ReSkipNet: Skip Connected Convolutional Autoencoder for Original Document Denoising

***Abstract*—Data pre-processing, data analysis, and Optical Character Recognition need a huge amount of clean data, and document images are usually a good source for this. However, document images frequently exhibit blurring and various other forms of noise, which can pose challenges in their manipulation and analysis. To denoise and deblur such document images, autoencoders have been used for a long time. For this task, we propose a novel Convolutional Autoencoder Network which is composed of multiple skip-connected residual blocks and other layers for supporting the encoder and decoder parts. This model not only uses less computational power to denoise existing document image datasets but also performs well. While prior research primarily concentrates on optimizing evaluation metrics, our approach additionally prioritizes larger resolution input sizes. This characteristic of using larger image sizes enhances its practicality and usability as real-world documents are typically characterized by a higher word density. Moreover, in order to further advance the development of our model, we produced an original dataset and proceeded to train our model on this dataset, resulting in satisfactory outcomes.**

***Index Terms*—Convolutional Autoencoder, Residual Block, Skip Connections, Denoising, Documents, Image Processing, Original Dataset**

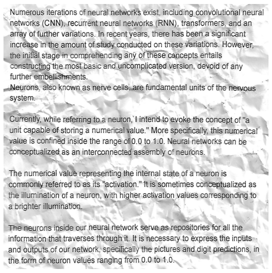
I. Introduction

Most modern documents are digitized, but many older ones are not. Typically, these old papers are preserved in the form of scanned images or photographs. The textual content present in the document images cannot be directly recognized or accessed using existing search and analysis technologies. Optical Character Recognition (OCR) helps to effectively analyze these documents and it refers to the systematic procedure of organizing and digitizing handwritten or printed documents. Nevertheless, the clarity of document images and the quality of image recognition are typically compromised by factors such as noise, blur, inkblot, fading, paper aging, food stains, and other similar issues. This is why many important documents are stored offline. Preprocessing document images is necessary before optical character recognition. In particular, data preprocessing and analysis require clean and denoised images. Preprocessing improves image quality to permit further processing. This is done by reducing document picture noise and blur. A typical degraded document from our original dataset is shown in Fig. 1.

The field of image restoration techniques has garnered increasing attention in recent decades. The objective is to generate a new image that exhibits reduced levels of noise and blur, while also closely resembling the original image. Similar to prior studies, the distorted image of poor quality can be expressed as:

$$y \ = \ D(x) \ + \ n \tag{1}$$

where y represents the degraded image, x represents the original image of good quality, D denotes the degradation function, and n represents the noise as denoted in equation (1). The process of image restoration is sometimes referred to as an inverse issue, that is, the estimation of variable x based on the observation of variable y.



Numerous iterations of neural networks exist, including convolutional neural networks (CNN), recurrent neural networks (RNN), transformers, and an array of further variations. In recent years, there has been a significant increase in the amount of study conducted on these variations. However, the initial stage in comprehending any of these concepts entails constructing the most basic and uncomplicated version, devoid of any further embellishments.

Neurons, also known as nerve cells, are fundamental units of the nervous system.

Currently, while referring to a neuron, I intend to evoke the concept of "a unit capable of storing a numerical value." More specifically, this numerical value is confined inside the range of 0.0 to 1.0. Neural networks can be conceptualized as an interconnected assembly of neurons.

The numerical value representing the internal state of a neuron is commonly referred to as its "activation." It is sometimes conceptualized as the illumination of a neuron, with higher activation values corresponding to a brighter illumination.

The neurons inside our neural network serve as repositories for all the information that traverses through it. It is necessary to express the inputs and outputs of our network, specifically the pictures and digit predictions, in the form of neuron values ranging from 0.0 to 1.0.

Fig. 1. Conventional Degraded Document Image

This study proposes restoring noisy document images with a convolutional auto-encoder. Autoencoders have an encoder and a decoder. This model uses residual blocks because we plan to use deep neural architecture. Using a deep architecture for computer vision is beneficial. This is largely due to deeper networks' higher parameter count, which improves their feature representation learning. However, we have made deliberate efforts to optimize our model parameters. The second benefit of increasing network depth is a wider network receptive field, which helps convolutional layer kernels learn more contextual information. However, the vanishing gradient problem often arises as network depth increases. This vanishing gradient problem is solved by using skip connections between residual blocks in our model.

