, P. (2018). A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences*, *9*(15), 3169. https://doi.org/10.3390/app9153169

[4]CheCampbell, R. (2018). Autonomous RF DAta Collection with Software Defined Radio. https://core.ac.uk/download/159487652.pdf

[5]Chen Zhong, Yuhang Wang, Decheng Zhang, Ke Wang, “Handwritten digit recognition based on corner detection and convolutional neural network”, Journal of Physics: Conference Series, Volume 1651, The 2020 second International Conferenceon Artificial Intelligence Technologies and Application (ICAITA) 2020 21-23 August2020, Dalian, China

[6]Concessao, R. (2021, November 11). Digital Image Processing. EDUCBA. https://www.educba.com/digital-image-processing/

[7]D. Beohar and A. Rasool, "Handwritten Digit Recognition of MNIST dataset using Deep Learning state-of-the-art Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 542-548, doi: 10.1109/ESCI50559.2021.9396870.

[8]D. -y. Ge, X. -f. Yao, W. -j. Xiang, X. -j. Wen and E. -c. Liu, "Design of High Accuracy Detector for MNIST Handwritten Digit Recognition Based on Convolutional Neural Network," 2019 12th International Conference on Intelligent Computation Technologyand Automation (ICICTA), Xiangtan, China, 2019, pp. 658-662, doi: 10.1109/ICICTA49267.2019.00145.

[9]Deng, L. (2012). “The mnist database of handwritten digit images for machine learning research [best of the web]”, IEEE Signal Processing Magazine, 29(6), 141-142.

[10] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, Qinfeng Shi, “From Motion Blur to Motion Flow: a Deep Learning Solution

for Removing Heterogeneous Motion Blur.” Computer Vision and Pattern Recognition 2016

[11] Elvanco. (2024). *How to use cross-validation to evaluate a machine learning model in 2024?* https://elvanco.com/blog/how-to-use-cross-validation-to-evaluate-a-machine

[12] Garg, A., Gupta, D., Saxena, S., Sahad, P.P. (2019). Validation of Random Dataset using an efficient CNN model trained on MNIST handwritten Dataset, Department of Computer Science & Engineering, IIIT Bhubaneswar, Bhubaneswar, Odisha, 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)

[13] Garma, R. J. D. (2015). Image Quality Modeling and Characterization of Nyquist Sampled Framing Systems with Operational Considerations for Remote Sensing. https://core.ac.uk/download/232140468.pdf

[14] Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I., Shen, C., Anton, V. D. H., & Shi, Q. (2017). *From Motion Blur to Motion Flow: A deep learning solution for removing heterogeneous motion blur*. https://openaccess.thecvf.com/content_cvpr_2017/html/Gong_From_Motion_Blur_CVPR_2017_paper.html

[15] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik, “EMNIST: extending MNIST to handwritten letters”, The MARCS Institute for Brain, Behaviourand Development Western Sydney University.

[16] Guo, Qing, Wei Feng, Ruijun Gao, Yang Liu, and Song Wang. "Exploring the effects of blur and deblurring to visual object tracking." IEEE Transactions on Image Processing 30 (2021): 1812-1824.

[17] Haijian Shao, Edwin Ma, Ming Zhu1, Xing Deng, Shengjie Zhai1, “MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization”, Department of Electrical and Computer Engineering, University of Nevada Las Vegas, Las Vegas, 89154, NV, USA.

[18] Hemmatian, J., Hajizadeh, R., & Nazari, F. (2025). Addressing imbalanced data classification with Cluster-Based Reduced Noise SMOTE. PLoS One, 20(2), e0317396.

[19] Hernández-Betancourt, N. (2018). Evaluation and comparison of techniques for reconstructing the point spread function of images blurred by uniform linear motion. TecnoLóGicas. https://doi.org/10.22430/22565337.789

[20] Huihui, Yu, Li Daoliang, and Chen Yingyi, "A state-of-the-art review of image motion deblurring techniques in precision agriculture." Heliyon 9, no. 6 (2023).

[21] Hur, S., & Park, Y. (2019). Application of Deep Convolutional Neural Networks and Smartphone Sensors for Indoor Localization. Applied Sciences, 9(11), n/a.

