Embedding Synthetic Patterns into Raw Images while Preserving Natural Image Statistics

1st Karthik
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

2nd Shaheen Khan
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

3rd Ananya Khed
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

4th Radhika Horti
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

5th Dr. Prabha C.Nissimagoudar
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

6th Prof. Sahana Punagin
dept. of Electronics and Communication
KLE Technological University
Hubballi,India

***Abstract*—The focal point of this report revolves around the innovative practice of embedding synthetic patterns into raw images, introducing a pioneering methodology to enrich visual data with artificially generated elements. The primary objective is to seamlessly incorporate computer-generated patterns into the raw image datasets, thereby achieving a harmonious fusion of naturalistic imagery and digitally augmented components. This exploration delves deep into the technical intricacies associated with the embedding process, dissecting its implications on image quality, authenticity, and its potential applications across diverse domains.**

The study not only scrutinizes the theoretical foundations that underpin the integration of synthetic patterns but also offers practical insights into the challenges and opportunities encountered during implementation. By providing a thorough examination of the results obtained through experimentation, this report makes substantial contributions to the ever-evolving fields of computer vision and image processing. Through a nuanced analysis, the research endeavors to unravel the intricacies of synthetic pattern embedding, shedding light on its transformative potential in reshaping visual content manipulation and enhancement. This comprehensive investigation aims to broaden our understanding of the implications and applications of embedding synthetic patterns into raw images, thereby pushing the boundaries of technological innovation in the realm of visual data processing

I. Acknowledgement

The success of this initiative can be ascribed to the efforts and direction of a lot of people.Throughout the research, we were lucky to receive helpful advice from the Guides, Professors Prabha C. N. and Sahana Punagin.We would like to thank them from the bottom of our hearts. We are also appreciative to KLE Technological University, Hubballi, our prestigious university, for giving us this chance.

Dr. Suneeta V Budihal, Head of the School of Electronics and Communication, has worked relentlessly to inspire and ensure the success of this attempt, and we would like to extend our sincere gratitude to her.

We also want to express our gratitude to the entire CIM-NVIDIA team for their help with this project. Last but not least, we would like to thank everyone who contributed significantly to this endeavour no matter how big or small.

II. Introduction

In computer vision, image processing, and related domains, embedding synthetic patterns onto pictures serves a variety of applications. Some examples may be: A dataset can be supplemented with synthetic patterns. Data augmentation is a typical strategy for increasing the quantity and variety of a training dataset, which can help machine learning models perform better. Fresh training examples that help the model generalize better by adding synthetic patterns to photos. Adding synthetic patterns to photos can be used to assess the resilience of image processing techniques or computer vision systems. These artificial patterns can be used to simulate real-world noise, distortions, or artifacts that the system may face in practice. Image processing or computer vision techniques must frequently be evaluated under controlled settings by researchers and engineers. They may evaluate how well these algorithms function in certain contexts by embedding synthetic patterns with known features. Synthetic patterns are useful for testing and troubleshooting. By providing predictable inputs, they can assist in identifying and resolving difficulties with image processing pipelines, object recognition, tracking, and other activities. Synthetic patterns can be used to watermark photos or movies in the context of digital media. Watermarks, which can be visible or invisible, are frequently used to protect intellectual property, authenticate material, and manage digital rights.

Synthetic patterns can be used to produce targets or markers that computer vision systems can easily identify. This is often employed in robotics, augmented reality, and computer vision applications that need the tracking of objects or markers. Synthetic patterns are employed in specific industries, such as computer graphics and virtual reality, to replicate natural textures, lighting conditions, or environmental impacts in order to create realistic and immersive virtual worlds. QR codes and barcodes are examples of synthetic patterns that

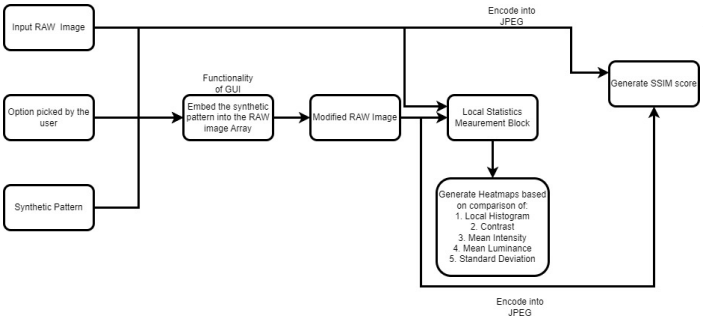

Fig. 1. Functional Block Diagram.

are used for security and authentication. These patterns may include encoded information that machines can simply read and analyze.

