FACE COUNTERFEIT DETECTION IN NATIONAL IDENTITY CARDS USING IMAGE STEGANOGRAPHY

Senthilkumar K[1], Dr. Vignesh Ramamoorthy H[2], Dr. S. Uma Shankari[3], Dr. R. Nallakumar[4]

Article History: **Received:** 12.12.2022 **Revised:** 29.01.2023 **Accepted:** 15.03.2023

Abstract

When we talk about "identity card," we're referring to a government-issued photo ID that can be used as such at least in Germany. Smart to travel documents, electronic IDs, electronic signatures, municipal cards, key cards for accessing protected areas or company infrastructure, social security cards, etc. are just some of the furthermost prevalent uses for smart cards. There are a amount of safeguards included into these documents. Fight the practise of document falsification. Criminal attacks against identity verification systems currently rely on illegally obtaining real documents and modifying facial pictures because these security mechanisms are hard to defeat. Having a system of trusted identities is crucial to any functional society. These governments and identity manufacturers should regularly update and enhance their security protocols to reduce the likelihood of fraud. For this reason, we deliver StegoCard, the first practical steganography approach tailored specifically for photos typically found on standard ID cards. StegoCard is a full-stack facial image steganography model that uses a Deep Convolutional Auto Encoder to create a portrait of a Stego that conceals a message and a Deep Convolutional Auto Decoder to decode the image. Able is the decoder. The Stego image serves as a message decoder. This holds true even if the image was printed out and afterwards digitised. Comparisons of He StegaStamp and StegoCard encoded face photographs show that the latter are of higher perceptual quality. Peak signal-to-noise ratio, concealing power, and inaudibility scores on the test set are used as presentation metrics.

Keywords: Deep Learning, Steganography, Recurrent proposal Network (RPN), Binary Error-Correcting Code Algorithm, Deep Convolutional Auto Encoder, Deep Convolutional Auto Decoder

[1]Assistant Professor Master of Computer Applications Karpagam College of Engineering Coimbatore, India
[2]Assistant Professor Department of IT and Cognitive Systems Sri Krishna Arts and Science College Coimbatore-6421008
[3]Assistant Professor Department of Computer Science and Applications SRM Institute of Science and Technology Ramapuram
[4]Associate professor Department of Artificial Intelligence and Data Science Karpagam Institute of Technology

Email: [1]senthilkumar.k@kce.ac.in , [2]hvigneshram@gmail.com, [3]umabalajees@gmail.com, [4]nallakumar.ai@karpagamtech.ac.in

DOI: 10.31838/ecb/2023.12.s3.063

1. Introduction

As things stand, it is possible to transmit a message that will only be seen by the intended recipient and sender by encoding it within a picture or a text. Steganography refers to the practise of secretly hiding information and revealing it when needed. The term "cover image", whereas the term "stego-image" refers to the image itself after it has been hidden. To prevent an intruder from deciphering a secret message, cryptography encrypts the text. Hence, combining steganography and encryption adds an extra safeguard. By using image compression, we may lessen the file size of the message and make it more covert [1]. Steganography, in which an image is used as a cover for a hidden message, is a common form of cryptography. Because to their widespread presence online, digital photographs are frequently used as cover objects because they provide enough redundant bits in pixels that can be used to conceal the secret information without degrading the image's aesthetics. Beginning with techniques for concealing information in the image's least significant bits (LSBs) [2, 3], the field of image steganography has progressed to more sophisticated methods like content-adaptive steganography, in which embedding costs are dynamically adjusted for each cover pixel by means of a distortion function. Cover image embedding is achieved by minimising a distortion function [4, 7]. It is vital to conceal the dispatch's environment and actuality from unauthorised beneficiaries to strengthen the security of secret dispatches in an open system setting. It is known as the "science of unseen or hidden communication" [8]-[10]. The three criteria of capacity, transparency, and robustness are necessary for any steganographic technique to be considered successful. The capacity of a file is the most sensitive data that may be stored within it [11]. If the info of the cover and the stego files are the same, then the steganographic method is completely secure and undetectable. Last but not least, the stego file must be strong, meaning it can withstand a variety of attacks and still reveal the secret message with minimal loss of information, as described in [12–15]. There is a tradeoff between the two of these values; increasing the hiding capacity reduces the robustness of the secret message and the transparency of the stego file, and vice versa. Hence, balancing them is difficult [12, 13]. Hence, improving steganographic volume and boosting safety and stealthiness while keeping robustness intact [16]–[18] is the primary goal of steganography.