[22] Insight Software Consortium. (n.d.). *ITK: itk::WienerDeconvolutionImageFilter< TInputImage, TKernelImage, TOutputImage, TInternalPrecision > class template reference*. https://itk.org/Doxygen44/html/classitk_1_1WienerDeconvolutionImageFilter.html

[23] J. Flusser, T. Suk and S. Saic, "Recognition of blurred images by the method of moments," in IEEE Transactions on Image Processing, vol. 5, no. 3, pp. 533-538, March 1996, doi: 10.1109/83.491327.

[24] Jain and B. K. Sharma, "Analysis of activation functions for convolutional neural network based mnist handwritten character recognition," International Journal of Advanced Studies of Scientific Research, vol. 3, no. 9, pp. 1–7, 2019.

[25] Jeon, W. S., Cielniak, G., & Sang-Yong, R. (2018). Semantic Segmentation Using Trade-Off and Internal Ensemble. https://doi.org/10.5391/IJFIS.2018.18.3.196

[26] Ji, Q., Huang, J., We, H., & Sun, Y. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. Algorithms. https://doi.org/10.3390/a12030051

[27] Jian Sun, Wenfei Cao, Zongben Xu, Jean Ponce, "Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal." Computer Vision and Pattern Recognition 2015

[28] Jin, Y., Zhou, C., Dai, W., & Dai, W. (2024). Sharpness-Based Distance Detection. Applied Sciences, 14(19), 8913.

[29] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, Hongdong Li, "Deep Image Deblurring: A Survey." International Journal of Computer Vision 2022

[30] Kandel, I., & Castelli, M. (2020). *Transfer learning with convolutional neural networks for diabetic retinopathy image classification: A review*. *Applied Sciences, 10*(6), 2021. https://doi.org/10.3390/app10062021

[31] Karri, C., Cheikhrouhou, O., Harbaoui, A., Zaguia, A., & Hamam, H. (2021). Privacy Preserving Face Recognition in Cloud Robotics: A Comparative Study. Applied Sciences, 11(14), 6522. https://doi.org/10.3390/app11146522

[32] Kayar, Y. (2024). Innovative Approaches to Clinical Diagnosis: Transfer Learning in Facial Image Classification for Celiac Disease Identification. Applied Sciences, 14(14), 6207. https://doi.org/10.3390/app14146207

[33] Kussul, E., & Baidyk, T. (2004). “Improved method of handwritten digit recognition tested on MNIST database. Image and Vision Computing”, 22(12), 971-981.

[34] Latif, S., Nayyer, M. Z., Raza, I., Hussain, S. A., Jamal, M. H., Hur, S., & Ashraf, I. (2022). Cloudlet Federation Based Context-Aware Federated Learning Approach. IEEE Access. https://doi.org/10.1109/access.2022.3212550

[35] Lee, C. H., & Qiu, Z. (2016). Motion Image De-blurring System Based on the Effectiveness Parameters of Point Spread Function. https://doi.org/10.1145/3033288.3033308

[36] Lokhande, S. D., Agrawal, D. P., & Gohokar, V. D. (2006). *Identification of parameters and restoration of motion blurred images*. In *Proceedings of the 2006 ACM symposium on Applied computing (SAC '06)*. BibSLEIGH. http://bibtex.github.io/SAC-2006-LokhandeAG.html

[37] Manzoor, S., Ye-Chan, A., Zhang, Y., & Kim, S. (2023). *SPT: Single pedestrian tracking framework with re-identification-based learning using the Siamese model*. *Sensors, 23*(10), 4906. https://doi.org/10.3390/s23104906

[38] Mohsenzadegan, K., & Kyamakya, K. (2020). A Visual Sensing Concept for Robustly Classifying House Types through a Convolutional Neural Network Architecture Involving a Multi-Channel Features Extraction. Sensors, 20(19), 5672.

