Copyright ⓒ 2025 KSII

Video Frame Denoising via CNN and GAN Methods

Ahmet Yapici[1*], and M. Ali Akcayol[2]
[1] Information Systems, Gazi University
Ankara, 06680, Turkey
[e-mail: ahmet.yapici@gazi.edu.tr]
[2] Computer Engineering Department, Gazi University
Ankara, 06560, Turkey
[e-mail: akcayol@gazi.edu.tr]
[*]Corresponding author: Ahmet Yapici

Received May 30, 2023; revised October 17, 2023; revised June 27, 2024; accepted February 24, 2025; published March 31, 2025

Abstract

Video and image denoising techniques aim to eliminate noise while preserving image details. However, in the process of noise reduction, while some image texture may get lost, residual noise artifacts can persist. So far, no single model has achieved universal success across all types and levels of noise. In this study, we present a noise reduction model that combines deep learning methods using limited hardware resources. Specifically, we leverage a combination of Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN) architecture, which has demonstrated effectiveness in preserving image structure as the depth of the model increases. Additionally, we enhance visual quality by incorporating GAN model after the CNN network. Our evaluation of the proposed approach reveals superior performance at low noise levels compared to previous neural network-based methods. By utilizing the findings of this study, we gain insights into the more efficient utilization of deep learning and traditional noise reduction methods, explore their behavior at different noise levels, and compare results obtained from diverse datasets. The proposed method outperforms other CNN-based methods at certain noise levels, thereby providing valuable prior knowledge to researchers in this field.

Keywords: Convolutional neural network, deep neural network, image denoising, video denoising

1. Introduction

Image degradation is one of the main problems encountered in computer vision applications. Due to the environmental conditions, transmission channel and environmental factors, images encounter noise. The presence of noise can alter the relationship between adjacent pixels in an image, and cause difficulties for denoising algorithms to determine which pixels are part of the true image and which are noise.

The distribution of noise on the image varies based the effect on the image. Some noise components show non-uniform spatial characteristics. Noise in images is generally expressed as additive white gaussian noise (AWGN), expressed by the following equation. The observed value $v(i)$ and the noise-free value $\mathrm{u}(i)$ and the noise is shown by $\mathrm{n}(i)$.

$$v(i) = \mathrm{u}(i) + \mathrm{n}(i) \tag{1}$$

Image denoising requires extensive attention, whereas video-specific algorithms are relatively scarce. Video denoising relies on the fundamental assumption of strong inter-frame correlations in video sequences. A commonly used, video denoising method involves temporally averaging consecutive frames, which delivers impressive results for static scenes but introduces motion blur and artifacts. There are different approaches used over the years for image and video denoising problems. Traditional methods mainly utilize both spatial-domain methods and transform-domain methods. Even though the most traditional methods obtain a good performance in noise reduction in images, they have been recently replaced by deep learning methods since they have manually adjustable parameters and require the best improvement methods during training. Artificial neural networks have become popular because of their advances in the GPU field and their performance compared to other methods. During our studies we realized deep learning methods are not flexible enough to different noise levels and noise types and they are not sufficient in terms of performance and success when dealing with high noise levels [1-5]. Also, deep learning methods use much more memory and GPU during training period [6]. In our study, we aim to achieve successful results using less training time and resources during denoising the images. Therefore, in order to eliminate the negative effects of neural networks method we introduced an automated denoising approach using CNN and GAN, enabling flexible and efficient video denoising capable of effectively eliminating noise without prior knowledge of the noise. By training the model in two stages, we reduced the training time and number of parameters and eliminated the noise with these two models.

There are methods such as mean square error (MSE), peak signal-to-noise ratio (PSNR), feature similarity index for image quality assessment (FSIM), and root mean squared error (RMSE) to measure image quality. Different methods are used to measure the image structure and texture similarity in different ways to evaluate the quality of the image after noise removal. Studies have shown that PSNR and SSIM methods are more sensitive to Gaussian noise [7]. During the noise removal phase, the Google Colab environment was utilized. In the Colab environment, there are certain limits on GPU, RAM, disk, and their usage durations. Memory constraints restrain the processing of large-scale datasets or the training of deep neural networks. Limited GPU resources or running within a restricted time window necessitates completing operations within a specific timeframe. The time constraint also requires keeping training durations within certain limits. Aware of these limitations, two denoising models were created using less training data and small image sizes, consuming fewer resources. In this architecture, we combined spatial and temporal filtering. For the spatial denoising part, we

selected CNN method and for the temporal part we selected GAN method. It is aimed to create a model that can denoise images and video frames by using limited resources. The contributions of the work are listed:

1. Hybrid Model Development:
    a. A hybrid model was developed using CNN and GAN methods to perform video noise removal. The architecture combines spatial and temporal filtering to enhance denoising performance.
    b. The developed model splits the denoising process into two separate stages, enhancing the denoising effectiveness. The CNN method was selected for spatial denoising, and the GAN method was chosen for temporal denoising. The first stage involves denoising the first frame using a CNN model. The second stage involves denoising neighboring frames using a GAN model.
    c. This approach replaces the traditional motion estimation algorithm with a more efficient two-stage process.

2. Resource Efficiency and Adaptation to Computational Constraints:
    a. The proposed model is designed to use less training time and resources, making it suitable for environments with limited computational resources.
    b. Two denoising models were created using less training data and smaller image sizes, optimizing resource consumption. By implementing a two-stage training process, the model minimizes training time and parameter count while effectively eliminating noise.
    c. The proposed model was developed considering the constraints of the Colab environment, such as limited GPU, RAM, disk space, and usage durations. The model's design ensures it can operate within the constraints of environments with restricted GPU and memory resources.

3. Automated Denoising Approach:
    a. An automated denoising approach using CNN and GAN was introduced to handle noise removal without prior knowledge of the noise characteristics.
    b. This approach enables flexible and efficient video denoising.

By addressing these key aspects, the study contributes an innovative and efficient approach to video denoising, leveraging the strengths of both CNN and GAN methods while optimizing for computational efficiency and resource limitations.

2. Related Works

In the process of image denoising, various traditional noise removal methods are employed. These methods include incorporating information about the statistics of natural images, modeling the smoothness of the image, or leveraging deep learning algorithms that have been trained on large datasets of noisy and denoised images.

2.1 Traditional Denoising Methods

Traditional methods used for image denoising can be divided into two groups: transform domain methods and spatial domain methods [8].

2.1.1 Spatial Domain Filtering

In spatial domain filtering, the image is divided into frames, and filtering is done separately on each frame. This method is applied to both images and video frames; similar frames are grouped, and image denoising is done on these frames. Spatial domain filtering uses linear filters, but image textures can be lost. In addition to Gaussian noise elimination, average filtering can be employed; however, it may excessively smooth out the image, particularly in cases of high-noise images [9]. To address this, Wiener filtering can be utilized, although it carries the drawback of potentially blurring the image's sharp edges. Alternatively, non-linear filters such as median filtering and weighted median filtering can effectively eliminate noise without necessitating prior knowledge of the noise level [10]. Bilateral filtering is widely used to prevent image degradation [11]. Non-linear partial differential equation-based edge preservation is one of the well-known examples of noise removal, designed to remove noise while preserving edges in images. Rudin–Osher–Fatemi (ROF) model, which is a denoising model on a general finite directed and connected graph, has been extended and modified in various ways to improve performance in other image restoration tasks [12]. These models are used in image restoration problems such as noise removal, de-blurring, blind noise removal, and image inpainting.

The non-local means (NLM) method can be used in image denoising and inpainting problems. NLM preserves image details, robustness to different types of noise, adaptability, ease of implementation, and suitability for large images make it a popular choice for image denoising applications [13]. Block matching and 3D filtering (BM3D) technique use block matching to group similar frames in static images and videos by filtering. As the noise level increases, the noise reduction performance of BM3D decreases, especially in flat areas, causing artifacts. VBM3D, and VBM4D are used to remove noise from videos using block-matching and 3-dimensional filtering. Non-local noise removal models are used in image restoration problems; noise removal, inpainting, addition, and image synthesis. These models represent the images with a series of local neighborhoods or patches and try to predict the noise. The maximum a posteriori (MAP) can be used to predict a clean patch from the noisy image [14].

