

U-Net Model for Watermark and Logo Removal: A Comparative Study

Bilge Kağan Özkan
Graduate School of Informatics
Middle East Technical University
Ankara, Turkey
ozkan.bilge@metu.edu.tr

Abstract—This study aims to remove disruptive watermarks from TV series and movies using the U-Net model. It also seeks to address the gap in the literature by examining the performance of the U-Net model on logos and watermarks of varying opacity and colors. The results of this study could serve as a roadmap for future research and applications.

Index Terms—watermark removal, U-Net, deep learning, image processing

I. INTRODUCTION

The proliferation of digital content has led to an increased use of watermarks in visual media such as TV series and movies. Watermarks are commonly employed by content creators to enhance brand recognition and protect copyright. However, these watermarks can be disruptive to viewers and negatively impact the viewing experience. Therefore, effectively removing watermarks has become a crucial requirement for improving user experience and facilitating content reuse scenarios.

Watermark removal is a highly complex task. Traditional methods, often relying on manual editing or simple image processing techniques, frequently fail to deliver satisfactory results. In this context, the application of deep learning, particularly deep neural network architectures like U-Net (Figure 1), holds the potential to develop more effective and automated solutions. The U-Net model has been demonstrated to be successful in various image processing tasks, and this study focuses on the hypothesis that it can also yield effective results in watermark removal.

This study aims to remove disruptive watermarks from TV series and movies using the U-Net model. In this context, it also seeks to address the gap in the literature by examining the performance of the U-Net model on logos and watermarks with varying opacity and colors. Enhanced versions of the U-Net model have been shown to more effectively remove watermarks with different characteristics [1]. Furthermore, it is well-known that deep learning methods are more robust and efficient compared to traditional methods [2]. Some studies [3] have demonstrated the effectiveness of multi-stage models in watermark removal, which supports our objective of investigating the effectiveness of a single-stage U-Net model.

The subsequent sections of this study will comprehensively cover the problem definition, environmental setup, datasets used, experimental results, and the findings obtained. Additionally, how the results of this study can form a roadmap for future research and applications will be discussed.

II. PROBLEM DEFINITION

The task of watermark removal is technically challenging. Watermarks can have different opacities and colors, come in various shapes and sizes, and are often placed in critical regions of the image. Additionally, filling in the area where the watermark was removed poses a significant problem. These factors complicate the watermark removal process and limit the effectiveness of existing methods. Traditional watermark removal techniques often require manual intervention or rely on simple image processing techniques, which frequently fail to produce satisfactory results.

Deep learning methods, particularly deep neural network architectures like U-Net, have made significant advancements in the field of image processing in recent years. The U-Net model has been successfully applied in various domains, such as medical image segmentation, and shows potential for effective watermark removal as well. However, the performance of the U-Net model on watermarks with varying opacities and colors has not been sufficiently explored, representing a significant gap in the literature.

The primary challenge of this study is to develop a method capable of effectively removing watermarks from TV series and movies. Specifically, the study will investigate how the U-Net model can remove watermarks with different opacities and colors. This research aims to address the existing gaps

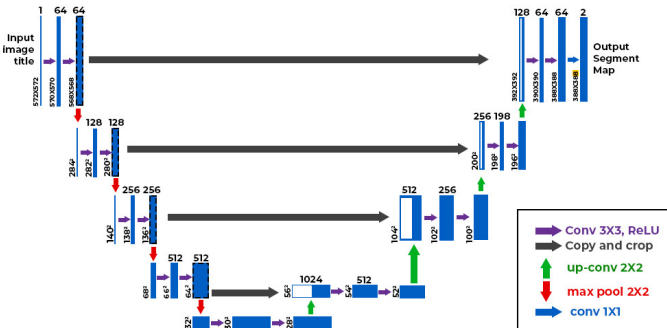


Fig. 1: Classical U-Net Model

in the literature and provide a detailed analysis of the U-Net model's performance. Solving this problem will contribute significantly to academic literature and enable higher quality and aesthetically superior images in practical applications.

III. LITERATURE REVIEW

In this section, various studies examining the use and performance of U-Net and its derivatives in the field of watermark removal will be discussed, and the existing gaps in the literature will be highlighted.

A. An Improved U-Net for Watermark Removal [1]

U-Net is a widely used architecture among deep learning-based image processing models. This model has achieved high accuracy rates, particularly in complex tasks such as medical image segmentation. The primary advantage of U-Net is its ability to produce effective results even with a small amount of data, and its capability to process both low and high-level features simultaneously. Fu et al. [1] proposed an improved version of U-Net called IWRU-net, demonstrating its superior performance in the task of watermark removal. IWRU-net aims to obtain more robust information through a serial architecture while enhancing the model's adaptability to real-world conditions by using randomly distributed blind watermarks. This study indicates the potential of effectively removing watermarks with various opacities and colors.

B. Visual Watermark Removal Based on Deep Learning [2]

Wei et al. [2] proposed AdvancedUnet, a single-stage neural network designed to simultaneously remove watermarks and reconstruct the image. AdvancedUnet is built on a Y-shaped structure consisting of an encoder and two decoders. One decoder extracts the binary mask of the predicted watermark, while the other produces the reconstructed image without the watermark. The model was trained using RSU modules that incorporate operations such as skip connections, upsampling, and downsampling. During training, a deeply supervised hybrid loss function was used. The model's performance was evaluated using a hybrid loss consisting of binary cross-entropy (BCE), structural similarity index (SSIM), and IoU losses. Although the performance of AdvancedUnet was compared to a baseline model, the specific baseline model and its performance levels were not disclosed. The researchers provided insights into the algorithm's consistency by observing the output loss graphs of the decoders at different stages.