[39] Monk, S. D., Jorgensen, E., McCulloch, A., Peacock, R., Sangprachatanaruk, I., Azeta, J., Bolu, C., & Taylor, C. J. (2022). A Prototype Telepresence Robot for Use in the Investigation of Ebola and Lassa Virus-Threatened Villages in Nigeria. https://core.ac.uk/download/574335154.pdf

[40] Navarro, Fernando, Francisco J. Serón, and Diego Gutierrez, "Motion blur rendering: State of the art." In Computer Graphics Forum, vol. 30, no. 1, pp. 3-26. Oxford, UK: Blackwell Publishing Ltd, 2011.

[41] Nazaré, Tiago S., Gabriel B. Paranhos da Costa, Welinton A. Contato, and Moacir Ponti. "Deep convolutional neural networks and noisy images." In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, November 7–10, 2017, Proceedings 22, pp. 416-424. Springer International Publishing, 2018.

[42] Nguyen, H., Wang, Y., & Wang, Z. (2020). *Single-shot 3D shape reconstruction using structured light and deep convolutional neural networks*. *Sensors*, *20*(13), 3718. https://doi.org/10.3390/s20133718

[43] Norman Mu, Justin Gilmer, "MNIST-C: A Robustness Benchmark for Computer Vision", ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning, 5 Jun 2019.

[44] Nurul Izzatie Husna, M. F. (2023). Midrange exploration exploitation searching particle swarm optimization with HSV-template matching for crowded environment object tracking. https://core.ac.uk/download/613052556.pdf

[45] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, Jiri Matas, "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks." Computer Vision and Pattern Recognition CVPR 2018

[46] Owhadi, H., & Scovel, C. (2017). Qualitative Robustness in Bayesian Inference. https://doi.org/10.1051/ps/2017014/

[47] PGYER. (2024). *Luqo Start Lite APK for Android download*. https://www.pgyer.com/apk/apk/com.luqo.start

[48] Price, K. (n.d.). *Co-author index for Saic, S.* Keith Price Bibliography. Retrieved May 24, 2025, from https://visionbib.com/bibliography/coauth/saic.html

[49] R. Lokhande, K. V. Arya, P. Gupta, "Identification of Parameters and Restoration of Motion Blurred Images", SAC '06: Proceedings of the 2006 ACM symposium on Applied computing, 23 April 2006.

[50] Review History for Enhancing software defect prediction: a framework with improved feature selection and ensemble machine learning [PeerJ]. https://peerj.com/articles/cs-1860/reviews/

[51] Rosalina, R., Hutagalung, J. P., & Sahuri, G. (2020). Hiragana Handwriting Recognition Using Deep Neural Network Search. https://core.ac.uk/download/482969143.pdf

[52] Runwal, N. R., Purve, N. S. J., & Chandak, N. M. (2023). Classification of Handwritten Digits on the web using Deep Learning. *INTERNATIONAL JOURNAL OF NEXT-GENERATION COMPUTING*. https://doi.org/10.47164/ijngc.v14i1.1003

[53] S. S. Kadam, A. C. Adamuthe and A. B. Patil, "CNN model for image classification on MNIST and fashionMNIST dataset," Journal of Scientific Research, vol. 64, no. 2, pp.374–384, 2020

[54] Shanghang Zhang, Xiaohui Shen, Zhe Lin, Radomír Měch, João P. Costeira, José M. F. Moura; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6586-6595.

[55] Sharabas, D., Varlet, M., & Grootswagers, T. (2023). An online browser-based attentional blink replication using visual objects. PLoS One, 18(8), e0289623.

[56] Solé-Casals, J., Travieso, C. M., Ferrer, M. A., Alonso, J. B., & Briceno, J. C. (2009). Automatic Recognition of Leaves by Shape Detection Pre-Processing with Ica. https://core.ac.uk/download/50524476.pdf

[57] Sun, J., Cao, W., Xu, Z., & Ponce, J. (2015). Learning a convolutional neural network for non-uniform motion blur removal. https://doi.org/10.1109/CVPR.2015.7298677

[58] Survey Instrument Services. (2020). *Pix4Dfields 1.8 introduces the advanced layer visualization, giving you more control over your data*. https://www.ssass.co.za/post/pix4dfields-1-8-introduces-the-advanced-layer-visualization-giving-you-more-control-over-your-data

[59] Uličný, Matej, Jens Lundström, and Stefan Byttner. "Robustness of deep convolutional neural networks for image recognition." In Intelligent Computing Systems: First International Symposium, ISICS 2016, Mérida, México, March 16-18, 2016, Proceedings 1, pp. 16-30. Springer International Publishing, 2016.