In the sparse approach method, the sparse representation vector is used to find object classes. Support Vector Machines and Kernel-Based Sparse Representation are important methods that use this approach. The nonlocally centralized sparse representation (NCSR) model is one of the widely used image noise removal methods, and the non-local central sparse representation has been successful in reconstructing both smooth and textured regions [15]. However, the repetitive dictionary learning and unknown sparse coefficients of sparse approach techniques increase the calculation time and limit the usability of this method in applications. Low rank approaches for noise reduction are divided into two groups: low-rank matrix factorization-based methods and nuclear norm minimization-based methods (NNM). Low-rank matrix factorization is effective in noise reduction [16], but it can cause loss of details at very low or very high noise values. K-means clustering singular value decomposition (K-SVD), is the generalized version of the k-means clustering method, which sparsely encodes input data based on a valid dictionary and updates the dictionary atoms to better fit the data iteratively. The weighted nuclear norm minimization (WNNM) has advanced noise reduction performance and its success at high noise levels has enabled its widespread use in fields such as artificial intelligence, image processing, pattern recognition, and computer vision [17].

2.1.2 Transform Domain Filtering

Originally, field transformation techniques were devised utilizing the Fourier transform. The

functions of independent component analysis and principal component analysis have proven effective in reducing noise for data sets that do not follow a normal distribution [18, 19]. Wavelet field methods in image noise reduction can achieve successful results with simple algorithms. Developed methods in this field are analyzed in 4 categories. The first method is linear filters, which are numerical filters used for noise reduction. The Wiener filter, a type of adaptive linear filter, is preferred for reducing Gaussian noise [20]. The second method is Wavelet Transform, which is a nonlinear coefficient threshold-based method and is used to remove additional Gaussian noise [21]. In the third method, the wavelet coefficient approach, the multi-resolution properties of the Wavelet Transform are used. This method is successful in removing noise, but it is computationally complex [22]. The fourth method, undecimated wavelet transform (UDWT) has also been used to separate the noise signal. This method is computationally efficient for noise reduction, but it can also lead to over-smoothing of the image [23].

2.2 Deep Learning Methods Used in Image Denoising

The utilization of artificial neural networks is favored for the purpose of easy use, efficient resource allocation, and capability to adapt to diverse noise types [24]. Convolutional neural network (CNN) based noise reduction methods generally obtain better results than other methods [25, 26]. CNN-based methods such as denoising convolutional neural network (DnCNN) [27] obtain noise-free images by training a deep neural network using clean and noisy data pairs to remove white Gaussian noise. With the residual learning strategy, DnCNN implicitly removes the latent clean image in the hidden layers. In unsupervised learning methods, noise is directly learned from data without the need for clean image data or noisy image data for training. Noise2Noise (N2N) applies models learned from matched corrupted observations to remove noise, Noise2Self (N2S) and Noise2Void (N2V) adopt a self-supervised strategy for blind noise removal and training a regression model [28]. In a particular study, both wavelet transform and CNN architecture were utilized to achieve successful outcomes [29]. In a study, a convolutional neural network was employed, incorporating principal component analysis, to address the task of reducing noise in images [30]. GAN architectures are also used to improve noise reduction performance by using a maximum a posteriori method-based adversarial network, to predict noise and for functions such as image inpainting and super-resolution [31]. The BRDNet network aims to combine two networks to increase the width and obtain more features [32]. The network uses expanded convolutions to remove noise from images and obtain more features. FFDNet is a noise reduction method used on images with unknown noise levels [2]. In SCUNet [33] model the data obtained from the convolutional layer is passed through the filter using the U-Net network, and similar patches are grouped into blocks using sliding windows. In a study, primarily the non-local patch method is used, and then CNN methods are used to remove noise on the video and image [34]. U-Net differs from traditional autoencoders in that it also includes skip connections that allow it to preserve high-level features from the input image during the encoding and decoding process. The MWCNN model integrates wavelet transforms and convolutional layers within a U-Net architecture [35]. Efficient sub-pixel convolutional neural network (ESPCN) is used to obtain high-resolution images from low-resolution ones [36]. The distinctive aspect of ESPCN is the use of sub-pixel convolution layers, instead of directly upscaling the low-resolution image using traditional interpolation techniques, ESPCN employs convolutional layers to increase the resolution. Enhanced convolutional neural denoising network (ECDNet) method is based on residual learning, Batch normalization (BN) and dilated convolution [37]. Deep video denoising network (DVDNet) combines CNN-based denoising method with motion

estimation technique [3]. In this method similar images from different frames in the video are grouped and denoised, eliminating flickering between image transitions. Fast deep video denoising network (FastDVDNet) is an enhanced version of the DVDNet method, this approach aims to perform denoising more quickly by combining spatial and temporal filtering [38]. First image than video network (FITVNet) model utilizes a two-stage CNN architecture to obtain clean video frames. Noise maps are used in this model, which indicates the level of noise in different regions of an image or video [39]. Task-oriented flow (ToFlow) is a method aimed at denoising noisy videos using the optical flow [40]. ViDeNN is a denoising model that uses convolutional neural networks without prior knowledge of noise distribution [41].

3. The Developed Model

The prepared method is inspired by these CNN, U-Net and GAN architectures. In the first stage of the prepared model, data augmentation is performed to increase the amount of training data. Noisy data is cleaned with the deep learning methods, in the first denoising phase CNN model is used to denoise each video frame, in the second phase the neighboring frames are cleaned with the GAN model. **Fig. 1** shows the main flow of the noise reduction model.

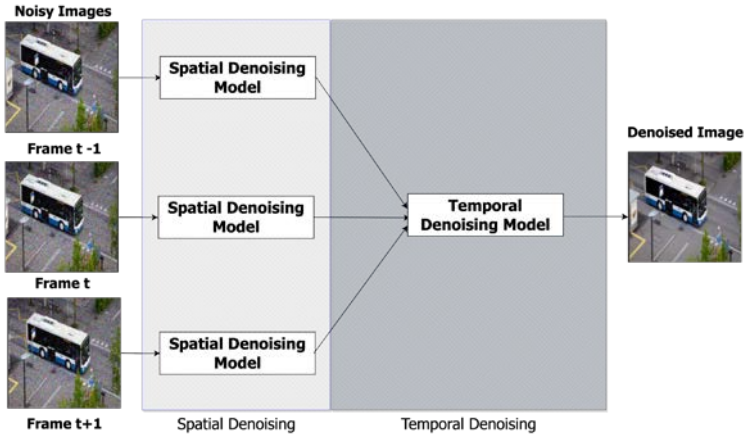

Fig. 1. Denoising model design

3.1 Spatial Model

The spatial model uses CNN and U-Net architectures and consists of 27 layers. When the noise level σ = 25, optimal parameter steps are selected using the DAVIS dataset with 180x180 image sizes, and the denoising results are shown in **Table 1**. The deep learning components considered during the preparation of the model are mentioned below.

- **Data Augmentation**: Studies on noise reduction show that working with augmented data sets can affect the performance of the model. Data augmentation is done by using methods such as rotation, translation, scaling, and cropping on image data. When the training data is increased, it is seen that the spatial model performs better on the increased data, and the loss value decreases. The noise reduction values obtained during comparison of data augmentation are shown in **Table 1**, in step 1, 2 and 28.
- **Increasing the receptive field**: The aim is to capture more contextual information by increasing the receptive field. To expand the receptive field, one approach is to augment the depth and width of the networks. However, this strategy comes with a trade-off: it

leads to higher computation costs and increased memory consumption. To solve this problem, dilated convolution is used, to provide more edge information. Using more convolution layers can cause the model to over-fit. To prevent this problem a double dilated convolution layer is added and comparison results of adding dilated convolution layers are shown in **Table 1**, step 2, 3, 4.

- **Loss Function**: The loss function measures how effectively the model is working so that the weights can be updated correctly. In the prepared model, L2 loss function is used, which minimizes the sum of squares of the differences between the predicted and actual values and produces more accurate results in matrix operations with different values. The values obtained from the application of the loss curves to the model are shown in **Table 1**, step 2, 5, 6, 7, 33. Loss of functions greatly affects the behavior of denoising neural networks. Finding a well-suited loss function for human visual system is also a crucial research topic in deep learning.
- **Optimization Function**: Gradient descent algorithms specify how many samples from the training data to use at each step to calculate the derivative of the loss for each parameter in the model. ADAM optimization function is used in the model, which can adjust itself according to error values. The values obtained by applying the optimization functions to noisy images are shown in **Table 1**, step 2, 8, 9, 10, 11.
- **Optimizer (Momentum)**: The optimizer is used for faster convergence of the loss function and helps to accelerate the model with less slope. When the momentum value is selected as 0.9 in the model, more successful results are obtained. The values obtained by applying the optimization value to the model are shown in **Table 1**, step 2, 12, 13, 27, 29, 34, 35.
- **Activation Function**: Introduce non-linearity to the model, which allows the neural network to learn more complex functions and representations. It allows the model to perform complex decision-making by combining multiple features of the input data. Denoising results of applying activation functions to the model are shown in **Table 1**, step 2, 14, 15, 16, 17, 18, 19, 30, 31, 32. The mish function obtains the best results because it is a non-linear function, it can capture non-linear relationships between inputs and outputs.
- **Initial Weight Values**: Initialization of the weights affect the learning rate of the model. Starting all weights with zero causes neurons to learn the same features during training, resulting in the cost oscillating around the minimum value. Using very large weights can cause an exploding gradient problem when used in backpropagation. Normalized Xavier uses a uniform probability distribution to calculate the initial values. The values obtained by applying weight functions to the model are shown in **Table 1**, step 2, 20, 21, 22, 30, 31, 32.
- **Batch Size**: Using larger batch sizes can lead to faster training but also increase memory usage; smaller batch sizes can lead to slower training but may result in models that generalize better. The batch size must fit within the GPU memory. Depending on the image size, the batch size can be chosen larger. The loss values obtained by using different batch sizes on the model are shown in **Table 1**, step 2, 23, 24. With small batch numbers for this size of images, a smaller batch number yields slightly better results.
- **Building Deep Neural Networks**: A deeper denoising model has been developed to preserve the texture of the image and prevent the blurring of boundary values. U-Net employs skip connections that directly connect the encoder (downsampling) and decoder (upsampling) paths. These connections help in preserving fine-grained details during the upsampling process. By integrating CNNs into the encoder part of the U-Net, we can leverage their feature extraction capabilities to obtain high-level representations of the

