International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Reinforcing Visual Content Integrity through Image Restoration and AI Recognition

M. Gangappa[1], Eloori Karthik[2], Gaddampally Shreyas Reddy[3], Gudimetla Satya Manjunadha Reddy[4], Mohammad Shoaib[5]

[1]Associate Professor, Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engg & Technology, Hyderabad, Telangana, India
[2,3,4,5]Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engg & Technology, Hyderabad, Telangana, India

Abstract
A number of factors can cause photographs to deteriorate over time, losing their clarity and aesthetic appeal. The goal of this research is to help restore priceless memories because images act as memory triggers and better image quality can increase clarity and visual comfort. This research is important because it may aid in the recovery and preservation of special memories that were captured in damaged photos. Restoring an image's visual quality has the power to arouse feelings and create closer ties to the past.AI has made great progress in several areas in the recent past.However, questions concerning the veracity and authenticity of visual content have been raised due to the possibility of false information and the spread of deepfakes.We set out to create an automated system that could accurately distinguish between photographs taken by hand and images created by artificial intelligence as a solution to this problem.Furthermore, there are some useful advantages to improving image quality, including better vitality, higher clarity, and less eye strain when viewing the restored photos. All things considered, this study shows how deep learning using GANs works well for image restoration, emphasising its usefulness in retrieving priceless memories and improving visual content for wider uses. The findings of this study demonstrate the potential benefits that AI-driven image restoration methods could have on our daily interactions with visual information in the digital age as well as our personal life.

Keywords: Deep Learning, GAN, Machine learning, Image In-painting, Deepfake .

I. INTRODUCTION

Since images are efficient tools for expression, documentation, and communication, they are fundamental to our visual environment. However, images aren't always perfect; they can contain a variety of defects, such as sections that are missing or broken. Image restoration is the process of bringing these images back to life by filling in or repairing the damaged or missing portions. One fascinating and challenging part of image restoration can be accomplished with the help of picture inpainting. Image inpainting is the art and science of selectively painting or reconstructing missing or deteriorated sections of images to bring them back to life. It comprises adding new content to damaged or unwanted portions of the image while preserving the image's general integrity and coherence. There are several applications for picture inpainting, including photo enhancement, object removal, and image

completeness.

Numerous methods and approaches for picture inpainting have been developed over time; each has pros and cons of its own. Image inpainting has come a long way, from simple patch-based techniques to sophisticated deep learning systems. These methods are being applied in a wide range of applications, such as medical imaging, computer vision, multimedia, and image processing.[1].

This study looks at the current state of picture restorationusing image inpainting, emphasising significant advancements and techniques that have impacted the field. We will examine the evolution of inpainting methods from the earliest human interventions to the present era of data-driven deep learning. We will also discuss the challenges and future directions, giving ageneral picture of this quickly advancing field's future and its potential implications for picture restoration and related applications. The possibilities for image restoration and modification are continually growing in the dynamic and fascinating subject of image inpainting research.This research demonstrates the application of patch GANs for contextual analysis and picture restoration in image restoration.

In an era of rapidly advancing AI and machine learning technology, creating remarkably lifelike artificial intelligence (AI)-generated content—such as images, videos, and texts—has become normal practice. Sometimes referred to as "deepfakes" or "synthetic media," these artificial intelligence (AI)-generated workshave the ability to mix fact and fiction. While AI- generated content has legitimate uses in automation and the creative arts, there are serious concerns about inaccurate information, privacy violations, and maliciousintent.

Determining AI-generated content is therefore essential to preserving the integrity of digital data and media.

Artificial Intelligence (AI)-generated content is typically produced using techniques like deep neural networksand generative adversarial networks (GANs), which can produce visually identical images and videos to real ones. These artificial intelligence (AI) components can be used for a number of activities, such as creating entirely fictional text, manipulating images, and creatingfalse movies.

The vital topic of detecting AI-generated data and images is examined in this overview of the paper. This field is always evolving as detection techniques advance and the calibre of AI-generated media rises.