[60] VBT Academie. (n.d.). *What is cross-validation and why is it important in machine learning?* https://academie.voetbaltrainer.nl/forums/topic/what-is-cross-validation-and-why-is-it-important-in-machine-learning/

[61] Wang, H., Ming-hui, S., Zhang, H., & Li-yan, D. (2022). LHPE-nets: A lightweight 2D and 3D human pose estimation model with well-structural deep networks and multi-view pose sample simplification method. PLoS One, 17(2), e0264302.

[62] Weiwei Jiang, “MNIST-MIX: a multi-language handwritten digit recognition dataset”, Weiwei Jiang 2020 IOPSciNotes 1 025002.

[63] Wulveryck, O. (2017, July 7). *A “Smart” CCTV with TensorFlow, and Inception? On a Raspberry Pi?* Unladen Swallow. https://blog.owulveryck.info/2017/07/07/a-smart-cctv-with-tensorflow-and-inception-on-a-rapsberry-pi.html

[64] Xiangyu Zhang∗, Xinyu Zhou∗ Mengxiao Lin Jian Sun Megvii Inc, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices”

[65] Xinyi Zhang, Hang Dong, Zhe Hu, Wei-Sheng Lai, Fei Wang, Ming-Hsuan Yang. “Gated Fusion Network for Joint Image Deblurring and Super-Resolution.” Computer Vision and Pattern Recognition BMVC 2018

[66] Y. Wang, F. Li, H. Sun, W. Li, C. Zhong et al., “Improvement of MNIST image recognition based on CNN,” in IOP Conf. Series: Earth and Environmental Science. 7th Annual Int. Conf. on Geo-Spatial Knowledge and Intelligence, Guangzhou, China,vol. 428, no. 1, 2020. https://doi.org/10.1088/1755-1315/428/1/012097

[67] Y. Yitzhaky, I. Mor, A. Lantzman, and N. S. Kopeika, "Direct method for restoration of motion-blurred images," J. Opt. Soc. Am. A 15, 1512-1519 (1998), https://opg.optica.org/josaa/abstract.cfm?URI=josaa-15-6-1512.

[68] Yang Gong, Pan Zhang, “Research on Mnist Handwritten Numbers Recognition basedon CNN”, Journal of Physics: Conference Series, Volume 2138, International Conference on Artificial Intelligence and Big Data Applications (ICAIBD 2021) 24-25September 2021, Hubei, China.

[69] YANMIN, Z. (2019). Split-and-Merge Algorithm for Motion Deblurring Based on DeblurGAN. https://core.ac.uk/download/226439978.pdf

[70] Yin, J., Shi, Y., Deng, W., Yin, C., Wang, T., Song, Y., Li, T., & Li, Y. (2023). *Internet of Things intrusion detection system based on convolutional neural network. Computers, Materials & Continua, 75*(1), 2119–2135. https://doi.org/10.32604/cmc.2023.035077

[71] Zahin, M. A. (2023). Multi-headed self-attention mechanism-based Transformer model for predicting bus travel times across multiple bus routes using heterogeneous datasets. https://doi.org/10.32469/10355/96193

[72] Zhang, H., Zhang, Y., Wang, L., Hu, Z., Zhou, W., Tsang, P. W., Cao, D., & Poon, T. (2020). Study of Image Classification Accuracy with Fourier Ptychography. *Applied Sciences*, *11*(10), 4500. https://doi.org/10.3390/app11104500

[73] Zhao, S., Zhang, Z., Hong, R., Xu, M., Yang, Y., & Wang, M. (2022). FCL-GAN: A Lightweight and Real-Time Baseline for Unsupervised Blind Image Deblurring. http://arxiv.org/abs/2204.07820

[74] Zhi, L., Jia, Z., Yang, J., & Kasabov, N. (2020). Low Illumination Video Image Enhancement. https://doi.org/10.1109/JPHOT.2020.3010966