In summary, embedding synthetic patterns onto images is a versatile technique with various applications in image processing, computer vision, data augmentation, testing, security, and more. These patterns are valuable tools for improving, evaluating, and extending the capabilities of image-based systems and technologies.

III. Methodology

ARGUS is an API for acquiring images and associated metadata from cameras. The fundamental ARGUS operation is capturing that is acquiring an image from a sensor and processing it into a final output image. For input, first the RAW image frames are extracted from the live capture.Other inputs include the option selected by the user and the type of of the synthetic pattern to be embedded. The options available to the user are as follows:

- Simple user input (defines the position where the synthetic pattern needs to be embedded).
- Based on matching the contrast of the synthetic pattern to the best suited position in the RAW image(defines only the size of the pattern that needs to be embedded).
- Based on matching the contrast of the synthetic pattern to the desired location so as to not disturb the input RAW image (defines the position where the synthetic pattern needs to be embedded).
- Based on matching the mean intensity of the synthetic pattern to the desired location so as to not disturb the input RAW image (defines the position where the synthetic pattern needs to be embedded).

The synthetic pattern array that is selected is then embedded onto the input image array and the modified RAW image array is generated. The input RAW image and the modified RAW image is then fed into the local statistics measurement block that compares:

- Local histogram
- Contrast
- Mean luminance
- Mean intensity
- Standard Deviation

A. Generating the Synthetic Patterns

In order to add unique visual features to a picture for uses such as data concealing, watermarking, or image enhancement, one must generate synthetic patterns for embedding onto images.In computer vision and image processing, the Image-to-Image (img2img) diffusion process is a method used for image creation, image-to-image translation, and image editing.We make use of this method to generate synthetic patterns:

- Dot Pattern: TensorFlow is used in Python to build an img2img diffusion setup. In order to optimize the generation of a dot pattern over a gray backdrop, it defines a generator and discriminator network. The generator first generates random noise, which it then shapes to fit the form of the dot pattern and combines it with the backdrop and dot pattern to create a picture. After that, the discriminator evaluates these created pictures using both fictitious and authentic examples. The networks' parameters are adjusted during training to reduce the disparity between produced and real pictures over a period of epochs. Ultimately, a picture with a dot pattern is produced by the trained generator and stored as an png image
- Uniform Color Patch: This setup uses an img2img diffusion technique to create a homogeneous color patch. It starts with a picture of random noise and, over a predetermined number of steps, diffuses it towards a desired color. The picture converges to the desired color by the repetitive adjustment of pixel values by the diffusion process. Once the picture has undergone 100 diffusion stages, it is transformed into a PIL image using the uint8 format and saved as an png image. To manage the diffusion process and the final look of the picture, the user may choose the target color (RGB values between 0 and 1), the number of diffusion stages, and the image size.
- QR Code This setup generates a QR code using the URL "https://www.nvidia.com/en-in/" using the 'qrcode' library. It configures the version, error correction level, box size, and border of QR codes. The QR picture is created by the code with a white backdrop and a black foreground, and it is saved as "qr.jpg". It further uses the built-in image viewer to display the QR code. This script is handy for creating QR codes with text, contact information, or website URLs on them. It also lets you customize the characteristics of the QR code and the data that it encodes.

B. Framework for Embedding the Synthetic Patterns

- Simple User Input: Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.Check is the position defined by the user is within the image1.Compare the pixel values at the corresponding spots in picture 1 with those in image 2 multiplied by

256.Update the value of the pixel in image1 at that position to that the value of image2 multiplied by 256 if the pixel value in image1 is greater than the pixel value in image2 multiplied by 256.Then the pattern is embedded.