II. BACKGROUND

As images can be utilised for visual communication across all media, archiving, and artistic expression, they are fundamental to modern society. However, imagestypically experience a range of degradations, from physical damage such as scratches and stains to digital artefacts like loss of data and compression artefacts. Restoring or improving these images to make them more visually appealing, instructive, or complete is the aim of image restoration.

"Image inpainting" is a specific area of image restoration that deals with replacing missing or damaged picture portions. This job is challenging since it requires the production of a credible that complements the existing image without detracting from its overall coherence and structure. In the fields of computer vision, image processing, and allied sciences, image inpainting has gained popularity. Its applications are numerous and include image completion, object removal, and photocorrection.

Over the course of several decades, image inpainting has evolved, with early methods relying on manually constructed rules and heuristics. These techniques were unable to handle the challenging

inpainting jobs. However, the advent of machine learning and deep learning caused a significant upheaval in the field. Picture inpainting has seen a radical transformation thanks to Convolutional Neural Networks (CNNs) and other deep learning architectures, which can learn from large datasets and produce results that are realistic.

Two recent developments in deep learning-based inpainting models that have significantly improved the quality and realism of inpainted images are Generative Adversarial Networks (GANs) and U-Net. These models understand the context of the image, allowing them to produce content that fits in perfectly with the surroundings.

Although picture inpainting has come a long way, problems remain, such as handling large inpainting regions, preserving global coherence, and real-time image processing.The field is still in its infancy, investigating novel techniques, efficient algorithms, and applications in a variety of industries, including the creative arts and medical imaging.

Artificial intelligence-generated content, also referred to as synthetic media or deepfakes, is a result of the rapidly developing field of artificial intelligence, particularly in the area of deep learning. These artificial intelligence (AI) elements may produce convincing photos, movies, and text that nearly exactly match real content. These algorithms include Generative Adversarial Networks (GANs). There are legitimate applications for artificial intelligence (AI)-generated media in automation, entertainment, and the creative arts, but there are also significant ethical concerns.

The proliferation of content produced by artificial intelligence has given rise to worries about deception, invasions of privacy, and potential misuse. Deepfakes created by artificial intelligence (AI) have the potential to spread misleading information, impersonate people, change visual evidence, and deceive viewers. Therefore, it is increasingly crucial to create techniques and tools for detecting content generated by artificial intelligence in order to maintain the legitimacy and authenticity of digital media.

To identify material produced by artificial intelligence, a complex approach is needed (AI). This method may involve the use of various signs and artefacts left by the generative algorithms, as well as forensic investigation, metadata analysis, and machine learning models. Researchers and technologists are continuously working to stay ahead of the curve as AI-generated content becomes increasingly complicated and difficult to discern from real facts.

The development of AI content detection techniques is essential not only for combating false information and protecting privacy, but also for ensuring the security of digital environments and verifying the authenticity of media in a variety of fields, such as journalism, entertainment, social media, and law enforcement. Furthermore, the ethics of using and regulating such technology is a vital topic of discussion in this emerging business.

III. RELATED WORK

It required looking up and analysing pertinent scholarly publications in the IEEE Explorer and Google Scholar databases. The extensive coverage of scientific and technical literature in these databases is the reason they were selected. Boolean operators and keywords were used to find relevant results for our project.

Few models which help image restoration:

The application of Generative Adversarial Networks (GANs) to picture inpainting has advanced the area. GAN-based inpainting models have been created by researchers, and they can produce high-quality

content for areas that are missing or contaminated.[17].

[6] study shows how famous models like Partial Convolutional Neural Networks (PCN) and Context Encoders (CE) have used GAN architectures to produce realistic inpainted images. PCN use partial convolution layers to handle missing data adaptively, while CE uses an adversarial loss to promote the creation of realistic material.

The combination of GANs' generative power with diffusion models' sequential refinement process has been investigated in some recent studies as a potential synergy for image inpainting.[1].These models produce global coherence and fine details in high-quality inpaintings by utilising a GAN to direct the diffusion process. New model designs are being developed as research in this field continues to advance.[11].