[75] Srinivasan, K., Garg, L., Datta, D., Alaboudi, A. A., Jhanjhi, N. Z., Agarwal, R., & Thomas, A. G. (2021). Performance comparison of deep CNN models for detecting driver’s

distraction. Computers, Materials & Continua, 69(3), 3453–3468. https://doi.org/10.32604/cmc.2021.016736

[76] Kiruthika Devi, S., Albraikan, A. A., Al-Wesabi, F. N., Nour, M. K., Ashour, A., & Hilal, A. M. (2023). Intelligent deep convolutional neural network-based object detection model for visually challenged people. *Computer Systems Science and Engineering, 47*(2), 1905–1920. https://doi.org/10.32604/csse.2023.036980

[77] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6848–6856

[78] Arora, A. (2020, August 2). DenseNet architecture explained with PyTorch implementation from TorchVision. Retrieved from https://amaarora.github.io/posts/2020-08-02-densenets.html

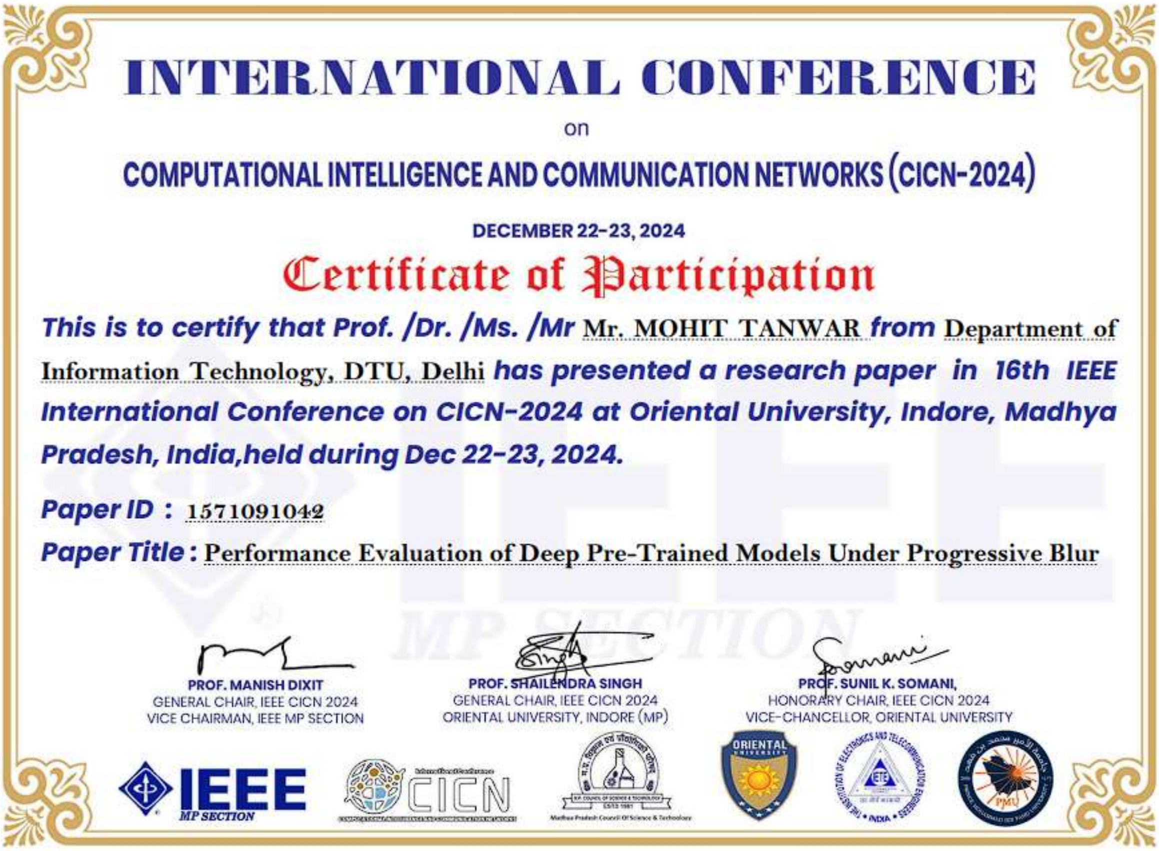
[79] Tiwari, R. S. (2021, September 2). *Transfer learning — Part 4.0!! VGG-16 and VGG-19*. Becoming Human: Artificial Intelligence Magazine. https://becominghuman.ai/transfer-learning-part-4-0-vgg-16-and-vgg-19-d7f0045032de

