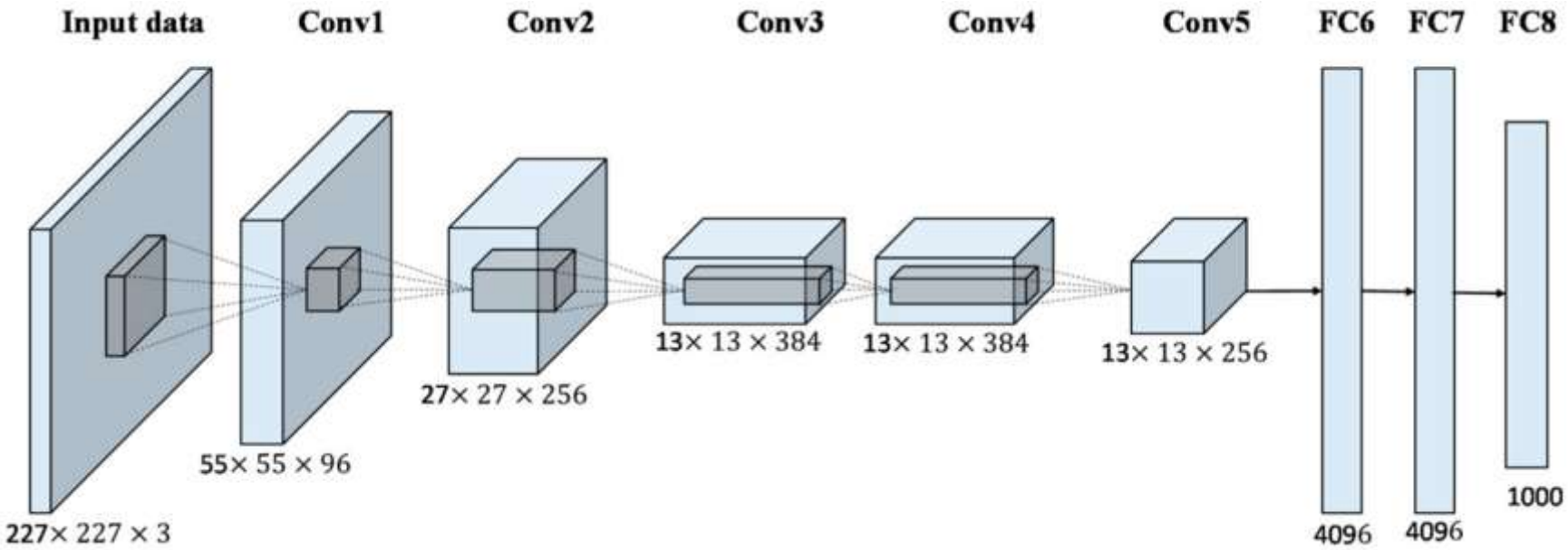
Information Hiding using Deep Generative Models

Rohit Gandikota
IIST batch 2015
Scientist -SC, NRSC
Under Guidance of Dr. Deepak Mishra
(CVVR Lab, IIST)

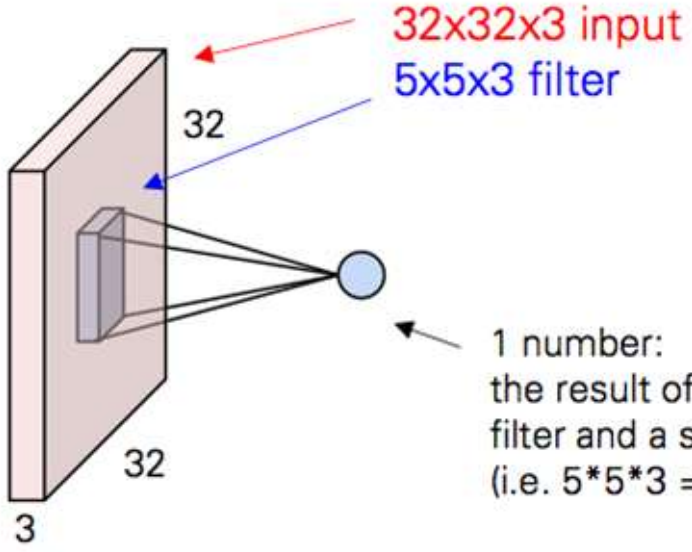
Indian Institute of Space science and Technology

Government of India
National Remote Sensing Centre
Indian Space Research Organisation
ISO 9001:2015

Convolutional Neural Network

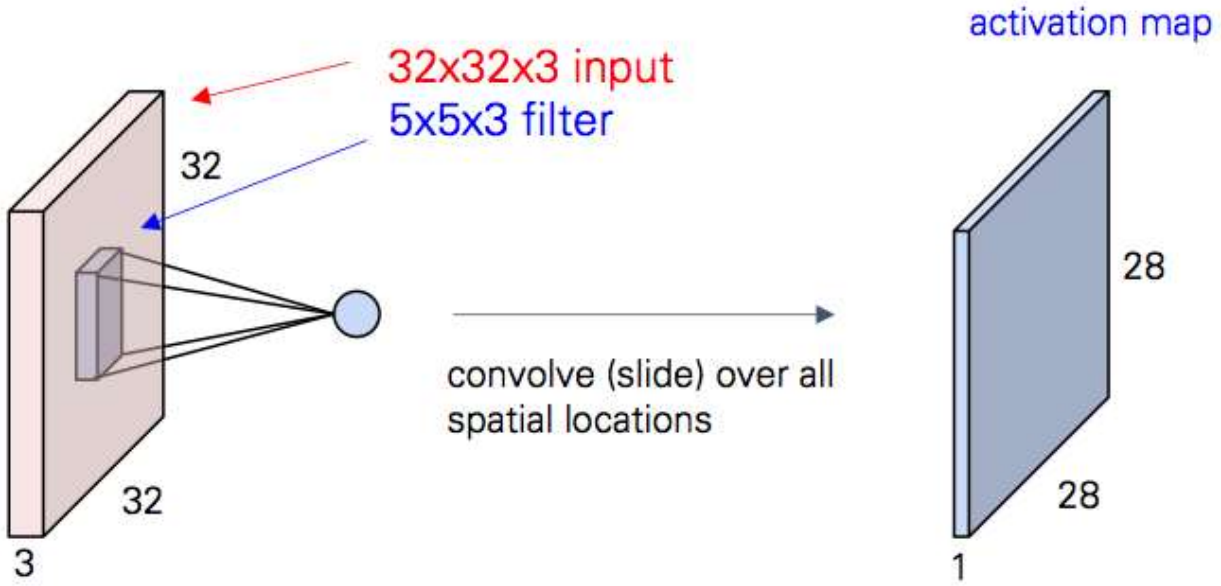


Convolutional Layer



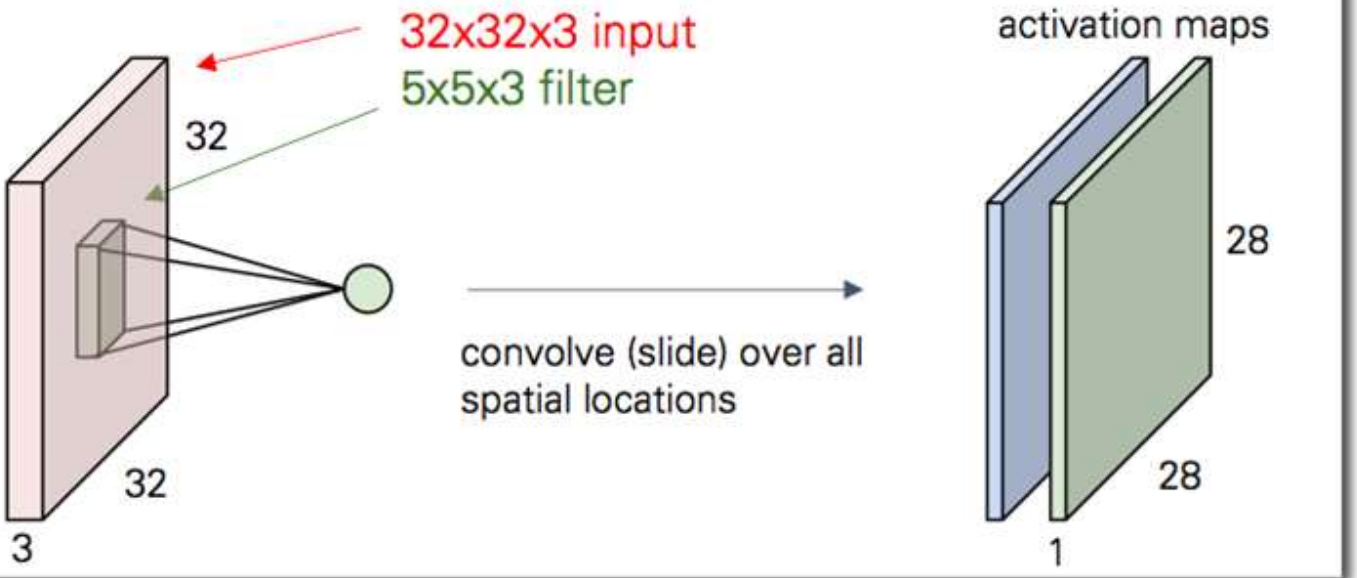
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the input
(i.e. 5*5*3 = 75-dimensional dot product + bias)