Because Convolutional Neural Networks (CNNs) can learn from enormous datasets and collect spatial information, they are frequently employed in inpainting. CNNs are used by models such as U-Net and DeepFill to perform inpainting tasks.[1].To efficiently create inpaintings, the popular architecture U-Net uses an encoder-decoder structure with skip connections. Conversely, DeepFill makes use of a CNN to forecast absent pixels while taking the entire context into account.

These sophisticated inpainting methods are useful in many fields, such as picture editing, medical image restoration, video post-production, and artistic creation. In these domains, the capacity to repair damaged or insufficient photos is essential.

Few models which help AI generated content detection :

CNNs are useful for identifying irregularities in content since they can be used to examine the visual components of pictures and movies. CNNs are specifically used to compare the input image or video with recognised patterns or attributes linked to artificial intelligence (AI)- generated content.[22].To distinguish between actual and artificial intelligence (AI)-generated images or videos, pre-trained CNN models might be adjusted. Large datasets of both artificial intelligence (AI)-generated and genuine content are used to train these models so they can identify trends and attributes that point to manipulation. The model assesses the input media during inference and produces a probability or classification indicating the possibility of manipulation.[1].

Paper [28] explains how disparities in the generated material can be exposed by training a discriminator GAN model to distinguish between actual and AI- generated content. This allows GANs to be employed in a counter-GAN or adversarial strategy for AI-generated content detection.Real and AI-generated media are included in the dataset used to train the discriminator GAN model. The model gains the ability to recognise minute variations and inconsistencies in AI-generated information throughout training. In actuality, the discriminator is able to assess new media content and offer a rating or categorization according to its veracity.

Diffusion models can be used to analyse the statistical characteristics and pixel-level attributes of media in order to discover AI-generated content. Their objective is to detect irregularities in the information that depart from typical trends[11].To comprehend the statistical distribution of naturally occurring media material, diffusion models are trained. They examine the distribution of textures, pixel values, and other visual characteristics in the input material during the detection process. Content that may have been produced by artificial intelligence can be identified by deviations from predicted statistical features.

An image or video's colour channels are analysed for statistical characteristics and patterns in order to use colour space channel combinations for AI-generated content recognition. Through a close

examination of the channels (Red, Green, and Blue), abnormalities and inconsistencies that could point to information generated by artificial intelligence can be identified.Red, Green, and Blue are an image's three major colour channels, and they are separated by the RGB colour space. Finding differences and irregularities can be aided by analysing these channels independently.[24].Comparing an image's individual colour channels is one popular technique. AI- generated material could have inconsistent colour schemes or patterns throughout different media. Suspicion of manipulation may arise, for instance, if a face's Red channel differs from its Green or Blue channels in terms of features.

IV. METHODOLOGY1.Build model

Key methods:

build_inpaint_net:

Takes an incomplete image and mask as input.Builds a two-stage network:

Stage 1: Extracts features from the input usingconvolutional layers.

Stage 2: Employs contextual attention to refine theinpainted region and generates the final output.

Returns the predicted image at both stages and the offsetflow (if applicable).

build_sn_patch_gan_discriminator:

Builds a patch-based discriminator network withSpectral Normalization for GAN training.

build_gan_discriminator:

Builds the overall discriminator network, potentially combining the patch-based discriminator with additional layers.

build_graph_with_losses:

Constructs the full computational graph for training orinference.

Generates a mask, defines losses (L1 and GAN), andbuilds the discriminator if training.

Returns trainable variables for both generator anddiscriminator, along with the loss dictionary.

build_infer_graph:

Builds the inference graph for generating inpaintedimages.

Takes an incomplete image (optionally with edgeguidance) as input.

Returns the completed image.

build_static_infer_graph:

Similar to build_infer_graph but uses a fixed maskinstead of generating one.

build_server_graph:

Builds the graph for the server-side inference.

Takes an incomplete image (optionally with edgeguidance and mask) as input.

Returns the completed image.

Overall, this code implements a context-aware inpainting model that can be trained and used for image completion tasks.