[80] Rismiyati, & Luthfiarta, A. (2021). VGG16 transfer learning architecture for salak fruit quality classification (Klasifikasi kualitas buah salak dengan transfer learning arsitektur VGG16). Telematika: Jurnal Informatika dan Teknologi Informasi, 18(1), 37–48. https://doi.org/10.31515/telematika.v18i1.4025

PUBLICATION DETAILS

1. M. Tanwar and S. Susan, "Performance Evaluation of Deep Pre-Trained Models Under Progressive Blur," 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN), Indore, India, 2024, pp. 1254-1258, Doi: https://doi.org/10.1109/CICN63059.2024.10847407
2. Mohit Tanwar, Seba Susan, "Enhanced Image Restoration Using Hybrid Wiener–Lucy-Richardson Deblurring and Histogram Equalization", ICCNet-2025 (LNNS-Scopus indexed Springer series) 1st-3rd August 2025. (Accepted)

PUBLICATION CERTIFICATE



INTERNATIONAL CONFERENCE

on

COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS (CICN-2024)

DECEMBER 22-23, 2024

Certificate of Participation

This is to certify that Prof. /Dr. /Ms. /Mr Mr. MOHIT TANWAR *from* Department of Information Technology, DTU, Delhi *has presented a research paper in 16th IEEE International Conference on CICN-2024 at Oriental University, Indore, Madhya Pradesh, India,held during Dec 22-23, 2024.*

Paper ID : 1571091042

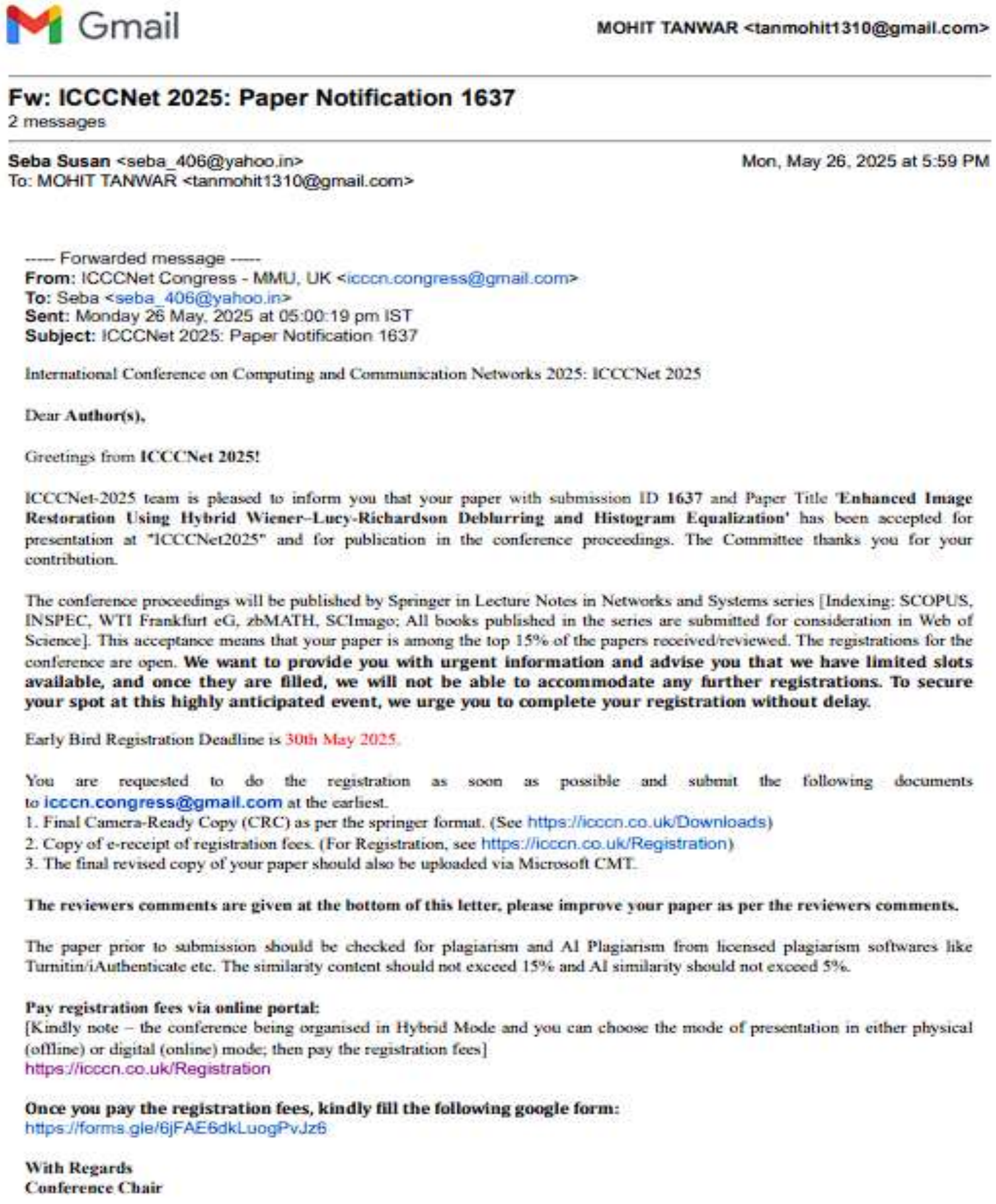
Paper Title : Performance Evaluation of Deep Pre-Trained Models Under Progressive Blur