- Matching the Contrast of the Synthetic Pattern to the best suited location in the input image: Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.Calculate the standard deviation & determine the contrast of the pattern & Retrieves the dimensions (height and width) .Extract a block of the same size as the pattern from the image data array and then compare the contrast of the block with the pattern. If the contrast of the current block is smaller than the previous one, update the coordinates to embed the pattern to the current coordinates.
- Matching the Contrast of the Selected block in the input image to the Synthetic Pattern: Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.The function calculates the contrast of the selected pattern and the contrast of the specified block from the image.It adjusts the contrast of the pattern iteratively until it aligns within a threshold (in this case, within 5 units).Once the contrast is adjusted, it embeds the modified pattern into the specified block in the image
- Matching the Mean Intensity of the Selected block in the input image to the Synthetic Pattern: Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.The function calculates the mean intensity of the selected pattern and the mean intensity of the specified block from the image.It computes the intensity difference between the block and pattern, then normalizes the pattern array.The function then adjusts the intensity of the pattern by applying the intensity difference and normalization. Then the pattern is embedded.

C. Final Design

We have chosen to implement four different variations to embed synthetic pattern into the input RAW image as mentioned above to be able to compare the local and global statistics. The local statistics that are being compared are local histogram, contrast, mean luminance, mean intensity and standard deviation. The statistics are displayed in the form of a heat map for better visual representation.Then the RAW images are encoded into JPEG format to be able to compare the global statistic in the terms of SSIM (Structural Similarity Index).The methods are then compared on the basis of these statistics to find the most optimized method to embed synthetic pattern onto the RAW image

IV. Implementation

A. Algorithm

The algorithm of the solution for dynamic display of histogram and tone-curve has the following steps:

- Step1:From the live capture obtain the RAW Image frames.
- Step2:The user then picks an option from the available methods
  - Simple user input (defines the position where the synthetic pattern needs to be embedded).
  - Based on matching the contrast of the synthetic pattern to the best suited position in the RAW image(defines only the size of the pattern that needs to be embedded).
  - Based on matching the contrast/ intensity level of the synthetic pattern to the desired location so as to not disturb the input RAW image (defines the position where the synthetic pattern needs to be embedded).
- Step3:Prompt the user to select the Synthetic Pattern to be embedded amongst the available options (QR Code, Dot Pattern, Uniform Color Patch and Noise schedule).
- Step4:Input the above three values into the GUI for processing.
- Step5:Select the framework for embedding based on the option selected by the user.
  - Simple user input
    - Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.The function calculates the mean intensity of the selected pattern and the mean intensity of the specified block from the image.It computes the intensity difference between the block and pattern, then normalizes the pattern array.The function then adjusts the intensity of the pattern by applying the intensity difference and normalization. Then the pattern is embedded.
  - Based on matching the contrast of the synthetic pattern to the best suited position in the RAW image
    - Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.Calculate the standard deviation & determine the contrast of the pattern & Retrieves the dimensions (height and width) .Extract a block of the same size as the pattern from the image data array and then compare the contrast of the block with the pattern. If the contrast of the current block is smaller than the previous one, update the coordinates to embed the pattern to the current coordinates.

- Based on matching the contrast of the synthetic pattern to the desired location so as to not disturb the input RAW image
  - Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.The function calculates the contrast of the selected pattern and the contrast of the specified block from the image.It adjusts the contrast of the pattern iteratively until it aligns within a threshold (in this case, within 5 units).Once the contrast is adjusted, it embeds the modified pattern into the specified block in the image.
- Based on matching the mean intensity of the synthetic pattern to the desired location so as to not disturb the input RAW image
  - Capture RAW image and convert the raw file into array.Obtain synthetic pattern from diffusion model and convert the 8 bit image into an array. Obtain position and size of the pattern through the user.The function calculates the mean intensity of the selected pattern and the mean intensity of the specified block from the image.It computes the intensity difference between the block and pattern, then normalizes the pattern array.The function then adjusts the intensity of the pattern by applying the intensity difference and normalization. Then the pattern is embedded.