2. Train and test the model

- Functions for generating random bounding boxes, creating masks based on bounding boxes, and creating brush stroke-like masks.
- Functions for cropping local patches, resizing masks, and handling different types of masks.
- Flow Visualization:
- Functions for converting flow maps to images, highlighting flow, and converting images to edges.
- Testing:

- I n c l u d e s a t e s t i n g f u n c t i o n (test_contextual_attention) that demonstrates theusage of the contextual attention layer on two input images.
- Command-Line Interface:
- The script can be run from the command line with three arguments: --imageA, --imageB, and --imageOut.

3. Input preprocessing

- Adds a batch dimension to the image and mask tensorsfor compatibility with TensorFlow models.
- Concatenates the image and mask along the channel axis to create a combined input.

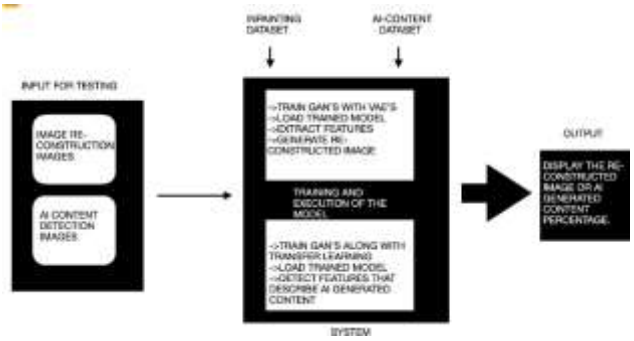
4. Generating output

The output is generated once the command line arguments are executed with the mask and the image to be inpainted are provided.This is output is stored locally with the name result.jpg .

5. Displaying it to the user

The final reconstructed image is displayed over an interface using flask.

V. ARCHITECTURE



Frontend:

The website's frontend includes the user interface where users can upload images or interact with the inpainting and content detection features.

Backend:

The backend of the website is responsible for handling user requests, processing images, and managing the integration of the two models.

Image Inpainting Model:

- One model is dedicated to image inpainting, the process of filling in missing or damaged parts of an image. This model takes an input image, identifies and understands the areas to be inpainted, and generates a visually coherent and complete output.
- It uses patch Gan's to implement image reconstruction function to the image that user gives as input.
- It is mostly based on contextual analysis which helps in improving the efficiency as it knows the context thus fill in the required ares based on it.

AI Content Detection Model:

- The second model is focused on content detection using artificial intelligence. It analyzes the uploaded images to identify and classify objects, scenes, orany relevant content within the image.

- We used GAN's to develop AI content detection model as GAN's have the best accuracy among all the methods available.

Integration:

- The website's backend coordinates the interaction between the inpainting and AI content detection models. When a user uploads an image, the backend first sends the image to the content detection model to identify objects and content within the image.
- The identified content is then used to guide the inpainting process. For example, if a detected object is damaged or missing in the image, the inpainting model can prioritize restoring or completing that specific object.
- The output from the inpainting model is a visually enhanced and completed image that incorporates both the original content and the inpainted areas.

Feature Extraction:

Both the inpainted image and the content detection results can be treated as features. These features can be extracted and used for further analysis, display, orany other relevant application.

User Interaction:

The website's frontend presents the inpainted image and the content detection results to the user. Users can explore the completed image and the identified content, providing an interactive and informative experience.

Feedback Loop:

The website may include features for user feedback, allowing users to provide input on the quality of inpainting or the accuracy of content detection. This feedback can be used to improve and fine-tune the models over time.

By integrating these two models into a cohesive architecture, the website provides users with a powerful tool for enhancing and analysing images with inpainting and AI content detection capabilities.

VI.IMPLEMENTATION

PatchGANs are a variant of GANs commonly used for image-to-image translation tasks, such as image inpainting. In PatchGANs, the discriminator classifies individual patches of the input image as real or fake, providing a more detailed and localised feedback. Here are the key equations for a PatchGAN :

Generator (G) Equation:

Similar to standard GANs, the generator (G) takes random noise z as input and generates synthetic datafake Xf :X =G(z,$). (1)

Patch Discriminator (D) Equation :

In PatchGANs, the discriminator (D) assesses theauthenticity of individual patches in the input image:

$$D(X,\$D)=\sum d(X,\$D)$$

Here d represents discriminator function , $D representsthe parameters of the entire discriminator.