PROF. MANISH DIXIT
GENERAL CHAIR, IEEE CICN 2024
VICE CHAIRMAN, IEEE MP SECTION

PROF. SHAILENDRA SINGH
GENERAL CHAIR, IEEE CICN 2024
ORIENTAL UNIVERSITY, INDORE (MP)

PROF. SUNIL K. SOMANI,
HONORARY CHAIR, IEEE CICN 2024
VICE-CHANCELLOR, ORIENTAL UNIVERSITY

ACCEPTANCE LETTER



Gmail

MOHIT TANWAR <tanmohit1310@gmail.com>

Fw: ICCCNet 2025: Paper Notification 1637

2 messages

Seba Susan <seba_406@yahoo.in>
To: MOHIT TANWAR <tanmohit1310@gmail.com>

Mon, May 26, 2025 at 5:59 PM

----- Forwarded message -----
From: ICCCNet Congress - MMU, UK <icccn.congress@gmail.com>
To: Seba <seba_406@yahoo.in>
Sent: Monday 26 May, 2025 at 05:00:19 pm IST
Subject: ICCCNet 2025: Paper Notification 1637

International Conference on Computing and Communication Networks 2025: ICCCNet 2025

Dear **Author(s),**

Greetings from **ICCCNet 2025!**

ICCCNet-2025 team is pleased to inform you that your paper with submission ID **1637** and Paper Title '**Enhanced Image Restoration Using Hybrid Wiener–Lucy-Richardson Deblurring and Histogram Equalization**' has been accepted for presentation at "ICCCNet2025" and for publication in the conference proceedings. The Committee thanks you for your contribution.

The conference proceedings will be published by Springer in Lecture Notes in Networks and Systems series [Indexing: SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago; All books published in the series are submitted for consideration in Web of Science]. This acceptance means that your paper is among the top 15% of the papers received/reviewed. The registrations for the conference are open. **We want to provide you with urgent information and advise you that we have limited slots available, and once they are filled, we will not be able to accommodate any further registrations. To secure your spot at this highly anticipated event, we urge you to complete your registration without delay.**

Early Bird Registration Deadline is 30th May 2025.

You are requested to do the registration as soon as possible and submit the following documents to **icccn.congress@gmail.com** at the earliest.
1. Final Camera-Ready Copy (CRC) as per the springer format. (See https://icccn.co.uk/Downloads)
2. Copy of e-receipt of registration fees. (For Registration, see https://icccn.co.uk/Registration)
3. The final revised copy of your paper should also be uploaded via Microsoft CMT.

The reviewers comments are given at the bottom of this letter, please improve your paper as per the reviewers comments.