We trained our model on the NoisyOffice dataset [1] and achieved better results than the original model employed for this particular dataset. To analyze the effectiveness of our model on the NoisyOffice dataset [1], it is important to highlight that our trained model exhibits greater computational efficiency and can handle larger image sizes or resolutions without compromising evaluation measures. We also created a new dataset to advance our model. Our dataset used actual crumpled, folded, and wrinkled papers to preserve the authenticity of the noise in document pictures. Our model is tested on this newly acquired dataset, yielding favorable outcomes. This clarifies our model's adaptability.

To conclude, our primary contributions lie within the following areas:

- Proposed a convolutional autoencoder model optimized for the existing denoising document image dataset called NoisyOffice [1].
- The proposed model uses skip connections between residual blocks to solve the vanishing gradient problem.
- Our model has lower parameters while keeping the performance metrics intact.
- Bigger dimension input sizes are employed for training the document images, rendering them more suitable for practicality.
- Produced an original dataset to test the adaptability of our model.

II. Literature Review

The primary emphasis of traditional restoration algorithms is on natural scene photos. However, due to the significant increase in the demand for optical character recognition (OCR), recent studies have been conducted to address document restoration [2]–[7]. The approach described by Chen et al. [3] utilizes document picture foreground segmentation as its foundation. The study conducted by Cho et al. [4] compares document images to natural-scene images and adds their unique aspects to the optimization process. The L0-regularized intensity and gradient prior are employed in [5]. The methods described in [7] and [8] utilize deep neural networks as their underlying framework. The basis for [9] is derived from the two-tone before. In a broad sense, the field of image restoration encompasses various components, including denoising [10]–[14], deblurring [3]–[5], [7], [15], debayering [16], [17], and super-resolution [18], [19], among others. The two salient facets of this study encompass denoising and deblurring.

Elad et al. [20] use sparse, redundant representations with taught dictionaries to eliminate zero-mean white and homogeneous Gaussian additive noise from images. Block-matching and 3D filtering (BM3D) effectively reduce noise in images impacted by Additive White Gaussian noise (AWGN) [10]. Local sparse representation of an image in the transform domain grounds the BM3D. The first stage is categorizing linked 2D image fragments into 3D data arrays. For 3D groups, collaborative filtering is employed. Finally, the denoised image is obtained using an inverse 3D transform.

In recent years, learning-based photo restoration algorithms have dominated. Deep neural network-based solutions have become the norm in this discipline. According to Vincent et al., [21], the Stacked Denoising Autoencoders (SDA) approach builds deep architectures by piling up layers of autoencoders. Locally trained autoencoders remove noise from faulty input images. The work uses a basic multilayer perceptron (MLP) on picture samples, yielding greater results than the BM3D approach [11]. The authors in [13] introduce a random field-based structure called shrinkage fields. The picture model and optimization method are combined in this architecture to improve computing productiveness and caliber of reconstruction. The authors of [22] developed a deep convolutional network design to capture unique features. They supervised-trained two submodules for deconvolution and artifact removal. A study by [8] used a generative adversarial network (GAN) to improve low-resolution photos by producing clear, high-resolution outputs. A generator and two discriminators are also used to create a specialized GAN to analyze facial and textual images. The researchers used BM3D with a CNN in their study [23].

The most comparable methodologies to our research are put out in [24] and [7]. Our approach employs an encoder-decoder architecture similar to that described in reference [24]. However, there are notable distinctions as we do not utilize a deconvolution layer within the network. The methodology described in [7] utilizes a convolutional neural network with 15 layers. In contrast to the study conducted by [7], our network incorporates batch normalization layers [25] and skip-connections [26]. The most similar study to ours can be found is SCDCA by Zhao et al. [27]

III. Proposed Methodology

We suggest a skip-connected residual deep convolutional autoencoder, illustrated in Fig. 2, for restoring or denoising document images. The goal of the proposed approach is to use low-quality document images to learn a noise or fade reduction function from beginning to end. The encoder and decoder make up our model. Multiple skip-connected residual blocks bring initial information to a bottleneck layer in the encoder. The bottleneck layer contains extracted image features without blur or noise. Finally, the decoder denoises the encoded input.