Convolutional Layer

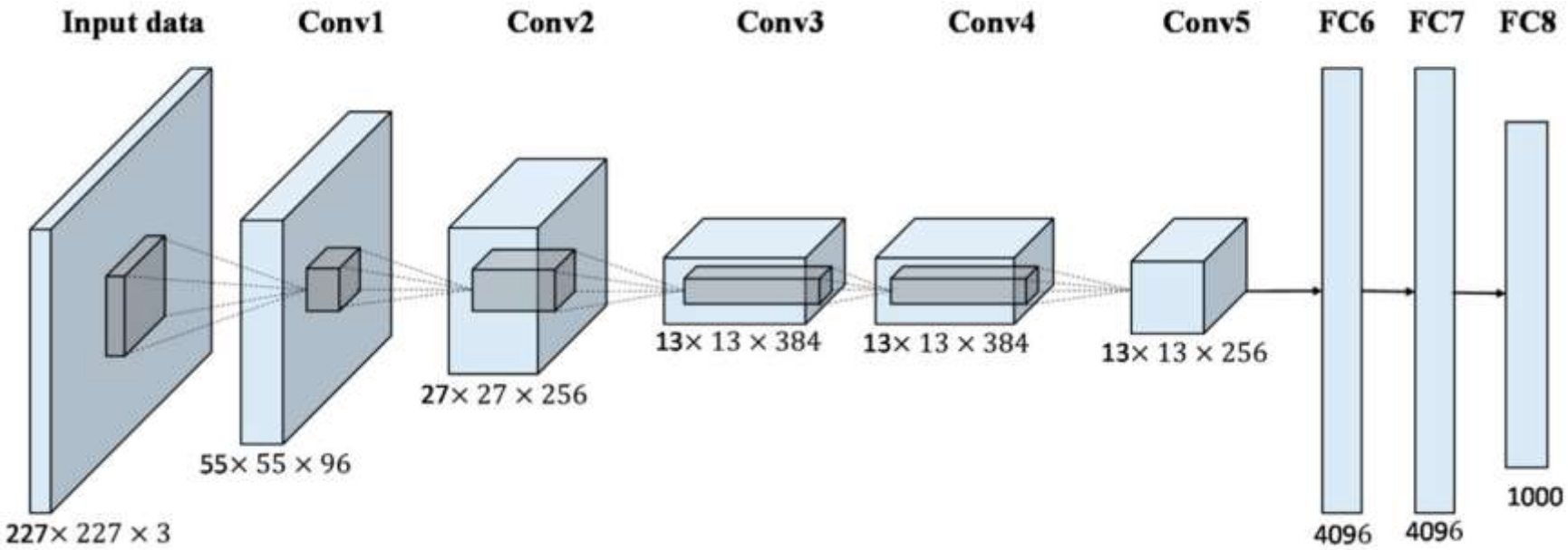


Convolutional Layer

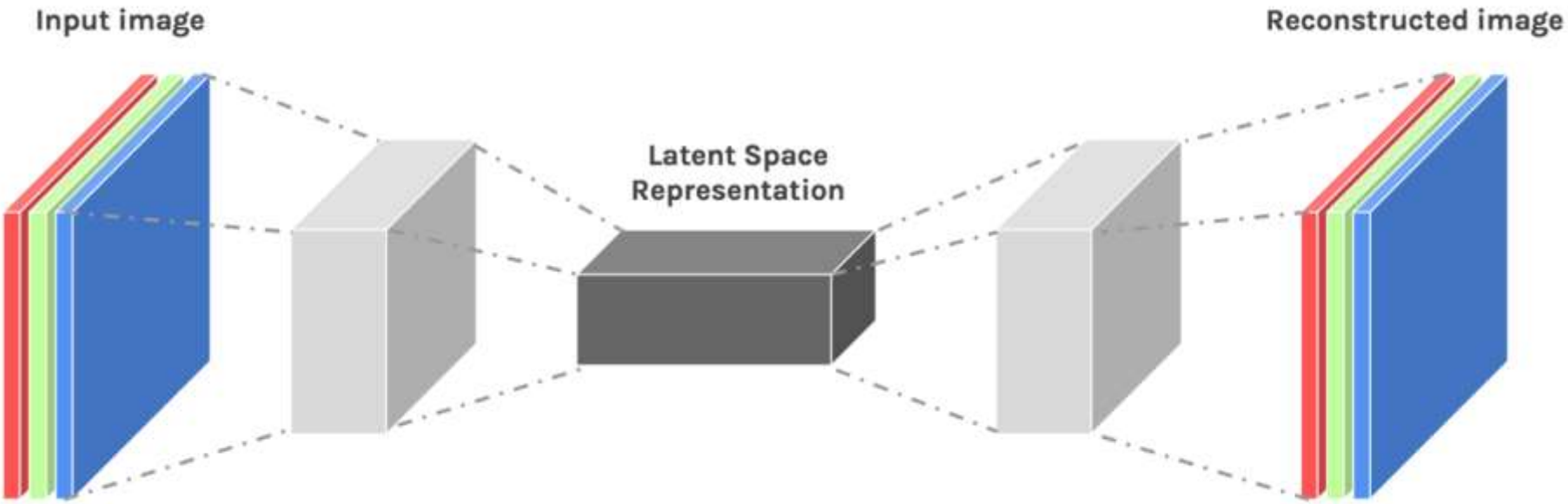
consider a second, green filter



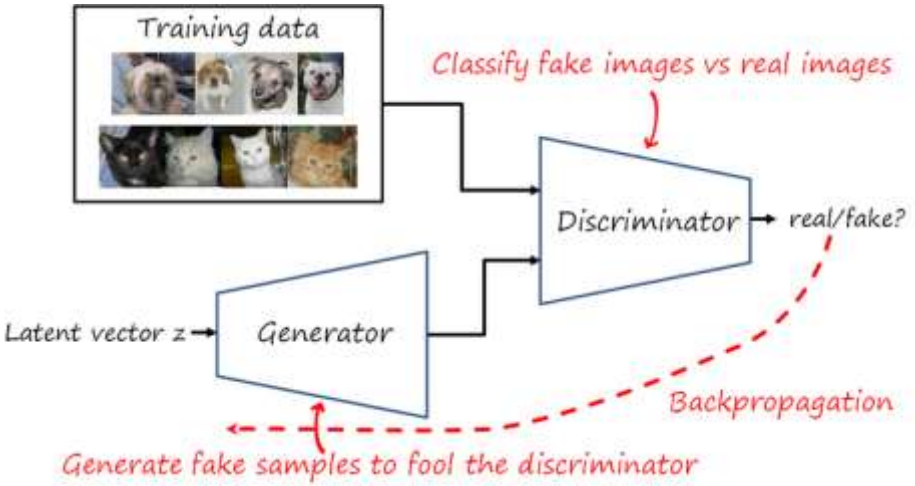
Classification Network (A classical variant of CNN)



Autoencoder (another variant of CNN)

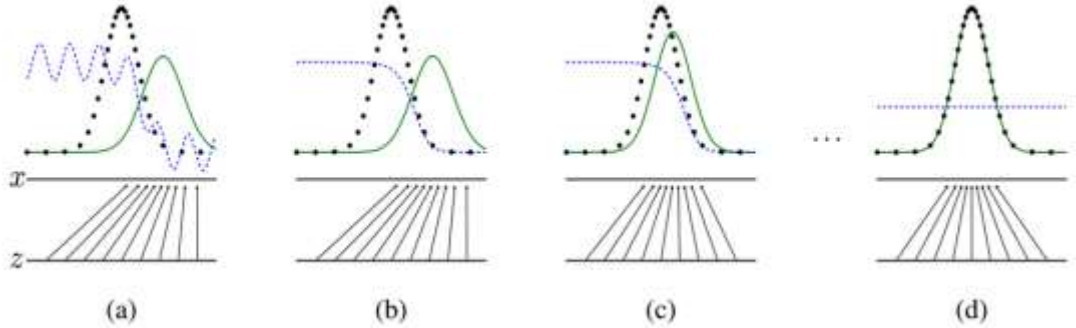


GANs : Generative Adversarial Networks



(Goodfellow et al, 2013 NIPS)

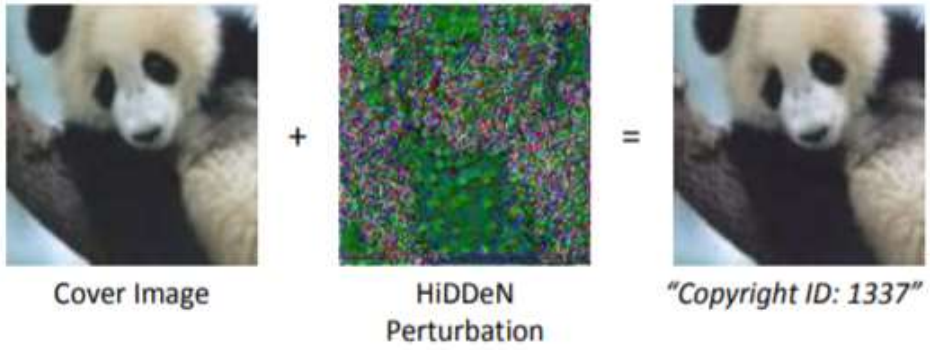
- Learning the data distribution in an adversarial way.
- No direct learning for generator.
- Gradients from discriminator to generator.
- Dying gradients (Mode collapse)



Data Hiding

Steganography

The goal is secret communication between a sender and a receiver, while an adversary person cannot tell if the image contains any message.

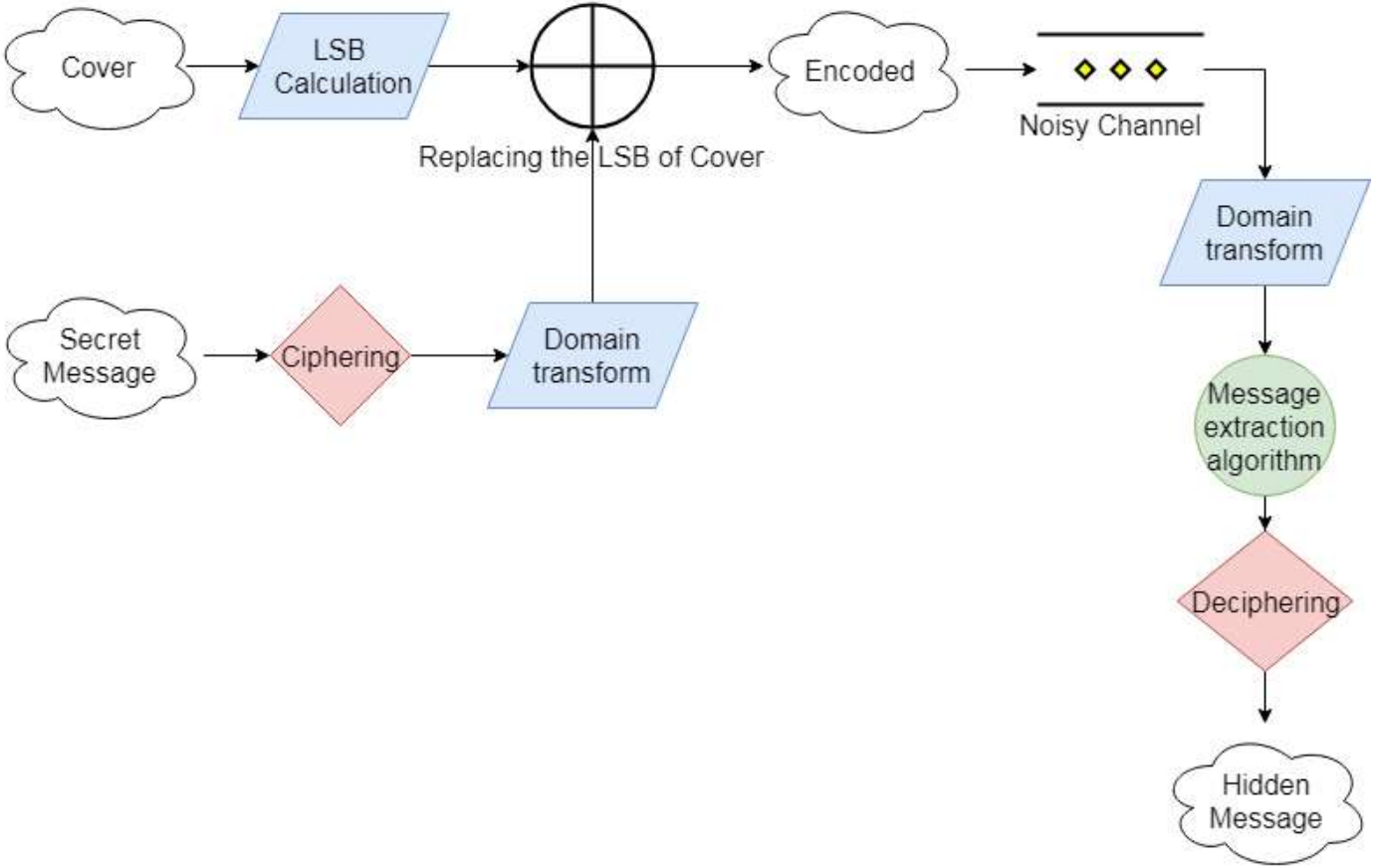


(Fei Fei Li et al, 2018 ECCV)

