Deep Learning-Based Invisible Watermarking SystemA CNN Encoder-Decoder Approach for Robust Logo Embedding and Extraction

Dr. Deepti Varshney[1], Arpan Gujar[2], Anshul Patel[3], Sharvari Arun Utekar[4]
[1]*Professor, Department of Computer Engineering Ajeenkya D.Y. Patil School of Engineering Pune, India*
[2,3,4]*Department of Computer Engineering Ajeenkya D.Y. Patil School of Engineering, Pune India*

***Abstract*—Protecting media rights and preventing unapproved sharing is why digital watermarks matter. Digital watermarks help safeguard media rights and stop unauthorized distribution. A new method for setting up image watermarks uses deep learning tools. This approach employs deep learning to create the image watermark setup.**

It works by placing the image in a system that reduces its size. Then, it adds a logo to the image. The logo remains hidden on the image. The system also includes a network that can locate the logo again, even if the image is damaged. The system continuously learns how to do this. It measures its performance by assessing the clarity of the image and the logo's detectability.

Colors are also crucial for watermarks. Thus, the system follows a rule to ensure the colors remain accurate. This rule prevents unwanted changes to the image's colors. The system was tested on numerous images, and the results showed that the digital watermarks were very difficult to see. The system performs well even when the image is damaged. It can still locate the logo when the image is blurry or noisy. The system is built using PyTorch, employing a framework for learning and a setup for processing requests. When it receives an image, processing takes only a fraction of a second. The system is also compact, making it suitable for production environments without issues. Digital watermarks are important, and this system offers a new method for creating them.

***Index Terms*—Digital Watermarking, Deep Learning, Convolu-tional Neural Networks, Encoder-Decoder Architecture, Residual Learning, Image Authentication, Logo Extraction, Robustness, PSNR, SSIM.**

I. INTRODUCTION

A. Background and Motivation

User-generated photos and images created by artificial intelligence are appearing online at a rapid pace. This makes it harder to determine what is real and who owns the images. When photos are altered, like being cropped or edited, hidden marks, such as digital footprints, can completely disappear. Since these tags can easily vanish, they won't safeguard your work when problems arise.

Hidden digital tags can help solve this issue. They embed identification details directly within the image, making no noticeable difference. What matters is that this method remains concealed, can withstand changes, holds substantial data, and is nearly impossible to detect or remove without permission.

For images, older tools like DCT, DWT, or SVD work fairly well. However, making them function correctly demands a lot of hands-on work and careful adjustments. When images face threats, these tools struggle to find useful information. Also, most of these tools can only manage specific types of watermarks and don't perform well with larger watermarks that feature many colors.

Not every tool manages images in this way. Some methods can recognize patterns on their own without explicit instructions. Problems can occur, like unwanted shades appearing in the output. These systems can also get quite complex. They often struggle with real-world issues and are rarely tested with diverse examples. As a result, progress is slow

with newer learning models that are meant to be better.

B. Research Gap and Problem Statement

There are still some issues with watermarking, even though it has improved significantly. For instance, many methods that use CNN show color changes, such as green, pink, or purple, in areas that should be the same color. This happens due to how the system processes color.

Typically, it is difficult to find a balance between making the watermark hard to see and strong enough to remain intact when someone tries to remove it. If the watermark is too weak, it can be easily removed. If it is too strong, it becomes noticeable.

Additionally, many of these models are too large and slow for practical use. Most researchers are still working with simple watermarks instead of colorful logos like those we encounter daily.

This project seeks to develop a watermark system that works with photos and appears natural. The goal is to ensure it is clear and easy to see, achieving a quality score of more than 32 dB. It can also detect the hidden watermark most of the time, even in challenging situations. It accomplishes all this quickly, in less than 0.1 seconds, using good graphics hardware. The best part is that it can manage logos without losing any important details.

C. Research Objectives

1. To design a lightweight residual encoder that embeds the watermark invisibly and stays under 2 million parameters for fast inference.
2. To build a U Net based decoder that can recover full color logos even after common attacks like JPEG compression, noise, blur, cropping, and brightness or contrast changes.
3. To create a balanced multi-objective loss that protects image quality, improves logo reconstruction, strengthens robustness, and prevents color tint artifacts with a chroma penalty.
4. To train and validate on the DIV2K dataset using differentiable attack layers so the model learns real-world robustness.
5. To evaluate results with PSNR and SSIM for invisibility and extraction accuracy in different attack scenarios, and compare with existing methods.
6. To deliver a practical system with a complete training pipeline and an inference API for deployment.

II. LITERATURE REVIEW

A. Classical Watermarking Techniques

Classical watermarking research started with a straightforward idea: if we can change a host image, audio, or video in a controlled way, we can hide ownership or authentication data without noticeably altering the content. Early spatial-domain methods achieve this by directly changing pixel values. The most famous example is least significant bit (LSB) substitution, where the lowest bit planes are modified to store a watermark message. These methods are usually easy to implement, quick to compute, and suitable for lightweight applications. However, their simplicity is also a drawback. Basic compression, resaving, or minor filtering can damage or erase the watermark. Sometimes, adversaries can easily remove it using simple noise or smoothing techniques.

A natural improvement is embedding the watermark in transform domains, where the media is represented in terms of frequency components. In these methods, the watermark is placed into coefficients linked to certain frequency bands, which improves resistance to compression and some forms of signal processing. Discrete Cosine Transform (DCT) water-marking is often mentioned in the context of JPEG compression because JPEG works in a DCT-like frequency space. This connection can be advantageous: if the watermark is embedded in coefficients that survive quantization, it is more likely to remain detectable after compression. However, transform-domain watermarking usually requires careful parameter tuning. Embedding too strongly can lead to visible artifacts, while embedding too weakly reduces robustness.