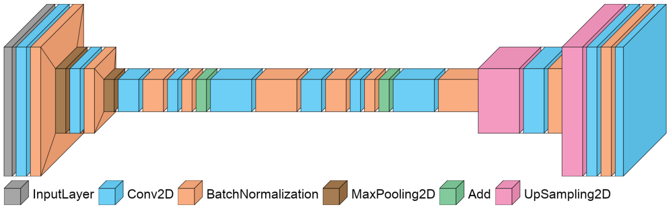


Fig. 2. Model architecture

A. Encoder Network

The encoder network extracts input features into a latent space. The autoencoder helps in reducing the bottleneck layer feature map size. To do this, two Max Pooling layers following each convolutional layer reduce feature map sizes. Convolutional layers stabilize feature size handling. We utilize the ReLU activation function with convolutional layers because it adds non-linearity to deep learning models and solves the vanishing gradients problem. All convolutional layers have 3x3 kernels. Moreover, the dimensions of the output for each image remain consistent with those of the original image. This is done by keeping the stride and padding of each convolutional layer the same. Finally, the feature map that has been extracted so far in the encoder part of the model is stored and transferred to the residual blocks.

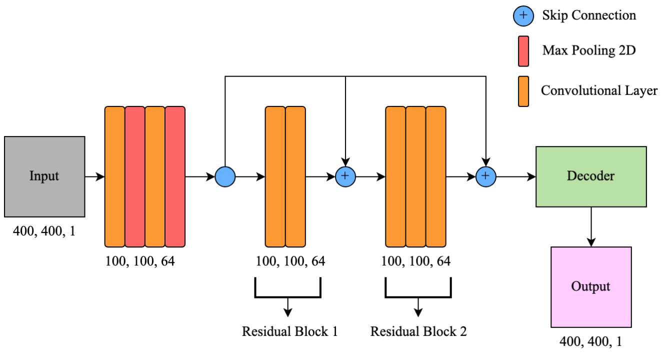


Fig. 3. Mechanism of the Encoder part

1) Residual Block: One of the primary units of the encoder for this model is the residual blocks. Residual blocks were introduced in [26] to address the vanishing degradation issue in image classification. Numerous studies have shown that this structure effectively addresses low-level vision issues [28]–[31], leading to the development of several residual block variations. We employed two residual blocks in our model. The first residual block has 2 convolutional and batch normalization layers [25]. Batch normalization layers enhance the efficiency and stability of training artificial neural networks by the normalization of input layers through means of re-centering and re-scaling. The second residual block is composed of three convolutional layers. As before, every convolutional layer is followed by a batch normalization layer. The number of filters in the first convolutional layers of the model is 64 which is doubled for each upcoming residual block. For example, in the initial residual block, the quantity of filters begins at 128, but in the subsequent residual block, this value increases to 256. However, the number is normalized to 64 (to match the first layers) at the end of each residual block to satisfy the additive nature of the skip connections among the residual blocks.

2) Skip Connection: As network depth increases, the training rate decreases. For this, we use skip connections. Skip connections organize network layers into blocks and ease data flow in a way that has certain benefits. First, each block improves data, helping the model learn while providing more information. Second, shorter paths help gradients reach each network layer. This accelerates model training. Third, it makes the model more modular, making block additions easier. Now, let us discuss our proposed model, where skip connections are implemented between the two residual blocks. The features are stored in the initial part of the encoder before passing them into the first residual block. Prior to proceeding with the second residual block, it is important to add the features obtained from the initial skip connections to the features gained from the first residual block. Finally, before the information is transferred to the decoder, features of the second residual block are added to the skip connections as well as shown in Fig. 3.