C. WDNNet: Watermark-Decomposition Network for Visible Watermark Removal [3]

Liu et al. [3] introduced a model called WDNNet, making significant advancements in the removal of visible watermarks. WDNNet roughly localizes and separates the watermarks, then refines these results using a small RefineNet. The model's two-stage architecture makes the task of watermark removal more precise. Liu and colleagues also created a new dataset called CLWD (Colored Large-scale Watermark Dataset), which contains colored watermarks and is more suitable for real-world

applications, playing a crucial role in enhancing WDNNet's performance. The performance of WDNNet was validated using visual quality metrics such as the structural similarity index (SSIM), peak signal-to-noise ratio (PSNR), and other visual quality measures. The authors compared WDNNet's performance with models like Pix2pix, SRNet, ABD-Net, and SAD-Net, demonstrating WDNNet's superior performance in terms of watermark removal and visual quality.

D. Gaps in the Literature and Contributions of This Study

In the existing literature, there are numerous studies on deep learning-based watermark removal models. However, the performance of U-Net and its derivatives on watermarks with varying opacity, colors, and sizes has not been sufficiently explored. The primary aim of this study is to address these gaps and to provide a detailed analysis of the performance of the U-Net model on watermarks with different opacities, colors, and sizes.

IV. EXPERIMENTAL SETUP

In this section, the methods used for training and evaluating the U-Net model to remove watermarks and logos of different colors, opacities, and sizes from images are detailed. The methodology includes the creation of the dataset, the design of the loss function, the structure of the U-Net model, pre-processing techniques, model initialization and learning rate optimization, hyperparameter optimization, and performance evaluations.

A. Dataset

Three different datasets were created to measure and compare the performance of the U-Net model on watermarks and logos with high opacity levels. The methods and details used to create these datasets are outlined below.

1) *Dataset Preparation*: A suitable pre-existing dataset for our research could not be found, so the datasets were manually created. The training datasets were obtained from twenty-three different films, each approximately one hour long, while the test datasets were obtained from five different films of similar length. The same films were used for both the training and test datasets, and images were captured from the films at every 60th frame to create the datasets. Subsequently, for each of the three training and test datasets, watermarks and logos with varying opacities, colors, and sizes were added to these images according to the purpose of the dataset creation. This method ensured that the backgrounds in all training and test datasets remained the same while the added watermarks and logos were different from each other.

2) *Image Capture and Processing*:

- **Image Capture**: Images were extracted from each video at every 60th frame. This process was applied to ensure consistency within the datasets.
- **Image Cropping**: The extracted images were cropped from the center and resized to 1024x1024 pixels. This was done to ensure that all images are of the same size, facilitating easier processing.

- **Watermark and Logo Placement:** Similar to real TV shows, watermarks and logos were placed within a %10 area from each corner of the image. Each image was randomly assigned between 1 to 5 watermarks and logos. The size of the watermarks and logos was set to not exceed %10 of the image size. This optimization ensures the impact of the watermarks and logos on the images is balanced. A pre-existing dataset [4] was used for the logos.

3) Dataset Types:

- **No Logo Low Opacity:** Contains watermarks of low opacity with varying colors and sizes. This dataset was created to test the model's performance in scenarios where watermarks are less noticeable.
- **No Logo High Opacity:** Contains watermarks of high opacity with varying colors and sizes. This dataset is used to evaluate the model's effectiveness in situations where watermarks are more prominent.
- **Logo High Opacity:** Contains watermarks and logos of high opacity with varying colors and sizes. This dataset was designed to examine the model's performance on watermarks and logos with different opacity levels and colors.

4) Dataset Sizes:

- **Training Dataset:** Each training set contains 31,993 pairs of images with and without watermarks. These datasets were created to be used during the training phase of the models.
- **Test Dataset:** Each test set contains 7,481 pairs of images with and without watermarks. These datasets were used to evaluate and validate the performance of the models.

The datasets created in this manner will be used to assess the model's performance and measure its success on various watermarks and logos. This method provides an appropriate basis for analyzing the impact of watermarks and logos with different colors, sizes, and opacity levels. These datasets are critical for evaluating the overall performance of the U-Net model during both the training and testing phases.

B. U-Net Model

The classic U-Net model (Figure 1) is a fully convolutional neural network widely used in image segmentation. The U-Net model learns features of the input image at different scales and converts these features into a segmentation map. The model structure includes both downsampling and upsampling paths.

The structure of the classic U-Net model used can be summarized as follows:

- **Input Layer:** Receives the input image and passes it to the first convolutional layer.
- **Downsampling Path (Encoder) Layers:**
 - **Convolutional Layers:** Each contains two 3x3 convolutional filters. After each convolutional layer, Batch Normalization is applied, followed by the ReLU activation function.

- **Max Pooling Layers:** Reduces the size of the feature maps by half using 2x2 max pooling operations..

- **Upsampling Path (Decoder) Layers:**

- **Up-Convolutional Layers:** Doubles the size of the feature maps using 2x2 up-convolution operations.
- **Concatenation Layers:** Combines the relevant feature maps from the encoder path.
- **Convolutional Layers:** Each contains two 3x3 convolutional filters. After each convolutional layer, Batch Normalization is applied, followed by the ReLU activation function.

- **Output Layer:** The final layer uses a 1x1 convolutional filter to produce the output segmentation map.

Appendix A, shows the input and output channel numbers and the filter sizes used for each layer.