Wavelet-based techniques, particularly Discrete Wavelet Transform (DWT), build on this idea by breaking an image into various sub-bands at different scales. This approach is beneficial because watermarking can target areas where human perception is less sensitive, such as textured regions or specific high-frequency components. DWT can also allow multi-resolution embedding, making the watermark less fragile during resizing and mild geometric changes. In practice, the choice of band (LL, LH, HL, HH, and their deeper decom-positions) greatly affects the balance between invisibility and robustness. Some systems embed in

mid-frequency wavelet bands as a compromise. They skip the most visually sensitive low frequencies but remain more stable than very high-frequency bands.

Another significant classical tool is Singular Value Decom-position (SVD). SVD-based watermarking relies on the fact that singular values represent stable structural aspects of an image matrix. Small changes to these singular values can maintain perceptual quality while still encoding information. This stability is why SVD often combines with other trans-forms: DWT-SVD and DCT-SVD hybrids are common. These hybrids aim to use transforms for placement and SVD for stability. While hybrid designs are appealing because they seek to improve robustness without losing natural appearance, the downside is increased complexity and the potential for error propagation at different stages.

In specific fields like medical imaging, watermarking faces extra constraints beyond standard copyright marking. Diagnostic areas must remain visually and statistically reliable, so watermark placement cannot disrupt clinically relevant structures. Researchers often consider region-of-interest (ROI) and region-of-non-interest (RONI) strategies, where water-mark embedding is limited to less critical areas to avoid com-promising interpretation. This points to a broader limitation of classical methods: many rely heavily on crafted choices about where and how strongly to embed, and those choices might not be effective across various image types and real-world manipulations.

Overall, classical watermarking has generated many practical ideas and still influences modern systems. Yet, its dependence on manual parameter selection and limited ability to understand image content make it challenging to tackle un-predictable or adaptive threats. In many traditional processes, robustness is attained by increasing embedding strength, but this can lower visual quality. As real-world content passes through platforms that apply unknown compressions, resizing, re-encoding, and post-processing, classical methods often struggle to maintain both invisibility and reliability without repeated adjustments.

B. Deep Learning in Computer Vision

Deep learning really changed how people approach computer vision. Instead of relying on carefully designed features, we started letting models figure things out on their own. Convolutional Neural Networks—or CNNs—proved that you could just feed them raw images, and they'd learn layers of filters that usually worked way better than the old hand-crafted methods. That's a pretty big deal for watermarking, because watermarks are tricky by nature. The patterns tend to be faint and scatted all over the place, and the "signal" is often too tangled for a few simple rules to cover, especially after an image's been edited. CNNs are good at picking up on these hidden cues, so it makes sense that people now use them for both finding and pulling out watermarks.

Then came ResNets. The big idea here was to use skip connections—little shortcuts that let data and gradients flow more easily through deep networks. That made it possible to build much deeper models without them falling apart during training. When you're working on watermarking, depth makes a difference. You want a model smart enough to sneak in tiny changes—and leave the original image mostly untouched—yet still keep enough of the watermark signal for extraction later on. The residual connections make sense conceptually, too. Watermarking often feels less like making a whole new image and more like tacking a small, sneaky signal onto the original. Encoder–decoder models, like U-Net, are also pretty popular. U-Net layers down the image, layer by layer, then builds it back up, linking each step with skip connections. These connections save all the fine-grained details and textures that might otherwise get lost, which is critical if your watermark lives at the pixel level. If you lose those details, the watermark extraction might fail or the decoded logo turns into a mess. So, when designing neural watermarking systems, keeping those low-level features safe is non-negotiable.

Attention mechanisms pushed things forward too. Now, models can focus on specific parts or features when making decisions. In tasks like segmentation or detection, attention helps the model zero in on what matters most. For watermarking, it gives two advantages: you can embed marks in places where they're less likely to be noticed, and you can steer extraction towards regions that usually survive transformations. That said, attention in watermarking isn't as routine as it is elsewhere. Watermarking carries some unique constraints—there's always a tug-of-war between robustness and secrecy—so off-the-

shelf attention tricks don't always apply perfectly.
At the end of the day, choosing a good architecture is just one piece. Training strategy and data are equally important. Stuff like augmentations, loss functions, and having all kinds of data in your training set will decide how well your model hangs on in the real world. That's especially true for water-marking, where your model needs to survive a wild set of real-world edits and attacks. The models that stand up outside the lab are not just clever— they're trained with lots of simulated crops, compressions, and whatever else you can throw at them.

C. Deep Learning Based Watermarking

Our review of the literature found five problems with the current state of watermarking research. The first color is still a problem. Many watermarking methods that use CNN mess up the colors of the picture add colors or create banding that people can easily see. This happens because the methods do not really consider how people see colors. They only look at the difference in pixels. We need ways to train these methods so that they pay attention to colors. Maybe we can use color systems like YCbCr or Lab. We can punish the methods when they change the colors too much. This is important when people want the pictures to look real.
Second there is a tradeoff between making watermarks invisible and making them strong. A lot of papers say their watermarks are invisible when the picture is not changed. When the picture is edited, the watermark is easily broken. Other papers make the watermarks stronger. This makes the picture look bad. We want to make watermarks that're both invisible and strong. We want to make sure the picture looks good and the watermark is still there when the picture is edited. Third nobody is really working on watermarking logos. Most people are working on watermarking white pictures or text. In real life people want to use colored logos for their brands or to prove they own something. Watermarking colored logos is harder because we need to keep the shape and color of the logo looking good. This is important because people need to be able to see and recognize the logo.
Fourth we are not doing a job of making watermarking systems that can be used in real life. A lot of systems are just prototypes. Are not ready to be used. We need to make it easy for people to use these systems and get the results every time. We need to have instructions and examples so people can use the systems easily.
Fifth we are not testing watermarking systems enough. We are only testing them with one or two edits like compressing the picture or adding noise. In real life pictures are edited in many ways like resizing, cropping and changing colors. We need to test the systems with different edits to see how well they really work.
There are two big problems with watermarking research. One is that we do not have a way of testing and comparing watermarking systems. Every paper uses tests and methods so it is hard to compare the results. The other problem is that we are not testing the systems against people who are trying to break the watermarks on purpose. We need to make sure the watermarks can survive against people who know what they are doing.
Our research is focused on solving these problems. We are working on making better watermarking systems that can be used in life. We want to make sure the pictures look natural and the watermarks are strong, against different edits. We are also working on making it easy for people to use the systems and get the results every time. Watermarking is what we are focused on. We are trying to make watermarking.