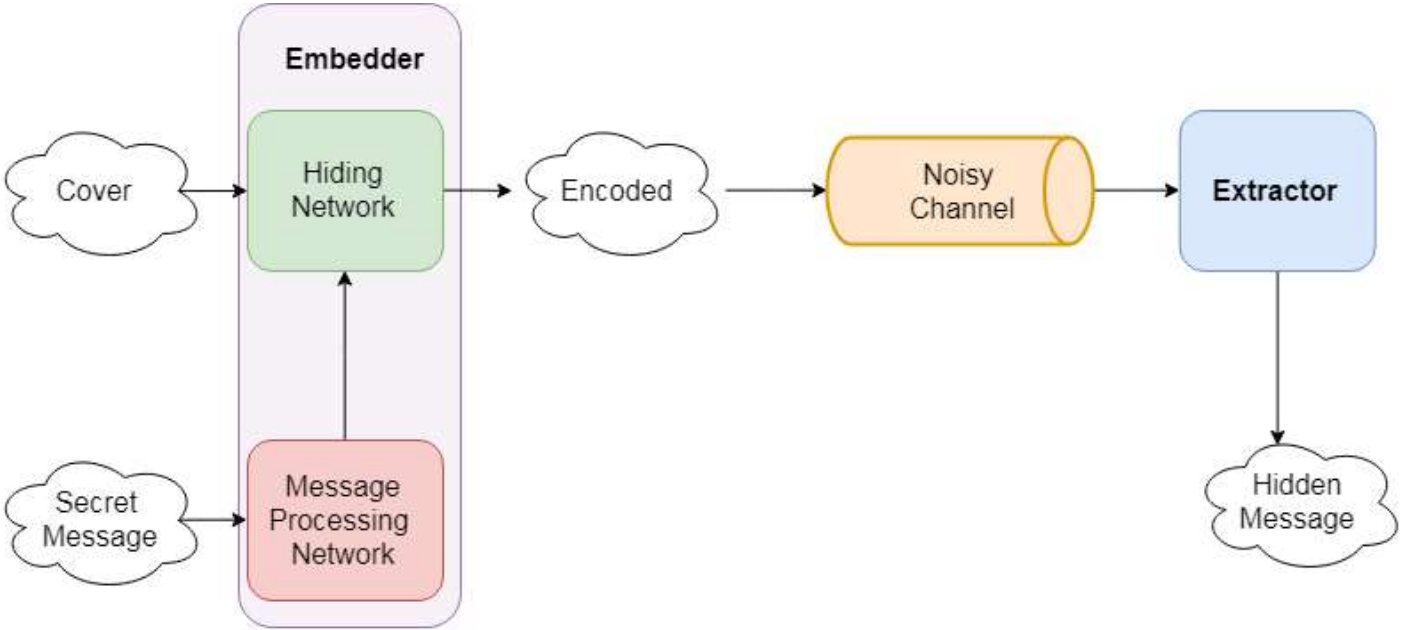
Watermarking

The goal is to encode information robustly. Even though an adversary distorts the image, the receiver can extract the message.

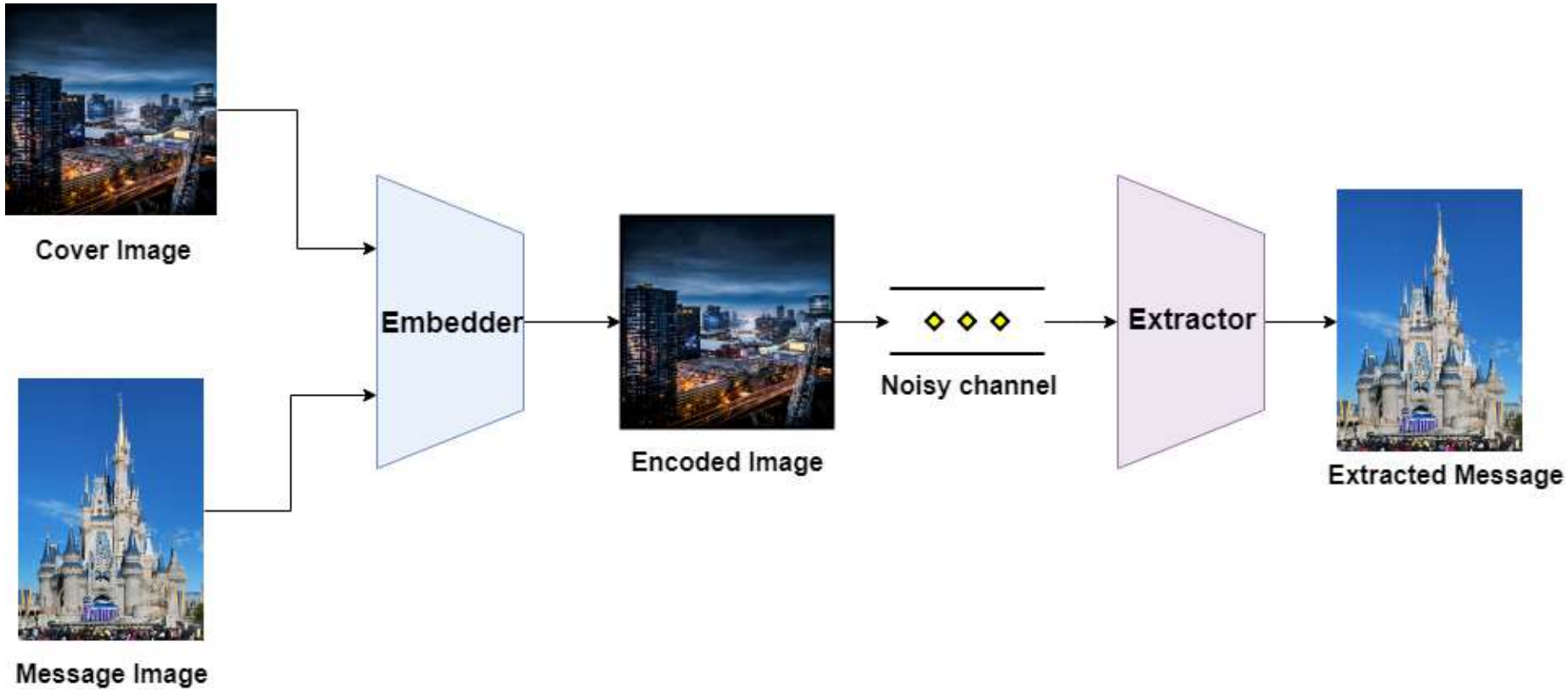
Traditional Data hiding



Data hiding using Deep Learning



Hiding Image in Image



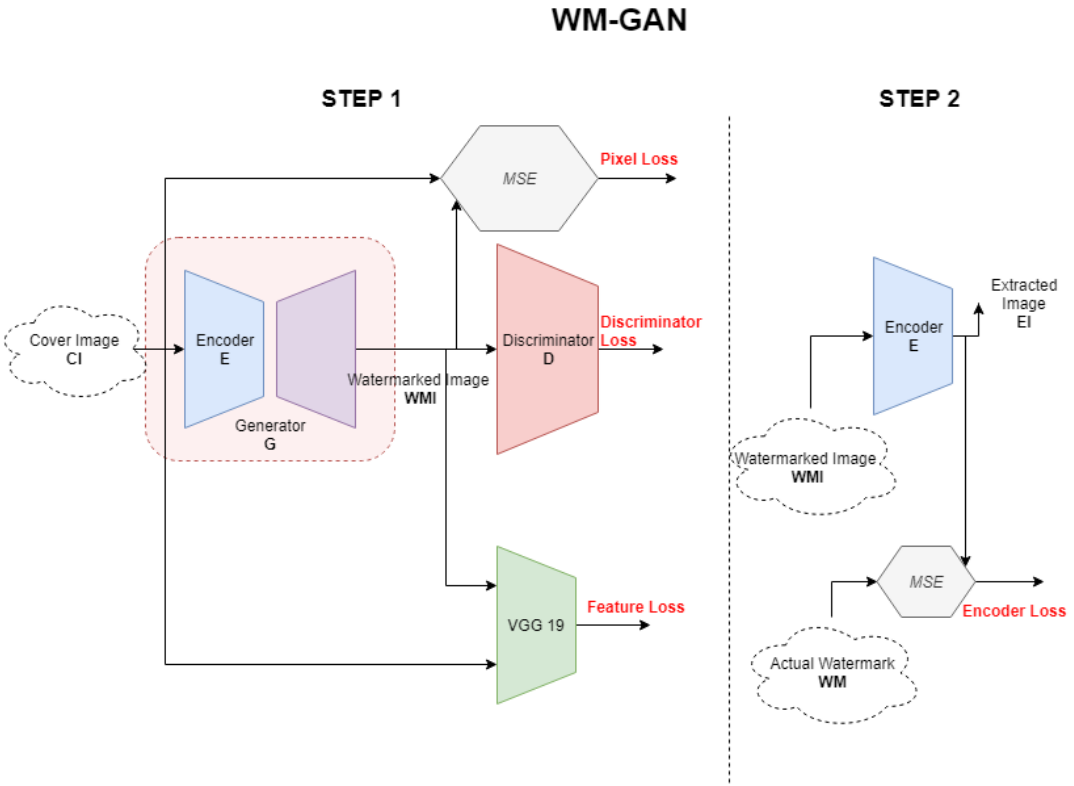
Watermarking

- Watermarking is typically used to identify ownership or copyright the data in the content creation industry.

- Usually requirement for a single watermark by a user. Therefore no need for a generic embedding.

- Since used majorly by content creation industry, invisibility of the watermark is as important as robustness towards attacks to remove the watermark.

Proposed Method



- Implicit 2 stage training of WM-GAN
    1. Train the generator **G,** to output image similar to input.
    1. Train the encoder of generator **E,** to output the watermark given the output of **G.**
- A cover image passed through **G** is “embedded” since, the encoder of the **G** has information about the watermark.
- The output of **E** is the embedded message as the input of stage 2 is output of stage 1 that is “encoded”.

