Multiple Image Steganography Using Deep Neural Networks

Saptashwa Mandal
Reg: 220911454
B.Tech. IT
MIT,Manipal

Venkata Sai Vishwanath
Reg: 220911508
B.Tech. IT
MIT,Manipal

Navaneeth L
Reg: 220911568
B.Tech. IT
MIT,Manipal

Bhanu Shashank
Reg: 220911572
B.Tech. IT
MIT,Manipal

***Abstract*—Steganography is the art of concealing a secret message in a standard, publicly visible message. The traditional way to embed a lower resolution image inside a higher resolution image has been LSB manipulation. In this work, we're attempting to utilize deep neural networks for encoding and decoding multiple hidden images inside of the same cover image, at the same resolution.**

I. Introduction

Steganography is a technique for hiding secret messages in a non-secret message to avoid detection during transmission. At the receiver end, this secret data is extracted from the cover message. Steganography can also be used in combination with encryption to provide an added layer of protection for data security. Conventional application uses the method to embed low-resolution images in high-resolution images using such simple techniques as LSB manipulation.

Aiming at the extension of work area from related papers, we propose methodologies to encode multiple secret images into one cover image. Our approach also differs from the conventional approach as we embed images of the same resolution. Here, neither is the reduction in image quality and change in resolution effects of embedding. A key challenge is to make any changes to the cover image imperceptible to the human eye, preserving its visual integrity to the extent that even statistical analysis would have difficulty identifying modifications. Simultaneously, we aim for high fidelity in the decoded images, making them as clear and understandable as possible upon retrieval. Our approach, thus, balances the minimum detected in the cover image encoded with maximum possible intelligibility in decoded images, rubbing at the very limits of modern steganography about visual secrecy and integrity over data.

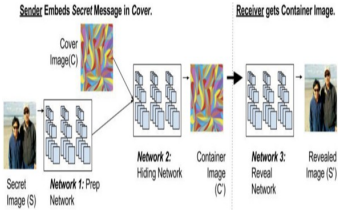


Fig. 1. The three components of the full system.

II. Related Works

Among several these, two of the implementations closely resemble our goal. 3.1. Hiding Images in Plain Sight-Deep Steganography In the paper by Baluja in 2017, a full-colored image is to be embedded inside another of the same size. This approach trains deep neural networks, jointly in the embedding and extraction as unified tasks. The model is trained on ImageNet database images, which are found successful with diverse natural images. Contrary to the common traditional steganography method encoding data in all the least significant bits LSBs of the cover image, Baluja's approach compresses the secret image's data and diffuses it for overlaid at all bits in the cover image without distortion. It encompasses three key networks: Preparation Network: This module preparesthe secret image for hiding. As generally, the size of thesecret image is much smaller than that of cover image's N×N, it will resize it to equal N×N, so the secret data willbe spread out over each pixel in the whole N×N grid of pixels. Hiding Network. This network takes the Preparation Network's outputs and cover image as inputs to produce the "Container" image, which embeds both images. Here, the N×Npixel inputs consist of RGB channels from the cover and the modified channels from the secret image. Reveal Network: At the receiver end, the decoder network is employed to extract the hidden image strictly only from the Container image, without access to the original cover or secret images. The Reveal Network filters the efficiency of the image reconstruc- tion of the secret image in eradicating the cover. Baluja's paperpresents the requirement of compression and embedding secretimage information in a cover image into the network least detectable regions. However, it does not mention anything about the hiding of this information from machines. Theirsteganalysis approach proceeds by training binary classifiers on the original ImageNet images as negative and Container images as positive, thus offering a baseline for encoding a single secret image. In addition, it does not investigate the encoding of more than one image within a single cover.

III. Methodology

A. Concepts

- **Steganograhy** - Steganography is a technique of encoding a secret message inside a cover message. In

images, traditionally, the message is encoded in the least significant bit (LSB) and the message is retrieved using classical algorthims.

- **CNNs (Convolutional Neural Network)** - CNNs is a type of neural network which uses a convoltuional filter ofgiven channels and sizes as the weights. CNNs is usually used on images as it can easily extract features of images.
- **Deep Steganography** - CNNs are used to get a representation of the cover(Image used to hide the secret) and the secret images. This encoder generates the cover image that is to be sent to the decoder. This representation is then passed to another CNN which acts as the decoder and generates the secret images.