input image. U-Net's architecture helps in reconstructing the denoised image while preserving image details, using skip connections. Deeper noise removal networks require more than normal memory resources and cause overfitting and vanishing gradient problems. To overcome the vanishing gradient problem, models like U-Net architectures can be used because it performs the convolution operation. FastDVDNet and SCUNet are also using CNN and U-Net architecture to effectively denoise image patches and to preserve fine details. The test results for determining the depth of the model are shown in **Table 1**, step 2, 25, 26.

Table 1. The parameter selection spatial model with noise level σ = 25

Step	Training Set Size	Dilated Convolution Layer Count	Loss Function	Optimization Method	Momentum	Activation Function	Weight Initialization Function	Batch size	Layer Count	PSNR (dB)	SSIM
1	400	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	26.76	0.85
2	480	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	27.63	0.88
3	480	0	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	27.05	0.87
4	480	1	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	27.50	0.88
5	480	2	Huber	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	27.61	0.88
6	480	2	L1	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	20.88	0.68
7	480	2	MSLE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	26.62	0.87
8	480	2	MSE	RMSProb	0.9	Sigmoid	Normalized Xavier	16	27	22.56	0.70
9	480	2	MSE	AdaGrad	0.9	Sigmoid	Normalized Xavier	16	27	17.30	0.45
10	480	2	MSE	AdaDelta	0.9	Sigmoid	Normalized Xavier	16	27	9.52	0.18
11	480	2	MSE	SGB	0.9	Sigmoid	Normalized Xavier	16	27	17.10	0.41
12	480	2	MSE	ADAM	0.0	Sigmoid	Normalized Xavier	16	27	24.55	0.72
13	480	2	MSE	ADAM	0.1	Sigmoid	Normalized Xavier	16	27	25.18	0.78
14	480	2	MSE	ADAM	0.9	tanh	Normalized Xavier	16	27	24.28	0.78
15	480	2	MSE	ADAM	0.9	Relu	Normalized Xavier	16	27	26.97	0.86
16	480	2	MSE	ADAM	0.9	Softmax	Normalized Xavier	16	27	22.76	0.67
17	480	2	MSE	ADAM	0.9	Swish	Normalized Xavier	16	27	27.28	0.87
18	480	2	MSE	ADAM	0.9	elu	Normalized Xavier	16	27	24.01	0.77
19	480	2	MSE	ADAM	0.9	LeakyRelu	Normalized Xavier	16	27	24.08	0.77
20	480	2	MSE	ADAM	0.9	Sigmoid	He	16	27	27.32	0.87
21	480	2	MSE	ADAM	0.9	Sigmoid	Xavier	16	27	27.48	0.87
22	480	2	MSE	ADAM	0.9	Sigmoid	Orthogonal	16	27	27.01	0.87
23	480	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	32	27	27.53	0.88
24	480	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	64	27	27.56	0.88
25	480	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	26.56	0.86
26	480	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	25	26.64	0.87
27	480	2	MSE	ADAM	0.1	Sigmoid	Normalized Xavier	16	29	26.48	0.86
28	540	2	MSE	ADAM	0.9	Sigmoid	Normalized Xavier	16	27	26.63	0.85
29	480	2	MSE	ADAM	0.9	tanh	He	16	27	27.12	0.85
30	480	2	MSE	ADAM	0.9	Sigmoid	He	16	27	27.32	0.87
31	480	2	MSE	ADAM	0.9	tanh	Xavier	16	27	27.30	0.86
32	480	2	MSE	ADAM	0.9	Relu	Xavier	16	27	27.32	0.86

33	480	2	Huber	RMSProb	0.9	Sigmoid	Normalized Xavier	16	27	25.05	0.80
34	480	2	MSE	ADAM	0.5	Sigmoid	Normalized Xavier	16	27	26.15	0.81
35	480	2	MSE	ADAM	0.8	Sigmoid	Normalized Xavier	16	27	26.87	0.81
36	**480**	**2**	**MSE**	**ADAM**	**0.9**	**Mish**	**Normalized Xavier**	**16**	**27**	**27.65**	**0.88**
37	540	2	MSE	ADAM	0.9	Mish	Normalized Xavier	16	27	26.63	0.85
38	480	0	MSE	ADAM	0.9	Mish	Normalized Xavier	16	27	27.05	0.87
39	480	2	MSE	Huber	0.9	Mish	Normalized Xavier	16	27	27.61	0.88
40	480	2	MSE	ADAM	0.9	Mish	He	16	27	27.12	0.85

The network depth is determined as 19 in CNN layers and 8 in U-Net layers to capture enough spatial information and shorten the noise reduction time. The loss function is used to learn mapping and the weights are initialized using the Normalized Xavier method. The model uses a momentum of 0.9. The prepared model is trained with 50 repetitions. In the CNN part of the model, the Mish activation function is used. In deep networks, single-fold functions are preferred to reduce loss values, and the Mish activation function is preferred in the U-Net part. In the encoding and decoding part of the model, 4x4 filters are used. In the encoder portion of the architecture, each block comprises a series of operations, including Convolution, Batch Normalization, and Rectified Linear Unit activation. Conversely, in the decoder part, each block performs Transposed Convolution, Batch Normalization, Dropout, and Mish activation operations. Skip connections are present between the encoding and decoding layers. During tests, it was observed that as the number of layers in the network increases, image denoising performance is decreasing, so we decided to use skip connections. The first 4 layers of the model uses U-Net architecture, 2 layers are used for image dimension reduction, and in the second half of the model, the dimension is increased. The U-Net architecture is used at 1-4 and 24-27 layers of the model. The model gets noisy image as an input and returns a noise tensor as an output. Subtract operation takes a list of tensors as input, both of the same shape, and returns denoised image. The subtract operation is applied to input image and noise tensor. Spatial model layers are shown in **Fig. 2**.

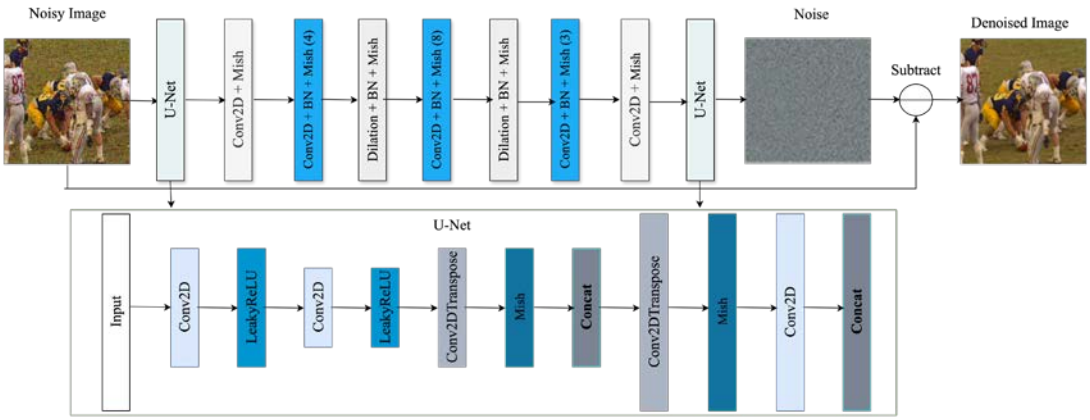

Fig. 2. Spatial model

The architectural design of this neural network is a synthesis of encoding, decoding, convolutional, and dilation layers. **Table 2** shows the layers used in the spatial denoising model.