Rohit G, Deepak Mishra, WM-GAN: Watermarking using Adversarial Network, Under peer review in pattern recognition letters

Training of WM-GAN: Stage 1

- First stage is to train the generator **G,** to output an image ***WMI*** similar to the input cover image ***CI.*** This embedding was done using adversarial loss, reconstruction pixel loss and feature loss.

$$D_{Loss} = log\mathbf{D}(CI) + log(1 - \mathbf{D}(WMI))$$

$$G_{Loss} = \frac{1}{cH \times cW} \sum_{i=1}^{cH} \sum_{j=1}^{cW} (WMI_{i,j} - CI_{i,j})^2$$

$$F_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (\chi(WMI)_{i,j} - \chi(CI)_{i,j})^2$$

Where $WMI = Generator_\theta(CI)$ *and* $\mathbf{D}$ *is Discriminator*

Train the generator with parameters θ such that:

$$\max_D \min_\theta \ \beta_1 G_{Loss} + \beta_2 F_{Loss} + \beta_3 D_{Loss}$$

Training of WM-GAN: Stage 2

- Second stage is to extract the watermark ***WM*** with the encoder **E** of the generator **G** from the watermarked image ***WMI***.

$$E_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (WM_{i,j}^{extracted} - WM_{i,j}^{actual})^2$$

$$Where\ WM^{extracted} = Extractor_{\lambda}(WMI)$$

Train the extractor with parameters λ such that:

$$\min_{\lambda} \beta_1 E_{Loss}$$

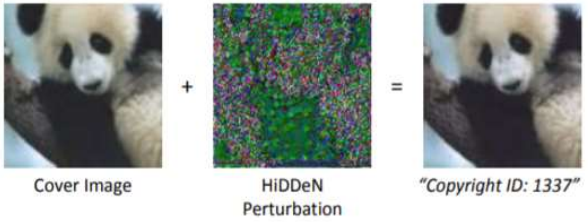
Contributions to the watermarking society

- To our knowledge, WM-GAN is the first end-to-end trainable model of Generative Adversarial Network solely dedicated for watermarking purpose.

- Proposing an extractor as a sub-model of the embedder and adding the adversarial loss in the extractor’s training.

- Had reduced the required number of networks, trainable parameters, training time and improved efficiency over the existing state-of-the-art methods.

Steganography

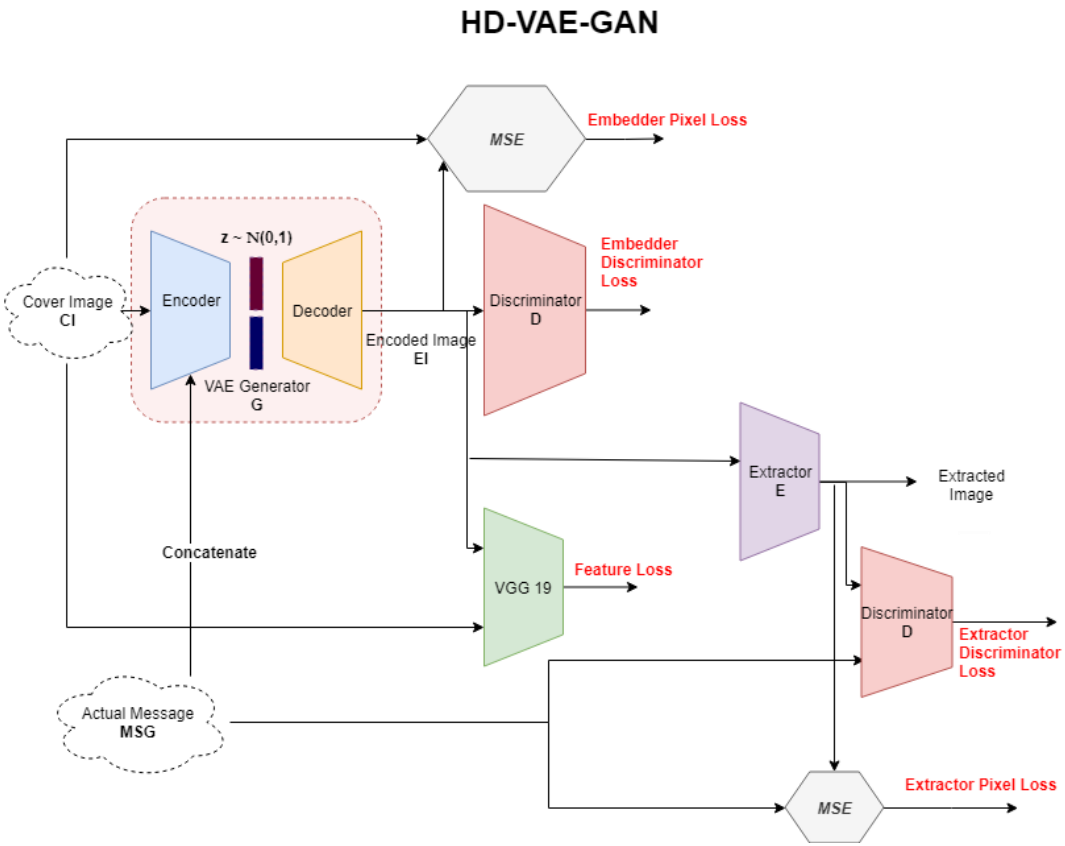


- Steganography is secret communication between a sender and a receiver, while an adversary person cannot tell if the data contains any message.

- Usually requirement for embedding multiple messages by a user. Therefore a need for generic embedding. (To embed any message into any cover)

- A possibility for an attacker to try and extract or destroy the message. Therefore, robustness is also a minor requirement.

Proposed Method



- Implicit 2 stage training of HD-VAE-GAN

    1. Train the VAE generator **G,** to output image similar to cover image given message and cover as inputs.

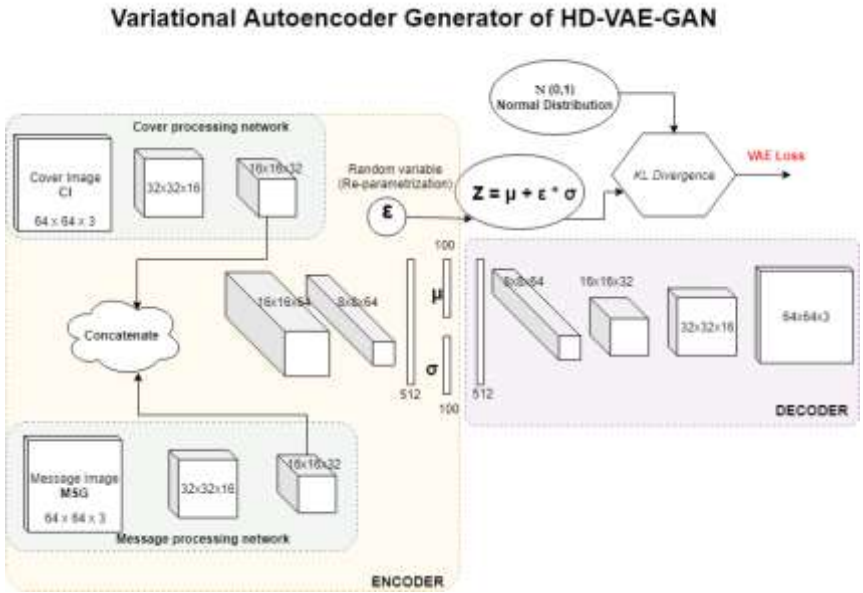
    1. Train the extractor **E,** to output the message given the output of **G.**

- The “encoding” is done by concatenation process in the encoder of the VAE.

G Rohit, D Mishra, HD-VAE-GAN: Hiding Data using Variational Autoencoder Generative Adversarial Network, submission to IEEE-SPL

Training of HD-VAE-GAN: Stage 1

- First stage is to train the generator **G,** to output an image ***EI*** similar to the cover image ***CI*** with message ***MSG*** and ***CI*** as inputs***.*** This embedding was done using variational loss, adversarial loss, reconstruction pixel loss and feature loss.



Training of HD-VAE-GAN: Stage 1

- First stage is to train the generator **G,** to output an image ***EI*** similar to the cover image ***CI*** with message ***MSG*** and ***CI*** as inputs***.*** This embedding was done using variational loss, adversarial loss, reconstruction pixel loss and feature loss.

$$V_{Loss} = KL((\mu + \epsilon * \sigma), \mathbb{N}(0,1))$$

Where μ and σ are outputs of $Encoder(CI, MSG)$

$$D_{Loss} = log\mathbf{D}(CI) + log(1 - \mathbf{D}(EI))$$

$$G_{Loss} = \frac{1}{cH \times cW} \sum_{i=1}^{cH} \sum_{j=1}^{cW} (EI_{i,j} - CI_{i,j})^2$$

$$F_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (\chi(EI)_{i,j} - \chi(CI)_{i,j})^2$$

Where $EI = Generator_\theta(CI)$ and $\mathbf{D}$ is Discriminator

Train the generator with parameters θ such that:

$$\max_{D} \min_{\theta} \beta_1 G_{Loss} + \beta_2 V_{Loss} + \beta_3 F_{Loss} + \beta_4 D_{Loss}$$

Training of HD-VAE-GAN: Stage 2

- Second stage is to extract the message image ***MSG*** with the extractor **E** of from the encoded image ***EI***.

$$E_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (MSG_{i,j}^{extracted} - MSG_{i,j}^{actual})^2$$

$$D_{Loss}^{extractor} = log\mathbf{D}(MSG^{actual}) + log(1 - \mathbf{D}(MSG^{extracted})$$

$$Where\ MSG^{extracted} = Extractor_{\lambda}(EI)$$

Train the extractor with parameters λ such that:

$$\max_{D} \min_{\lambda} \beta_5 E_{Loss} + \beta_6 D_{Loss}^{extractor}$$

$$Loss = \min_{E} \min_{G} \max_{D} \beta_1 G_{Loss} + \beta_2 V_{Loss} + \beta_3 F_{Loss} + \beta_4 D_{Loss} + \beta_5 E_{Loss} + \beta_6 D_{Loss}^{extractor}$$

Contributions to the steganography society

- To our knowledge, HD-VAE-GAN is the first end-to-end trainable model of Variational Autoencoder Generative Adversarial Network for hiding images in images.

- Proposing variational inference for the embedder and adding the adversarial loss in the extractor’s training.

- Improved the robustness for steganography compared to the state-of-the-art methods. Making our method more convincing to opt.

Experiment

- We have used the DIV2K dataset to train and evaluate our model. The dataset consists of 900 images (800 train, 100 test) to evaluate our models.

Table 3.1: Comparison of emdedding and extraction capacity without any attacks on DIV2K dataset

Method	SSIM	PSNR-CI (in dB)	PSNR-EX (in dB)
WM-GAN (N=50)	0.9875	36.02	23.04
WM-GAN (N=100)	0.9899	37.02	23.28
WM-GAN (N=200)	0.9930	38.12	24.01
HD-VAE-GAN (N=800)	0.9059	27.89	27.34
SteganoGAN Res (N=800)	0.9000	38.25	-
SteganoGAN Basic (N=800)	0.6700	24.62	-
DeepStego †	-	35.29	37.79
WM-GAN (N=50) †	0.9801	36.05	25.78

*N – Number of samples used in training

- HD-VAE-GAN has poor embedding but efficiency due to the variational inference, making the embedded images very close in distribution.