Patch-wise Inpainting Objective:

For image inpainting using PatchGANs, a patch-wise reconstruction loss (Lrecon) is introduced, focusing on thequality of individual patches:

$$L_{total} = \lambda . L_{adv} + (1 - \lambda) . L_{recon}$$

The balance between the adversarial loss and the reconstruction loss is controlled by the

hyperparameterλ.
These equations highlight the principles of PatchGANs in image inpainting, where the discriminator assesses the realism of individual patches, providing a more localized and detailed feedback for the generator to improve the inpainting process.

Algorithm:
# Define the generator network class Generator:def__init_(self):
def forward(self, x):
def__init_(self):
def forward(self, x):
G = Generator()
D = Discriminator()
def adversarial_loss(output, target):
def reconstruction_loss(output, target):
Optimizers optimizer_G = Adam(G.parameters, learning_rate, betas=(0.5, 0.999))
optimizer_D = Adam(D.parameters, learning_rate, betas=(0.5, 0.999))
masked_images = mask_function(real_images) Generate labels real_labels = ones(batch_size, 1)
fake_labels = zeros(batch_size, 1)
real_output = D(real_images)
d_loss_real = adversarial_loss(real_output, real_labels) Forward pass masked images through G and then Dfake_images = G(masked_images)
fake_output = D(fake_images.detach())
d_loss_fake = adversarial_loss(fake_output, fake_labels)

# Total discriminator loss
d_loss = (d_loss_real + d_loss_fake) /2d_loss.backward() optimizer_D.step()

# 2. Update Generator optimizer_G.zero_grad()# Forward pass masked images through G
fake_images = G(masked_images) fake_output =D(fake_images)

# Generator adversarial loss
g_loss_adv = adversarial_loss(fake_output, real_labels)
f G
G is the generator function , z is the random noise vector and $G represents the parameters of the generator learnedduring training.
# Generator reconstruction loss
g_loss_rec = reconstruction_loss(fake_images, real_images)

# Total generator loss
g_loss = g_loss_adv + lambda_rec * g_loss_recg_loss.backward() optimizer_G.step()

Print log info

```
if (batch_index % log_interval == 0):
print(f"Epoch [{epoch}/{num_epochs}], Batch[{batch_index}/{len(dataloader)}], " f"D Loss: {d_loss:.4f}, G Loss: {g_loss:.4f}")

# Define a function to create masked imagesdef mask_function(images):

# Implement masking strategy (e.g., randomly occludepatches) pass

# Hyperparameters
num_epochs = 100 batch_size = 64 learning_rate =0.0002 lambda_rec = 10

# Weight for reconstruction losslog_interval = 10

# Data loader
dataloader = DataLoader(dataset, batch_size=batch_size,shuffle=True)

# Training proceduretrain()
```

Fine-tuning small parameters can significantly improve the performance of image inpainting algorithms. Hereare some parameters that we adjusted to get maximum performance.

Learning Rate:
Initial Learning Rate: We started with a suitable learning rate of 0.0001and settled at 0.0003 to improve the training progress .
Learning Rate Decay: We Implemented learning rate decay to reduce the learning rate over time, which helped in stabilizing the training process.

Batch Size:
Small vs. Large Batch Size: Experiment with different batch sizes. Smaller batch sizes lead to noisier updates while larger batch sizes benefited from more stable gradient estimates.

Optimizer Parameters:
Beta1 and Beta2 for Adam Optimizer: Adjusted the beta1 and beta2 parameters for the Adam optimizer to balance the trade-off between speed and stability.
Weight Decay: We set a small weight decay toregularize the model and prevent overfitting.
Dropout Rate:
Dropout Probability: Adjusted the dropout rate to 0.4 during training to prevent overfitting and improve generalization. Kernel Size and Stride:
Convolutional Layers: Adjusted the kernel size to 3x3 and stride of 1 and 2 of the convolutional layers to balance the trade-off between detail preservation andcomputational efficiency.