C. Preprocessing Techniques

In addition to the preprocessing techniques used during dataset creation, a transformation pipeline was established for model training. This pipeline includes the following techniques:

- **Image Resize:** To prevent COD (Cuda Out of Memory) errors during image processing and reduce training time, images were resized to 128x128 pixels.
- **Tensor Transformation:** Images were converted to tensor format to be processed by the model.
- **Zero Mean Normalization:** To improve model performance and make the learning process more efficient, each image was normalized using the zero mean normalization technique, ensuring that pixel values have a distribution with a mean of zero.

This transformation pipeline is crucial for ensuring that the model's input data is consistent and of high quality.

D. Loss Function

A specialized loss function has been developed to ensure the model focuses on areas with watermarks and logos. This loss function, called Weighted MSE Loss, is a mean squared error loss function that gives more weight to the watermark regions.

Weighted MSE Loss calculates the weighted MSE loss between the model output and the target images. The loss function aims to minimize the error rate in the watermark regions while reducing the focus on other areas. The weights are determined by a mask used to identify the watermark regions in the input images. This mask highlights the areas where the difference between the input and target images exceeds a certain threshold. The details of the loss function are explained below:

- **Watermark Weight:** Applies extra weight to the watermark regions. For this study, the value was set to 10.0.
- **Threshold:** Identifies the regions where the difference between the input and target images exceeds a certain threshold. This threshold is used to create the watermark mask. For this study, the value was set to 0.05.

The Weighted MSE loss function enhances the overall performance of the model, leading to more effective results in the task of watermark removal.

E. Model Initialization and Learning Rate Optimization

Determining the initial weights of the model and optimizing the learning rate are critical factors that directly impact the model's performance. In this study, the **Xavier Initialization** method was used to determine the initial weights of the model. Xavier Initialization ensures that the weights are appropriately scaled for each layer of the network, facilitating faster and more stable learning.

For optimizing the learning rate, the **StepLR** method was used. StepLR gradually reduces the learning rate using a specified step size and decay rate (gamma). This approach ensures more stable and effective learning throughout the training process. Specifically, a high learning rate is used at the beginning of the training process to enable rapid learning, while the learning rate is decreased in later epochs to allow the model to learn more finely-tuned and precise details.

The application of these methods has contributed to the effective optimization of the model's initial weights and learning rate, thereby enhancing the model's performance.

F. Hyperband Usage and Hyperparameter Optimization

In this study, the Hyperband technique was used to optimize the model's performance. Hyperband is a hyperparameter optimization algorithm that efficiently evaluates a large number of hyperparameter combinations to find the best one. This section explains how the Hyperband technique was used and how the model's hyperparameters were optimized.

Hyperparameter trials were conducted using Ray Tune. The hyperparameter search space evaluated in this study included:

- **lr (learning rate):** Ranged from $1e-4$ to 1, on a logarithmic uniform distribution.
- **batch size:** Options included 16, 32, and 64.
- **step size:** Options included 5, 10, and 15.
- **gamma:** Options included 0.1, 0.5, and 0.9.
- **epoch:** Options included 30, 40, and 50.

During the hyperparameter trials, specific CPU and GPU resources were allocated for each trial. After the trials were completed, the trial with the lowest validation loss was selected. The hyperparameter configuration from this trial was used to ensure the best model performance. This process effectively utilized the Hyperband technique to optimize the models' performance and identify the best hyperparameter combination. As a result, the best hyperparameter configuration was determined, and the models were trained with these configurations.

G. Cross Validation

To better evaluate the model's performance, the **K-Fold** cross-validation technique was used. In this study, a fold value of 5 was chosen, ensuring that in each fold, %80 of the dataset was used for training and %20 for validation.

H. Performance Evaluation

When evaluating the performance of the U-Net model, the following metrics were used in addition to the Weighted MSE Loss metric:

- **Structural Similarity Index (SSIM):** SSIM is a metric that measures the structural similarity between two images. It is used to evaluate how well the model preserves the structural integrity of the image while removing watermarks. SSIM values range from 0 to 1, with 1 indicating that the two images are completely identical.
- **Peak Signal-to-Noise Ratio (PSNR):** PSNR is a metric that measures the difference between the original and reconstructed images. High PSNR values indicate that the image quality is well preserved. PSNR is expressed in decibels (dB), and higher values indicate better image quality. In the literature, values of 40 dB and above are generally considered good.

These metrics quantitatively evaluated the model's ability to remove watermarks and logos while preserving the underlying image quality. Throughout the training process, both training and validation losses were monitored, and performance evaluations were conducted. Additionally, all three metrics were used during model testing.

V. RESULTS

In this study, the performance of the U-Net model was evaluated on three different datasets. The models were trained and validated using various folds, and the average values of the performance metrics, including Training Weighted MSE Loss, Validation Weighted MSE Loss, Validation PSNR, and Validation SSIM, were recorded.

A. Hyperband Results

The best hyperparameters obtained from the Hyperband optimization are presented in Table I

Parameter	No Logo Low Opacity U-Net	No Logo High Opacity U-Net	Logo High Opacity U-Net
Learning Rate	0.0035750314 13129566	0.0041113275 43636456	0.0008114977 372445823
Batch Size	16	16	16
Step Size	5	5	5
Gamma	0.5	0.9	0.9
Epochs	50	40	50
Best Trial Final Validation Loss	0.0005798752 343252999	0.0009434317 536943127	0.0007185489 579569548

TABLE I: Hyperband Results for Three Models

B. Training and Validation Results

1) *No Logo Low Opacity:* Appendix B contains the average train loss, validation loss, PSNR, and SSIM results for each epoch. The average training and validation loss results are summarized in Figure 2.