Table 2. The spatial model layers

Layer Number	Layers
1-2 (Encode)	Conv2D + Mish (2)
3 (Decode)	Conv2DTranspose + Mish + Concat (Layer 3)
4 (Decode)	Conv2DTranspose + Mish + Conv2D + Concat (input)
5 (CNN)	Conv2D + Mish (1)
6 .. 9	Conv2D + BN + Mish (4)
10	Dilation + BN + Mish (1)
11 .. 18	Conv2D + BN + Mish (8)
19	Dilation + BN + Mish (1)
20 .. 22	Conv2D + BN + Mish (3)
23 (CNN)	Conv2D + Mish (1)
24-25 (Encode)	Conv2D + Mish (2)
26 (Decode)	PixelShuffle + Mish + Concat (Layer 24)
27 (Decode)	PixelShuffle + Mish + Conv2D + Concat (Layer 23) + Conv2D + Subtract

Encoding layers (1-2 and 24-25) are composed of convolution (Conv2D) layers and activation functions (Mish). These layers are primarily tasked with feature extraction and spatial dimension reduction in the input data. The decoding layers (3, 4, 26, 27), are responsible for expanding the spatial dimensions of the data. Transposed convolution (Conv2DTranspose) layers, purposefully chosen for their capacity to perform up sampling. Mish function is applied, to produce values close to the input near zero, effectively preserving small or negative values. These layers employ concatenation operations to amalgamate features extracted from earlier encoding layers with those generated in the decoding phase. CNN layers (5 and 23) are used for feature extraction and spatial dimension reduction. Convolutional layers with batch normalization and mish function (6-9, 11-18, 20-22) are used for feature extraction and nonlinear transformations. Dilation layers (10 and 19) are used to enable convolutional filters to have larger receptive fields without necessitating an increase in the number of parameters.

3.2 Temporal Model

According to our practice observations and further studies, we decided to GAN model in the temporal model. Using GAN model, we can compare sequential image frames using pure deep neural networks. GAN method can capture complex noise patterns, generate visually pleasing results and GAN method prevents video frame flickering during image transition. After denoising a frame using a deep learning method, we applied the temporal model to the output of the spatial model. While spatial denoising model takes a single frame, temporal model aims to reduce noise by grouping neighboring frames of the video. Temporal model provides good denoising results by preserving the edges and fine details in the image. To reduce noise in images and decrease the training time of the model and use fewer hardware resources in high-resolution images. To shorten the denoising time and achieve successful results at high noise levels, a GAN model is preferred in the spatial denoising stage. The Pix2Pix model [42] is used in generator network. The U-Net model is used to prevent overfitting by skip connections. By inspiring the VGGNet model architecture from the Deep Convolutional Generative Adversarial Network model, only convolution and activation layers are used. In the generator part of the network, sequential video frames are concatenated to build a block, and noise reduction is applied to this block. In this work, three consecutive frames are grouped to build a single tensor. Block denoising preserves image features and improves overall image quality, while preventing flickering in image transitions. The network depth was maintained at a

certain level to prevent model from overfitting. The generator network is shown in **Fig. 3**.

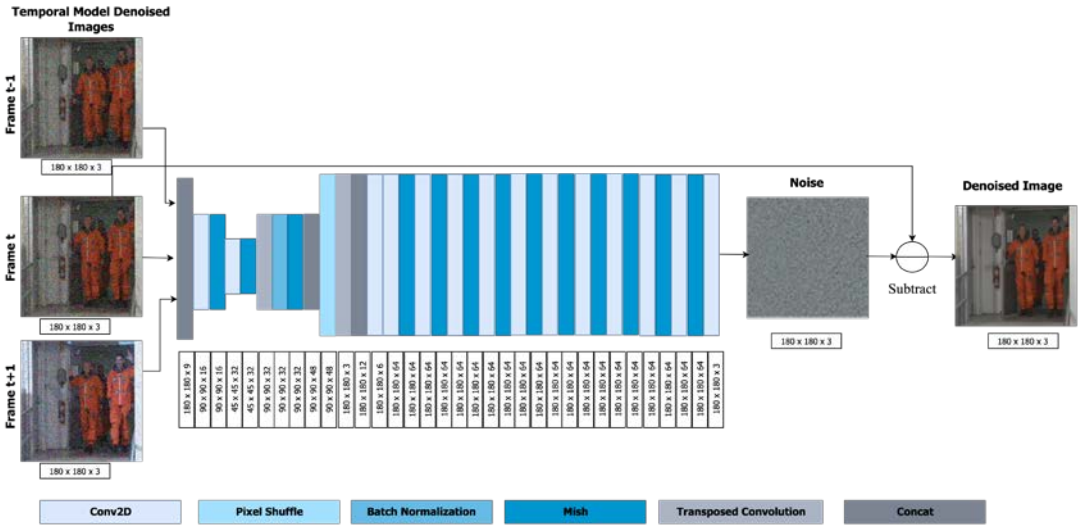

Fig. 3. Layers of the generator network

In the U-net model pixel shuffling is used to increase the spatial resolution of the image. Using the DAVIS dataset with dimensions of 180x180 images, it is observed that the pixel shuffling achieves successful results according to transposed convolution. Pixel shuffling operation is shown in **Fig. 4**.

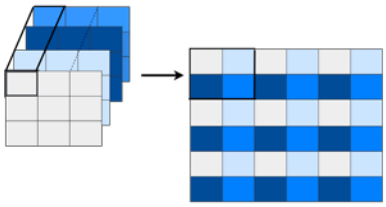

Fig. 4. Pixel shuffling operation

The comparison results of 2D transposed convolution and pixel shuffle are shown in **Table 3**.

Table 3. Pixel shuffle and 2D transposed convolution usage results in the generator model

Pixel Shuffle	PSNR (dB)	SSIM	FSIM	RMSE
-	26.98	0.88	0.98	0.00001545
+	27.99	0.91	0.98	0.00000976

Pixel shuffling is used to upscale images in a way that enhances its resolution while preserving important details. It involves rearranging and redistributing pixel values from a lower-resolution grid to a higher-resolution grid. Transposed convolution and pixel shuffling are both techniques used in deep learning for tasks like image super-resolution, image generation, and image-to-image translation. In this model pixel shuffle method gets better results in PSNR, SSIM and RMSE. FSIM value did not change both in pixel shuffle and transposed convolution approach. FSIM takes into account various visual features, including luminance, contrast, and structural information, to determine how similar the images are in

terms of their structural content. Pixel shuffling brings some improvements in the image during image denoising but this method also leaves noise residues in some areas of the images. We are using LeakyReLU in the generator model. Primary motivation for using LeakyReLU is to address the vanishing gradient problem. Activation functions like sigmoid and hyperbolic tangent (tanh) squish their input into a very narrow range. This causes gradients to become extremely small during backpropagation, which can lead to slow or ineffective training. ReLU based functions do not saturate in a large range of values.

Discriminator network is build using PatchGAN system. PatchGAN is used to classify if each patch in an image is similar to original one. The discriminative part of the model takes the generated image and the real image as inputs and these two input vectors are concatenated to obtain a single vector. **Table 4** shows the layers of the discriminator network.

Table 4. Layers of the discriminator network

Layer Number	Layers
1	Conv2D + BatchNorm
2	Conv2D + BatchNorm + LeakyReLU
3	Conv2D + BatchNorm + LeakyReLU
4	Conv2D

The discriminative network, assess the quality of the denoised images generated by the generator network. This network gets generator model generated image and original image as inputs and returns a probability score that helps the generator network to produce more realistic, noise-free images during the training process. In the convolution layer, features of the images are extracted using 3x3 filters. Convolution and batch normalization layers are used to extract essential features from the input data. Batch normalization is applied after convolution layer to improve the stability and speed of training. Leaky rectified linear unit activation function is used to introduce non-linearity in the model and this function helps network to learn effectively. The layers in the discriminative model are illustrated in **Fig. 5**.

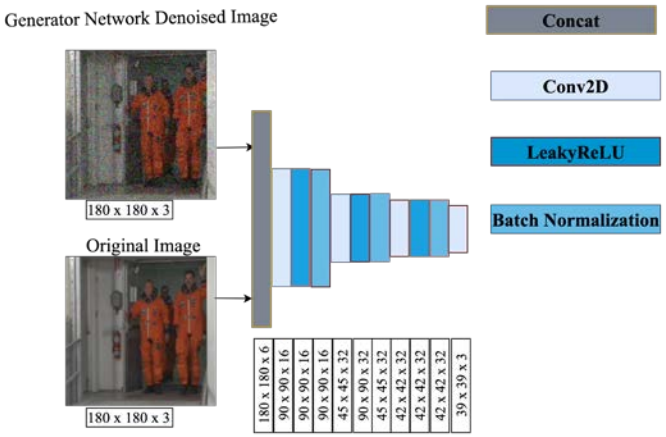

Fig. 5. Layer size of the discriminator network

As a result of the conducted tests, it was observed that there is a 1 dB increase in the PSNR value when the temporal denoising model is applied after the spatial denoising process.