Literature Review

Credit cards, driver's licences, and other forms of identification are all useful, but identity cards are essential in identifying individuals and combating fraud. Unfortunately, security risks and financial losses are inevitable results of ID card forgery. Steganography has been investigated as a means of increasing the safety of identification cards by concealing sensitive information on the cards themselves. Steganography refers to the practise of concealing data in a digital media like a picture or audio file so that it is undetectable by the senses. To get a feel for the various methods of steganography that can be applied to digital photographs, see Singh and Gupta (2016). The authors explain the difficulties of steganography, such as the detection and extraction of the concealed data, and go over the various types of steganography, such as spatial domain and frequency domain techniques. Madhu et al. (2018) present a steganographic approach to facial recognition. The method uses a classifier to identify a face after it has been concealed within a picture. Face identification and fake-spotting using steganography: a method developed by Naveen Kumar and coworkers in 2017. (PCA) and Discrete Wavelet Transform (DWT) are used together by the authors to mask the faces in a picture. For national identification cards, Agarwal et al. (2020) suggest a new method of image steganography. The authors apply the (DCT) and the Arnold Transform together to hide the faces in plain sight. The results of the investigation demonstrate the feasibility and efficacy of the suggested method in detecting and preventing the use of counterfeit identity cards. The authors demonstrate that their method is more accurate and secure against attacks than other existing steganography methods by comparing them. Applications of their method, such as digital watermarking and copyright protection, are also discussed. In summary, steganography shows promise as a method for strengthening the safety of national identity cards. This paper reviews research showing how steganography can be used to secretly embed facial traits within a picture, allowing for more reliable face recognition and the identification of fakes. By their enhanced steganographic method, Agarwal et al. (2020) demonstrate the viability of preventing fake ID cards. Increased security for identification cards and other forms of personal information can result from continued study of steganography.

Proposed Approach

StegoFace is the name of the planned system. StegoFace is an ID and MRTD-relevant model for encoding and decoding covert communications using images of people's faces. We are really proud of our model, as it is the first model industrialized specifically as a security approach for document

portrait authentication. With the use of a steganographic representation. The two steps that make up StegoFace are: Input/Output Devices (Encoder/Decoder).

Recurrent Proposal Network (RPN)
It is possible to anticipate both object boundaries and non-object scores at each place with the help of a Region (RPN), which is a fully The RPN receives consistent training to ensure it suggests reliable regions. RPN was created to effectively foresee regional proposals across scales and aspect ratios. As a point of reference, RPN employs an anchor box that can be resized and reshaped to fit a variety of screen sizes. In order to avoid enumerating images or filters that have numerous scales or aspect ratios, this technique might be thought of as a pyramid of recursive references.

Binary Error-Correcting Codes algorithm
During encoding, a Binary Error-Correcting Codes procedure is used to adapt a secret message into a binary message. At the decoding stage, the same Binary Error-Correcting Code method converts the binary communication to a string containing the secret message.

Deep Convolutional Auto Encoder
The encoder network constitutes the initial stage of the generator. An encoder's training should aim to maximise the perceptual accuracy with which the input image may be recovered. The skill of a decoder to unearth coded information. As initial input, the encoder is given a picture of a face and a coded message. In order to create the encoded face image, the encoder application's final step is to use the pre-trained encoder perfect to embed the message into the cropped face. To prevent identity theft, the encoded cropped image is printed on ID cards instead of the original facial image.

Deep Convolutional Auto Decoder
Face photos can be used to send and receive secret messages, and decoders are created to decipher those signals. A digital camera is used by the decoder to read the encoded picture of your face on your ID card. The encoded portion of the face image is subsequently received by the StegoFace decoder network, where a face recognition module deciphers it to reveal the secret message. As a final step, the received message is checked using checksum verification technique to ensure its authenticity and give a means of verifying the identity and integrity of the MRTD face portrait..