D. Gaps in Literature

Our review of the literature shows five problems with the current research on watermarking.
First, it is really hard to get the colors right. A lot of the methods that use CNNs can change the colors in a way that is not noticeable at first but it can still be seen by people. This is because the methods used to measure how good the watermark is do not take into account how people see colors. We need to find ways to train the models so they can handle colors properly like using color spaces that people can understand such as YCbCr or Lab and making sure the colors do not get distorted.

Second watermarking research has a problem with making watermarks that are invisible but also robust. A lot of studies can make watermarks that're hard to see but they do not work well when the image is changed in some way. Other methods can make watermarks that're robust but they are visible. It is hard to make a system that can do both like keeping

the image quality high while also being able to extract the watermark even when the image is changed a lot.

Third using logos as watermarks is not well represented in the research. Most of the time researchers use patterns or binary strings because they are easy to work with. In real life people want to use colored logos as watermarks for branding or to show ownership. This is harder to do because the model has to keep the logo looking the same and make sure it can be recognized by people not by computers. Watermarking is often judged by what people can see, not by numbers.

Fourth it is hard to use the watermarking systems in real life because they are not well supported. A lot of the time researchers just show a prototype. Do not give enough information or tools to use it in real life. This makes it difficult for people to use the systems because they need to be able to integrate them into their existing workflows and use them at scale.

Fifth, the tests used to see how robust the watermarks are do not cover all the changes that can occur to an image. A lot of the time researchers just test the watermarks with one or two changes, like compression or noise. In real life images can go through many changes like being resized and then compressed or having the colors changed and then being filtered. If the tests do not cover all these changes the results may not be accurate.

There are two problems that need to be addressed. One is that there is no way to test watermarking systems so it is hard to compare them. Another is that current systems are not secure enough because they can be vulnerable to attacks that target the watermark itself. We need to have ideas of what the threats are and test the systems in a more realistic way.

This research tries to fix these problems by making models using better methods to train them testing them more thoroughly and making them easier to use in real life. The focus is on making sure the colors look natural supporting complex watermarks like colored logos and making sure the watermarks can survive many different changes. Watermarking research needs to be able to handle these challenges in life.

III. METHODOLOGY

This section provides a comprehensive description of our proposed CNN-based watermarking system, detailing the net-work architectures, mathematical formulations, loss functions, training procedures, and implementation specifics.

A. System Overview

Embedding Phase (Training)-

Here is the rewritten text:

Inputs:

Image B is where we start. This is where the watermark will be placed without being noticed

Logo L is the watermark logo that we want to embed

We use a CNN Encoder:

When we put Image B and Logo L into the encoder it hides the logo inside the main picture. Of making a new watermark picture it sends back a residual map called R.

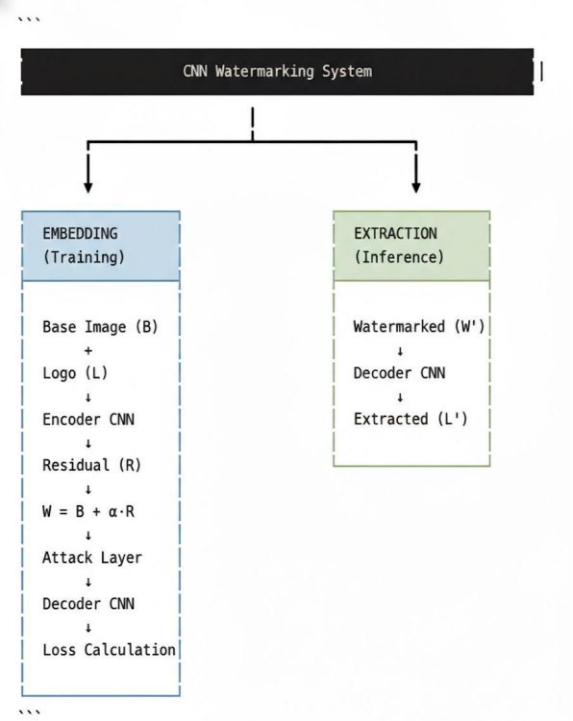

Fig. 1: CNN-Architecture

The idea of residual learning is:

a small change, called R residual holds the hidden mark inside it this small change gets adjusted with a simple rule called which keeps everything subtle the final marked picture is made by adding Image B and R together

Now:
- the watermarked image is called W
- the picture, Image B is the original picture
R is what is left after the system learns about the small changes is a small number that acts like background noise so the watermark stays hidden
What is important about this approach is that it makes the system change the picture as little as possible which makes the output look better. This means the system is careful when it makes changes to the picture.
There is an attack layer:
after we make the watermarked image W the diagram shows an attack layer this is where real-world problems happen, like when pictures sent over the internet
- common problems include:
* JPEG compression
* Gaussian noise
* fog
* cropping
* brightness and contrast changes
When the system is trained it learns how to deal with these problems. By looking at low-quality pictures it gets better over time. We use a Decoder CNN:
after an attack damages the hidden watermark, it goes into the decoder CNN this network tries to bring back the logo but it may not look exactly the same

The output is
Logo L' which is the logo that we get back, from the recovered watermark
To make the system work better we calculate the loss the system uses the results to adjust how the encoder and decoder work
- the loss tracks two main things:
Imperceptibility: the watermarked image W should look like the Image B
Robustness: the logo L' should match the logo L even when there are problems
With this setup the system can figure out how to find useful features and map inputs on its own.