4. Experimental Results

4.1 Datasets and Experimental Setup

We trained our proposed model using four datasets, Davis 2017 [43], CBSD400 [44], Derf and Vimeo [45]. From the Davis dataset, 480 images are used to train the model. The folder names and the number of images in the dataset are: Car-turn – 80, Mbike-trick - 79, Motorcross-bumps - 60, Sheep - 68, Soccerball - 48, Train - 80, Upside-down - 65. The Bus folder with 80 images is used as test dataset. Second dataset Vimeo consists of real-life images. From the Vimeo dataset 420 images are used during training phase. These images are in the 00001 and 00002 folders. In the Vimeo dataset, 70 images are used as test data. The images are in the 0004, 0007, 0008, 0012, 0016, 0019, 0022, 0025, 0029, 0031, 00003 folders. The third dataset CBSD400 includes 400 colorful images. CBSD68 dataset is used to test the model and contains 68 test data. The fourth database Derf contains video sequences in the uncompressed YUV4MPEG format. The city and foreman videos are used to train the model. The training data contains 600 frames. The crew folder is used as test dataset, that contains 300 frames. To test the denoising performance of the proposed methods PSNR, SSIM, FSIM, and RMSE quality metrics are used. We performed our experiments using Python 3 Google Compute Engine backend (GPU).

We compared the proposed model with DnCNN, CDAE, FastDVDNet, SCUNet and ViDeNN models. State-of-art models are trained using different datasets. Waterloo exploration database, contains 4744 pristine natural images and 94,880 distorted images. DIV2K dataset contains 1000 images with different scenes. Flick2K dataset contains 2650 RGB images of 2K resolution. SCUNet model used waterloo exploration database [46], DIV2K [47], and Flick2K [48] for training. FastDVDNet model uses the 2017 DAVIS dataset for training. Davis dataset contains a total of 150 videos, 60 for training, 30 for validation, 60 for testing. The frame sizes are 4219 for training, 2023 for validation and 2037 for testing totaling 10459 annotated frames. VideNN uses Waterloo Exploration Database for training.

We compared training data size and layer size of the state-of-art methods. Since different mathematical operations are performed within the layers of each model, the approximate model layer sizes and training data size are listed in the **Table 5**.

Table 5. Training data size and layer size of the methods

Method	Training Data Size	Model Layer size
Proposed	1900	27
CDAE	1900	14
DnCNN	1900	19
FastDVDNet	4219	38
SCUNet	8394	35
ViDeNN	4744	40

FastDVDNet, VideNN and SCUNet stand out in the obtained results. These models use deep neural networks and more training data to achieve better results. Both our proposed model and the other models (CDAE and DnCNN), are trained with 1900 images. FastDVDNet and ViDeNN use twice as much training data as CDAE, DnCNN and our proposed model. SCUNet model uses the most training data. Our proposed model, DnCNN and CDAE architectures consist of smaller layer sizes compared to FastDVDNet, SCUNet and ViDeNN architectures. According to the obtained results, FastDVDNet achieves better results than the others;

therefore, we can conclude that the network architecture and the training data size are critical for the successful denoising results.

In the preparation phase Gaussian noise is added to original images. Gauss distribution can be defined with three parameters, mean (μ), variance(σ^2) and standard deviation (σ). Gauss curve has a normal distribution which has 0 mean value and standard deviation value is 1 as shown in equation (2).

$$\sigma^2 = \frac{\sum_{i=1}^{N}(X_i-\mu)^2}{N} \tag{2}$$

The noise value expressed as σ and this value can be between [0-255]. The sigma value is proportioned to the maximum pixel value which is 255, and a normally distributed gauss noise value is added to the relevant pixels of the original image.

4.2 Experimental Results and Analysis

In this section, we present the denoising results of the proposed method and the state-of-art methods. The highest PSNR values for each noise level are highlighted in bold. First, we start by comparing traditional methods and deep learning methods. Using these results, we compare our proposed method with the methods which produce the best results. In a study, the traditional methods BM3D and WNNM are compared with the deep learning methods DnCNN, FFDNet and the proposed model. The results of noise reduction for deep learning methods and traditional noise reduction methods are shown in **Table 6**.

Table 6. PSNR results of denoising models using bsd68 test dataset [49]

Noise Level (dB)	Proposed	BM3D	WNNM	DnCNN	FFDNet
σ = 15	**31.74**	31.07	31.37	31.72	31.62
σ = 25	**29.27**	28.57	28.83	29.23	29.19
σ = 50	26.15	25.62	25.87	26.23	**26.30**
σ = 75	25.74	24.21	24.40	**26.64**	24.78

The comparison was conducted under different noise levels: 15 dB, 25 dB, 50 dB, and 75 dB. At lower noise levels of 15 dB and 25 dB, our developed model outperformed the traditional methods. The deep learning model demonstrated superior noise reduction capabilities, maintaining image quality and preserving fine details. PSNR metrics showed higher values compared to BM3D, WNNM, DnCNN, and FFDNet. At a higher noise level of 50 dB, FFDNet achieved better results. While our model still performed adequately, FFDNet's specialized architecture for handling high noise levels gave it an edge in maintaining image clarity. At an even higher noise level of 75 dB, DnCNN outperformed our model at this extreme noise level. The robustness of DnCNN in handling severe noise conditions contributed to its superior performance in reducing noise while preserving image integrity. The deep learning model's ability to learn intricate patterns and features enabled it to excel in lower noise environments. The performance superiority at these noise levels demonstrates the model's effectiveness in practical scenarios where noise levels are typically lower. At higher noise levels, FFDNet and DnCNN's architecture, which is specifically designed for diverse noise levels, provided better results.

BM3D method divides the noisy image into overlapping blocks. Instead of processing the entire image at once, this method operates on these smaller blocks, to allow for better local noise adaptation. By leveraging the concept of nuclear norm minimization with weighted terms, WNNM method can effectively reduce noise while preserving important image structures. According to PSNR values, the DnCNN method obtains more successful results, as

the noise level increases, the FFDNet method achieves better results. As seen from the table deep learning methods yields better results than traditional methods.

State-of-art models such as DnCNN and CDAE are trained with same datasets. All the models are trained using CBD400 training data and CBSD68 test data is used to compare the models. Test and train data dimensions are 176x144, and the models are tested by adding 25 dB AWGN. The training time, total parameters and denoising results of DnCNN, CDAE, and our model are displayed in **Table 7**.

Table 7. Denoising results of DNCNN, CDAE, and proposed model using CBSD68 test dataset with noise level of 25 dB

Method	PSNR (dB)	SSIM	Total Parameter	Elapsed Training Time (sec)
DnCNN	26.94	0.83	**560,320**	605
CDAE	23.62	0.73	1,349,059	**50**
Proposed	**27.88**	**0.83**	639,008	1205

As a result, we observed that DnCNN method has better denoising results compared to CDAE method. The CDAE autoencoder method has a fast-training time and a simple architecture. On the other hand, it was observed that some of the details on the denoised image are lost. The DnCNN method has also a slightly longer training time than the CDAE method. Our proposed model yields better PSNR and SSIM results compared to DnCNN and CDAE methods. Because our model utilizes deep architecture and it causes higher training time and higher total parameters counts compared to the other methods.

In another study, we compared the proposed model with the state-of-the-art method DnCNN method. The vimeo test dataset is used to train the models, the denoising results of this comparison are shown in **Table 8**.

Table 8. Denoising results of the proposed model and the DnCNN using 180x180 Vimeo dataset

Noise Level (dB)	Method	PSNR (dB)	SSIM	FSIM	RMSE
σ = 10	**Proposed**	**33.45**	**0.96**	**0.98**	**0.00000595**
	DnCNN	31.23	0.95	0.99	0.00000654
σ = 20	**Proposed**	**31.28**	**0.94**	**0.98**	**0.00000670**
	DnCNN	28.50	0.88	0.96	0.00000919
σ = 30	**Proposed**	**27.96**	**0.93**	**0.97**	**0.00000976**
	DnCNN	27.35	0.88	0.96	0.00001102
σ = 40	Proposed	25.03	0.84	0.93	0.00001772
	DnCNN	**25.78**	**0.90**	**0.95**	**0.00001266**
σ = 50	Proposed	23.19	0.81	0.91	0.00001691
	DnCNN	**24.07**	**0.86**	**0.95**	**0.00001549**
σ = 75	Proposed	22.94	0.81	**0.91**	0.00001701
	DnCNN	**23.59**	**0.83**	0.91	**0.00001890**
σ = 100	Proposed	21.97	0.75	0.82	0.00002311
	DnCNN	**22.29**	**0.79**	**0.84**	**0.00002170**

It has been seen that our model preserves important image data, but as the noise level increases, noise residues begin to appear on the image. At higher noise levels, the DnCNN method yields better results. DnCNN, has the ability to adapt to different noise levels in images. The learning process allows the model to understand the characteristics of noise at different levels and adjust its denoising filters accordingly. DnCNN uses ReLU activation function which enabling it to capture complex relationships within the data.