Figure 2 For the No Logo Low Opacity U-Net model, the training and validation losses over 50 epochs are shown.

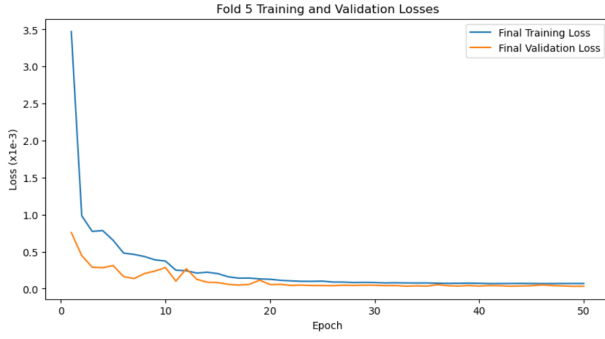


Fig. 2: Training and Validation Results for No Logo Low Opacity U-Net Model

The graph indicates that both losses decrease smoothly and converge to a low value, suggesting that the model effectively completed the learning process with minimal overfitting. The noticeable decrease in losses during the first few epochs indicates a rapid learning phase at the beginning. Towards the end of the epochs, the stabilization of both losses indicates that further training did not reduce the losses significantly, thus reaching a stable state.

2) *No Logo High Opacity*: Appendix C contains the average train loss, validation loss, PSNR, and SSIM results for each epoch. The average training and validation loss results are summarized in Figure 3

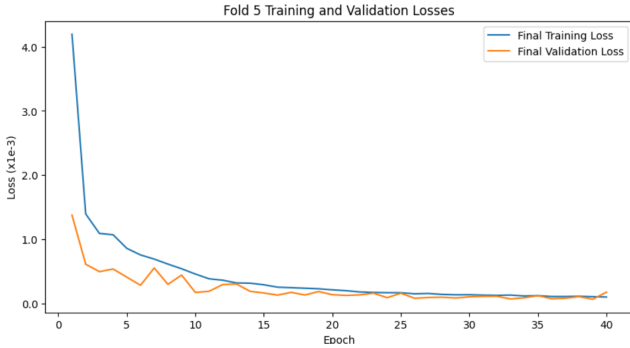


Fig. 3: Training and Validation Results for No Logo High Opacity U-Net Model

Figure 3 For the No Logo High Opacity U-Net Model, the training and validation losses over 40 epochs are presented. Significant decreases in both training and validation losses are observed within the first few epochs, indicating a rapid learning phase at the beginning. Additionally, the validation loss closely follows the training loss with minimal difference, suggesting that the model is not overfitting and performs well on the validation data. The stabilization of both training and validation losses towards a minimum value over time indicates that the model underwent a good training process and successfully generalized on the validation data.

3) *Logo High Opacity U-Net Model*: Appendix D contains the average train loss, validation loss, PSNR, and SSIM results

for each epoch. The average training and validation loss results are summarized in Figure 4



Fig. 4: Training and Validation Results for Logo High Opacity

Figure 4 For the No Logo High Opacity U-Net Model, the training and validation losses over 50 epochs are shown. The training loss starts at a high value, indicating a significant initial error rate for the model. While the training loss rapidly decreases at the beginning of the training process, the validation loss exhibits slight fluctuations. These fluctuations are due to the model optimizing its overall performance on the validation data while showing varying performance on different validation set examples during specific epochs. This can be attributed to the model adapting to the uncertainty and diversity in the validation data during the learning process. However, over time, the validation loss also decreases and stabilizes. This indicates that the model performs well on the training data and has improved its generalization ability on the validation data. The convergence of both training and validation losses to low values suggests that the model has successfully completed its learning process.

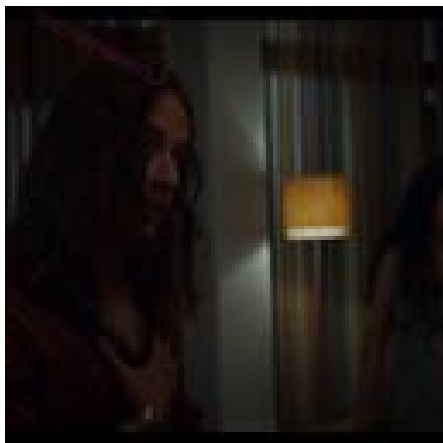
C. Test Results

Table II shows the test results for three different U-Net models. For each model, the average test loss, average PSNR (Peak Signal-to-Noise Ratio), and average SSIM (Structural Similarity Index) values are provided.

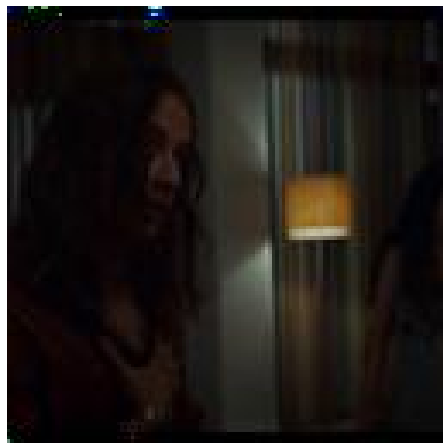
Model	Average Test Loss	Average PSNR	Average SSIM
No Logo Low Opacity U-Net	3.169995306961937e-05	54.96872464043579	0.9901973127689295
No Logo High Opacity U-Net	0.00015139468893884885	46.90792103210165	0.9702861880244207
Logo High Opacity U-Net	0.00014737646051798947	49.51239989339584	0.9670536850707767