B. Model Architecture

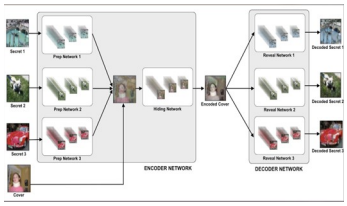


Fig. 2. Model Architecture

Fig. 3. Encoder Diagram

Encoder is used to hide the three images into the cover image. It consists of three preperation networks each with a similar architecture. Each of these prep networks consists of two blocks of the layers. Each of these layer consists of three Conv2d layers with kernel sizes as 3,3,5. These Conv2d layers have output channel dimensions as 50,10 and 5 respectively. Since the output image must have the same dimensions as the input image, the padding is chosen as 1,1,2 respectively and the stride is kept as 1 for all the Conv2d layers. Relu acivation function is applied after each convolution layer. Input image is passed to each layer and the final output from these layers is concatenated and passed on the next block of the prep network. This is carried out for a single image and for a single prep network. Similar procedure is carried out for the remaining two images and we a get three images of dimensions **(batch size,65,64,64)**. Note that, here our channels are 65 (Due to concatenation of 5,10 and 50 channel outputs).

Next, after getting the representation for three images, we need to **hide** these three images into our cover image. For this, we have a hiding network. First, we concat our representations of three images with the cover image, giving as a tensor of dimensions **(batch_size, 198, 64,64)**. Here, the channels are 198, due to concatenation of these four tensors:

- Cover Image (batch_size, 3, 64,64)
- Representation of first secret Image (batch size, 65, 64,64)
- Representation of second secret Image (batch_size, 65, 64,64)
- Representation of third secret Image (batch size, 65, 64,64)

This tensor is passed on the hiding network which consists of five blocks of layers. Each of these layers is similar to the blocks used in the prep network. The final layer of the hiding network should output a tensor of dimensions **(batch_size, 3, 64, 64)**. This is the encoded image **OR** the cover images which is used to hide our three secret images.

Fig. 4. Decoder Network(Tree Diagram)

Decoder is used to reveal the secret image from the encoded cover image. It consists of the three reveal networks for each of the three secret images. Each of the reveal networks consists of five blocks of convolutional layers. The underlying architecture of each block is the same as in the encoder. Since the output of each bock is an image of dimension **(batch_size,65,64,64)**, we need to change the number of channels to 3 for an RGB image. So, we apply another conv2D layer at the end of each of the reveal networks to get the image of dimesion **(batch size, 3, 64, 64)**. This is achieved by setting the output channels as 3 inthis last conv2D layer. This final image is the decoded secret image obtained from the reveal networks. Thus we get three decoded secret images from each of the three reveal networks.

C. Dataset

Dataset used for this paper is Tiny Image Net. This dataset consists of some greyscale images too. These have been removed by running the script create_dataset.py . This script is used for creating the training and validation datasets. We sample 10 images from each of the 200 classes of tiny imagenet dataset for training and 4 images for testing. In total,we have **2000** training images, and **800** testing images.

· Here, we describe the method to train the model. Initially, this seems like a standard encoder-decoder model. But, the training works in this way: First, we train the encoder, and keep the decoder's parameters as untrainable. This helps the encoder to learn and create the encoded hidden image. This model is evaluated using the loss for all the reveal images as well as the hidden image.

· After getting the output from the encoder, we train the decoder (Consisting of three reveal networks). This decoder, is evaluated only on the basis of the reveal loss.

· Now, the parameters of the deocder's networks are shared with the reveal networks while the encoder is being trained.

· This ideally helps the network to learn much better as the encoder and the decoder network become smaller and task is way more focused in terms of optimization instead of training the whole model in a joint manner.