TABLE I
MODEL DESCRIPTION OF CONVOLUTIONAL AUTO ENCODER

Layer Type	Output shape	Parameter	Activation
Input Layer	(None, 400, 400, 1)	0	-
2D Convolution Layer	(None, 400, 400, 64)	640	ReLU
BatchNormalization	(None, 400, 400, 64)	256	-
2D Max Pooling Layer	(None, 200, 200, 64)	0	-
2D Convolution Layer	(None, 200, 200, 64)	36928	ReLU
BatchNormalization	(None, 200, 200, 64)	256	-
2D Max Pooling Layer	(None, 100, 100, 64)	0	-
Residual Block 1			
2D Convolution Layer	(None, 100, 100, 128)	73856	ReLU
BatchNormalization	(None, 100, 100, 128)	512	-
2D Convolution Layer	(None, 100, 100,64)	73792	ReLU
BatchNormalization	(None, 100, 100, 64)	256	-
Add	(None, 100, 100, 64)	0	-
Residual Block 2			
2D Convolution Layer	(None, 100, 100, 256)	147712	ReLU
BatchNormalization	(None, 100, 100, 256)	1024	-
2D Convolution Layer	(None, 100, 100,128)	295040	ReLU
BatchNormalization	(None, 100, 100,128)	512	-
2D Convolution Layer	(None, 100, 100, 64)	73792	ReLU
BatchNormalization	(None, 100, 100, 64)	256	-
Add	(None, 100, 100, 64)	0	-

TABLE II
MODEL DESCRIPTION OF DECODER NETWORK

Layer Type	Output shape	Parameter	Activation
2D Convolution Layer	(None, 100, 100, 256)	147712	ReLU
BatchNormalization	(None, 100, 100, 256)	1024	-
2D UpSampling Layer	(None, 200, 200, 256)	0	-
2D Convolution Layer	(None, 200, 200, 128)	295040	ReLU
BatchNormalization	(None, 200, 200, 128)	512	-
2D UpSampling Layer	(None, 400, 400, 128)	0	-
2D Convolution Layer	(None, 400, 400, 64)	73792	ReLU
BatchNormalization	(None, 400, 400, 64)	256	-
2D Convolution Layer	(None, 400, 400, 1)	577	Sigmoid

B. Decoder Network

Max Pooling layers reduce the feature map in the encoder. Decoders must decode and upscale feature maps to their original proportions. Thus, following the first two convolutional layers, the decoder network has Batch Normalisation and Upsampling layers for each. Upsampling is a weightless layer that doubles input dimensions. The last convolutional layer uses the sigmoid function. To convert input values to a range between 0 and 1, the sigmoid function is used. The final convolutional layer also terminates with a single filter, mirroring the initial input convolutional layer which similarly employs a single filter. This design choice is attributed to the model's processing of grayscale images that possess only one channel.

IV. EXPERIMENT

A. Dataset

1) NoisyOffice: The dataset called NoisyOffice [1] we train our model on includes 144 document images from 18 different fonts. The primary purpose of this dataset is to facilitate the training and evaluation of supervised learning techniques for the tasks of cleaning, binarization, and augmentation of noisy images including grayscale text documents. On average, there are 8 images for each font. There are also two image sizes and eight different types of sludge. There are 144 clean photos that correlate to the training images.

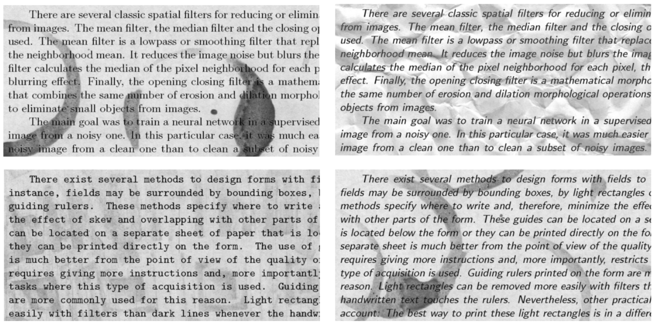
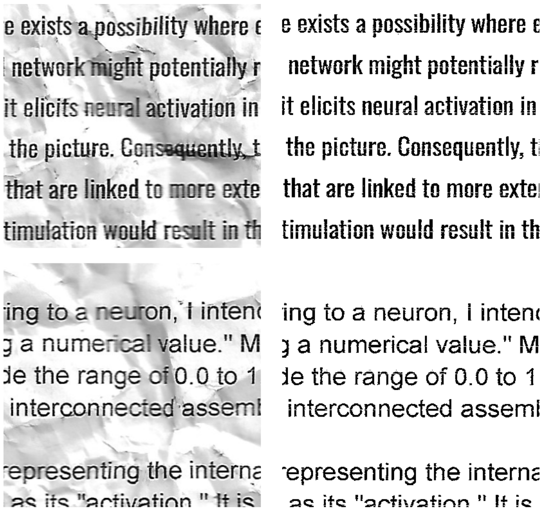