In another study, the SCUNet model is compared with the proposed model. The comparison results of the SCUNet and proposed model using the Davis dataset with the first 10 BUS images of size 180x180 are shown in **Table 9**.

Table 9. Denoising results of the proposed model and the SCUNet model using DAVIS dataset

Noise Level (dB)	Method	PSNR (dB)	SSIM	FSIM	RMSE
σ = 10	**Proposed**	**31.78**	**0.94**	**0.99**	**0.00000991**
	SCUNet	30.72	0.95	0.99	0.00001812
σ = 20	Proposed	29.17	0.92	0.97	0.00001543
	SCUNet	**29.35**	**0.91**	**0.94**	**0.00002068**
σ = 30	Proposed	28.76	0.87	0.95	0.00001734
	SCUNet	**29.01**	**0.91**	**0.94**	**0.00002137**
σ = 40	**Proposed**	**25.61**	**0.79**	**0.93**	**0.00001908**
	SCUNet	24.41	0.80	0.89	0.00004124
σ = 50	Proposed	24.17	0.83	0.93	0.00002310
	SCUNet	**24.22**	**0.83**	**0.93**	**0.00003829**
σ = 75	**Proposed**	**22.09**	**0.80**	**0.85**	**0.00002885**
	SCUNet	21.75	0.79	0.88	0.00005086
σ = 100	**Proposed**	**21.45**	**0.77**	**0.81**	**0.00003293**
	SCUNet	19.42	0.76	0.81	0.00006658

The SCUNet model is designed to process high-resolution images while maintaining denoising accuracy. This is achieved using a perceptual loss function that measures the difference between the denoised image and the original image in terms of perceptual features rather than pixel-wise differences. SCUNet model and proposed model's architectures are composed of dilated convolutions and U-Net model. In U-Net architecture encoder and decoder layers are connected through skip connections. It has been observed that, the SCUNet model yields better results at low noise levels. As the noise level increases the denoising success rates decreases.

The proposed model and the other state-of-art models, DnCNN, FastDVDNet, SCUNet and ViDeNN, are compared using the Derf test dataset with an image size of 180x180. The comparison results of the state-of-art models and proposed model are shown in **Table 10**.

Table 10. Denoising results of the proposed model and the other models using Derf dataset

Noise Level (dB)	Model	PSNR (dB)	SSIM	FSIM	RMSE
σ =10	Proposed	33.23	0.96	0.98	0.00000524
	DnCNN	28.60	0.87	0.96	0.00000912
	FastDVDNet	34.94	0.97	0.96	0.00000755
	SCUNet	34.82	0.97	0.94	0.00000417
	ViDeNN	**34.99**	**0.97**	**0.98**	**0.00000430**
σ =20	Proposed	30.52	0.93	**0.96**	0.00000729
	DnCNN	25.41	0.83	0.95	0.00001585
	FastDVDNet	**34.44**	0.93	0.86	**0.00000606**
	SCUNet	30.71	0.86	0.88	0.00000669
	ViDeNN	32.30	**0.94**	0.87	0.00000585
σ =30	Proposed	29.47	0.90	**0.89**	0.00000891
	DnCNN	27.12	0.86	0.82	0.00001053
	FastDVDNet	**32.87**	**0.90**	0.83	0.00000916
	SCUNet	29.50	0.88	0.81	0.00000983
	ViDeNN	30.59	0.91	0.85	**0.00000713**

σ =40	Proposed	27.89	0.85	**0.82**	0.00001997
	DnCNN	27.42	0.87	0.72	0.00001051
	FastDVDNet	**32.21**	**0.89**	0.71	**0.00000884**
	SCUNet	28.80	0.87	0.71	0.00001060
	ViDeNN	29.20	0.89	0.73	0.00000838
σ =50	Proposed	25.90	0.79	0.85	0.00002013
	DnCNN	27.87	0.84	0.73	0.00001088
	FastDVDNet	**30.82**	**0.85**	**0.89**	0.00001246
	SCUNet	28.79	0.79	0.61	0.00001279
	ViDeNN	28.08	0.86	0.72	**0.00000953**
σ =60	Proposed	24.43	0.77	0.84	0.00002201
	DnCNN	27.35	0.84	0.72	0.00001048
	FastDVDNet	**29.96**	**0.83**	**0.86**	**0.00000805**
	SCUNet	27.43	0.84	0.72	0.00001050
	ViDeNN	27.17	0.84	0.71	0.00001059
σ =70	Proposed	24.10	0.75	0.83	0.00002301
	DnCNN	26.91	0.83	0.71	0.00001103
	FastDVDNet	**28.65**	**0.78**	**0.84**	**0.00001001**
	SCUNet	25.65	0.82	0.70	0.00001110
	ViDeNN	26.36	0.82	0.70	0.00001162
σ =80	Proposed	23.85	0.73	0.82	0.00002314
	DnCNN	24.76	**0.82**	0.71	0.00001642
	FastDVDNet	**28.36**	0.77	**0.83**	**0.00001388**
	SCUNet	24.23	0.80	0.70	0.00001702
	ViDeNN	25.61	0.80	0.74	0.00001268
σ =90	Proposed	23.69	0.71	0.81	0.00002334
	DnCNN	24.10	0.77	0.74	0.00001288
	FastDVDNet	**27.32**	0.72	**0.81**	**0.00001077**
	SCUNet	23.62	0.67	0.53	0.00001429
	ViDeNN	24.91	**0.78**	0.78	0.00001374
σ =100	Proposed	22.98	0.70	0.80	0.00002412
	DnCNN	22.31	**0.79**	0.74	0.00002175
	FastDVDNet	**26.12**	0.76	0.79	**0.00001483**
	SCUNet	22.15	0.78	0.74	0.00002149
	ViDeNN	24.16	0.76	**0.88**	0.00001498

DnCNN, FastDVDNet, SCUNet, ViDeNN and the proposed model are compared using Derf test dataset. While using the Derf dataset, the FastDVDNet method achieved better results compared to other methods. FastDVDNet's success can be attributed to its use of noise maps and two-stage denoising. The ViDeNN method yielded better results at the 10 dB noise level. ViDeNN has a convolutional denoising architecture and this model yielded successful results at low noise levels. The DnCNN model denoised the video frames without distorting of edges and textures. The SCUNet model groups similar patches and denoises these frames. In this dataset the SCUNet model achieves successful results between 0 dB and 80 dB noise levels. The developed model, achieves better results between the 0 dB and 30 dB noise levels. However, at higher noise levels, proposed model obtains less successful results. If we look at FSIM values, it can be seen that the developed model achieves successful results between 10 dB and 20 dB noise levels. Examining SSIM values, it can be seen that the ViDeNN model achieves successful results at 10 dB and 20 dB noise levels.

4.3 Execution Times

Some frame rates are commonly used in video production are as follows: 24 frames per second (fps) for cinema videos, 30 fps for standard videos, 60 fps for slow motion videos. If the input video is captured at 24 fps, the online image denoising model should be able to process each frame and output the denoised frame within 41 milliseconds (msec) or less. Real-time models should be fast enough to provide real-time denoising with low latency and low computational complexity. Execution times for image denoising operations are shown in **Table 11**.

Table 11. Runtime results of the models using Davis test data

Method	Prediction Time (msec)
Proposed	63
FastDVDNet	105
ViDeNN	90
DnCNN	**27**
SCUNet	72

Training the DnCNN model on the Berkeley segmentation dataset (CBSD400) using a single NVIDIA Tesla K40 GPU took about 12 hours [27]. The training time of SCUNet is expected to be similar to other U-Net based models, such as U-Net, Residual U-Net, and Attention U-Net. The training time of SCUNet model on the 400-image Berkeley segmentation dataset (CBSD400) using a single high-end GPU can take several hours to a few days. The actual training time can vary depending on the specific hardware and software configurations used.

The obtained results show that only DnCNN can be used in real time video denoising 24 fps and 30 fps videos. Other methods can be used for off-line image processing. The usability of a method in online image processing, the processing time, computational complexity, memory footprint, and accuracy of the method should also be considered.