Experiment

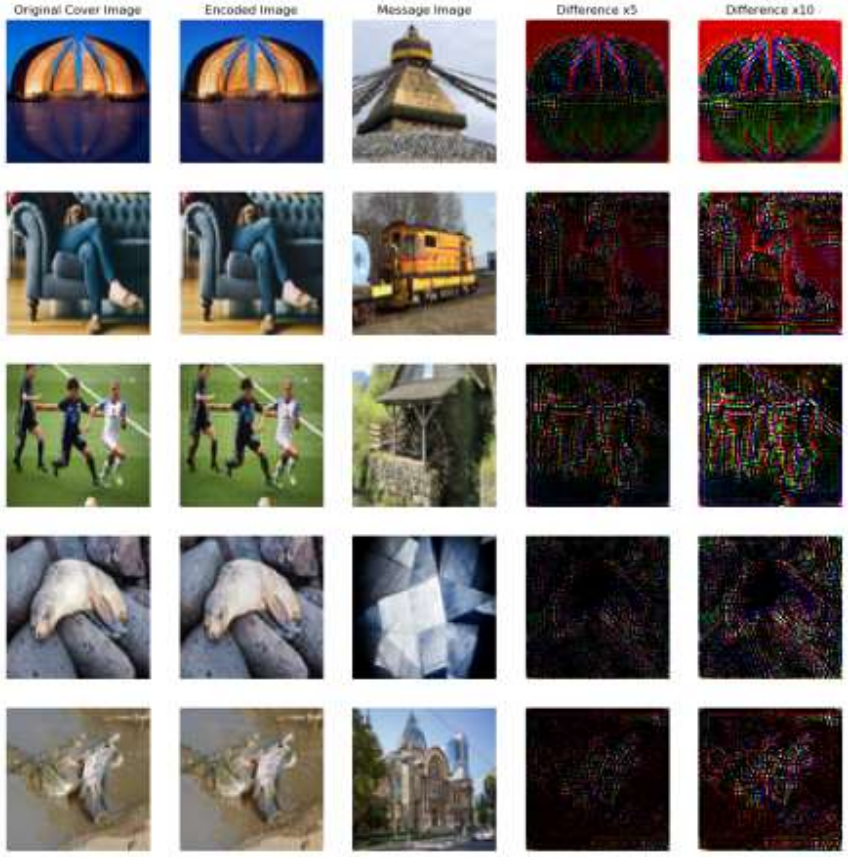
- The variational inference was a draw-back for the embedding process, however, this close distribution helped in a robust extraction.

Table 3.2: PSNR (in dB) of extracted watermarks from attacked encoded image before and after training on attacks

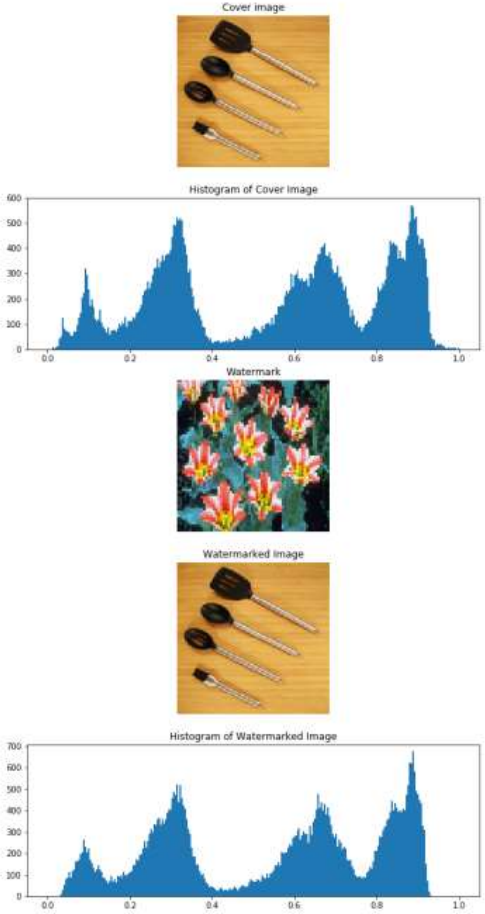
Simulated Attack	PSNR-EX before training		PSNR-EX after training	
	WM-GAN	HD-VAE-GAN	WM-GAN	HD-VAE-GAN
Gaussian Blur	16.55	20.77	36.96	40.41
Rotation 90°	7.48	9.57	32.49	35.57
Rotation 10°	7.62	6.68	31.14	37.51
Cropping 10%	8.34	12.13	32.03	36.08
Speckle	4.57	21.07	18.28	29.74
S&P Noise	9.88	16.15	33.30	39.83
Gaussian noise	7.01	15.90	20.84	26.86
Blur + Crop *	-	-	21.70	28.56
Gaussian + Rot90°†	-	-	19.42	24.74
Gaussian + Rot10°†	-	-	20.66	26.23
Crop +Rot 90°†	-	-	10.82	19.31
Crop +Rot 10°†	-	-	12.31	21.42

Analysis

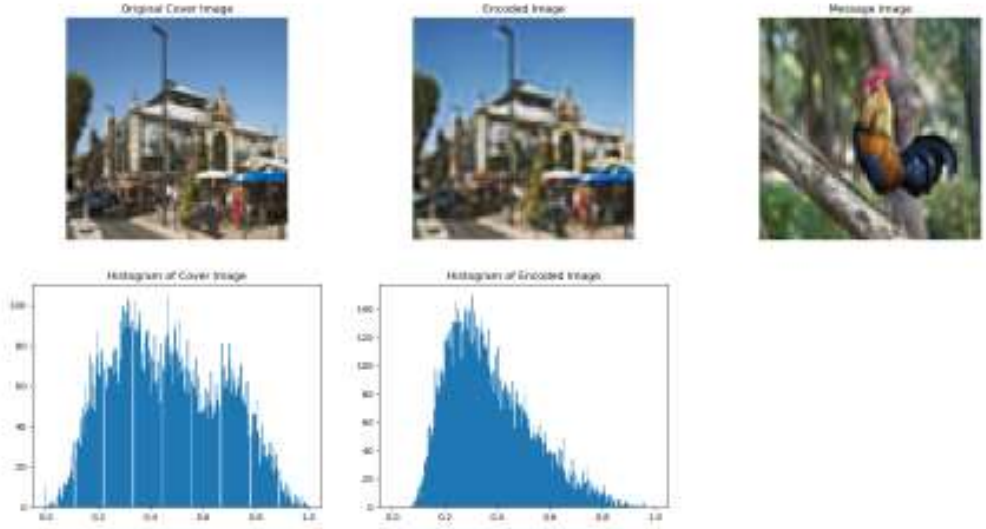
- Where is the encoding happening ?