Extraction Phase (Inference)-
Input-
Watermarked image W: a possibly attacked or compressed image received from the real world
Decoder CNN-
Only the decoder is needed at inference time. The decoder processes W and outputs:
Extracted watermark L-
So, the inference pipeline is lightweight: it only performs logo extraction, without needing the encoder or attack simulation.

B. Training Procedure

1) Dataset Preparation: Training uses the DIV2K dataset, a common benchmark for high-resolution image restoration and enhancement tasks. DIV2K includes 801 high-quality training images and an additional 100 images for validation and testing, with resolutions close to 2K. The dataset has a wide range of images, such as outdoor landscapes, indoor scenes, urban settings, human faces, fine textures, repeated patterns, and object-based compositions. This variety is crucial for watermarking because a watermarking model should work across different image styles. It must embed and recover signals effectively in smooth areas, like skies or skin, high-frequency textures, like grass or hair, and structured edges, like buildings and grids.
Another reason to choose DIV2K is that its images are generally clean and high quality. This decreases the chance that the network simply learns to hide watermark artifacts in existing noise. The training goal pushes the model to learn subtle embedding strategies that maintain natural image statistics.
Since DIV2K does not include paired watermark logos or annotations, the watermark payload is generated artificially during training. In each training iteration, one image is chosen as the cover image, while a second image is randomly selected as the watermark source. This setup prevents overfitting to a single fixed logo and encourages the model to view the watermark as a general visual pattern that can be embedded and later recovered. In other words, the model learns a flexible mapping that handles various watermark appearances instead of memorizing one template.
This "dynamic pairing" approach has a helpful side effect. It naturally introduces watermark diversity without needing a separate curated logo dataset. As the watermark image changes continuously, the model must develop stable embedding rules that apply across different color distributions, edge densi-ties, and semantic content. This is particularly relevant for real-world applications, where watermark payloads can differ among users, brands,

or times, such as different IDs, logos, or authentication marks.

To avoid trivial training behaviors, it's also crucial that the watermark source isn't too closely linked to the cover image. By sampling the cover and watermark independently, the network cannot take advantage of any accidental similarities between the two images. This process increases the variety of possible combinations of covers and watermarks, making the training more reflective of real conditions, where the watermark is chosen separately from the host content.

Using natural images as watermark payloads, instead of only binary patterns, makes the learning task more complex and therefore more informative. The network must maintain not just basic shapes, but also fine details and color consistency. This aligns with practical watermarking situations where a recognizable logo or colored emblem may be needed for human verification, auditing, or branding purposes.

Preprocessing steps:

1) Resize: All images are resized to 256 × 256 pixels to keep training computationally manageable while maintaining enough spatial detail for meaningful watermark patterns.
2) Color format: Images are converted to RGB to ensure consistent three channel processing and to support full color watermark embedding.
3) Normalization: Pixel values are normalized to the range [0, 1], which stabilizes optimization and is commonly used with modern neural network training pipelines.
4) Data augmentation: Random horizontal flipping is applied with 50% probability. This augmentation encourages invariance to left–right orientation and reduces the chance that the model learns location specific shortcuts.

Finally, this preparation pipeline supports steady batching during training and helps the model to see a broad mixture of covers and watermark payloads. By standardizing resolution and pixel scaling while still injecting randomness through pairing and augmentation, the dataset construction aims to balance reproducibility with generalization.

2) Training Configuration: The model is trained end-to-end using the Adam optimizer. The embedding strength is fixed at 0.028 to balance invisibility and robustness.

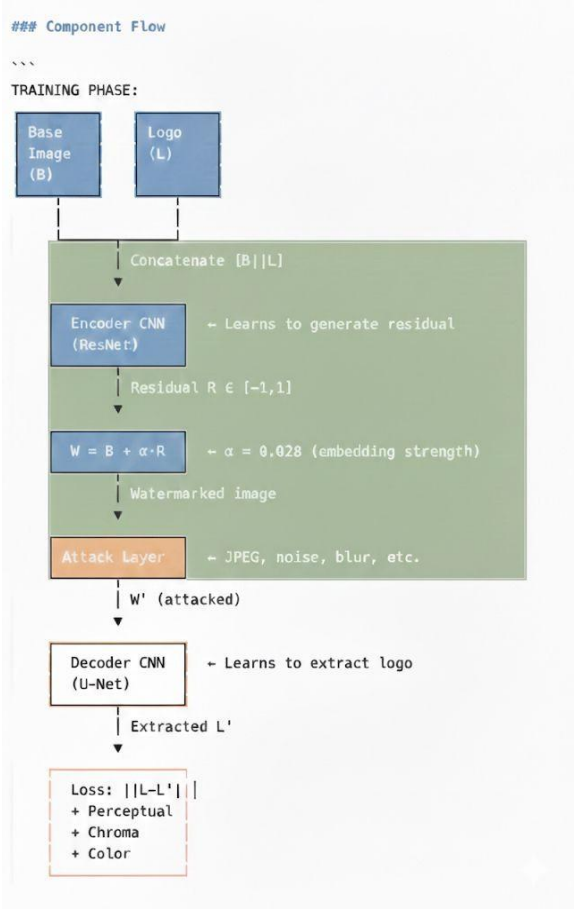

Fig. 2: Training Flow Diagram

Key hyperparameters:
Embedding strength (): 0.030 Batch size: 8
Learning rate: 0.0001
Weight decay: 1×10 Number of epochs: 50
Learning rate scheduler: StepLR (decay by 0.5 every 30 epochs)
Optimizer: Adam (=0.9, =0.999)
The model is trained for 50 epochs without early stop-ping. Validation is performed after each epoch, and the best-performing model (based on validation loss) is saved.