TABLE II: Model Test Sonuçları

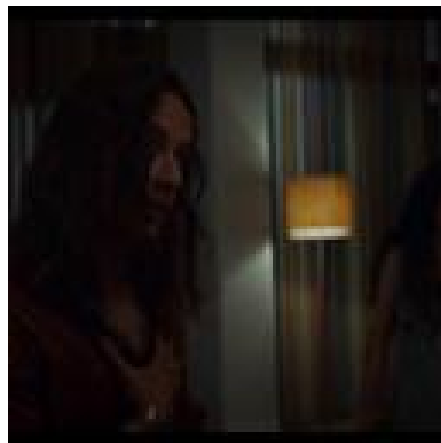
As seen in Table II, the No Logo Low Opacity U-Net model has a very low average test loss, a very high PSNR value, and an SSIM value that is nearly fully preserved. These findings indicate that the model is highly successful in removing low-opacity watermarks. The No Logo High Opacity U-Net model also has a low average test loss, with high PSNR and SSIM values, demonstrating its effectiveness in removing high-opacity watermarks. Similarly,



Watermarked Image

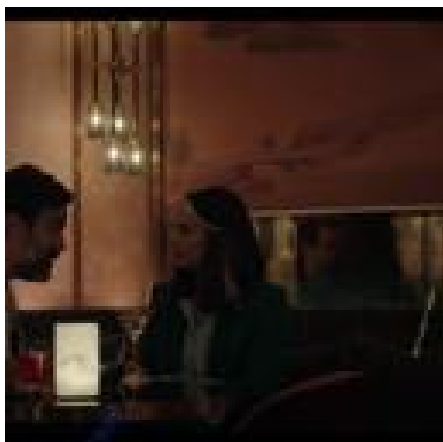


Model Output Image

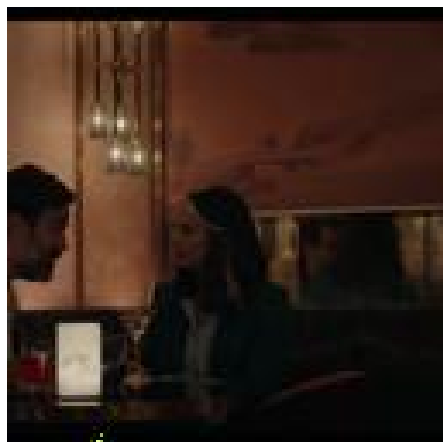


Original Image

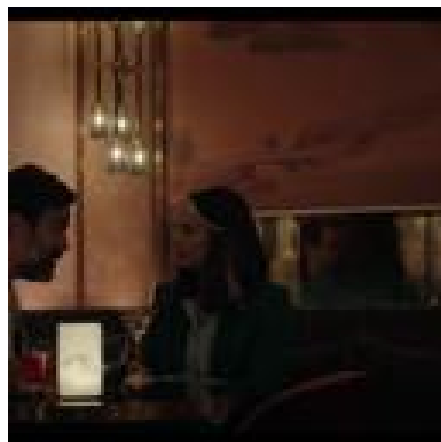
Fig. 5: No Logo Low Opacity U-NET Model Visualization Results



Watermarked Image

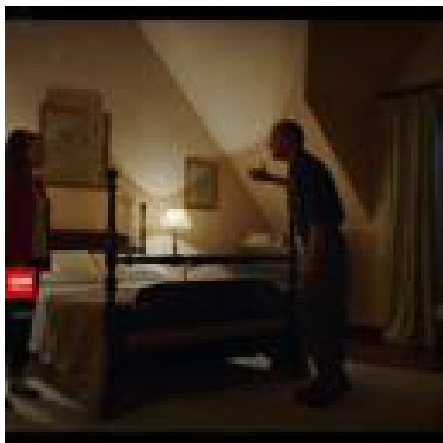


Model Output Image

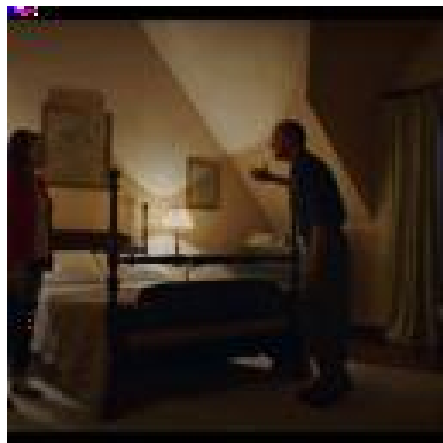


Original Image

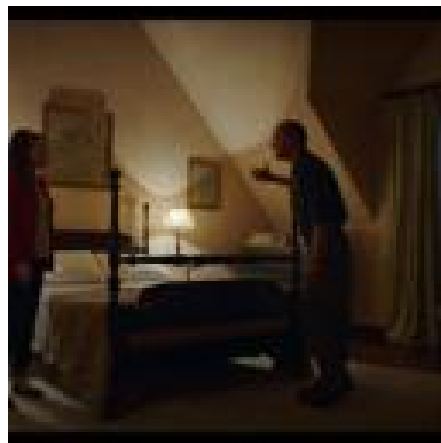
Fig. 6: No Logo High Opacity U-NET Model Visualization Results



Watermarked Image



Model Output Image



Original Image

Fig. 7: Logo High Opacity U-NET Model Visualization Results

the Logo High Opacity U-Net model shows low average test loss, high PSNR, and high SSIM values. These results indicate the model's success in removing high-opacity watermarks. Overall, the test results of all models show that the U-Net model is effective in removing logos and watermarks with different levels of opacity.