Modules Description
Generator Control Panel
Here, a government official uploads an ID card to Auto Encoder using the StegoFace web interface. The encoder begins by taking in both the face image and the coded message. Using a face detection model, the important part of the image is found and clipped. In parallel, a binary error correcting coding algorithm encodes the confidential message. The encoded secret message in the face photograph is resistant to noise and other distortions introduced by the image carrier. For this purpose, the decoder learns the settings of a noise simulation module. A digital camera on a ubiquitous mobile device can capture this invisible message, which can then be recognised and processed by a validation algorithm employing deep learning techniques.

Verifier Control Panel
In this step, a verifier accesses the StegoFace web dashboard and sends a photo ID to the decoder in the vehicle. Using the mobile device's camera, the decoding procedure begins by capturing an image of the document before afterwards identifying and cropping the image to remove the encoded area (portrait). The input from the decoder network is the trimmed encoded face, and it outputs the original binary her message. After that, the same binary -correcting coding technique transforms the binary message into a string holding the hidden message. We then check the fidelity of the portrait by analysing the recovered messages.

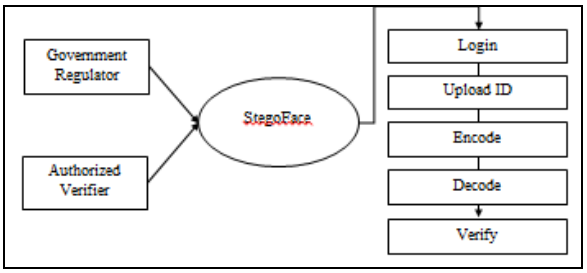

Fig. 1 Stego Face

Preprocessing Module
Processing times are cut in half and flawless matches are more likely with proper image pretreatment. To facilitate encoding, facial photographs are preprocessed. A preprocessing engine gathers features from the raw geometry of the cover and a hidden image before processing it. Redundant information is common in high-

resolution photographs; eliminating it by extracting the most relevant attributes helps keep the embedding network running smoothly. The input size should have the form m/m/n, where m, m, and n are the width, height, and depth, respectively. The m signifies that both the width and the height are identical. Any size secret image is acceptable as input. Seeing as how the cover and secret images need to be the same dimensions, the preprocessing programme scales the secret image down to 256x256. Use the skimage library's resize method to scale both the cover and secret images down to a fixed size of 256 by 256 pixels. The preprocessing engine takes the input images and transforms them from gradients to useful functions that the embedding network may use. The input layer is followed by three convolutional layers, each with progressively more filters, making up the preprocessing module. Depending on the use case, one may select a different number of filters, filter size, and increment. In order to accomplish this, the preprocessing module uses convolution layers of varying filter sizes to extract useful and relevant characteristics. At first, a smaller filter size is used to extract lower-level local characteristics like edges. To facilitate the model's acquisition of more sophisticated features, the filter size is enlarged. Eight, sixteen, and thirty-two filters are employed. Both the cover and the hidden image go through the simultaneously. As a final step, a merge layer is created to combine the features learned from the cover and hidden images.

Face Detection

If we want to use facial photos with concealed messages as part of a secure identity verification procedure, we'll need a facial recognition model to figure out where part of the face contains the secret information. Facial recognition models need to be transparent about which facial features are being used for encoding. The OpenCV Toolkit makes it simple to implement the time-tested method of Region Proposal Network (RPN) for any number of useful detection tasks. In addition, PRnet offers a unified service for identifying people in images by analysing their poses and detecting occlusions and stance changes. Then, decide on a PRnet technique that excels in these areas. Finally, we converted the model to TensorFlow-Lite format for embedding in web apps, which greatly improved the network's performance and allowed us to reduce its size without sacrificing accuracy.

1) Read Pictures:

The image reader will upload the ID card to the website for further processing.