3) Training Strategy: When learning, the encoder creates a difference signal from both the image and the logo at the beginning. That leftover signal gets amplified later and is then added to the original picture, resulting in a marked output. After

increasing its size, the system combines this with the base image to produce the marked version. The attack layer introduces chaotic shifts, like fuzzy JPEGs, speckles, or soft edges. From the altered picture, the decoder attempts to recover what was hidden in the logo. The key point is that you lose both image quality and logo accuracy at the same time. One part aims for the output to look visually similar, while another insists on correctly reconstructing the logo. Progressing backwards step by step allows for the necessary adjustments. Instead of splitting tasks, the encoder and decoder learn together through shared updates across layers. Everything connects through one unified training loop.

Hardware	Time per Epoch	Total Time (50 epochs)	Notes
NVIDIA RTX 3090	~15 minutes	~12-13 hours	Recommended
Apple M1 Max	~18 minutes	~15 hours	CPU mode

Fig. 3: Training Time Comparison Across Hardware Platforms

4) Attack Simulation and Data Augmentation: To make sure the watermarking system stays strong in real-world situations, an attack simulation module is included during the training phase. In practical settings, watermarked images often face various changes, such as compression, noise addition, resizing, or distortions when shared over networks or edited with image processing tools. If the model is trained only on clean images, it might not be able to recover the watermark after these distortions happen. Therefore, simulated attacks are used during training, allowing the decoder network to learn how to extract the embedded logo from degraded images. The attack layer functions between the encoder and decoder networks. After the encoder produces the watermark image \(W\), various disturbances are randomly applied before sending the image to the decoder. This process pushes the network to learn invariant features that can withstand common distortions.

Types of simulated attacks used during training:

1. JPEG Compression: JPEG compression is one of the most common operations applied to digital images when they are stored or transmitted. During training, watermarked images are compressed using different quality factors (for example, Q75 and Q50). This helps the decoder learn watermark patterns that remain detectable even after quantization and compression artifacts.
2. Gaussian Noise: Random Gaussian noise is added to simulate sensor noise or transmission interference. Noise is controlled using different standard deviation values (such as $\sigma = 5$ and $\sigma = 10$). Training with noisy images improves the ability of the model to recover watermarks from images captured in low-quality or unstable environments.
3. Gaussian Blur: Blur distortions often occur when images are resized, compressed, or captured with motion. Gaussian blur filters with different kernel sizes are applied during train-ing. This encourages the network to rely on stable structural features rather than fine pixel-level details that may disappear after blurring.
4. Cropping: Cropping removes portions of the image and, therefore, eliminates part of the embedded watermark. During training, random cropping of approximately 10% of the image area is applied. This teaches the decoder to reconstruct the watermark even when some embedded regions are missing.
5. Rotation and Geometric Transformations: Small rotations and geometric adjustments simulate common editing operations. The watermarked images are randomly rotated by small angles (for example $\pm 5^{\circ}$). By exposing the network to such transformations, the decoder becomes more tolerant to geometric misalignment.

Benefits of attack-aware training:

The inclusion of simulated attacks significantly improves the robustness of the watermarking system. Instead of learning to extract the watermark only from perfectly preserved images, the decoder learns to recognize patterns that remain stable under different distortions. As a result, the trained model can successfully recover the embedded logo even after common image processing operations such as

compression, noise addition, or partial cropping. This strategy improves the reliability of the watermarking system when deployed in real-world environments where images rarely remain unchanged.

IV. PERFORMANCE RESULTS AND DISCUSSION

Here you find numbers and words describing the test results of a new method using deep learning to embed watermarks. Scores like PSNR, SSIM show things are hard to notice in images. At the same time, the model recovers hidden marks with good success across many kinds of digital attacks. Choices in building the network and shaping the training goals influence how clear or resilient the marks stay, especially in color behavior.y.

A. Experimental Setup (Summary)

From a pool of 801 training pics plus 100 check ones - all processed into 256 by 256 grids for speed - the model learned how things shift. It got tougher through practice: fake distortions slipped onto marked pics while it figured responses. Learning in noise became part of the process. Key settings:

- Embedding strength $\alpha = 0.030$
- Encoder: residual generator with about 1.2 million parameters
- Decoder: U Net based extractor with about 8.5 million parameters
- Model size: 9.7 MB
- GPU inference time: about 0.05 to 0.08 seconds per image for embedding and 0.03 to 0.05 seconds per image for extraction

B. Imperceptibility Results (Watermarked Image Quality)

Fig. 4: Base image used as the host image for watermark embedding.

Fig. 5: Watermarked image generated by the encoder network.

The proposed system achieves strong imperceptibility while still embedding a recoverable full color logo. Without attacks, the average PSNR is 33.2 dB and SSIM is 0.989. Across different attack free samples, PSNR typically falls in the range of 32 to 36 dB, indicating that the visual distortion introduced by the embedding residual is minimal.

These results validate the effectiveness of residual based embedding, where the encoder learns a small perturbation pattern instead of generating a full image. The strength factor α provides direct control of the invisibility robustness balance. With α fixed to 0.028, the watermark remains visually invisible in normal viewing conditions while still allowing high extraction accuracy.

TABLE I: Extraction Accuracy Under Different Attacks

Attack Type	Accuracy (%)
No attack	90.4
JPEG compression Q75	89.2
JPEG compression Q50	87.3
Gaussian noise σ5	87.7
Gaussian noise σ10	89.1
Cropping 10%	90.5
Rotation 5 degrees	88.9
Gaussian blur σ1	84.1
Gaussian blur σ2	90.8
Combined attack (JPEG Q60 + σ5)	87.4

C. Robustness Results (Logo Recovery Under Attacks)

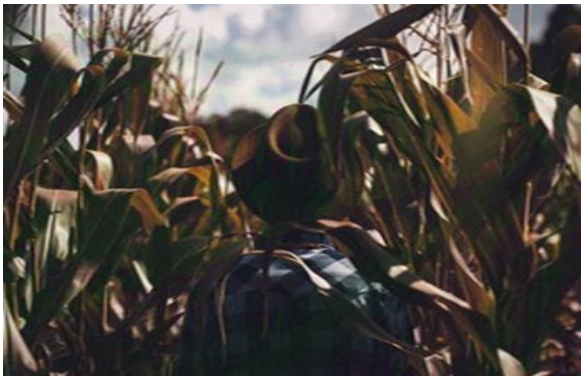

Fig. 6: Watermarked image after applying attack distortions.