Normalization Layers:

Batch Normalization Momentum: Adjusted themomentum parameter of batch normalization layers to
0.9 to control how much of the past information isretained.

Attention Mechanisms:
Attention Heads: In multi-head attention mechanisms, adjusted the number of heads (e.g., 4, 8) to find the right balance between model complexity and performance andfixed it to 8.
Attention Dropout: Implemented dropout within attention layers and tuned the dropout rate to prevent overfitting.
Loss Weights:
Adversarial Loss Weight: Adjusted the weight of the adversarial loss component in the total loss function to balance between realism and pixel accuracy.
Perceptual Loss Weight: Tuned the weight of perceptual loss to ensure the generated images are perceptually similar to the ground truth.
Patch Size:
Patch Size in GANs: Adjusted the patch size to(128x128) to capture different levels of image detail.
Number of Layers and Filters:
Depth of Network: Adjusted the number of layers in thenetwork to find the optimal depth that balances performance and computational cost.
Number of Filters: Tuned the number of filters in each convolutional layer to control the capacity of the network.
Training Iterations and Epochs:
Number of Epochs: Increased the number of training epochs based on the convergence of the model.
Early Stopping: Implemented early stopping criteria to halt training when performance stops improving on a validation set.
Masking Strategy:
Mask Size and Shape: Experimented with different mask sizes and shapes to simulate various occlusion patterns during training.And finally settled with a grey scale masking strategy to acquire optimum performance.
Interface:
This interface provides an option to choose file from the system of the user ignorer to reconstruct their image. Once they choose the image they need to click on upload button to upload the damaged image and expect the reconstructed image.
Let us suppose the image given is :
Once the input is given then the algorithm is executed to provide the reconstructed image for the damaged image. Once it is run successfully , the generated reconstructed image is then uploaded on to the interface .

The output image is the image to the right which solves the problem of dead pixels and replaces them with the pixels which are obtained by series of convolutional and de-convolutional layers which are implemented once the image is given as input to the model and produces the reconstructed image as above.

VII. CONCLUSION

Finally, consumers looking to enhance and analyse photographs have a flexible and strong option with the integrated design of a website that includes AI content identification and image inpainting.Users can upload photographs, have damaged or missing areas intelligently filled in through inpainting, and concurrently obtain insights into the content within the images through AI-based recognition when these two features are smoothly combined.

The frontend's user-friendly interface makes it simple to interact with and explore the content detection results and inpainted images. Inpainting and content detection work together in a symbiotic relationship that is facilitated by the backend, with content detectiondirecting the inpainting process.

In addition to producing visually complete and coherent photos, this integration offers insightful information about the scenes and objects that have been detected. The features that were retrieved from the two processes can be used for additional research or visualisation. Furthermore, by include user input mechanisms, the models may be continuously improved, guaranteeing a responsive and dynamic system.

In the end, this project improves the aesthetic appeal of photographs while simultaneously providing users with an all-inclusive tool for content analysis and image manipulation. This makes it an invaluable asset for a variety of applications, ranging from automated content detection to creative image editing.

VIII. FUTURE SCOPE

Enhancing the quality and realism of inpainted images could be achieved by extending the capabilities of inpainting techniques with cutting-edge deep learning models, like those that incorporate attention processes. Furthermore, the incorporation of semantic segmentationmodels may yield a more profound comprehension of the visual context, resulting in more accurate and contextually aware inpainting. Enhancing the user experience even further would be possible with real-time processing system optimisation, which would enable quick and easy image editing on the website. Furthermore, a more thorough and adaptable analysis is promised by expanding the architecture to accommodate various modalities, such as video or 3D content, and integrating AI content detection throughout these modalities.

Beyond aesthetic improvements, future work may incorporate customisable inpainting features that allow users to give particular objects or styles priority during inpainting. Integrating the architecture for

real-time inpainting and content analysis within AR interfaces could be beneficial for augmented reality applications. It may be possible to include cross-modal retrieval features that allow users to look for photos using both textual descriptions and visual content.