2) Detect Faces:

Initiating a Regional Proposal Network (RPN) In a first step, a Region Proposal Network (RPN) uses a backbone convolutional neural network to analyse an input image. It all starts with scaling the input image so that the straight side is 600 pixels and the longest side is no more than 1000 pixels. The output feature of the backbone network is normally significantly smaller than the input image, though this varies depending on the stride of the backbone network. Networks must be trained to determine whether or not an item exists at each pixel in the input image in order to generate an accurate output feature map, as well as an estimate of the size of that object. To do this, the input image is labelled with a set of 'anchors' corresponding to each site on the backbone network's output feature map. Objects of varying sizes and aspect ratios can be indicated by these anchors. The network iteratively examines if the k matching anchors across the input image include the item as it moves from pixel to pixel in the output feature map, and if not, it modifies the anchors' coordinates to ensure that the bounding box accurately represents the object. The term "object proposition" or "object of interest" is more appropriate.

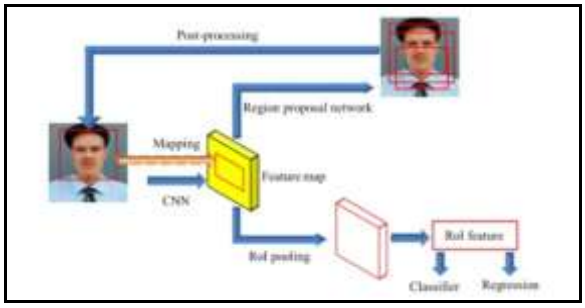

Fig. 2

Segmenting a face from its background is a necessary step in performing face detection. Before running the recognition algorithm, it is necessary to choose sub-regions (patches) of the image in order to locate the face. Region Proposal Network is used to generate these sub-regions. The region proposal network uses a convolutional layer and a ReLU activation to refine the feature maps supplied by the head network. The input and output channels of this convolutional layer are both 512. Background

and foreground class scores and probabilities, as well as bounding box regression coefficients, are generated by feeding this output into two (1,1) layers. In order to do their respective jobs, RPN networks generate prospective RoIs, whereas classification networks award object class ratings to each RoI. Consequently, the ground truth annotations, or the coordinates of the bounding boxes around the faces in a picture, are necessary for training this network. For the truth, we turn to the image dataset. The dataset's annotation file specifies the location of the object's bounding box and assigns a class name to each object seen in the images. Anchor Generation Layer and Region Proposal Layer make up the region proposal network.

3) Create Boxes around Faces:

Place white squares around the identified faces in the image. Anchorage Layer of Generational Anchors The anchors that are generated by this layer are bounding boxes that range in size and aspect ratio. The majority of the anchors won't be dispersed throughout the image in a way that allows them to surround the foreground objects (faces). The RPN network's focus is on learning to locate the face-enclosing anchors and determining the appropriate regression coefficients. Once an anchor is located, it is converted to a bounding box that more closely matches the shape of the face. We employ 4, 8, 16, and 32-by-anchors with scales of 0.5, 1, and 2. This means that each image grid has a total of 12 anchors. Where 16 is the sub sample ratio, a total of W x H x 12 anchors are formed. Anchors that were found to be outside of the image's borders have been removed.

4) Regional Proposal Level:

A "region proposal" that generates a sparse or dense feature set serves as input to the proposed system. This method use a region proposal network to rank candidate regions based on how likely it is that the region contains a face, and a sliding window methodology to generate a large number of regions from which to choose. The region proposal layer is responsible for classifying anchors as either background or foreground, and then using regression coefficients to adjust the foreground anchors so that they lie within the proposed regions' borders. The three parts that make up a region proposal layer are the proposal layer, the anchor target layer, and the proposal target layer. In order to limit the amount of anchors, the proposal layer uses non-maximal suppression based on the foreground score and then uses regression coefficients to output the altered bounding boxes generated by the anchor creation layer. To do. As the name implies, the anchor target layer is responsible for selecting talented anchors that may be utilised to train an RPN network to differentiate between foreground and background areas and produce good bounding box regression constants for foreground boxes. Foreground anchors are transformed to better fit face regions, and the network determines whether they are in the background or forefront based on the formulation of the RPN loss.

RPN Loss = Classification Loss + Bounding Box Regression Loss

Misclassified boxes are punished by the cross-entropy loss in the classification loss, while the distance between the actual and predicted regression coefficients is used in the regression loss. From the pool of potential ROIs generated by the proposal layer, the target layer chooses those with the highest likelihood of success. Using the feature map generated by the main layer, these potential ROIs are pooled together to form the RoI and passed on to the rest of the network, where they are utilised to derive the projected class values and the box regression coefficients. RoI pooling's primary goal is to train the entire system from input to output more quickly by reducing the encoding/decoding phase. To do this, the convolutional ROIs generated by the proposed target layer are retrieved. Once the feature maps have been recovered, they are sent to the rest of the network, where they are used to calculate the probabilities of various object classes and the corresponding regression coefficients for each ROI.