- Step6:After obtaining the modified raw image array, feed both the input RAW image and the modified RAW image into the Local Statistics measurement block.
- Step7: It determines dissimilar blocks based on contrast, mean brightness, mean intensity, and standard deviation differences using the Chi-square dissimilarity measure for histogram comparison.
- Step8: The algorithm iterates across blocks with predetermined sizes, comparing their properties, and storing the locations of blocks that differ. Then, by creating heatmaps that show the locations of the dissimilar blocks in each comparison, it visualizes these differences.
- Step9:Encode both the RAW images into JPEG format.
- Step10:Generate the SSIM score after comparing the two images for comparing the global statistics.

B. Flowchart for Embedding Synthetic Patterns into Raw Images while Preserving Natural Image Statistics.

The flowchart of the solution for Embedding Synthetic Patterns into Raw Images while Preserving Natural Image
As shown in the flowchart, the input to the GUI are taken in the form of the input RAW image, the option of embedding format from the user and the the synthetic pattern to be embedded selected by the user. After which the appropriate form of embedding is carried out based on the option selected by the user.The corresponding modified image array is obtained as result. The modified array contains the synthetic pattern embedded onto the input RAW image. The next step is comparison of the modified image and the input image which is carried out by comparison of local as well as global statistics. In terms of local statistics, we compare local histogram, contrast, mean luminance, mean intensity and standard deviation. This comparison is then presented in terms of a heatmap wherein the dissimilar blocks are represented as cold regions. For global statistics comparison we make use of SSIM

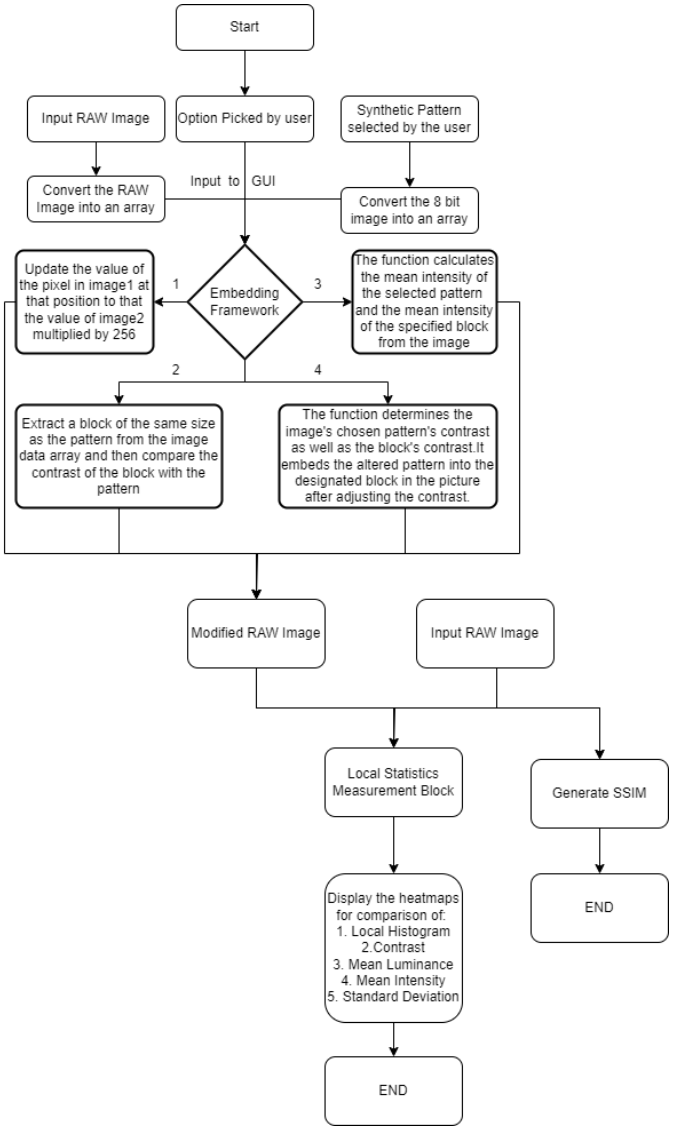

Fig. 2. Flowchart To Embed Synthetic Pattern

V. Results and Discussion

A. Result Analysis

We were able to obtain the modified RAW image with the synthetic pattern embedded onto it after executing our Sample Code.Both the RAW images were also converted in JPEG images using ARGUS sample application.The local statistics that were considered for comparison were local histogram,