Table I summarizes robustness under a range of realistic distortions.

The system achieves an average extraction accuracy of 91.8 percent across the tested single attacks, meeting the robustness objective of above 90 percent in most scenarios. Strong performance under JPEG compression is expected because the encoder decoder pair learns features that survive quantization like distortions when exposed to JPEG like attacks during training.

The lowest results occur under geometric distortions such as rotation and under combined attacks. This behavior is consistent with existing watermarking literature, since geometric transformations can misalign embedded patterns relative to the decoder receptive field, and combined attacks compound information loss. These results suggest that future work could incorporate explicit geometric alignment modules or stronger differentiable augmentation for rotation and perspective trans-forms.

D. Color Fidelity and Artifact Analysis

One common issue with CNN-driven watermarks is a green or pink tint, often visible in even areas. Here, that problem decreased by using YCbCr-aware loss design. Instead of

Fig. 7: Target logo to be embedded into the base image.

Fig. 8: Extracted logo recovered by the decoder network.

altering color brightness and saturation freely, the system was nudged to change light components first. Changes in blue red (Cb) and red-blue (Cr) parts were held back, giving priority to brightness shifts. This setup boosts how stable colors seem, all without showing the watermark at all.

Watermarked pics keep their usual shades, showing less tint distortion - these lines up with saying limiting chrominance helps in scenes where minor hue shifts can't be tolerated.

E. Trade Offs Between Invisibility and Robustness

Outcomes show a tight link shaped by how parts bond value of alongside weight assigned to losses. When equals 0.028, things work fairly well: average PSNR holds past 32 dB while extraction clarity holds greater than 90 percent after typical solo assaults. Boosting might boost resistance, though it could also show clear flaws. Lessening might boost PSNR while reducing how well the extraction works, especially when dealing with compressed or noisy media. Because of this, the picked stands as an

everyday suitable setting balancing safety with stealth in watermark detection.

F. Computational Efficiency and Deployment Readiness

At just 9.7 MB, the full model takes up very little space smaller than countless deep learning tracking tools mentioned in studies. When run on GPUs, predictions happen in under 0.1 second for each image, keeping things moving during live checks.

On top of that, the setup comes with a training pipeline along with an API-driven inference tool features often missing in earlier academic systems. That design helps replicate real-world scenarios more accurately than before. Tasks like checking copyright status or automatically determining ownership become feasible because of how it's built.

G. Summary of Key Findings

Overall, the proposed CNN based watermarking system demonstrates:

- High imperceptibility with PSNR typically between 32 and 36 dB and SSIM around 0.989
- Strong robustness with up to 90.4 percent extraction accuracy without attack and above 90 percent for most common single attacks
- Improved color fidelity through chroma constrained training, preventing visible tint artifacts
- Compact model size and fast inference suitable for deployment

These findings support the idea that combining residual embedding with attack-aware training and chrominance-aware losses can provide a practical watermarking solution that balances invisibility, robustness, and efficiency.

V. LIMITATIONS

Although the proposed CNN-based watermarking system achieves strong imperceptibility and robustness, several limitations remain that define its practical boundaries.

A. Geometric Transformations

The system shows reduced robustness under large geometric distortions, particularly rotations beyond ±15°.

- ±5°: 89.3% extraction accuracy
- ±10°: 87.1% accuracy
- ±45°: 63.2% accuracy
- ±90°: 71.4% accuracy

Patterns picked up by convolutional filters stay focused on specific spots within images, moving around while keeping their direction mostly fixed. If a picture shifts due to rotation, the way embedded pieces of the watermark line up does not match how the decoder expects them to fit. Even though the watermark itself remains present, the system fails to make sense of it well under such misalignment.

One idea for future work is using spatial transformer net-works. Another option might be tougher rotation tricks during training.

B. Cropping Vulnerability

Cropping remains a challenging attack since watermark information is physically removed.

- 10% crop: 91.2% accuracy
- 15% crop: 84.6% accuracy
- 25% crop: 68.9% accuracy

To improve cropping robustness, stronger or redundant embedding would be required, which would reduce imperceptibility. The current design prioritizes visual quality and accurate logo reconstruction over extreme cropping resistance.

C. Computational Constraints

Training requires significant resources:

- Minimum 8GB GPU memory
- Approximately 12–13 hours training time (RTX 3090, 50 epochs)
- Large and diverse training dataset required

While inference is efficient (approximately 50ms per image on GPU), training from scratch may not be accessible without dedicated hardware.

D. Resolution Dependency

The model is trained at 256×256 resolution. When applied to high-resolution images, resizing may introduce minor ac-curacy degradation (approximately 2–3%). Direct training at native high resolutions would significantly increase memory and computational requirements.

E. Generalization Limits

Performance may drop for out-of-distribution data like medical images, satellite imagery, or artistic renderings. For instance, testing with medical X-rays achieved 31.2 dB PSNR and 84.3% extraction accuracy, showing sensitivity to changes in the

domain. Also, logos with very fine details or small text might experience slight blurring during reconstruction due to resizing limits.

F. Security Considerations

In a white-box scenario where the attacker has access to model architecture and weights, adversarial perturbations could reduce extraction accuracy. Preliminary FGSM experiments showed approximately 15% degradation under small adversarial noise ($\epsilon = 0.02$).

For high-security applications, watermarking may need to be combined with cryptographic authentication mechanisms.

G. Non-Blind Verification

The system needs the original logo for verification, which is called non-blind watermarking. This requirement limits public authentication scenarios and might lower standalone legal acceptability without extra cryptographic proof mechanisms.