Cropper

It is possible to encode only the parts of the image containing faces. The face body is cropped from the original image with coordinates (0, 90) to (290, 450)..

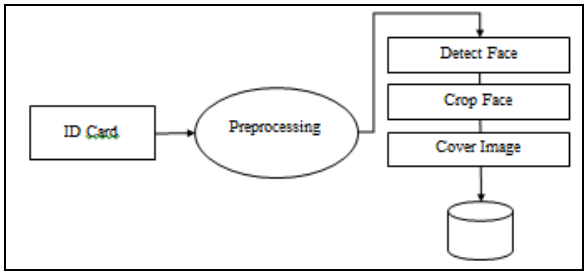

Fig. 3 Preprocessing

BECC Translator

A message can be transmitted as a binary number using the Binary Error Correction Code (BECC), which allows the original message to be recovered even if some bits are flipped during transmission. His ECC is particularly useful in preventing data corruption in storage, so you'll find them anywhere you send or receive messages. BECC comes in two varieties (Error Correcting Code)

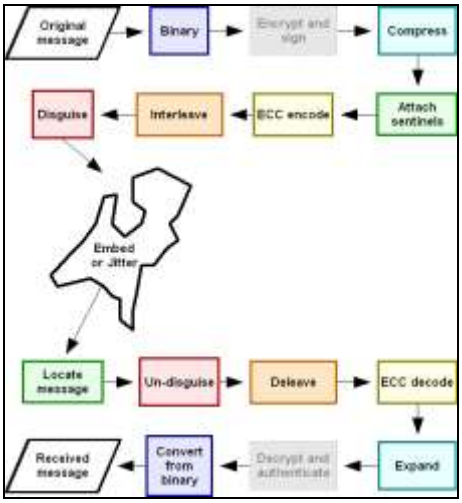

Fig. 4

1) Block Code:

Messages are contained in block codes in fixed-size blocks of bits. Add redundancy bits to correct and detect errors.

2) Convolutional Code:

The message consists of a data stream of random length and the parity symbols are produced by sliding a Boolean function on the data stream. Hamming code technology is used for error correction.

3) Hamming Code:

Hamming code is an example of block code. This code detects two concurrent bit errors and corrects a single bit error. In the Hamming encoding mechanism, the sender adds insignificant bits to the data to encode the message. These bits are added at specific positions in the message as they are extra bits for correction.

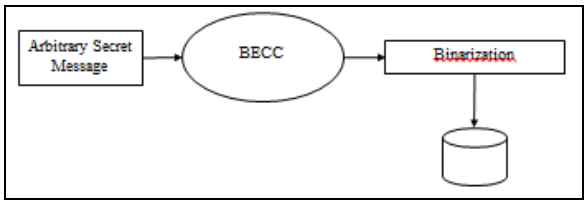

Fig. 5 BECC

Deep Convolutional ID Face Steganography

1) **Auto Encoder**:

The encoder network constitutes the initial stage of the generator. Encoder training aims to maximise the effectiveness of the decoder in extracting hidden messages while also maximising the likelihood of successful recovery of the input image's perceptual attributes. The selected encoder network architecture is based on UNet, but without the pooling layer. Saves sensitive data from

encrypted messages that could be erased during network training. Therefore, it accepts as input an aligned face and a random binary her message, and outputs message (upsampling) such that it fits the input size. Then, the encoder takes the face image as input and runs it through its various processing stages. Since the encoder lacks pooling layers, we need to take care when designing its architecture, fine-tuning the convolution's parameters by hand. Invalid layer connection.