Fig. 3. Input image

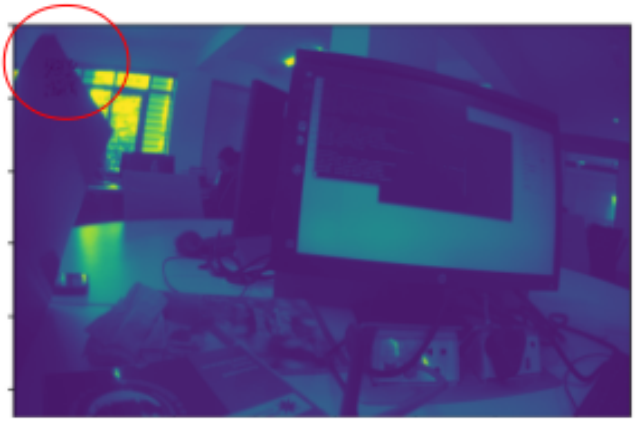

Fig. 4. Output image with pattern embedded at position (100,100)

contrast, mean luminance, mean intensity and standard deviation. To give the user a better idea of the dissimilar blocks present in the modified image, we displayed them in the form of a heatmap where the dissimilar blocks where represented as cold regions. We also verified the global statistics for the both images by making use of SSIM .The frames collected before and after the synthetic pattern was embedded onto them are shown in the following pictures, for indoor lighting conditions:

The below heatmaps clearly illustrate that there are

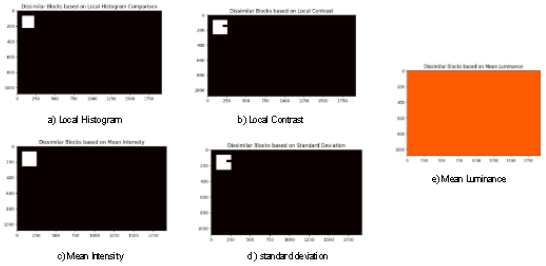

Fig. 5. Local Statistics measurement for simple user input

Fig. 6. Input image

Fig. 7. Output RAW Image (1212,707)

dissimilar blocks present where the synthetic pattern was embedded

The above frames represent the synthetic pattern embedded at the location of (1212,707) where best suited block is located. The heatmaps for this condition clearly illustrate that there are no dissimilar blocks present where the synthetic pattern was embedded.

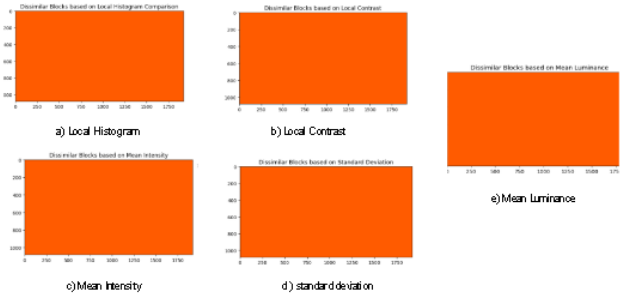

Fig. 8. Local Statistics Measurement for matching contrast of pattern

Fig. 9. Input image

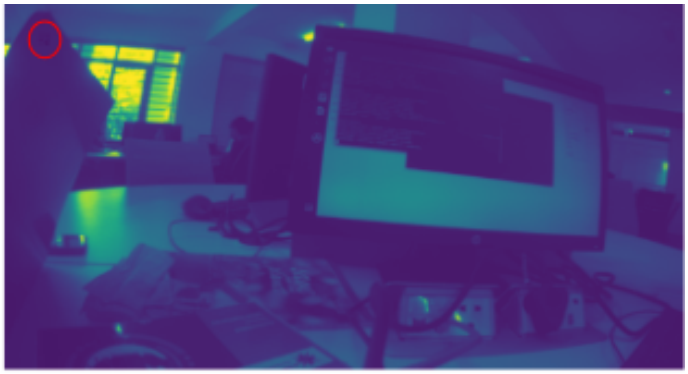

Fig. 10. Output RAW Image (100,100)