Analysis



- Histogram analysis



Visual samples

Generated image

Actual watermark Image

Extracted watermark

Actual Watermark

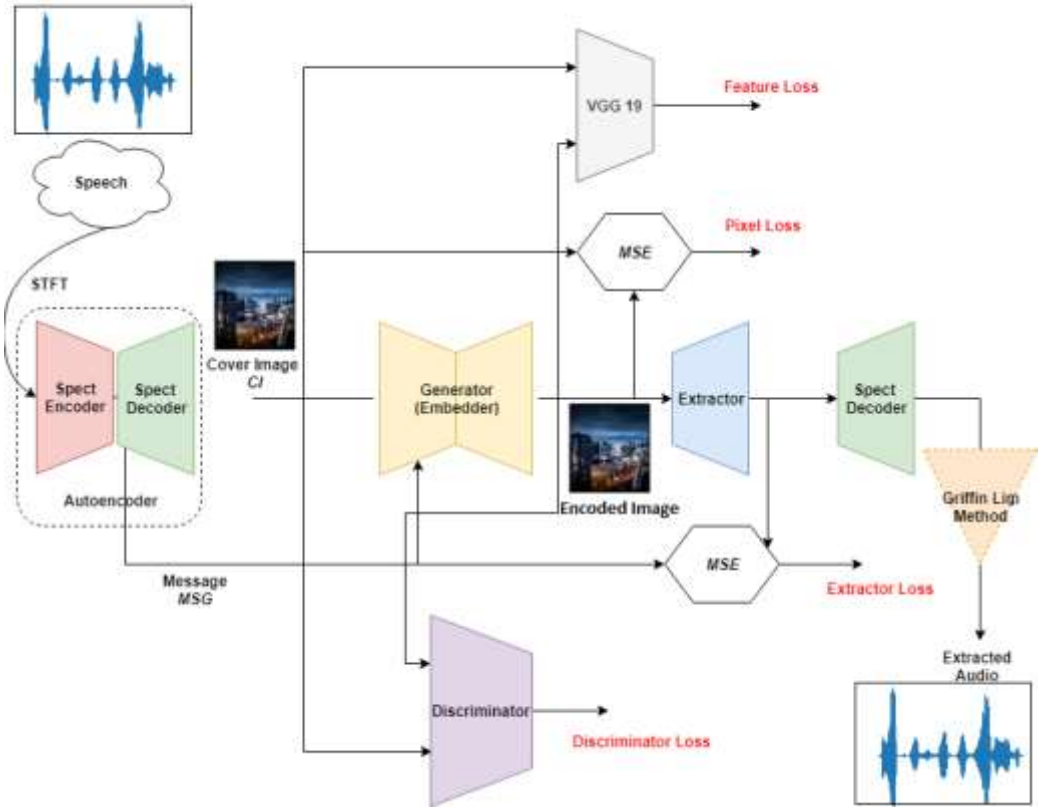
Generated Image

Actual Cover Image

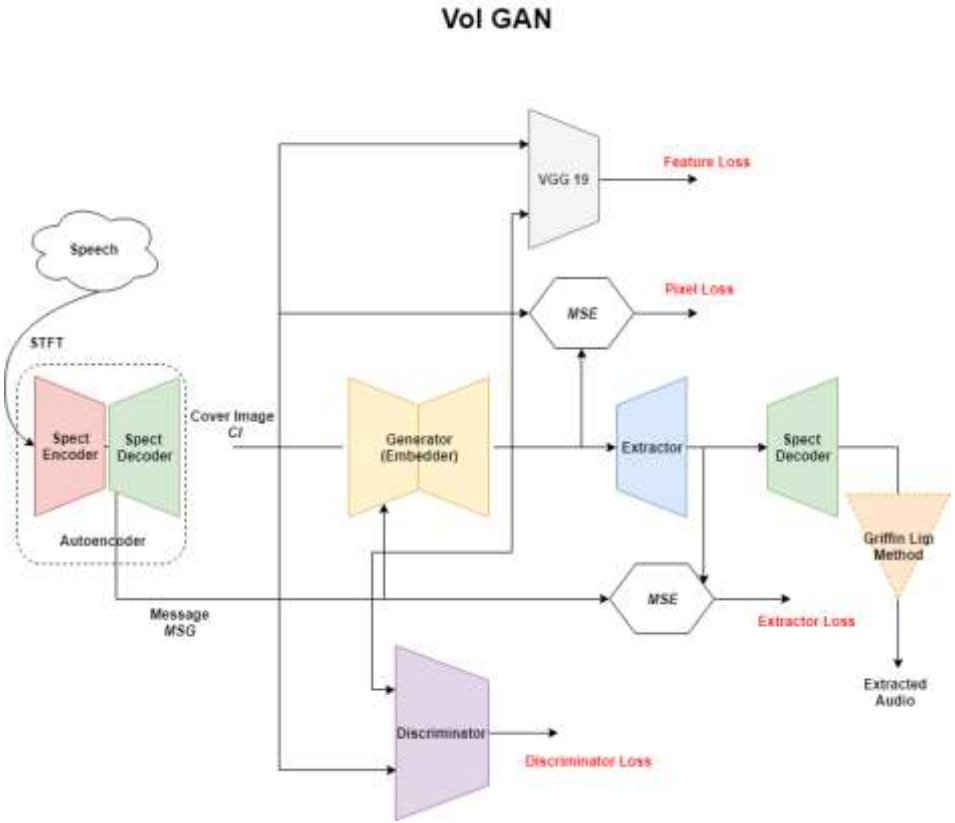
Extracted watermark

Actual Watermark

Hiding Audio in Image



Proposed Method

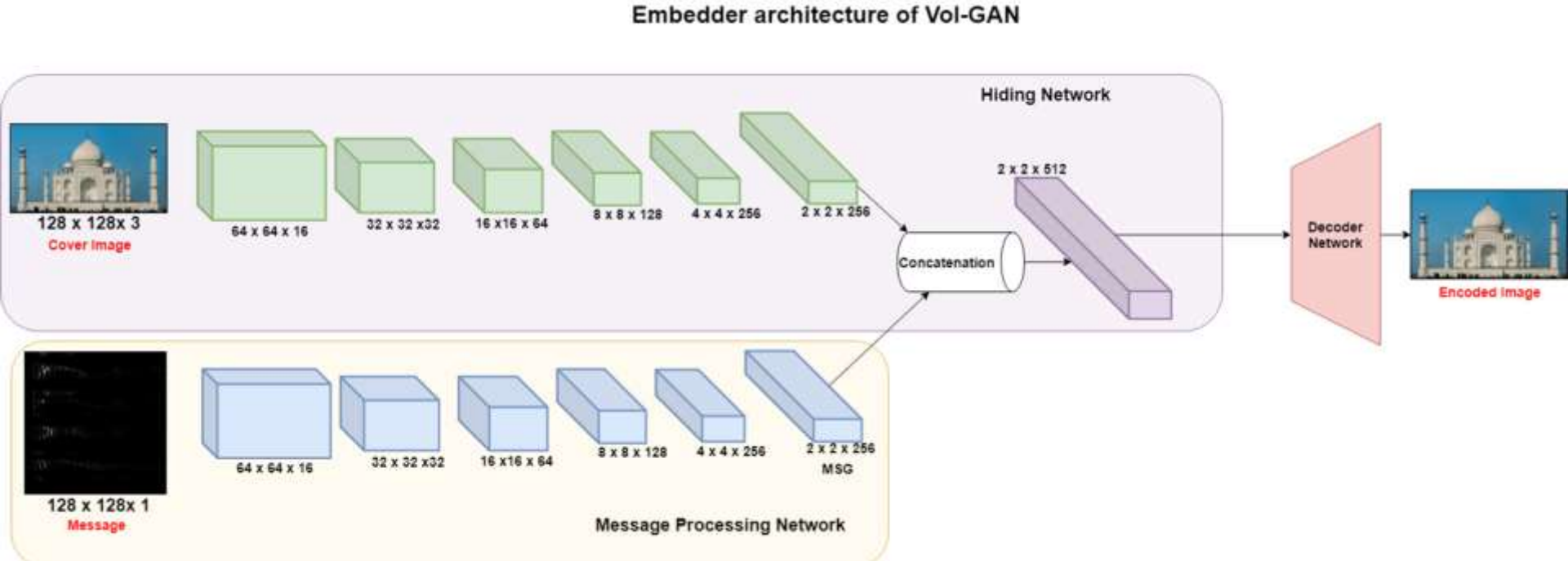


- 3 stage implicit training of VoI-GAN
    1. Train the autoencoder **AE,** to output image similar to input image.
    1. Trin the embedder **Emb,** to output an image similar to cover given cover and latent spectrogram as input.
    1. Train the extractor **Ex,** to output the latent spectrogram given the output of **Emb.**
- The “encoding” is done by concatenation process in the encoder of the **Emb**.

1. G Rohit, D Mishra, VoI-GAN: Hiding Audio in Images, submission to Human Intelligent Systems Integration, springer
2. G Rohit, D Mishra, Hiding Audio in Images: A Deep Learning Approach , under review 8th International Conference of PReMI

Training of HD-VAE-GAN: Stage 2

- Second stage is to train the embedder **Emb,** to output an image ***EI*** that is similar to the cover image ***CI*** with message ***MSG*** and ***CI*** as inputs**.** This embedding was done using adversarial loss, reconstruction pixel loss and feature loss.



Experiment

- We have used the DIV2K dataset to train and evaluate our model. The dataset consists of 900 images (800 train, 100 test) to evaluate our models and 800 audio files of 2 seconds length those are sampled at 8 kHz from a publicly available SignaLogic dataset.

Table 2.1: Embedding and extraction efficiency in terms of PSNR and SSIM on Div2K dataset. PSNR-EX is the measure of closeness between original and extracted spectrogram.

Method	PSNR-CI	PSNR-EX	SSIM
VoI-GAN	35.20	31.44	0.9828

Experiment

- Inherent robustness of the model towards multiple attacks due to the base learning and deep architecture.

Table 2.3: VoI-GAN's extraction efficiency of the hidden spectrogram from attacked encoded images in terms of PSNR (in dB). The last three columns corresponds to the types of noises.

Audio	Blur ($\sigma = 1$)	Rot90°	Rot10°	Crop 10%	S&P	Speckle	Gaussian
Female	55.26	26.86	51.89	58.17	54.80	54.32	54.12
Male	52.99	43.46	57.25	56.83	52.57	52.05	50.22
Car Noise	52.76	24.95	51.62	53.90	52.96	53.24	52.92
White Noise	51.99	20.81	57.73	52.04	51.33	51.79	51.40

Experiment

- To prove that the existing methods can not hide audio in images.

Table 2.4: Comparison of extraction efficiency in terms of SSIM, PSNR (in db) and Mean Opinion Score between different models.

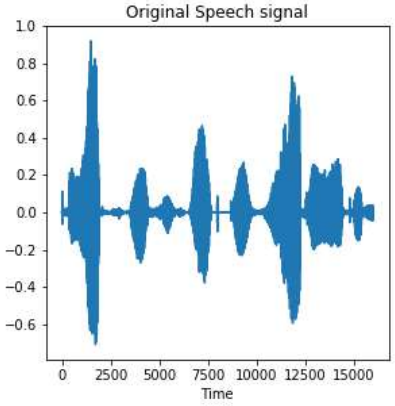
Model	SSIM	PSNR	Mean Opinion Score
Model 1	0.78	17.47	4.25
Model 2	0.81	21.42	0.00
VoI-GAN	0.998	53.45	4.53

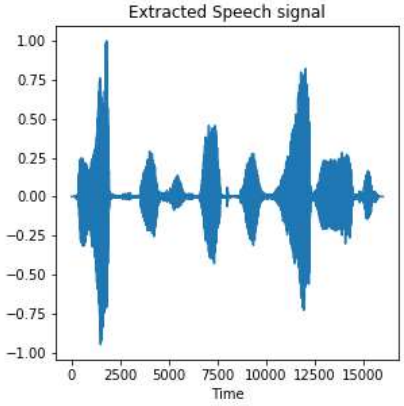
Table 2.5: Comparison of emdedding and extraction capacity of Model2 and exisiting image hiding methods on DIV2K dataset. We have tested Model2 to hide and extract random pairs of images from Div2K test dataset. This clearly proves that Model2 is similar to the existing work and hence can be taken as a reference image hiding method.

Method	**SSIM**	**PSNR-CI**	**PSNR-EX**
Model2 (N=50)	0.9875	36.02	23.04
Model2 (N=100)	0.9899	37.02	23.28
Model2 (N=200)	0.9930	38.12	24.01
SteganoGAN Res (N=800)	0.9000	38.25	-
SteganoGAN Basic (N=800)	0.6700	24.62	-
DeepStego †	-	35.29	37.79
Model2 (N=50) †	0.9801	36.05	25.78