4.4 Visual Comparisons

Following the process of image denoising, certain desirable outcomes should be achieved: the flat regions within the image should exhibit smoothness, the edges should be protected from blurring, the textures present in the image should be preserved, and importantly, no new noise should be introduced onto the image. Considering these rules denoising results are visualized in this part. Using the proposed model, a noise reduction study was conducted on the CBSD68 test dataset using images of 180x180 size with 50 dB noise added. The images obtained after the study are shown in **Fig. 6**. Generally, proposed models recover fine structures and preserves image sharpness However, the model also leaves a noise residue on the crocodile in the first image. In the second image the model fails to recover the green structure along the grass, while proposed model introduce some smoothness on the sky in the third image.



Noisy Images Denoised Images Original Images

Fig. 6. Denoising results of images from CBSD68 with noise level 50

State-of-art methods and proposed method are compared using 180x180 image size with DAVIS test dataset at 40 dB noise level. The obtained images are shown in **Fig. 7**.




Noisy Image
Proposed Model
DnCNN
FastDVDNet
SCUNet



Fig. 7. Visual comparison of state-of-art methods and proposed method using DAVIS test dataset

The FastDVDNet method achieved more successful results compared to other methods. When visually comparing images, it is observed that FastDVDNet model produces successfully denoised images, in line with PSNR and SSIM values. Upon examining FSIM values, both FastDVDNet and ViDeNN models are seen to achieve successful results. When SSIM values are examined, the FastDVDNet model produces successful results consistent with other quality metrics. Comparing RMSE values, in most cases, it is observed that the values from FastDVDNet and ViDeNN outperform those from SCUNet, ViDeNN, and our proposed model. RMSE is a measure used to quantify the amount of error between the denoised image and the real image. It is observed that there are noise residues in the SCUNet and DnCNN methods, and other models perform better in denoising the images. In proposed model flickering rendered by temporal denoising method in flat areas, but low- frequency noise appears in these areas on the white part of the bus. Blurry spots of the noise can be seen in some parts of the road. SCUNet produced smoother edges compared to the proposed model, but both of the models did not preserve some of the texture details on the road. As can be seen, the prepared model can recover the details of the regions at low noise levels, but it cannot eliminate high noise. Proposed method is successful in preserving edges but our model is less successful in preserving low-contrast details.

5. Discussions

Although deeper architectures in noise reduction have better results, deeper networks can cause overfitting and vanishing gradient problems. In this model CNN and GAN are used for denoising the image. This architecture improves the accuracy of the segmentation by reducing the noise in the image. This study shows that as the noise level in images increases, the proximity of the image that is obtained through noise reduction to the original image also decreases. In the generated convolutional neural networks, it is observed that the utilization of the following methods affects the success of the architectures positively:

- The receiver field was expanded to capture more context information. Dilated convolutions, which are proficient in enhancing edge details and preserving edge information by expanding the receiver field at the beginning of the model, are used.
- Deep neural networks have been preferred for better noise reduction results, and residual networks are chosen to enhance the performance of deep networks.
- Training data is seen as an important factor in the learning of the model, it is observed that data augmentation has an important role in the success of the model.
- It has been observed that the use of bias value can be used to increase the training time of the model also preventing overfitting in the model, resulting in better results.

- The choice of loss function also affects the success of the model. It is necessary to select the loss function depending on the structure of the model and the type of the problem.
- By detecting noise levels in the images, models can behave differently, and better results can be obtained at different noise levels. Detecting the noise level of the image and applying the model trained at this level produces more successful results.
- A hybrid model was created by using CNN model and GAN model. To reduce flickering in images, noise removal is done on similar image patches by using GAN method.

A new noise reduction method has been created by being inspired by CNN, GAN and U-Net methods. In the constructed model, the receiver field was expanded to provide more fine-grained information. By increasing the training data, the same model showed better learning and lower loss value when trained on the augmented data. According to the results obtained, the CNN-based method is successful in achieving low noise levels [0-49dB] for different images. For sequential images, the CNN-based method is successful in achieving low noise levels. Studies conducted with the CBSD400, DAVIS, and Vimeo datasets showed that the CNN-based model was less successful in the CBSD400, Derf and Vimeo datasets. It is observed that different models have achieved successful results in different images depending on color intensity and the number of objects. Both CNN and U-Net approaches have demonstrated state-of-the-art performance in image denoising tasks, with U-Net being particularly effective when dealing with more complex noise patterns or higher noise levels. However, the choice of which approach to use depends on the specific application and available data.

6. Conclusion

In our study, we aimed to achieve successful results using less training time and resources during image denoising. Therefore, in order to eliminate the negative effects of neural networks method, we introduced an automated denoising approach using CNN and GAN. This approach enables flexible and efficient video denoising capable of effectively eliminating noise without prior knowledge of the noise. By training the model in two stages, we reduced the training time and number of parameters and eliminated the noise with these two models. The behavior of proposed method at different noise levels is examined, and the results obtained from experiments with different data sets are compared. An ensemble method is used to combine CNN and GAN. Spatial model containing CNN and U-Net architecture is combined and their output is used as input of temporal denoising model. This approach reduced the risk of overfitting and improved denoising performance. However, this model does not exhibit good performance, and its effectiveness deteriorates as the noise level increases. Considering its flexibility and simplicity, the proposed model provides a practical solution to CNN denoising applications. The proposed method performs better (+0.15dB and +0.35dB) on some noise levels than other state-of-art CNN-based methods. However, in different datasets and noise levels state-of-art methods perform better results (+1.25dB and +1.45dB). It has seen that using noise maps, tracking images and denoising similar patches helps to obtain better results.

The studies did not reveal any mathematical model unknown in the literature other than how deep learning methods can be used more efficiently without traditional methods. Having limited resources, dividing model into parts can reduce required resources during the training of the model. Future studies will focus on investigating methods for denoising images across various noise levels, taking into consideration limitations such as the absence of real and noisy image pairs, system resource constraints, training time, and the preservation of proximity to

the original image. The aim is to develop techniques, that can effectively reduce noise while optimizing system resources and minimizing the need for paired data. Deep learning methods use significant amount of memory and GPU during training period. Consequently, future deep learning models should be designed to require less memory GPU, and CPU. Splitting the model across multiple GPUs or CPUs allows different parts of the model to run concurrently and it can help with memory limitations and designing larger models. On the other hand, training smaller models to mimic the behavior of larger models may require fewer resources while maintaining the performance of the larger model. Using hybrid models, combining deep learning with traditional algorithms, when possible, can leverage the strengths of both approaches, mitigating the limitations of each algorithm. We need strategies for creating such models that starts with a small and resource-efficient model and progressively increase its complexity during training based on the available resources.

References

[1] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising," *IEEE Transactions on Image Processing*, vol.27, no.9, pp.4608-4622, 2018. Article (CrossRef Link)

[2] M. Tassano, J. Delon and T. Veit, "DVDnet: A Fast Network for Deep Video Denoising," in *Proc. of the IEEE International Conference on Image Processing*, pp.1805-1809, 2019. Article (CrossRef Link)

[3] Y. Tai, J. Yang, X. Liu, "Image Super-Resolution via Deep Recursive Residual Network," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2790-2798, 2017. Article (CrossRef Link)

[4] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang and T. S. Huang, "Connecting Image Denoising and High-Level Vision Tasks via Deep Learning," *IEEE Transactions on Image Processing*, vol.29, pp.3695-3706, 2020. Article (CrossRef Link)

[5] F. Jia, W. H. Wong and T. Zeng, "DDUNet: Dense Dense U-Net with Applications in Image Denoising," in *Proc. of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp.354-364, 2021. Article (CrossRef Link)

[6] H. Liu, S. Liu, C. Wen and W. E. Wong, "TBEM: Testing-Based GPU-Memory Consumption Estimation for Deep Learning," *IEEE Access*, vol.10, pp.39674-39680, 2022. Article (CrossRef Link)

[7] D. Sadykova, A. P. James, "Quality assessment metrics for edge detection and edge-aware filtering: A tutorial review," in *Proc. of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp.2366-2369, Udupi, India, 2017. Article (CrossRef Link)

[8] V. Roy, S. Shukla, "Spatial and Transform Domain Filtering Method for Image De-noising: A Review," *International Journal of Modern Education and Computer Science*, vol.5, no.7, pp.41-49, 2013. Article (CrossRef Link)

[9] Z. Liu, W. Q. Yan and M. L. Yang, "Image denoising based on a CNN model," in *Proc. of the 4th International Conference on Control, Automation and Robotics (ICCAR)*, pp.389-393, 2018. Article (CrossRef Link)