The paper prior to submission should be checked for plagiarism and AI Plagiarism from licensed plagiarism softwares like Turnitin/iAuthenticate etc. The similarity content should not exceed 15% and AI similarity should not exceed 5%.

Pay registration fees via online portal:
[Kindly note – the conference being organised in Hybrid Mode and you can choose the mode of presentation in either physical (offline) or digital (online) mode; then pay the registration fees]
https://icccn.co.uk/Registration

Once you pay the registration fees, kindly fill the following google form:
https://forms.gle/6jFAE6dkLuogPvJz6

With Regards
Conference Chair

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis____

____

Total Pages ________________ Name of the Scholar____________________________________

Supervisor (s)

(1)______________________________________
(2)______________________________________
(3)______________________________________

Department____

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: _______________________ Similarity Index: ______, Total Word Count: ________

Date: ___________

Candidate's Signature **Signature of Supervisor(s)**

Seba Susan

PERFORMANCE EVALUATION OF DEEP LEARNING APPROACHES FOR BLURRED IMAGE PROCESSING

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98200302

Submission Date

May 28, 2025, 8:02 PM GMT+5:30

Download Date

May 28, 2025, 8:04 PM GMT+5:30

File Name

Dissertation (1).docx

File Size

1.7 MB

35 Pages

7,858 Words

44,026 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Small Matches (less than 10 words)

Match Groups

26 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

4 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
1%	Publications
9%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **26** Not Cited or Quoted 8%
  Matches with neither in-text citation nor quotation marks
- **4** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2%	Internet sources
1%	Publications
9%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	GGS IP University Delhi on 2025-05-27	6%
2	Submitted works	GGS IP University Delhi on 2025-05-24	<1%
3	Submitted works	GGS IP University Delhi on 2025-05-23	<1%
4	Submitted works	GGS IP University Delhi on 2025-05-23	<1%
5	Internet	ijireeice.com	<1%
6	Internet	papers.miccai.org	<1%
7	Internet	escholarship.org	<1%
8	Publication	Rizk, Rodrigue. "Deciphering Ancient Scripts with Scarce Annotations Using Deep ...	<1%
9	Internet	www.nature.com	<1%
10	Submitted works	Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering and Com...	<1%

11	Publication	
Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...		<1%

e-Receipt for State Bank Collect Payment

REGISTRAR, DTU (RECEIPT A/C)

BAWANA ROAD, SHAHABAD DAULATPUR, , DELHI-110042
Date: 26-May-2025

SBCollect Reference Number :	DUO1114404	**Category :**	Miscellaneous Fees from students
Amount :	₹3000		
University Roll No :	2K23/ITY/21		
Name of the student :	MOHIT TANWAR		
Academic Year :	2024-25		
Branch Course :	Infromation Technology		
Type/Name of fee :	Others if any		
Remarks if any :	M.Tech IT Dissertation Fees		
Mobile No. of the student :	9306260380		
Fee Amount :	3000		
Transaction charge :	0.00		

Total Amount (In Figures) :	3,000.00	**Total Amount (In words) :**	Rupees Three Thousand Only
Remarks :	M.Tech IT Dissertation Fees	**Notification 1:**	Late Registration Fee, Hostel Room rent for internship, Hostel cooler rent, Transcript fee (Within 5 years Rs.1500/- & $150 in USD, More than 5 years but less than 10 years Rs.2500/- & $250 in USD, More than 10 years Rs.5000/- & $500 in USD) Additional copies Rs.200/- each & $20 in USD each, I-card fee,Character certificate Rs.500/-.
Notification 2:	Migration Certificate Rs.500/-, Bonafide certificate Rs.200/-, Special certificate (any other certificate not covered in above list) Rs.1000/-,Provisional certificate Rs.500/-, Duplicate Mark sheet (Within 5 years Rs.2500/- & $250 in USD, More than 5 years but less than 10 years Rs.4000/- & $400 in USD, More than 10 years Rs.10000/- & $1000 in USD)		

Thank you for choosing SB Collect. If you have any query / grievances regarding the transaction, please contact us

Toll-free helpline number i.e. 1800-1111-09 / 1800 - 1234/1800 2100

Email -: sbcollect@sbi.co.in