Fig. 4. Sample from NoisyOffice Dataset

2) Original Dataset: For our original dataset, 14 A4 pages of text are generated using 5 different types of most used fonts including Times New Roman and Courier and then printed. In order to obtain accurate clean references of the document images, we initially scanned the photos.



Noisy Clean

Fig. 5. Sample from our Original Dataset

Then, to simulate authentic real-world noise and visual distortion, we proceeded to crease and crumple the papers, introducing wrinkles onto their surface. Subsequently, the simulation incorporates the replication of page wrinkles, smudges, and various forms of degradation. Following that, a second round of scanning was conducted for the noisy image references. While scanning both the clean and dirty images, we have made sure the dimensions are equal and individual pixels are equally proportioned in both images. 14 A4 pages are then cropped into patches of 400 by 400. Due to this, 256 patches are created for both noisy and clean references. After the exclusion of non-squared patches, a total of 165 patches measuring 400 by 400 remain available for training the model. Ultimately, this fresh new dataset is used for evaluating the novelty of our model.

B. Dataset Preparation

The process of data normalization is of utmost importance as it serves as a basic step in preserving the numerical stability of Convolutional Neural Network (CNN) models. The process of data normalization enables a Convolutional Neural Network (CNN) model to acquire knowledge quickly while concurrently guaranteeing the stability of its gradient descent. As a result, the pixel values of the photographs have been adjusted to a numerical range spanning from 0 to 1. This feature enhances the model's ability to maintain fairness by ensuring equitable treatment of pixel or feature values that are larger in magnitude. To assist the rescaling process, the pixel values were multiplied by a factor of 1/255. For the NoisyOffice [1] dataset, we resized the image size to 400 by 400. As we have already made the same size patches for our original dataset, we have not done any size conversion to match the experiment.

C. Experimental Setup and Training Details

We developed our training and prediction environment using Python Deep Learning Libraries like TensorFlow and Keras. For [1], we trained our model with 70 epochs as it reached convergence with Adam optimizer and batch size of 10. As our model is optimized for the [1] we kept everything similar for the Original dataset except for the epoch. Weights are updated more often with more epochs, improving convergence. Therefore, there is better loss function minimization and images are denoised properly. Zhao et al. [27] trained the skip-connected network for 300 epochs whereas other research can be found on image enhancement with models trained for up to 4000 epochs [32] to reach convergence. We trained our model on our original dataset for 210 epochs due to limited computational resources.

D. Result Analysis and Comparison

Engineering uses the Peak Signal-to-Noise Ratio (PSNR) to measure the ratio between a signal's maximum possible intensity and intervening noise that affects its representation. It is often expressed using the logarithmic decibel scale due to various sources' high dynamic range. We chose PSNR as our performance indicator since it is extensively used to evaluate lossy-compressed images. The PSNR, measured in decibel (dB), can be defined by the following equations (2), (3), and (4):

$$PSNR \; = \; 10 \; \cdot \; log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{2}$$

$$PSNR \; = \; 20 \; \cdot \; log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{3}$$

$$PSNR \; = \; 20 \; \cdot \; log_{10}\left(MAX_I\right) \; - \; 10 \; \cdot \log_{10}\left(MSE\right) \tag{4}$$

where MAX_I is the maximum possible pixel value of input image. Here, MAX_I is calculated by the equation below:

$$MAX_I \; = \; 2^B - 1 \tag{5}$$

where B represents bits per sample in linear PCM. Also, PSNR is defined using the mean squared error (MSE), which is computed by the equation:

$$MSE \; = \; \frac{1}{m\ n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{6}$$

where I is the given m x n monochrome image and K is its noisy approximation.