*N – Number of samples used in training

Visual and audio sample



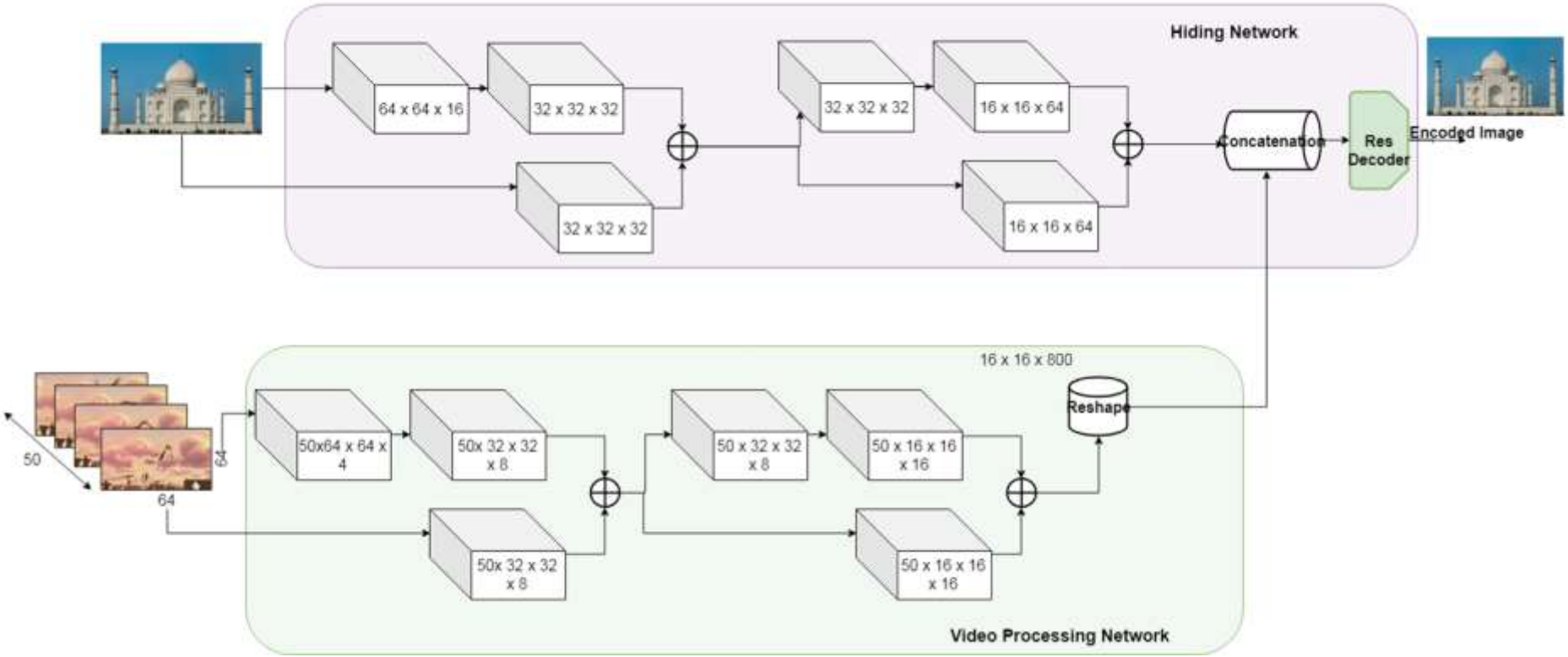


Original File

Extracted File by Model 2

Extracted File by VoI-GAN

Hiding Video in Image



Introduction

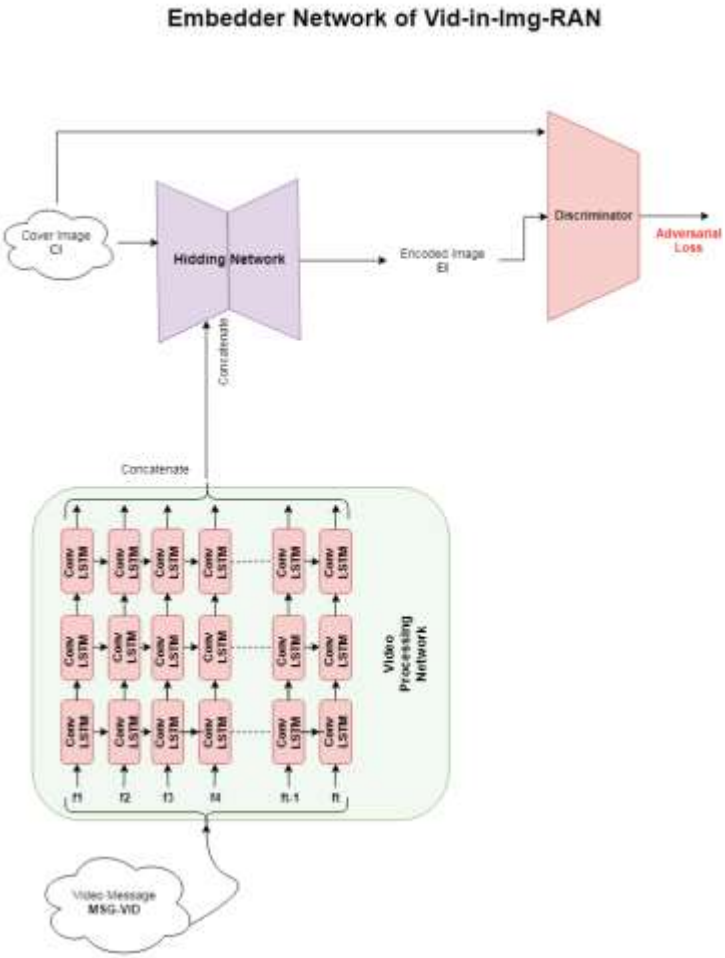
- This is a completely new variant of data hiding, the novelty of this work also includes the proposal of this new research area.

- We propose two working models using deep generative models.
  a. Using Recurrent Neural Networks.
  b. Using 3-D spatio temporal Convolutional networks.

- More detailed analysis and refining of these models are the possibilities for the future work.

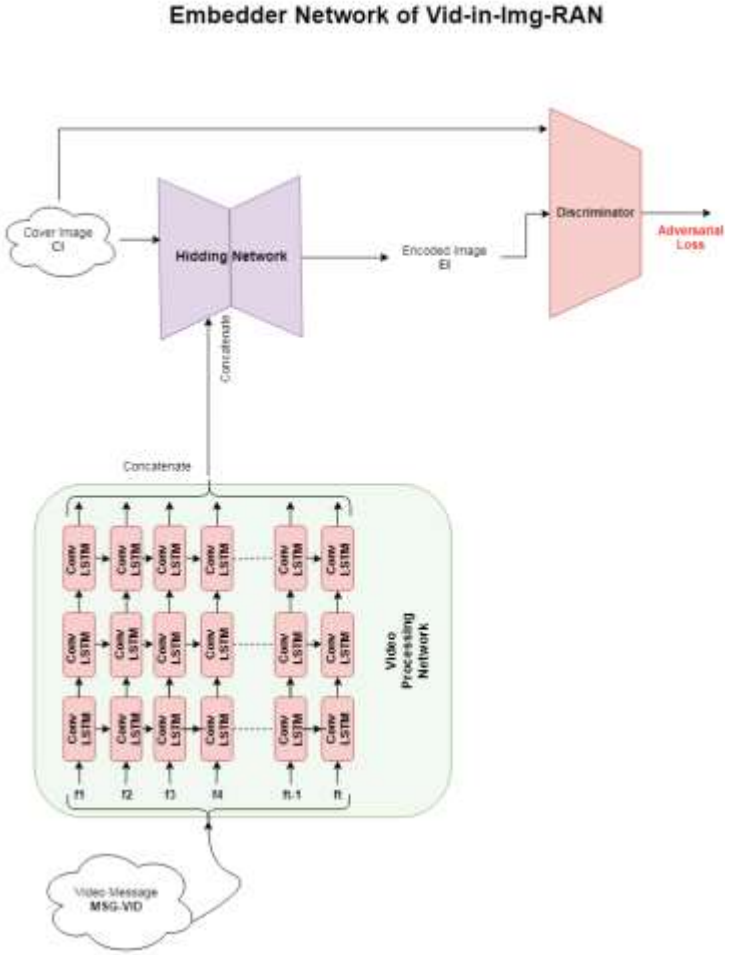
Rohit G, Deepak Mishra, Hiding Video in Images using temporal GANs, Submission to IEEE International Workshop on Information Forensics and Security (WIFS)

Proposed Method - RNN version (Vid-in-Img-RAN)



- The very first attempt to hide video in image.
- Video Processing Network (many-to-many RNN) to extract features of video and a shape them into 3D.
- Hiding Network (simple autoencoder) takes the cover image produces an encoded image by concatenating the video latent representations.
- Video Processing Network is built with stacks of LSTM cells with convolutions (Conv LSTM).
- Hiding Network is built with traditional 2D convolution layers.

Proposed Method - RNN version (Vid-in-Img-RAN)



$$D_{Loss} = log\mathbf{D}(CI) + log(1 - \mathbf{D}(EI))$$

$$Emb_{Loss} = \frac{1}{cH \times cW} \sum_{i=1}^{cH} \sum_{j=1}^{cW} (EI_{i,j} - CI_{i,j})^2$$

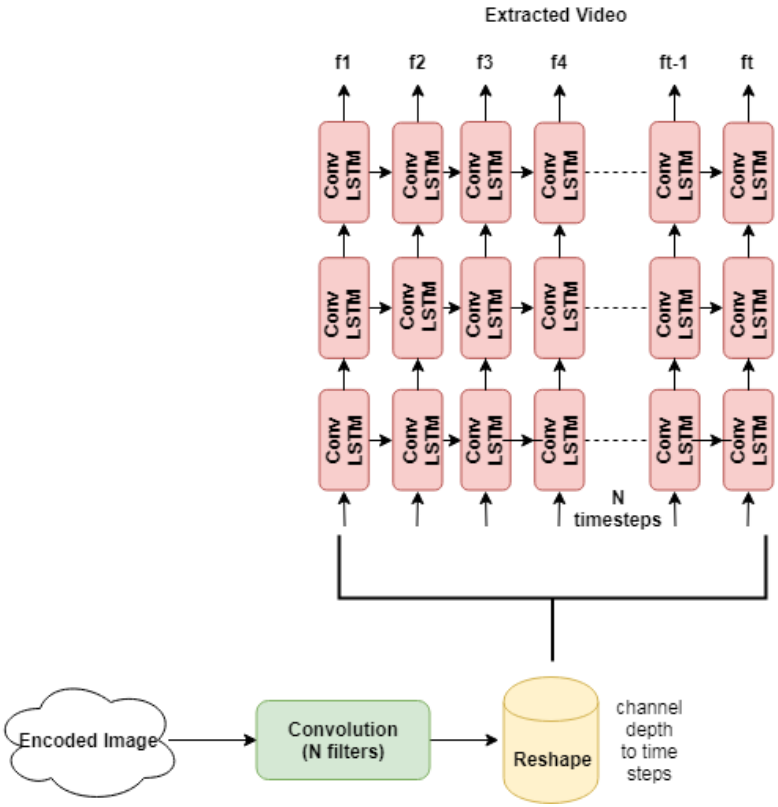
$$F_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (\chi(EI)_{i,j} - \chi(CI)_{i,j})^2$$

Train the generator with parameters θ such that:

$$\max_{D} \min_{\theta} \beta_1 Emb_{Loss} + \beta_2 F_{Loss} + \beta_3 D_{Loss}$$

Proposed Method - RNN version (Vid-in-Img-RAN)

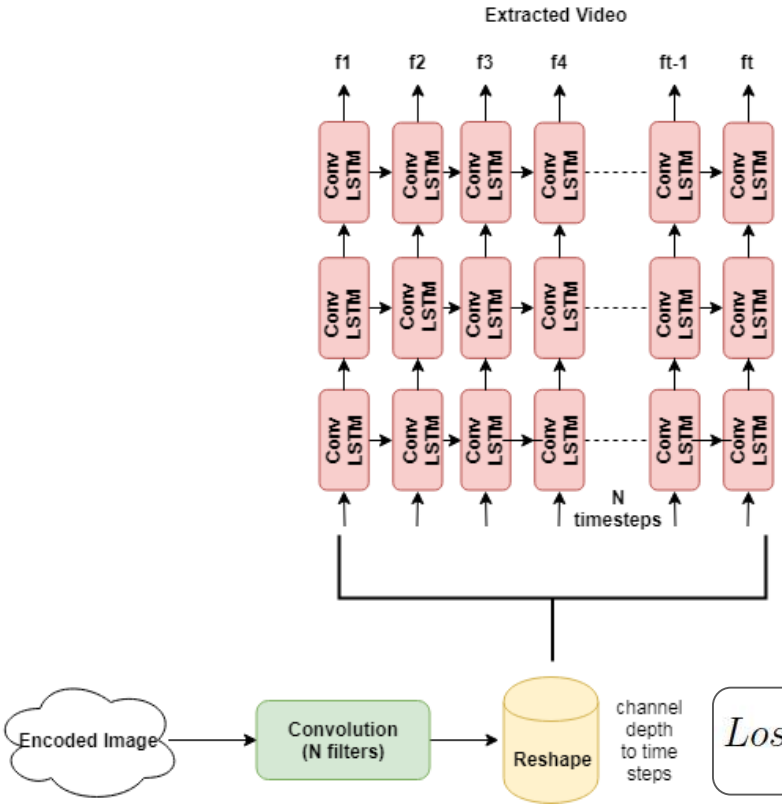
Extractor Network of Vid-in-Img-RAN



- Extraction is the most challenging task.
- A convolution layer that acts like temporal feature extractor.
- All the spatially channel activations are converted to temporal features.
- The main body of extractor is another RNN stack with convolutions.
- The output will be a 4D data with 3D temporal frames

Proposed Method - RNN version (Vid-in-Img-RAN)