[10] G. George, R. M. Oommen, S. Shelly, S. S. Philipose and A. M. Varghese, "A Survey on Various Median Filtering Techniques for Removal of Impulse Noise from Digital Image," in *Proc. of the Conference on Emerging Devices and Smart Systems (ICEDSS)*, pp.235-238, 2018. Article (CrossRef Link)

[11] C. He, C. Hu, W. Zhang and B. Shi, "A Fast Adaptive Parameter Estimation for Total Variation Image Restoration," *IEEE Transactions on Image Processing*, vol.23, no.12, pp.4954-4967, 2014. Article (CrossRef Link)

[12] X. Liao, "An improved ROF denoising model based on time-fractional derivative," *Frontiers of Information Technology & Electronic Engineering*, vol.21, pp.856-865, 2020. Article (CrossRef Link)
[13] T. Tasdizen, "Principal components for non-local means image denoising," in *Proc. of the 15th IEEE International Conference on Image Processing*, pp.1728-1731, 2008. Article (CrossRef Link)
[14] G. Chantas, N. Galatsanos and A. Likas, "Maximum a posteriori image restoration based on a new directional continuous edge image prior," in *Proc. of the IEEE International Conference on Image Processing*, pp.941-944, 2005. Article (CrossRef Link)
[15] X. Zhang, W. Zhou and Z. Duan, "Image super-resolution using nonlocally centralized sparse representation and fields of experts priors," in *Proc. of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp.1245-1249, 2017. Article (CrossRef Link)
[16] F. Xu, Y. Chen, C. Peng, Y. Wang, X. Liu and G. He, "Denoising of Hyperspectral Image Using Low-Rank Matrix Factorization," *IEEE Geoscience and Remote Sensing Letters*, vol.14, no.7, pp.1141-1145, 2017. Article (CrossRef Link)
[17] R. Cheng and H. Wang, "Image denoising based on learning sparse coding network," in *Proc. of the International Conference on Signal Image Processing and Communication*, vol.11848, 2021. Article (CrossRef Link)
[18] L. Jicheng and Z. Yi, "Image Denoising Based on Independent Component Analysis," in *Proc. of the International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp.18-20, 2009. Article (CrossRef Link)
[19] H. Zhong, C. Yang and X. Zhang, "A New Weight for Nonlocal Means Denoising Using Method Noise," *IEEE Signal Processing Letters*, vol.19, no.8, pp.535-538, 2012. Article (CrossRef Link)
[20] X. Zhang, "Image denoising using local Wiener filter and its method noise," *Optik*, vol.127, no.17, pp.6821-6828, 2016. Article (CrossRef Link)
[21] S. Ruikar and D. D. Doye, "Image denoising using wavelet transform," in *Proc. of the International Conference on Mechanical and Electrical Technology*, pp.509-515, Singapore, 2010. Article (CrossRef Link)
[22] L. Feng and J. Wang, "Research on Image Denoising Algorithm Based on Improved Wavelet Threshold and Non-local Mean Filtering," in *Proc. of the IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pp.493-497, 2021. Article (CrossRef Link)
[23] V.N. P. Raj and T. Venkateswarlu, "Denoising of medical images using undecimated wavelet transform," in *Proc. of the IEEE Recent Advances in Intelligent Computational Systems*, pp.483-488, India, 2011. Article (CrossRef Link)
[24] Z. Liu, W. Q. Yan and M. L. Yang, "Image denoising based on a CNN model," in *Proc. of the 4th International Conference on Control, Automation and Robotics (ICCAR)*, pp.389-393, Auckland, New Zealand, 2018. Article (CrossRef Link)
[25] A. E. Ilesanmi, T. O. Ilesanmi, "Methods for image denoising using convolutional neural network: a review," *Complex & Intelligent Systems*, vol.7, pp.2179-2198, 2021. Article (CrossRef Link)
[26] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, vol.131, pp.251-275, 2020. Article (CrossRef Link)
[27] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol.26, no.7, pp.3142-3155, 2017. Article (CrossRef Link)
[28] K. Kim, J. C. Ye, "Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images," in *Proc. of the NIPS'21: Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp.864-874, 2021. Article (CrossRef Link)
[29] L. Li, J. Wu and X. Jin, "CNN Denoising for Medical Image Based on Wavelet Domain," in *Proc. of the 9th International Conference on Information Technology in Medicine and Education (ITME)*, pp.105-109, Hangzhou, China, 2018. Article (CrossRef Link)
[30] H. Sun, M. Liu, K. Zheng, D. Yang, J. Li and L. Gao, "Hyperspectral Image Denoising via Low-Rank Representation and CNN Denoiser," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.15, pp.716-728, 2022. Article (CrossRef Link)

[31] F. Luo and X. Wu, "Maximum a Posteriori on a Submanifold: A General Image Restoration Method with GAN," in *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, pp.1-7, 2020. Article (CrossRef Link)
[32] C. Tian, Y. Xu and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol.121, pp.461-473, 2020. Article (CrossRef Link)
[33] M. C. Sheeba, and C. S. Christopher, "A Review On Video Denoising Methods," in *Proc. of the International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, pp.1-6, 2019. Article (CrossRef Link)
[34] P. Liu, H. Zhang, K. Zhang, L. Lin and W. Zuo, "Multi-level Wavelet-CNN for Image Restoration," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp.886-895, 2018. Article (CrossRef Link)
[35] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1874-1883, 2016. Article (CrossRef Link)
[36] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, N. Luo, "Enhanced CNN for image denoising," *CAAI Transactions on Intelligence Technology*, vol.4, no.1, pp.17-23, 2019. Article (CrossRef Link)
[37] M. Tassano, J. Delon, T. Veit, "FastDVDnet: Towards Real-Time Deep Video Denoising Without Flow Estimation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1351-1360, 2020. Article (CrossRef Link)
[38] C. Wang, S. K. Zhou, Z. Cheng, "First image then video: A two-stage network for spatiotemporal video denoising," *arXiv preprint arXiv:2001.00346*, 2020. Article (CrossRef Link)
[39] T. Xue, B. Chen, J. Wu, D. Wei, W. T. Freeman, "Video Enhancement with Task-Oriented Flow," *International Journal of Computer Vision*, vol.127, pp.1106-1125, 2019. Article (CrossRef Link)
[40] M. Claus, J. Van Gemert, "ViDeNN: Deep Blind Video Denoising," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp.1843-1852, 2019. Article (CrossRef Link)
[41] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, D.-P. Fan, R. Timofte, L. Van Gool, "Practical Blind Image Denoising via Swin-Conv-UNet and Data Synthesis," *Machine Intelligence Research*, vol.20, pp.822-836, 2023. Article (CrossRef Link)
[42] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5967-5976, 2017. Article (CrossRef Link)
[43] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung and L. V. Gool, "The 2017 DAVIS Challenge on Video Object Segmentation," *arXiv preprint arXiv:1704.00675*, 2017. Article (CrossRef Link)
[44] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. of the Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, pp.416-423, 2001. Article (CrossRef Link)
[45] S. Roth, M. J. Black, "Fields of Experts," *International Journal of Computer Vision*, vol.82, no.2, pp.205-229, 2009. Article (CrossRef Link)
[46] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, L. Zhang, "Waterloo Exploration Database: New Challenges for Image Quality Assessment Models," *IEEE Transactions on Image Processing*, vol.26, no.2, pp.1004-1016, 2017. Article (CrossRef Link)
[47] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp.1122-1131, Honolulu, HI, USA, 2017. Article (CrossRef Link)
[48] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp.1132-1140, Honolulu, HI, USA, 2017. Article (CrossRef Link)

[49] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol.2, 2019. Article (CrossRef Link)



AHMET YAPICI received BS degree from Ege University, Turkiye in 2004, while MS degree from Middle East Technical University of Software Engineering Department, Turkiye in 2007 and PhD degree from Gazi University of Information Systems, Turkiye in 2024. He is currently working as e-Government Director at Turksat. His research interests include IOT, data management, and artificial intelligence.



M. ALI AKCAYOL received BS, MS and PhD degrees from Gazi University, 1993, 1998 and 2002, respectively. In his dissertation, he worked on applying fuzzy logic and artificial neural networks to dynamic systems. He has been at Michigan State University, USA, 2004, for postdoctoral research. He is currently director of Big Data and Artificial Intelligence Laboratory that receives both national and international funding. The projects that he was principal investigator have been funded by various public and private sector organizations techniques, mobile wireless networks, and smart buildings. He has served as editor/field editor and reviewer for numerous international and national journals. He has served as session chairman, member of the organizing committee and member of the scientific committee at prestigious conferences around the world. He has ACM professional membership. His research interests include, artificial intelligence, deep learning, big data analytics, recommender systems, intelligent optimization systems, mobile wireless networks and smart buildings.