REFERENCES

1. Jam, J., Kendrick, C., Walker, K., Drouard, V., Hsu, J. G. S., & Yap, M. H. (2021, February). A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, *203*, 103147. https://doi.org/ 10.1016/j.cviu.2020.103147.
2. Isogawa, M., Mikami, D., Iwai, D., Kimata, H., & Sato, K. (2018). Mask Optimization for Image Inpainting. *IEEE Access*, *6*, 69728–69741. https:// doi.org/10.1109/access.2018.2877401
3. Guo, Q., Gao, S., Zhang, X., Yin, Y., & Zhang, C. (2018, June 1). Patch-Based Image Inpainting via Two-Stage Low Rank Approximation. *IEEE Transactions on Visualization and Computer Graphics*, *24*(6), 2023–2036. https://doi.org/ 10.1109/tvcg.2017.2702738
4. Lu, H., Liu, Q., Zhang, M., Wang, Y., & Deng, X. (2017, March 13). Gradient-based low rank method and its application in image inpainting. *Multimedia Tools and Applications*, *77*(5), 5969–5993. https:// doi.org/10.1007/s11042-017-4509-0
5. Xue, H., Zhang, S., & Cai, D. (2017, September). Depth Image Inpainting: Improving Low Rank Matrix Completion With Low Gradient Regularization. *IEEE Transactions on Image Processing*, *26*(9), 4311–4320. https://doi.org/ 10.1109/tip.2017.2718183
6. Elharrouss, O., Almaadeed, N., Al-Maadeed, S., & Akbari, Y. (2019, December 6). Image Inpainting: AReview. *Neural Processing Letters*, *51*(2), 2007– 2028. https://doi.org/10.1007/s11063-019-10163-0
7. Ding, D., Ram, S., & Rodriguez, J. J. (2019, April). Image Inpainting Using Nonlocal Texture Matching and Nonlinear Filtering. *IEEE Transactions on Image Processing*, *28*(4), 1705–1719. https://doi.org/10.1109/tip.2018.2880681
8. Krishna, P. R., & Rajarajeswari, P. (2022, August 31). EapGAFS: Microarray Dataset for Ensemble Classification for Diseases Prediction. *International Journal on Recent and Innovation Trends in Computing and Communication*, *10*(8), 01– 15.https://doi.org/10.17762/ijritcc.v10i8.5664
9. Fan, Q., & Zhang, L. (2017, September 2). A novel patch matching algorithm for exemplar-based image inpainting. *Multimedia Tools and Applications*, *77*(9), 10807–10821. https://doi.org/10.1007/ s11042-017-5077-z
10. K. Alilou, V., & Yaghmaee, F. (2016, March 3). Exemplar-based image inpainting using svd-based approximation matrix and multi-scale analysis. *Multimedia Tools and Applications*, *76*(5), 7213–7234. https://doi.org/10.1007/s11042-016-3366-6
11. Jiang, Y., Xu, J., Yang, B., Xu, J., & Zhu, J. (2020). Image Inpainting Based on Generative Adversarial Networks. *IEEE Access*, *8*, 22884–22892. https://doi.org/10.1109/access.2020.2970169
12. Quan, W., Zhang, R., Zhang, Y., Li, Z., Wang, J., & Yan, D. M. (2022). Image Inpainting With Local and Global Refinement. *IEEE Transactions on Image Processing*, *31*, 2405–2420. https://doi.org/10.1109/ tip.2022.3152624
13. Sridevi, G., & Srinivas Kumar, S. (2019, January 14). Image Inpainting Based on Fractional-Order Nonlinear Diffusion for Image Reconstruction. *Circuits, Systems, and Signal Processing*, *38*(8), 3802–3817. https://doi.org/10.1007/s00034-019-01029-w