Here are the hyperparameters which we have used for training the networks:

IMG_SIZE = 64
LEARNING_RATE = 0.001
COVER_LOSS_WEIGHT = 1
SECRET_LOSS_WEIGHT = 1
TRAIN_BATCH_SIZE = 16
VALID_BATCH_ SIZE = 1
EPOCHS = 1000
DECODER_LOSS_WEIGHT = 1

IV. Results

Performance was evaluated based on encryption and decryption time for each algorithm, with AES and DES being the fastest options.

Sample input output:

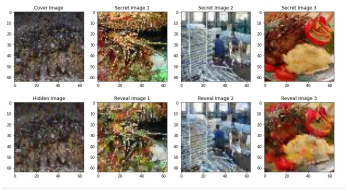


Fig. 5. Output 1

Fig. 6. Output 2

V. Conclusion

The development and implementation of Multi-Image steganography has resulted in a highly adaptable and secure messaging platform that provides users with various encryption algorithms to safeguard their communications.

VI. Limitations

There are two different losses for the full model and the deocder.

The full model loss is calculated by summation of MSE of the following pairs:

Cover Image, Hidden Image
Secret Image 1, Reveal Image 1
Secret Image 2, Reveal Image 2
Secret Image 3, Reveal Image 3 We weight the losses of cover image and secret images to 1:1.

For the decoder network, the reveal loss is summation of MSE of the following pairs:

Secret Image 1, Reveal Image 1
Secret Image 2, Reveal Image 2
Secret Image 3, Reveal Image 3

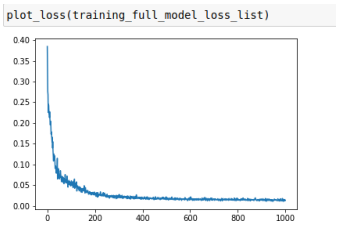


Fig. 7. Full Model Loss

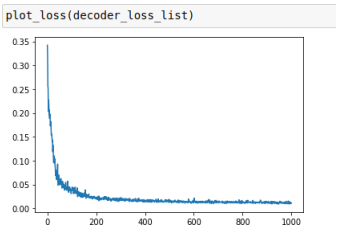


Fig. 8. Decoder Loss

VII. Future Scope

1) Future improvements may include extending the study and trying to build a web app around the algorithm for users to upload images and try around.
2) Greater amount of data: multi-image steganography conceals a higher volume of data spread across multiple images.
3) Integration with Video and Audio: With the advancement of the digital ecosystem, multi-image steganography mightnot be limited to images alone.

VIII. References

[1] Xin Liao, Jiaojiao Yin, Mingliang Chen and L. Adleman, "Adaptive Payload Distribution in Multiple Images Steganography Based on Image Texture Features,".
[2] Mingwei Tang, Jie Hu, Wen Song, "A high-capacity image steganography using multi-layer embedding,".

IX. FIGURES

FIG-1: The three components of the full system.

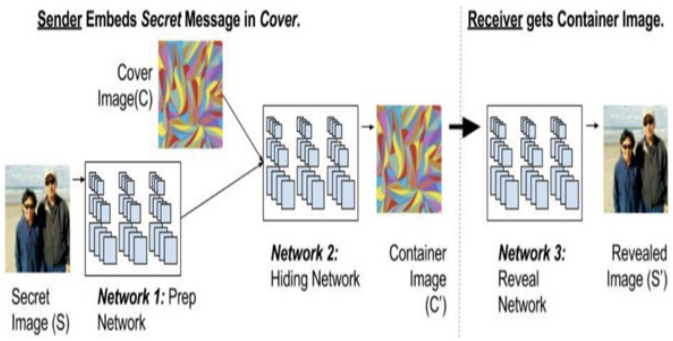


FIG-2: Model Architecture.