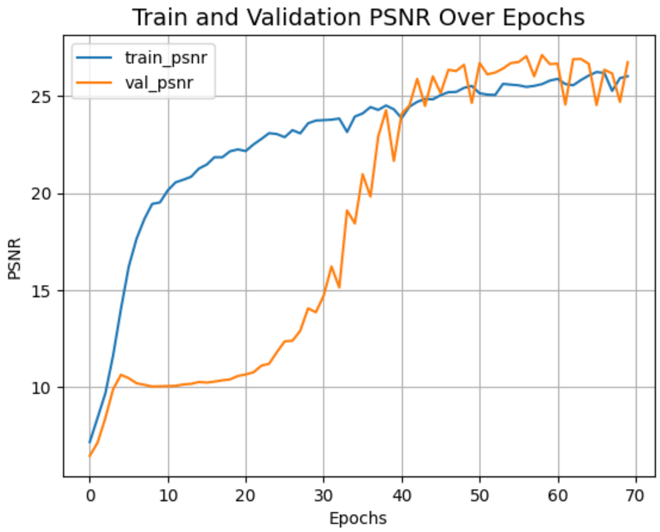
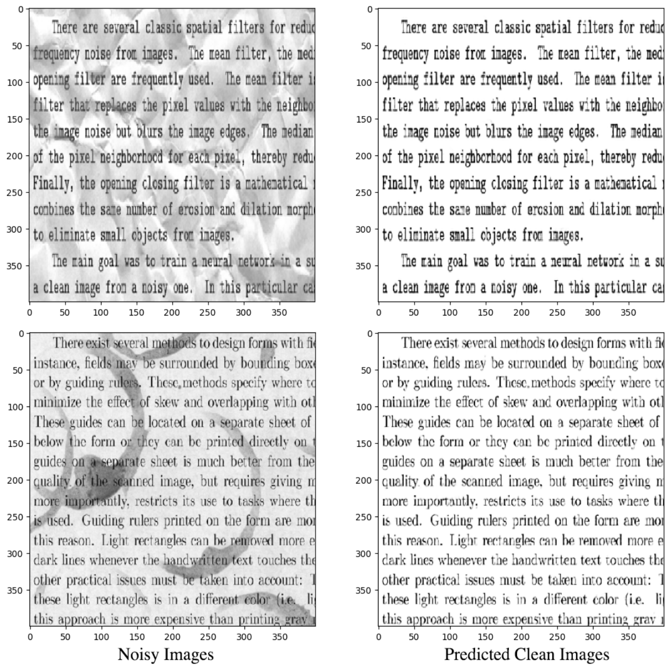


Fig. 6. Training vs Validation PSNR Curve

Our model is trained on the NoisyOffice dataset and achieved a training PSNR of 26.0266 and a validation PSNR of 26.9108. Furthermore, it achieved a training loss of 0.0968 and a validation loss of 0.0834. As we can see in Fig. 6, the performance metric PSNR starts to become more stable after 40 epochs.

1) Experiment on NoisyOffice Dataset: After running our model for 70 epochs on this dataset, we observe that the obtained denoised images are distinct and effectively remove noise from their respective noisy counterparts. The resulting images exhibit a lack of smudges or wrinkles, and the text displayed on them is consistently accurate. Fig. 7 displays these results where every letter is uniformly filled with black points, while the white gaps in the background are clearly discernible. Additionally, the shape of the letters is almost entirely correct, with negligible outliers. Another model, known as SCDCA [27], conducted a different test exclusively utilizing clean images from the NoisyOffice dataset. The images were subjected to additive Gaussian noise with a mean of zero and a standard deviation of 50. The resulting PSNR achieved by SCDCA was 26.57. In comparison, our model achieved a PSNR of 26.9108, which falls within the competitive range for this dataset and can be considered as standard. Additionally, with a lesser number of epochs, we are able to train our model with higher patch sizes, specifically 400 by 400, as opposed to the previous size of 40 by 40 in [27].