Extractor Network of Vid-in-Img-RAN



$$Ex_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (VMSG_{i,j}^{extracted} - VMSG_{i,j}^{actual})^2$$

$$Where\ VMSG^{extracted} = Extractor_{\lambda}(EI)$$

Train the extractor with parameters λ such that:

$$\min_{\lambda}\ \beta_4 E_{Loss}$$

$$Loss = \min_{EMB}\ \max_{D}\ \min_{EX}\ \beta_1 Emb_{Loss} + \beta_2 F_{Loss} + \beta_3 D_{Loss} + \beta_4 Ex_{Loss}$$

Drawback

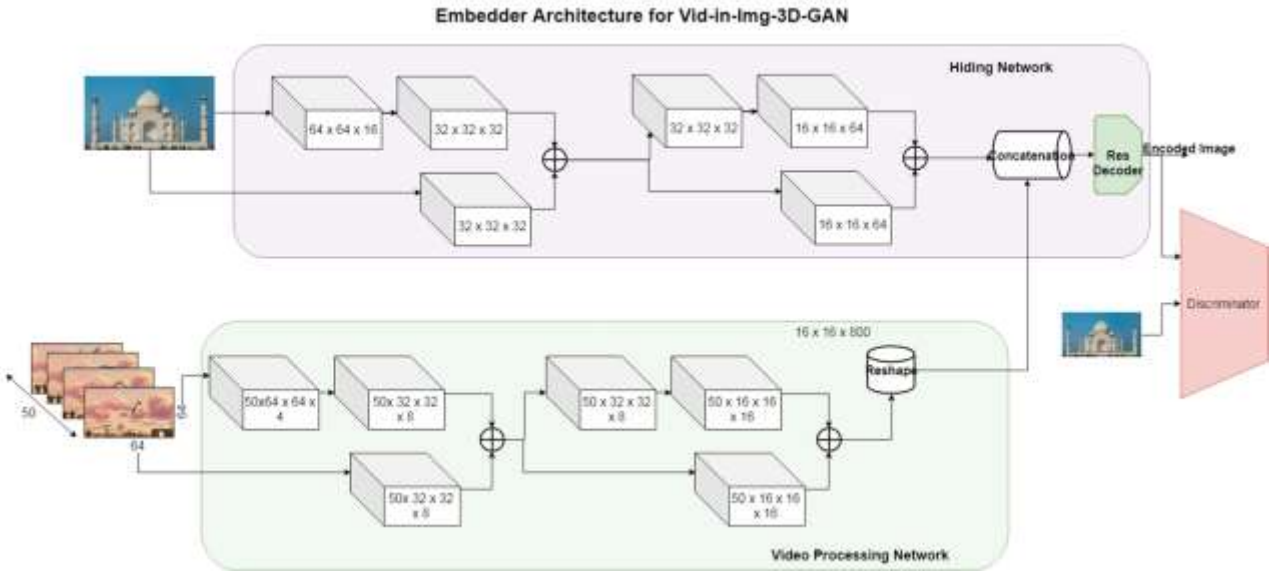
- The drawback is the structure of a vanilla LSTM. The first cell has no information except the input, while the last has all the important past info.

Solutions

➔ Using bi-directional LSTM.

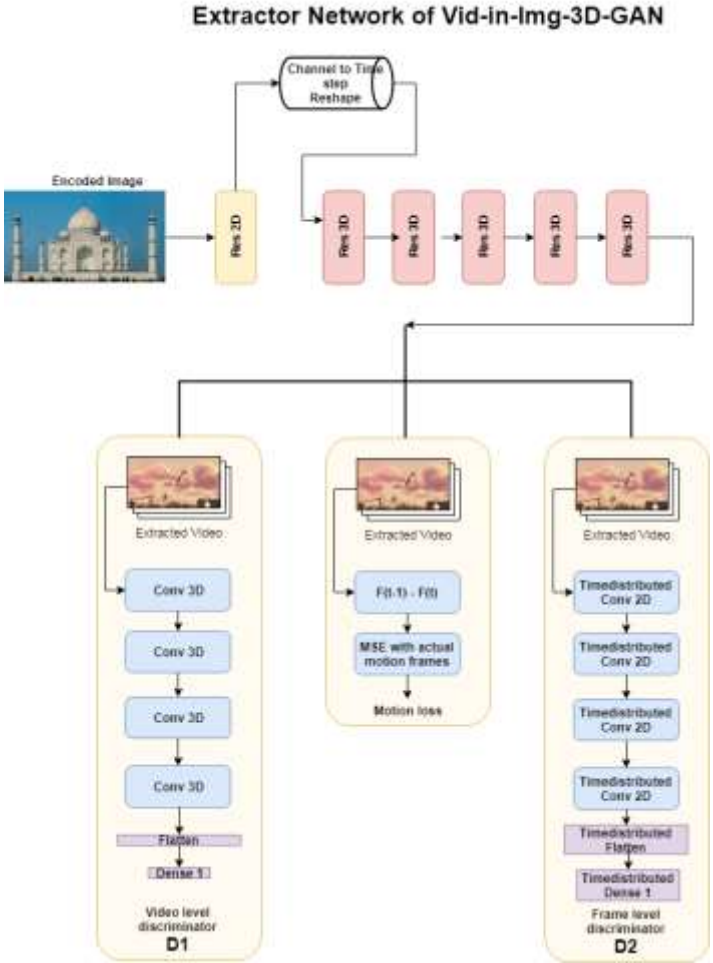
➔ Using 3D convolution layer.

Proposed Method - 3D GAN version (Vid-in-Img-3D-GAN)



- Slightly refined version with less number of computational operations and a more involved method.
- Video Processing Network (3D Conv net) to extract features of video and a shape them into 3D.
- Hiding Network (simple autoencoder) takes the cover image produces an encoded image by concatenating the video latent representations.

Proposed Method - 3D GAN version (Vid-in-Img-3D-GAN)

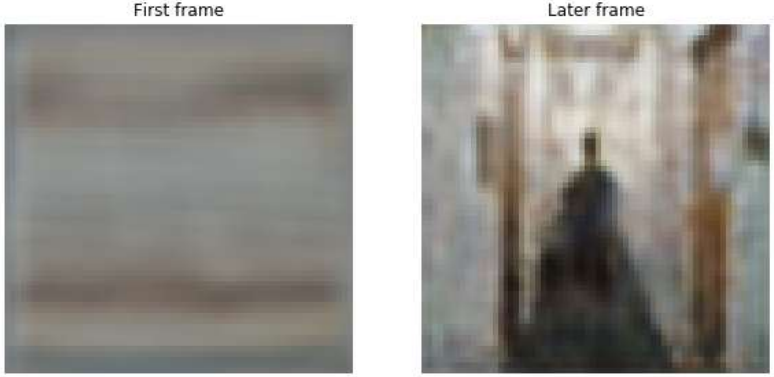


- A res layer with convolutions that acts like temporal feature extractor.
- All the spatially channel activations are converted to temporal features.
- The main body of extractor is stack with 3D convolutions.
- The output will be a 4D data with 3D temporal frames.
- We introduce a video level and a frame level discriminator.
- We also introduce a motion loss to maintain the temporal smoothness in the videos. (Compensation for RNN)

Analysis

- First frame vs a later frame

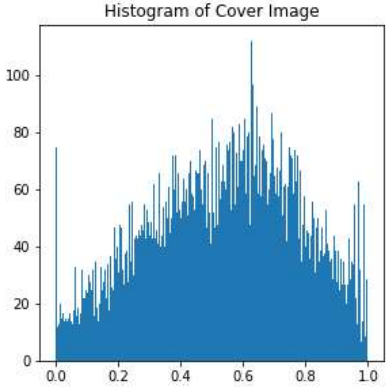
Vid-in-Img-RAN

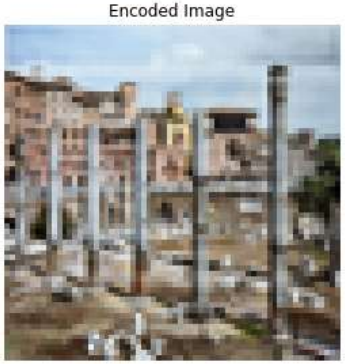


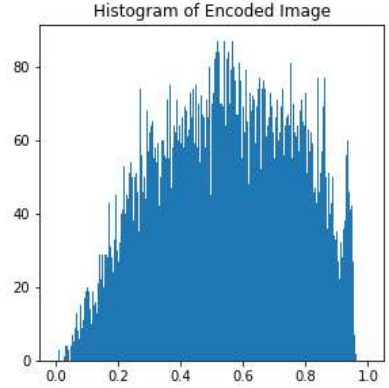
Vid-in-Img-3D-GAN

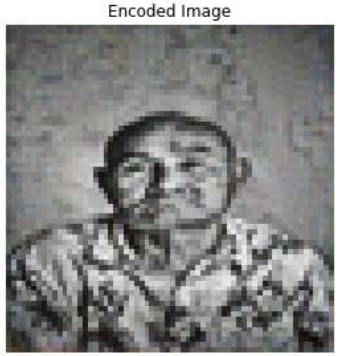
Analysis

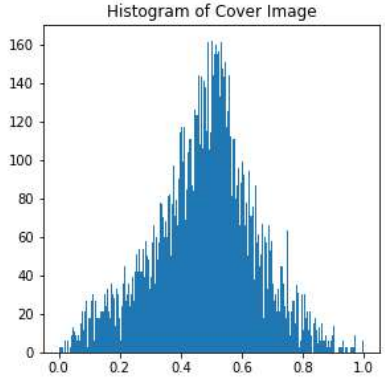
- Histogram analysis to check pixel-level distortions

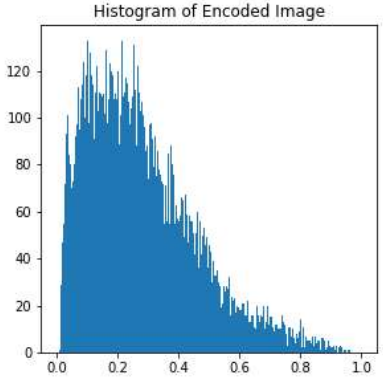












Visual Results

Vid-in-Img-RAN

Vid-in-Img-3D-GAN

Conclusion

- Proposed 5 novel deep generative models that are end-to-end trainable for multimedia data hiding.
- Proposed a novel research area of hiding video in images.
- All the methods had a similar base structure but due to the modality of multimedia, they had their own unique requirements.
- All the models can be used for strategic purposes and can be adopted by ISRO for information security, data hiding and watermarking.
- Although the proposed models are for the application of data hiding, they are generic and can be directly adopted into applications like data privacy, 3D mesh model watermarking, data security, cryptography etc.
- This work is fully open-sourced for further academic improvements on this research.

Future Work

- Improving the proposed video hiding techniques.
- Exploring the applications like 3D mesh watermarking, audio watermarking, medical information, biometrics, and data privacy.
- All the proposed methods are generic and can be directly adopted into mentioned applications with minimal architectural changes.
- Adopting deep learning ways may open new directions and find interesting applications in the field of Data Hiding.

nGraph

INTEL® NEURAL COMPUTE STICK 2
(INTEL® NCS2)

Thank You

Any doubts ??

(rohit_g@nrsc.gov.in)
(deepak.mishra@iist.ac.in)