D. Visualization Results

Figure 5 When examining the No Logo Low Opacity U-NET Model Visualization Result, it is observed that the watermark is largely successfully removed, and there is no significant distortion in the image compared to the original. However, some distortions are noted, particularly in the black areas where the watermark was in contact with the edges. Overall, the model's performance appears to be quite successful, as the structural integrity of the original image is maintained, and the watermark is nearly entirely removed.

Figure 6 When examining the No Logo High Opacity U-NET Model Visualization Result, it is observed that the watermark is largely successfully removed, and there is no significant distortion in the image compared to the original. However, some distortions are noted, particularly in the black areas where the watermark was in contact with the edges. Overall, the model's performance appears to be quite successful.

Figure 7 When examining the Logo High Opacity U-NET Model Visualization Result, it is observed that the watermark is successfully removed, and the overall structure of the image is preserved compared to the original. Although there are some minor distortions in the areas where the watermark was in contact, the model's performance is generally quite successful. The clarity and color accuracy of the image are largely maintained. The small distortions in the regions where the watermark was removed do not significantly impact the image quality, indicating that the model is effective at removing even high-opacity watermarks.

VI. DISCUSSION

In this study, the performance of the U-Net model was evaluated across three different datasets. The best hyperparameters obtained from the Hyperband optimization varied for each dataset. These differences clearly illustrate how the opacity levels of the watermarks within the datasets impact the model's learning process.

A. Interpretation of Findings

The No Logo Low Opacity U-Net model has shown the best performance in removing low-opacity logos. This indicates that the model can effectively clean thin and low-opacity watermarks. The results obtained for the No Logo Low Opacity U-Net model, as shown in Figure 2 demonstrate that the training and validation losses decrease smoothly and converge to a low value. The visual results in Figure 5 also support this finding. The model output, when compared to the original image, shows that the watermark has been largely removed successfully and there is no significant distortion in

the image. However, there are some distortions, particularly in the black areas where the watermark made contact with the edges. Additionally, as seen in Table II, the No Logo Low Opacity U-Net model has the lowest average test loss, highest PSNR, and highest SSIM values, indicating its high effectiveness in removing low opacity watermarks. However, for the High Opacity U-Net models, a performance decline is observed compared to the No Logo Low Opacity U-Net model.

For the No Logo High Opacity U-Net model, Figure 3 shows a significant decrease in training and validation losses within the first few epochs, followed by stabilization. The visual results in Figure 6 indicate that the model successfully removes most of the watermark compared to the original image, although some artifacts are observed, particularly in the black areas where the watermark contacted the edges. As shown in Table II, the No Logo High Opacity U-Net model performs moderately in removing high-opacity watermarks, with a slightly higher average test loss and relatively lower PSNR and SSIM values compared to the No Logo Low Opacity U-Net model.

For the Logo High Opacity U-Net model, Figure 4, shows significant improvement in training and validation losses, although there are some fluctuations in the validation losses. These fluctuations may result from the model trying to optimize its performance on the validation set, causing varying performance across different epochs. The visual results in Figure 7 indicate that the model can effectively remove high-opacity watermarks and logos, although some minor artifacts are present in areas where the watermark contacted the image. As shown in Table II, the Logo High Opacity U-Net model performs similarly to the No Logo High Opacity U-Net model in removing high-opacity watermarks, with comparable average test loss, PSNR, and SSIM values, achieving moderate success compared to the No Logo Low Opacity U-Net model.

Overall, the results for all models indicate that the U-Net model is effective in removing logos and watermarks with varying opacity levels; however, the No Logo Low Opacity U-Net model performs better. All three models experience artifacts, particularly in the black areas where the watermark touched the edges. This issue may be attributed to several factors:

- **Lack of Information at the Edges:** The edge regions of images typically contain less information than the central regions. As the U-Net model upsamples and downsamples feature maps, it may struggle to make accurate predictions in these areas due to the lack of information.
- **Effect of Watermarks at the Edges:** Since watermarks are often placed at the edges of images, the high contrast differences in these areas can negatively impact the model's performance. High-opacity watermarks, in particular, can lead to more noticeable artifacts in these regions.
- **Less Data:** The number of samples with watermarks in the edge regions in the training dataset may be fewer

compared to those in the central regions. This can make it more difficult for the model to accurately remove watermarks in these areas.

- **Border Effect:** Convolutional neural networks may exhibit a distortion known as the border effect due to the lack of features in the boundary regions. This effect can become pronounced, especially when removing fine details such as watermarks.

B. Comparison with Literature

Our study's results are consistent with other research in the literature. For instance, [1] also noted the U-Net model's success in removing low-opacity watermarks. However, studies on high-opacity watermarks in the literature are limited. The findings of this study provide significant contributions toward filling these gaps in the field.

C. Strengths and Limitations

One of the strongest aspects of this study is the comprehensive analysis conducted on three different datasets. This analysis has helped in understanding how the model removes watermarks with various levels of opacity. However, there are some limitations to the study. Notably, the decrease in model performance during the removal of high-opacity watermarks indicates that further improvements are needed in this area. Additionally, the manual creation of datasets and the reduction of image sizes from 1024x1024 to 128x128 may not fully reflect the model's performance in real-world applications.

D. Conclusion and Applications

The results of this study demonstrate that the U-Net model is an effective tool for watermark removal. The use of the U-Net model in removing watermarks from media content such as television programs and films can significantly enhance the viewer experience. The findings of this study provide an important foundation for future research and encourage further advanced studies in this area.

Overall, the performance of the U-Net model is highly successful in removing low-opacity watermarks and shows promising results for high-opacity watermarks as well. This increases the model's applicability in various scenarios and highlights areas for improvement. By using different variations of the U-Net model and advanced techniques, it is possible to further enhance the watermark removal processes.