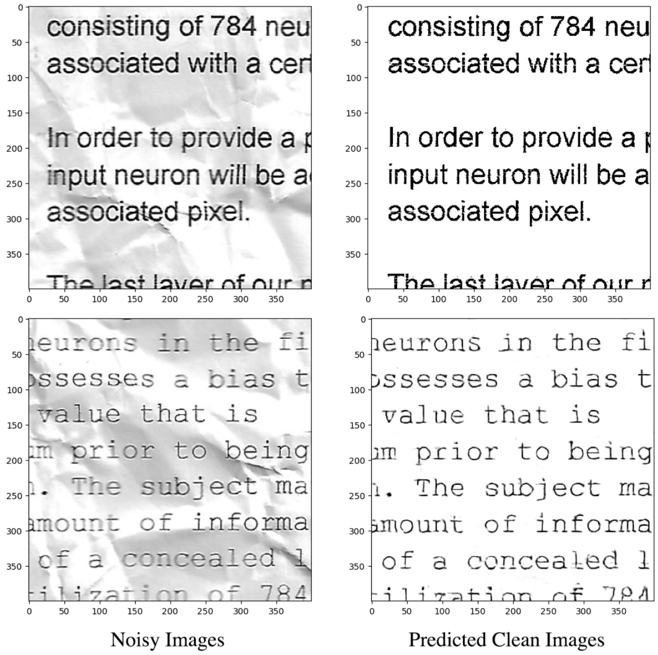

There are several classic spatial filters for reduc
frequency noise from images. The mean filter, the medi
opening filter are frequently used. The mean filter i
filter that replaces the pixel values with the neighbo
the image noise but blurs the image edges. The median
of the pixel neighborhood for each pixel, thereby redu
Finally, the opening closing filter is a mathematical
combines the same number of erosion and dilation morph
to eliminate small objects from images.
The main goal was to train a neural network in a su
a clean image from a noisy one. In this particular ca

There are several classic spatial filters for reduc
frequency noise from images. The mean filter, the medi
opening filter are frequently used. The mean filter i
filter that replaces the pixel values with the neighbo
the image noise but blurs the image edges. The median
of the pixel neighborhood for each pixel, thereby redu
Finally, the opening closing filter is a mathematical
combines the same number of erosion and dilation morph
to eliminate small objects from images.
The main goal was to train a neural network in a su
a clean image from a noisy one. In this particular ca

There exist several methods to design forms with fi
instance, fields may be surrounded by bounding boxe
or by guiding rulers. These methods specify where to
minimize the effect of skew and overlapping with ot
These guides can be located on a separate sheet of
below the form or they can be printed directly on t
guides on a separate sheet is much better from the
quality of the scanned image, but requires giving m
more importantly, restricts its use to tasks where th
is used. Guiding rulers printed on the form are mor
this reason. Light rectangles can be removed more e
dark lines whenever the handwritten text touches the
other practical issues must be taken into account: T
these light rectangles is in a different color (i.e. li
this approach is more expensive than printing gray r

There exist several methods to design forms with fi
instance, fields may be surrounded by bounding boxe
or by guiding rulers. These methods specify where to
minimize the effect of skew and overlapping with ot
These guides can be located on a separate sheet of
below the form or they can be printed directly on t
guides on a separate sheet is much better from the
quality of the scanned image, but requires giving m
more importantly, restricts its use to tasks where th
is used. Guiding rulers printed on the form are mor
this reason. Light rectangles can be removed more e
dark lines whenever the handwritten text touches the
other practical issues must be taken into account: T
these light rectangles is in a different color (i.e. li
this approach is more expensive than printing gray r

Fig. 7. Noisy vs Predicted Clean Images


consisting of 784 neu
associated with a cer
In order to provide a p
input neuron will be a
associated pixel.
The last layer of our n

consisting of 784 neu
associated with a cer
In order to provide a p
input neuron will be a
associated pixel.
The last layer of our n

neurons in the fi
ssesses a bias t
value that is
m prior to being
. The subject ma
amount of informa
of a concealed l
tilization of 784

neurons in the fi
ssesses a bias t
value that is
m prior to being
. The subject ma
amount of informa
of a concealed l
tilization of 784