Matching the mean intensity of the selected block by adjusting the mean intensity of the synthetic pattern. We have presented the frames to which the synthetic pattern was embedded on the location of (100,100). The below heatmaps clearly illustrate that there are dissimilar blocks present where the synthetic pattern was embedded

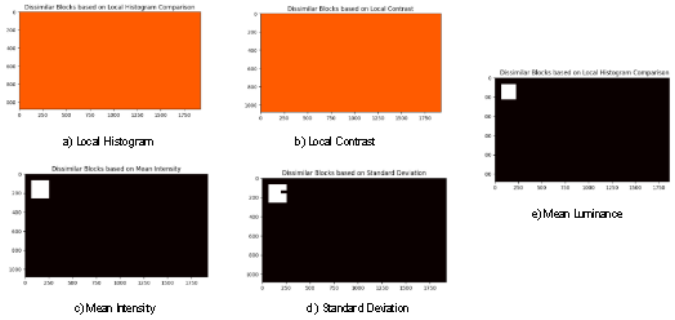

Fig. 11. Local Statistics Measurement for matching the contrast of desired location

TABLE I
COMPARISON OF SSIM SCORES OF THE EMBEDDING FRAMEWORKS

Embedding Framework	SSIM Score
Simple User Input	0.9973
Contrast of Pattern	1.0
Contrast of block	0.9988
Mean Intensity of block	0.9990

The table above compares the SSIM scores of the various embedding frameworks that the GUI makes use of. High similarity is shown across a range of characteristics in the examined picture comparison utilizing SSIM scores and the embedding architecture. Strong similarity is demonstrated by the remarkably high SSIM score of 0.9973 for simple user input. It is clear that the option two that is embedding based on the contrast of the pattern is the most efficient option with the SSIM score of 1.0, so as to be able to fully preserve the natural image statistics.Furthermore, there is a notable similarity between the contrast and mean intensity comparisons, with scores of 0.9988 and 0.9990, respectively, indicating almost comparable characteristics.Overall, the findings show a strong and constant similarity between the photos on a variety of measures, confirming their integrity and close likeness within the assessed limits.

VI. CONCLUSION

The technique provides a GUI for interactive picture modification and provides a variety of options for embedding artificial patterns into RAW photographs.Users select insertion methods—simple positioning, contrast matching, or preserving desired image features—to embed patterns.The method creates modified RAW photos and handles the embedding. Heatmaps are created for visual comparison by local statistic measures, which evaluate intensity differences, mean luminance, contrast, and histogram. Using ARGUS APIs, both RAW photos are converted to JPEG. To aid in comprehensive examination and assessment of the similarities between the original and changed pictures, the method also computes SSIM scores for global statistics comparison. This allows for well-informed decision-making in image processing jobs.

REFERENCES

[1] Shobhit Tyagi, Daivakar Yadav- "A Comprehensive Review on Image Synthesis with Adversarial Networks: Theory, Literature, and Applications- 27th october,2021

[2] Zhiyue Liu, Jinyuan Liu, Fanrong Ma -"Improving Cross-modal Alignment with Synthetic Pairs for Text-only Image Captioning - 14 December,2023.

[3] Luca Piras, Giorgio Giacinto -" Synthetic pattern generation for imbalanced learning in image retrieval -1 December 2012

[4] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah-" Diffusion models in vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023

[5] ZTong Li, Hansen Feng, Lizhi Wang, Zhiwei Xiong, and Hua Huang. "Stimulating the diffusion model for image denoising via adaptive embedding and ensembling. arXiv preprint arXiv:2307.03992, 2023

[6] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020