E. Suggestions for Future Research

Future work should explore different model architectures, especially generative AI model structures, to more effectively remove high-opacity watermarks because instead of the classical U-Net model, which deletes watermarks and logos with classical filtering logic, generative AI models that fill the gap with the prediction method may produce much different and interesting results. Additionally, larger and more diverse datasets should be used to improve the model's performance in removing watermarks of various colors and patterns. Moreover, in order to adapt the model to real-world conditions,

more complex and realistic watermark and logo data sets must be created and the visual dimensions must be increased. These improvements will help develop models more suitable for practical applications.

REFERENCES

- [1] L. Fu, B. Shi, L. Sun, J. Zeng, D. Chen, H. Zhao, and C. Tian, "An Improved U-Net for Watermark Removal," *Electronics*, vol. 11, no. 22, p. 3760, 2022. DOI: 10.3390/electronics11223760.
- [2] R. Wei, "Visual Watermark Removal Based on Deep Learning," *ArXiv preprint arXiv:2302.11338*, Feb. 2023.
- [3] Y. Liu, Z. Zhu, and X. Bai, "WDNet: Watermark-Decomposition Network for Visible Watermark Removal," *ArXiv preprint arXiv:2012.07616*, Dec. 2020.
- [4] <https://github.com/danzkigg/tvlogos>
- [5] <https://pytorch.org/>

APPENDIX A
U-NET MODEL LAYERS TABLE

Layer	Input Channel	Output Channel	Filter Size
Input Layer (DoubleConv)	3	64	3x3 (Conv)
Down1 (MaxPool + DoubleConv)	64	128	2x2 (MaxPool), 3x3 (Conv)
Down2 (MaxPool + DoubleConv)	128	256	2x2 (MaxPool), 3x3 (Conv)
Down3 (MaxPool + DoubleConv)	256	512	2x2 (MaxPool), 3x3 (Conv)
Down4 (MaxPool + DoubleConv)	512	1024	2x2 (MaxPool), 3x3 (Conv)
Up4 (UpConv + DoubleConv)	1024	512	2x2 (UpConv), 3x3 (Conv)
Up3 (UpConv + DoubleConv)	512	256	2x2 (UpConv), 3x3 (Conv)
Up2 (UpConv + DoubleConv)	256	128	2x2 (UpConv), 3x3 (Conv)
Up1 (UpConv + DoubleConv)	128	64	2x2 (UpConv), 3x3 (Conv)
Output Layer (OutConv)	64	3	1x1 (Conv)

APPENDIX B
NO LOGO LOW OPACITY U-NET MODEL TRAINING AND VALIDATION RESULTS TABLE

Fold	Epoch	Training Loss	Validation Loss	Validation Average PSNR	Validation Average SSIM
Fold 1	1	0.003473	0.000757	39.971	0.9099
	2	0.000984	0.000446	41.354	0.9018
	3	0.000773	0.000289	43.146	0.9398
	4	0.000785	0.000282	43.378	0.9385
	5	0.000653	0.000312	42.927	0.9499
	6	0.000480	0.000163	45.658	0.9702
	7	0.000463	0.000136	47.208	0.9814
	8	0.000434	0.000205	44.819	0.9645
	9	0.000389	0.000237	43.307	0.9323
	10	0.000373	0.000285	44.093	0.9724
Fold 2	1	0.000249	0.000102	47.808	0.9760
	2	0.000241	0.000269	44.276	0.9772
	3	0.000210	0.000126	46.445	0.9738
	4	0.000222	0.000086	49.330	0.9877
	5	0.000203	0.000082	49.016	0.9833
	6	0.000161	0.000057	51.141	0.9892
	7	0.000141	0.000049	52.497	0.9908
	8	0.000143	0.000057	51.168	0.9905
	9	0.000132	0.000114	48.597	0.9903
	10	0.000127	0.000055	51.093	0.9889
Fold 3	1	0.000112	0.000058	51.240	0.9905
	2	0.000105	0.000044	52.888	0.9908
	3	0.000099	0.000048	52.326	0.9904
	4	0.000099	0.000041	53.144	0.9915
	5	0.000101	0.000042	53.094	0.9912
	6	0.000089	0.000040	53.342	0.9911
	7	0.000088	0.000046	52.359	0.9912
	8	0.000083	0.000044	52.842	0.9921
	9	0.000085	0.000047	52.350	0.9920
	10	0.000083	0.000047	52.295	0.9913
Fold 4	1	0.000077	0.000041	53.054	0.9930
	2	0.000079	0.000042	52.676	0.9920
	3	0.000077	0.000033	54.512	0.9929
	4	0.000076	0.000038	53.637	0.9931
	5	0.000077	0.000034	54.198	0.9924
	6	0.000073	0.000054	51.242	0.9927
	7	0.000071	0.000040	53.214	0.9931
	8	0.000072	0.000035	54.245	0.9934
	9	0.000074	0.000043	52.640	0.9931
	10	0.000072	0.000035	54.250	0.9933
Fold 5	1	0.000068	0.000042	52.908	0.9929
	2	0.000069	0.000039	53.228	0.9928
	3	0.000069	0.000033	54.752	0.9931
	4	0.000071	0.000036	53.820	0.9930
	5	0.000069	0.000040	53.071	0.9926
	6	0.000068	0.000050	51.820	0.9928
	7	0.000068	0.000042	52.836	0.9928
	8	0.000069	0.000037	53.680	0.9927
	9	0.000069	0.000032	54.946	0.9932
	10	0.000069	0.000032	54.784	0.9930