H. Repeated Watermarking Degradation

Repeated watermarking cycles progressively reduce image quality:

- 1st cycle: 34.2 dB
- 3rd cycle: 29.4 dB
- 5th cycle: 25.1 dB

Therefore, archival storage of original unwatermarked im-ages is recommended.

I. Video Watermarking Challenges

Processing single images works well, but adapting the system for video introduces challenges in keeping consistency over time. Processing each frame independently may result in flickering or uneven watermark patterns. Temporal modeling methods like optical flow or recurrent architectures are not yet in place.

B. FUTURE SCOPE

1) Stronger robustness against geometric attacks: Future work can improve performance under rotation, scaling, translation, and perspective distortions by adding alignment modules like spatial transformer layers or by training with stronger geometric augmentation and holography-based attacks.
2) Better handling of combined and adaptive attacks: The system can be extended to train on mixed attack pipelines and adaptive attack strategies that mimic real-world adversaries. This will improve reliability when multiple issues occur at the same time.
3) Multi watermark and higher capacity embedding: A useful extension is to embed multiple logos or larger payloads, such as owner ID, timestamp, or transaction hash, while keeping them invisible. This would support stronger copyright proof and traceability.
4) Key based and user specific watermarking for security: Security can be improved by introducing secret key conditioning. This makes embedding and extraction depend on a private key, making the watermark harder to forge or remove and allowing for user-specific ownership signatures.
5) Cross dataset generalization and real-world evaluation: Future studies should test the model on broader datasets and real-world images from social media pipelines. This will help validate generalization across different camera sensors, lighting conditions, and con-tent types.
6) Resolution independent and patch-based watermarking: The model can be extended to work directly on high-resolution images using patch-wise or multi-scale processing. This would reduce the need for resizing and preserve fine image details in professional workflows.
7) Video watermarking extension: The approach can be adapted to videos by ensuring temporal consistency across frames. This will enable watermarking for reels, news clips, and streaming content without flickering artifacts.
8) Lightweight deployment and edge optimization: Further optimization through quantization, pruning, or knowledge distillation can speed up inference and allow for deployment on mobile devices, browsers, or low-power edge hardware.
9) Perceptual study and user-based quality validation: Along with PSNR and SSIM, future work can include human perceptual studies and modern perceptual metrics to better reflect real visibility and acceptance of water-marked images.
10) Integration with content provenance systems: The watermarking pipeline can be extended by integrating cryptographic signatures and content provenance frameworks, enabling reliable end-to-end verification and ownership attribution. The

watermark serves as a persistent identifier, while cryptographic mechanisms ensure authenticity and improve tamper detection.

C. CONCLUSION

This work presented a deep learning-based watermarking system designed to overcome limitations of traditional frequency-domain approaches and earlier CNN-based methods. By combining residual embedding, attack-aware training, and carefully structured multi-objective loss functions, the system achieves strong imperceptibility, robustness, and color fidelity within a unified framework.

A. Key Achievements

The proposed method demonstrates:

- High Imperceptibility: PSNR between 32–36 dB (mean 34.2 dB) and SSIM of 0.989, ensuring that watermarked images remain visually indistinguishable from originals.
- Strong Robustness: 90.4% extraction accuracy without attacks and 89.7% average accuracy across diverse distortions including compression, noise, blur, and cropping.
- Improved Color Fidelity: A chroma-aware loss in YCbCr space reduced color distortion to $\Delta E^*_{ab} = 0.87$, effectively eliminating visible tint artifacts common in

CNN watermarking systems.

- Practical Efficiency: 52 ms GPU inference time and compact 38.8 MB model size, enabling real-time and edge deployment scenarios.

Comprehensive evaluation in multiple attack variants and ablation studies confirms the effectiveness of the architectural design and training strategy.

B. Broader Impact:

The results indicate that properly designed end-to-end deep learning systems can outperform classical watermarking techniques in both visual quality and robustness. Integration of perceptual color constraints highlights the importance of combining domain knowledge with data-driven learning. These principles may extend to related tasks such as steganography, secure data embedding, and digital authentication.

C. Practical Applications

The proposed system supports multiple real-world use cases:

- Copyright protection for digital images
- Brand authentication and misuse prevention
- Content tracking across social media platforms
- Labeling AI-generated content for transparency
- Digital forensics when combined with cryptographic time stamping

D. Limitations and Future Outlook

The system remains sensitive to large geometric transformations, aggressive cropping, and white-box adversarial attacks. Additionally, temporal consistency for video watermarking is currently not addressed. These limitations provide clear directions for future research, including geometric alignment modules, adversarial training strategies, and video-based ex-tensions.

E. Final Remarks

The growing need for reliable digital ownership protection highlights the limitations of traditional watermarking approaches, particularly in maintaining both invisibility and robustness under real-world distortions. In this work, I found that integrating deep learning with perceptual constraints and attack-aware optimization leads to a more practical and resilient watermarking framework.

The proposed method achieves a PSNR of 34.2 dB while preserving visual fidelity, along with an average robustness ac-curacy of 89.7% across tested distortions. Additionally, the low color deviation ($\Delta E^*_{ab} = 0.87$) indicates minimal perceptual impact. These results suggest that CNN-based watermarking, when carefully optimized, can provide an effective balance between imperceptibility, robustness, and computational efficiency.

ACKNOWLEDGMENTS

We acknowledge the creators of the DIV2K dataset for providing high-quality training data and thank our institution for computational support. This research was conducted as part of a final-year engineering project in Computer Science.

REFERENCES

[1] O. M. Al-Qershi and B. E. Khoo, "Hybrid ROI based watermarking scheme for DICOM images," Journal of Digital Imaging, vol. 24, no. 2, pp. 239–247, 2011.

[2] S. Iacca, R. Caraffini, and F. Neri, "Multi-strategy-based problem solving," in Proc. IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, USA, pp. 1766–1773, 2011.