14. Wang, N., Zhang, Y., & Zhang, L. (2021). Dynamic Selection Network for Image Inpainting. *IEEE Transactions on Image Processing*, *30*, 1784–1798. https://doi.org/10.1109/tip.2020.3048629
15. R. K. Peddarapu, S. Balaga, Y. R. Duggasani, S. K. Potlapelli and S. C. Thelukuntla, "Liver Tumor Risk Prediction using Ensemble Methods," 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Dharan, Nepal, 2022, pp. 1077-1082, https://doi: 10.1109/I- SMAC55078.2022.9987419.
16. Liao, L., Hu, R., Xiao, J., & Wang, Z. (2019). Artist- Net: Decorating the Inferred Content With Unified Style for Image Inpainting. *IEEE Access*, *7*, 36921– 36933. https://doi.org/10.1109/access.2019.2905268
17. Wang, H., Jiao, L., Wu, H., & Bie, R. (2019). New Inpainting Algorithm Based on Simplified Context Encoders and Multi-Scale Adversarial Network. *Procedia Computer Science*, *147*, 254–263. https:// doi.org/10.1016/j.procs.2019.01.250
18. Abdal, R., Zhu, P., Mitra, N. J., & Wonka, P. (2021, May 5). Style Flow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. *ACMTransactions on Graphics*, *40*(3), 1–21. https:// doi.org/10.1145/3447648
19. Zeng, Y., Fu, J., Chao, H., & Guo, B. (2023, July 1). Aggregated Contextual Transformations for High- Resolution Image Inpainting. *IEEE Transactions on Visualization and Computer Graphics*, *29*(7), 3266– 3280. https://doi.org/10.1109/tvcg.2022.3156949
20. Guo, H., Hu, S., Wang, X., Chang, M. C., & Lyu, S. (2022). Robust Attentive Deep Neural Network for Detecting GAN-Generated Faces. *IEEE Access*, *10*, 3 2 5 7 4 – 3 2 5 8 3 . https://doi.org/10.1109/ access.2022.3157297
21. Singh, J. M., & Ramachandra, R. (2023). Deep Composite Face Image Attacks: Generation, Vulnerability and Detection. *IEEE Access*, *11*, 7 6 4 6 8 – 7 6 4 8 5 . https://doi.org/10.1109/ access.2023.3261247
22. Bhavsar, A. V., & Rajagopalan, A. N. (2012, April). Range map superresolution-inpainting, and reconstruction from sparse data. *Computer Vision and Image Understanding*, *116*(4), 572–591. https:// doi.org/10.1016/j.cviu.2011.12.005
23. Ramakrishna, P., & Rajarajeswari, P. (2023). Evolutionary Optimization Algorithm for Classification of Microarray Datasets with Mayfly and Whale Survival. International Journal of Online and Biomedical Engineering (iJOE), 19(13), pp. 17– 37. https://doi.org/10.3991/ijoe.v19i13.40145
24. Segawa, Y., Kawamoto, K., & Okamoto, K. (2018, May 14). First-person reading activity recognition by deep learning with synthetically generated images. *EURASIP Journal on Image and Video Processing*, *2018*(1). https://doi.org/10.1186/ s13640-018-0272-z
25. Mo, S., Lu, P., & Liu, X. (2022, October 27). AI- Generated Face Image Identification with Different Color Space Channel Combinations. *Sensors*, *22*(21), 8228. https://doi.org/10.3390/s22218228
25. Choi, J. G. (2000, December 1). Reliable watermark detection method based on analysis of correlation. *Optical Engineering*, *39*(12), 3308. https://doi.org/10.1117/1.1327900
26. ZHAI, D., ZUO, W., DUAN, W., YU, J., & LI, T. (2013, December 17). Image inpainting algorithm based on double-cross curvature-driven diffusion model. *Journal of Computer Applications*, *33*(12), 3 5 3 6 – 3 5 3 9 . https://doi.org/10.3724/sp.j.1087.2013.03536

27. Xiao, Y., & Zhao, J. (2024). Evaluating the effectiveness of machine learning models for detecting AI-generated art. *Journal of Emerging Investigators*. https://doi.org/10.59720/23-132
28. Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake Detection for Human Face Images and Videos: A Survey. *IEEE Access, 10,*1 8 7 5 7 – 1 8 7 7 5 . https://doi.org/10.1109/access.2022.3151186