APPENDIX C
NO LOGO HIGH OPACITY U-NET MODEL TRAINING AND VALIDATION RESULTS TABLE

Fold	Epoch	Training Loss	Validation Loss	Validation Average PSNR	Validation Average SSIM
Fold 1	1	0.004192	0.001377	37.357	0.8422
	2	0.001396	0.000614	41.168	0.9193
	3	0.001093	0.000498	41.795	0.9202
	4	0.001072	0.000539	41.016	0.8867
	5	0.000860	0.000412	42.517	0.9362
	6	0.000758	0.000286	45.557	0.9720
	7	0.000693	0.000555	40.995	0.9161
	8	0.000615	0.000300	44.328	0.9602
Fold 2	1	0.000542	0.000444	42.199	0.9308
	2	0.000461	0.000174	47.626	0.9762
	3	0.000387	0.000192	46.620	0.9716
	4	0.000365	0.000297	43.862	0.9618
	5	0.000322	0.000305	43.641	0.9762
	6	0.000318	0.000191	46.215	0.9770
	7	0.000293	0.000166	47.443	0.9776
	8	0.000256	0.000133	49.320	0.9848
Fold 3	1	0.000249	0.000176	46.528	0.9843
	2	0.000240	0.000134	48.525	0.9802
	3	0.000231	0.000188	46.573	0.9647
	4	0.000215	0.000139	47.727	0.9523
	5	0.000201	0.000128	48.284	0.9827
	6	0.000182	0.000136	48.065	0.9851
	7	0.000173	0.000163	46.552	0.9703
	8	0.000170	0.000093	51.716	0.9873
Fold 4	1	0.000169	0.000162	46.935	0.9833
	2	0.000153	0.000084	51.526	0.9869
	3	0.000158	0.000096	49.957	0.9799
	4	0.000143	0.000100	49.618	0.9839
	5	0.000137	0.000088	50.616	0.9870
	6	0.000138	0.000106	49.316	0.9707
	7	0.000133	0.000113	48.923	0.9862
	8	0.000130	0.000115	49.271	0.9855
Fold 5	1	0.000133	0.000072	52.130	0.9907
	2	0.000119	0.000091	50.252	0.9914
	3	0.000124	0.000124	48.079	0.9888
	4	0.000110	0.000076	50.880	0.9892
	5	0.000111	0.000084	50.497	0.9895
	6	0.000114	0.000109	48.503	0.9908
	7	0.000108	0.000068	51.982	0.9909
	8	0.000102	0.000178	45.668	0.9776

APPENDIX D
LOGO HIGH OPACITY TRAINING AND VALIDATION U-NET MODEL RESULTS TABLE

Fold	Epoch	Training Loss	Validation Loss	Validation Average PSNR	Validation Average SSIM
Fold 1	1	0.004165	0.000958	40.142	0.9225
	2	0.001235	0.000982	40.318	0.9327
	3	0.000978	0.000490	43.289	0.9513
	4	0.000917	0.000484	43.246	0.9525
	5	0.000817	0.000447	43.338	0.9477
	6	0.000704	0.001243	39.280	0.9429
	7	0.000643	0.000289	46.823	0.9745
	8	0.000581	0.000367	44.141	0.9710
	9	0.000548	0.000305	46.170	0.9675
	10	0.000491	0.000263	46.687	0.9692
Fold 2	1	0.000435	0.000233	47.220	0.9776
	2	0.000411	0.000323	44.102	0.9441
	3	0.000382	0.000285	45.363	0.9759
	4	0.000382	0.000254	46.020	0.9718
	5	0.000347	0.000235	46.958	0.9770
	6	0.000309	0.000222	47.474	0.9803
	7	0.000299	0.000199	47.763	0.9756
	8	0.000293	0.000244	47.211	0.9754
	9	0.000269	0.000188	48.594	0.9814
	10	0.000262	0.000181	49.590	0.9819
Fold 3	1	0.000262	0.000149	49.572	0.9819
	2	0.000238	0.000195	47.763	0.9759
	3	0.000225	0.000143	49.736	0.9835
	4	0.000223	0.000138	50.153	0.9840
	5	0.000219	0.000164	49.596	0.9852
	6	0.000210	0.000149	48.667	0.9787
	7	0.000197	0.000172	49.195	0.9862
	8	0.000191	0.000165	48.457	0.9851
	9	0.000183	0.000127	50.602	0.9856
	10	0.000191	0.000147	49.437	0.9840
Fold 4	1	0.000177	0.000105	50.581	0.9818
	2	0.000173	0.000143	48.904	0.9784
	3	0.000167	0.000099	51.712	0.9887
	4	0.000168	0.000107	50.599	0.9845
	5	0.000161	0.000100	51.682	0.9877
	6	0.000149	0.000117	49.127	0.9716
	7	0.000150	0.000113	50.715	0.9877
	8	0.000145	0.000116	49.841	0.9753
	9	0.000141	0.000101	51.130	0.9837
	10	0.000140	0.000101	51.188	0.9873
Fold 5	1	0.000137	0.000079	52.572	0.9906
	2	0.000132	0.000082	52.134	0.9899
	3	0.000127	0.000131	48.652	0.9888
	4	0.000128	0.000110	49.107	0.9747
	5	0.000128	0.000107	49.601	0.9868
	6	0.000120	0.000084	52.058	0.9901
	7	0.000117	0.000091	50.468	0.9834
	8	0.000113	0.000105	49.798	0.9840
	9	0.000111	0.000081	52.519	0.9903
	10	0.000114	0.000116	49.127	0.9765