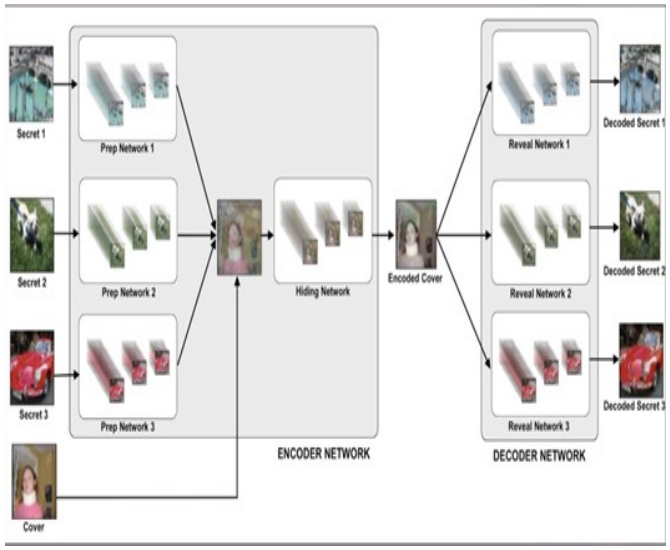


FIG-5: Output 1.

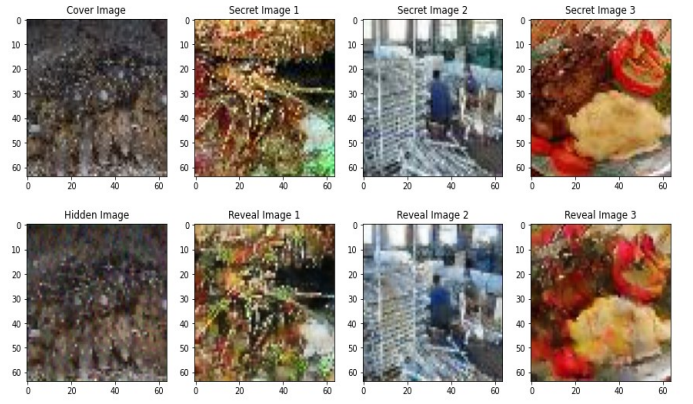


FIG-6: Output 2.

FIG-7: Full Model Loss.

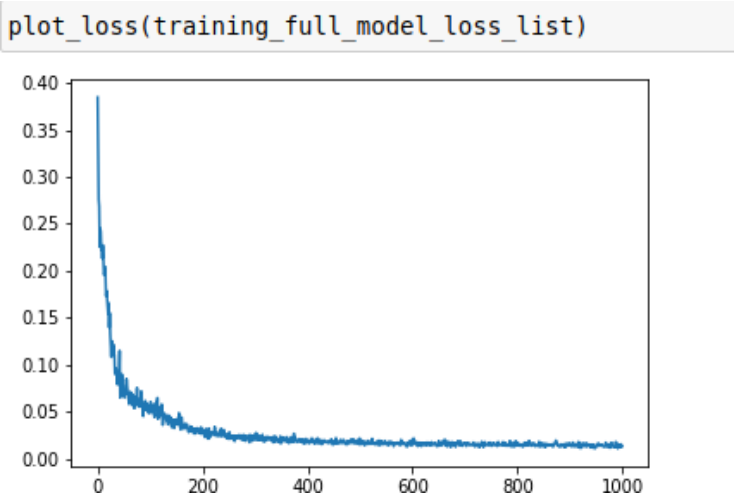


FIG-8: Decoder Loss.

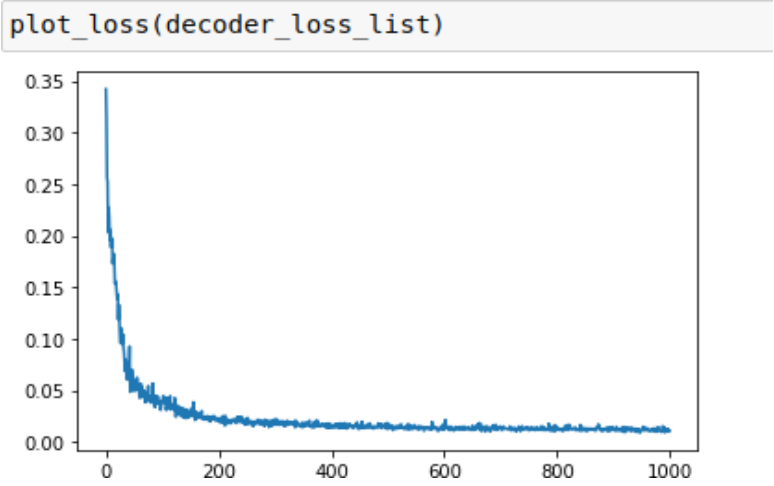


FIG-3: Encoder Diagram.