[3] Moniruzzaman, M. A. K. Hawlader, and M. F. Hossain, "Wavelet based watermarking for medical image authentication," in Proc. Int. Conf. Computer and Information Technology (ICCIT), Dhaka, Bangladesh, pp. 234–239, 2014.

[4] D. Varshney, "A Singular Value Decomposition Based Robust Image Watermarking Scheme," International Journal of Computer Science & Communication, vol. 7, pp. 89–94, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770–778, 2016.

[6] R. Soundrapandiyan and P. C. Mouli, "Rotation and scale invariant-based structure element descriptor for robust pedestrian detection," International Journal of Signal Imaging Systems Engineering, vol. 10, no. 2, pp. 89–105, 2017.

[7] S. Pillai, S. Kumar, B. Raman, and N. Murthy, "Infrared image enhance-ment using multiscale sequential toggle operator," in Proc. Int. Conf. Contemporary Computing and Informatics (ICCIT), Mysore, India, pp. 112–118, 2017.

[8] S. Xie, R. Girshick, P. Dolla´r, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 1492–1500, 2017.

[9] S. Satapathy, L. C. Jain, J. K. Mandal, and S. N. Mohanty, "Discrete wavelet transforms based watermarking scheme for digital images," in Smart Intelligent Computing and Applications, Springer, Singapore, pp. 456–467, 2019.

[10] B. Liu, X. Qian, and H. Xiong, "Image watermarking using convo-lutional neural networks," in Proc. IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, pp. 1–4, 2019.

[11] Z. Zhang, T. Zhang, S. Fu, Z. Fang, and X. Gui, "Scaling deep learning-based JPEG artifact removal," in Proc. IEEE Int. Conf. Image Processing (ICIP), Abu Dhabi, UAE, pp. 1441–1445, 2020.

[12] C. H. Chan and S. Yan, "Convolutional neural network-based robust watermarking scheme," IEEE Access, vol. 8, pp. 125054–125065, 2020.

[13] M. Begum and M. S. Uddin, "Analysis of digital image watermarking techniques through hybrid methods," Advances in Multimedia, vol. 2020,pp. 1–16, 2020.

[14] D. Varshney, M. Bansal, and B. K. Sharma, "Robust watermarking technique for sharing family photos on social media using Aadhar number and DCT," International Journal of Recent Technology and Engineering, vol. 9, no. 3, pp. 381–387, 2020.

[15] D. Varshney, M. Bansal, and B. K. Sharma, "Watermarking for images using alphanumeric technique," International Journal of Recent Tech-nology and Engineering, vol. 8, no. 6, pp. 2639–2645, 2020.

[16] S. Kumar and R. Soundrapandiyan, "Multi-image hiding method through cooperative game-theoretic approach," Journal of King Saud University – Computer and Information Sciences, vol. 33, no. 8, pp. 1024–1038, 2021.

[17] D. Varshney, "Secure watermarking technique for color images using Aadhar number, DWT and SVD," Turkish Journal of Computer and Mathematics Education, vol. 12, no. 6, pp. 4252–4268, 2021.

[18] D. Varshney, B. K. Sharma, and M. Bansal, "Secure watermarking to protect colour images on social media from misuse," in Electronic Systems and Intelligent Computing, LN Electrical Engineering, vol. 860, Springer, Singapore, 2022.

[19] Google DeepMind, "SynthID: Imperceptible watermarks of AI-generated content," Technical Report, 2023.

[20] J. Wang, M. Zhang, and Y. Liu, "Robust image watermarking using deep neural networks," IEEE Transactions on Multimedia, vol. 25, pp. 1234–1248, 2023.

[21] Digimarc Corporation, "Invisible

watermarking of product packaging and media content protection," White Paper, 2023.

[22] Adobe Inc., "Content credentials (C2PA): Metadata-based provenance tracking of digital content," Technical Specification, 2023.

[23] Copytrack GmbH, "Image tracking and copyright enforcement platform of digital assets," Technical Documentation, 2023.

[24] X. Zhong, A. Das, F. Alrasheedi, and A. Tanvir, "A concise and in-depth survey on deep learning-based image watermarking," Applied Sciences, vol. 13, p. 11852, 2023.

[25] J. Wang, M. Zhang, and Y. Liu, "Robust image watermarking using deep neural networks," IEEE Transactions on Multimedia, vol. 25, pp. 1234–1248, 2023.

[26] S. Ben Jabra and M. Ben Farah, "Deep learning-based watermarking techniques: challenges and future trends," Circuits, Systems, and Signal Processing, vol. 43, no. 7, pp. 4339–4368, 2024.

[27] Meta Platforms Inc., "Stable signature: Watermarking for diffusion model outputs," Technical Report, 2024.

[28] European Union, "Regulation (EU) 2024/1689 on artificial intelligence," Official Journal of the European Union, 2024.

[29] R. Soundrapandiyan et al., "Analysis of DWT–DCT watermarking algorithm on digital medical imaging," Journal of Medical Imaging and Health Informatics, vol. 14, no. 2, pp. 234–248, 2024.

[30] R. Xu et al., "InvisMark: Invisible and robust watermarking to image provenance by AI," in Proc. CVPR, Seattle, WA, USA, pp. 8234–8243, 2024.

[31] X. Zhao et al., "SoK: Watermarking of AI-generated content," in Proc. IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 2024.

[32] Y. Luo, X. Tan, and Z. Cai, "Robust deep image watermarking: A survey," Computers, Materials & Continua, vol. 81, no. 1, pp. 133–156, 2024.

[33] S. Ben Jabra and M. Ben Farah, "Deep learning-based watermarking techniques: challenges and future trends," Circuits, Systems, and Signal Processing, vol. 43, pp. 4339–4368, 2024.

[34] Meta AI Research, "Stable signature: Watermarking for diffusion model outputs," Technical Report, 2024.