Fig. 8. Noisy vs Predicted Clean Images from Original Dataset

2) Experiment on Original Dataset: Lastly, we run the proposed model on an originally produced dataset. Our model is optimized for the NoisyOffice dataset but still produces good results for this new dataset. It is important to note that these outcomes are achieved by the model by employing a mere 1.2 million (1,223,745) parameters, which is a very low number compared to the other research conducted in this field. In Fig. 8, the resulting denoised images of the experiment are shown. As we can see, the images are easy to understand with only a few trivial issues.

V. Conclusion and Future Work

In this study, we have put forth a convolutional autoencoder model that has been specifically optimized for the denoising of document images within the NoisyOffice dataset [1]. The suggested model incorporates skip connections between residual blocks as a means to address the issue of the vanishing gradient problem. It is important to recognize that our model demonstrates a decreased parameter size, enabling its execution on minimal processing power, while yet upholding its performance metrics. Moreover, larger dimension input sizes are utilized for training the document photographs, making them more practical in nature. Subsequently, an original dataset is generated in order to evaluate the effectiveness of our model in terms of its ability to adapt. Now, let us discuss the future work that we intend to undertake on this paper in order to further expand the scope of the study. Initially, our intention is to enhance the optimization of our suggested model specifically for the original dataset that we have developed, as it is currently only optimized for the NoisyOffice dataset [1]. Furthermore, it is necessary to perform a comparative study in the future by using our original dataset on additional state-of-the-art models.

References

[1] C.-B. M. Espaa-Boquera S., Pastor-Pellicer J. and Z.-M. F., "NoisyOffice," UCI Machine Learning Repository, 2015, DOI: https://doi.org/10.24432/C5G31N.

[2] J. Banerjee, A. M. Namboodiri, and C. Jawahar, "Contextual restoration of severely degraded document images," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 517–524.

[3] X. Chen, X. He, J. Yang, and Q. Wu, "An effective document image deblurring algorithm," in *CVPR 2011*, 2011, pp. 369–376.

[4] H. Cho, J. Wang, and S. Lee, "Text image deblurring using text-specific properties," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 524–537.

[5] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2901–2908.

[6] L. Xiao, J. Wang, W. Heidrich, and M. Hirsch, "Learning high-order filters for efficient blind deconvolution of document photographs," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 734–749.

[7] M. Hradis, J. Kotera, P. Zemcík, and F. Sroubek, "Convolutional neural networks for direct text deblurring," 09 2015.

[8] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang, "Learning to super-resolve blurry face and text images," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 251–260.

[9] X. Jiang, H. Yao, and S. Zhao, "Text image deblurring via two-tone prior," *Neurocomputing*, vol. 242, 02 2017.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[11] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2392–2399.

[12] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[13] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.

[14] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, dec 2016. [Online]. Available: https://doi.org/10.1145/2980179.2982399

[15] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "l_0 -regularized intensity and gradient prior for deblurring text images and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 342–355, 2017.

[16] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[17] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288–1300, 2016.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 184–199.

[19] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," 2016.

[20] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, dec 2010.

[22] L. Xu, J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1790–1798, 01 2014.

[23] D. Yang and J. Sun, "Bm3d-net: A convolutional neural network for transform-domain collaborative filtering," *IEEE Signal Processing Letters*, vol. 25, no. 1, pp. 55–59, 2018.

[24] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, 2016.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] G. Zhao, J. Liu, J. Jiang, H. Guan, and J.-R. Wen, "Skip-connected deep convolutional autoencoder for restoration of document images," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2935–2940.

[28] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1405–1414.

[29] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[30] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," 2017.

[31] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2790–2798.

[32] C. Zhang, Q. Yan, Y. zhu, X. Li, J. Sun, and Y. Zhang, "Attention-based network for low-light image enhancement," 2020.