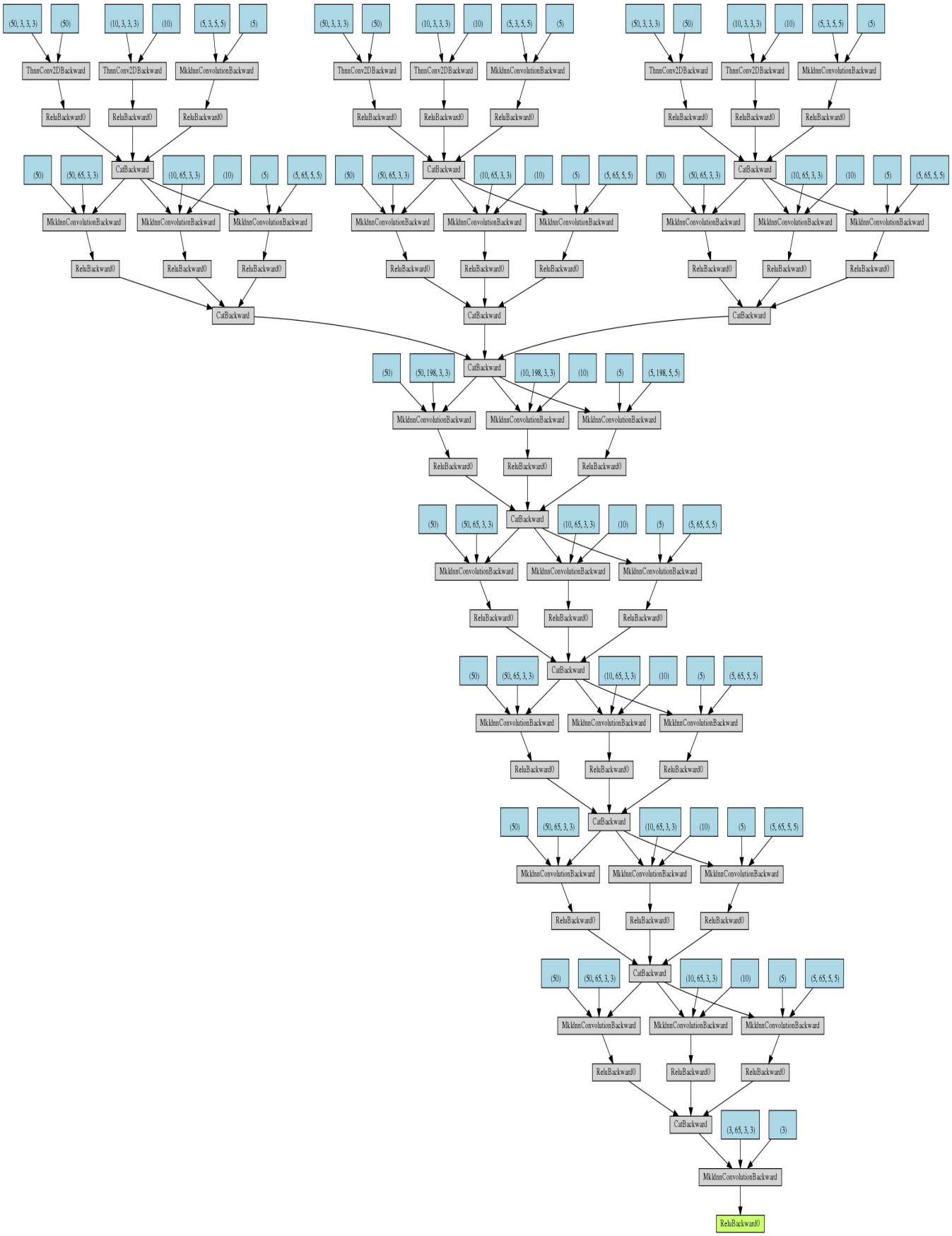
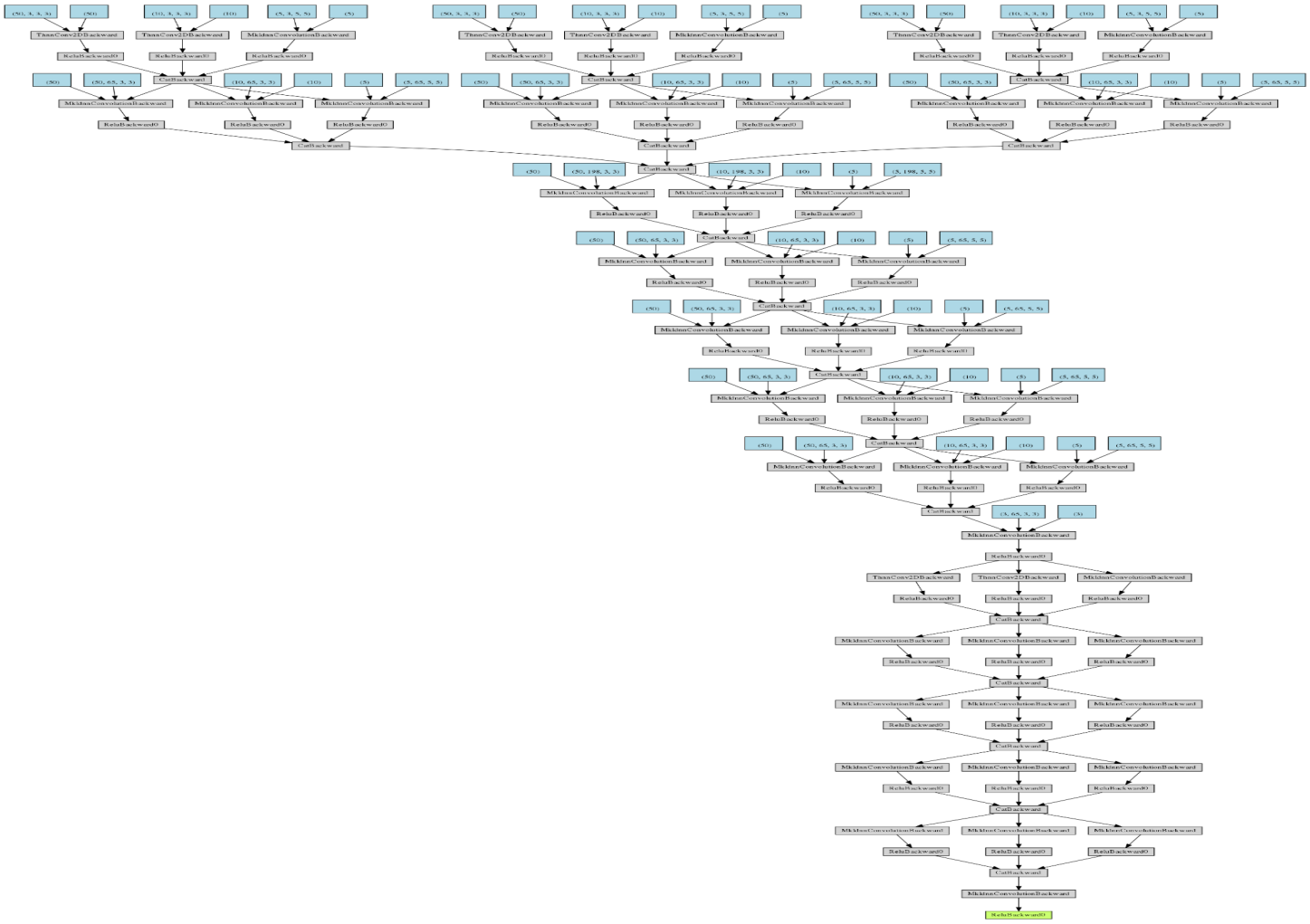


FIG-4: Decoder Diagram.



1s_plag.docx

ORIGINALITY REPORT

11%	10%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github.com Internet Source	10%
2	"Advanced Computer and Communication Engineering Technology", Springer Science and Business Media LLC, 2015 Publication	1%
3	Abdelsalam, A.A.. "Characterization of powerquality disturbances using hybrid technique of linear Kalman filter and fuzzy-expert system", Electric Power Systems Research, 201202 Publication	<1%

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		