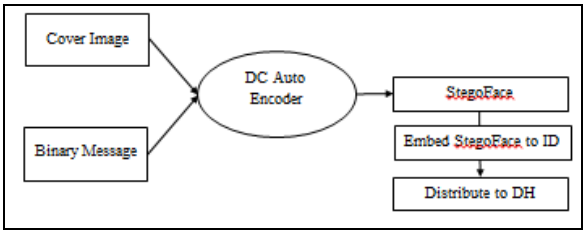

Fig. 6 DC Auto Encoder

Autoencoder architecture informs the joint development of the preprocessing unit and the embedding network. The preprocessing module and the embedding network both have an hourglass shape with growth and shrinkage phases. When given an input, the encoder component of an autoencoder network is used to extract features. An autoencoder's latent space is a feature representation of the input. Autoencoders have a decoder component that is used to retrieve the original image from the hidden representation. Applications of image steganography do not necessitate resizing. It is recommended that the latent space be a feature representation of both the cover and hidden images. The merged features from the preprocessing module are sent into an embedding network, which then creates a latent space and uses it to reconstruct a stegoface (similar to a cover picture). The entire secret image is concealed within the cover art, using every conceivable inch of space. Two convolutional layers with progressively more filters make up the embedding network's architecture. Finer-grained characteristics of both the cover and the combined secret picture are represented by the latent space at the encoder's output. As there is no need to transform dimensions, the decoder portion of the embedding network is only five convolutional layers deep. Both the encoder and decoder halves of the embedding network have large numbers of filters: 64 and 128 in the former case, and 32, 16, and 8 in the latter. Added after the convolutional layer, the ReLU activation introduces linearity via the specification of a maximum positive value and a negative zero. Training performance is enhanced by using ReLU since it solves the vanishing gradient problem prevalent in multi-layer designs. An expression for ReLU of the form h(c) D max is possible (0,c). An additional convolutional layer with three filters is added at the end of the embedding network to reduce the 2562568 feature vector to a 2562563 stego picture.

2) Auto Decoder:

When the images have been subjected to noise, the decoder network is included into the overall design. The face image is encoded with information, and the decoder is made to extract that information. To improve the network's performance, RPN is used to crop off the right area and normalise its scale. With the use of a learnable affine transformation and subsequent interpolation, it eliminates the spatial invariance of the encoded images.

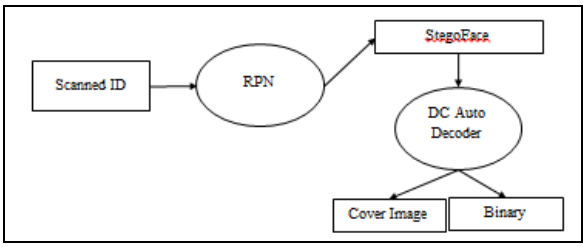

Fig. 7 RPN

When using DCAD, the RPN block comes first. The determination of the extraction network is to retrieve the stego image's hidden secret. Experiments show that hidden picture extraction with little information loss is best achieved by using a network architecture identical to the embedding network. There are growth and decline phases for the extraction network. The experimental outcomes are used to fine-tune the hyperparameters, such as the number of filters, filter size, stride, etc. This paper details the architecture that led to the best outcome. Five

layers with a growing number of filters make up the expanding encoder in the extraction network (8, 16, 32, 64, 128). Five convolutional layers with progressively fewer filters make up the decoder component (128, 64, 32, 16, 8). An ReLU activation function is built into each of the layers. To create the retrieved secret image, the decoder of the extraction network is shadowed by a convolutional layer with three filters.

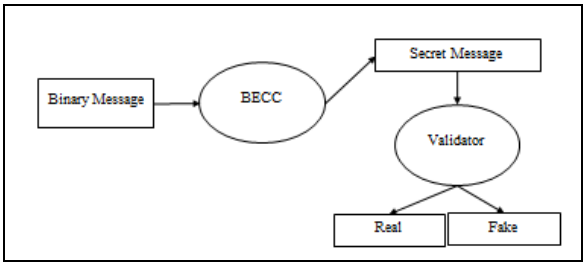

Fig. 8 BECC – Validator

Loss Function

The StegoFace decoder receives all signals sent forth by the StegoFace generator. As a means of boosting the model's efficiency, the decoder is built with a collection of loss functions. We use LPIPS (Learned Perceptual Image Patch Similarity) and face embedding as our primary loss functions. Instead of using a single input image and a single output image to reconstruct an image, the image steganography procedure uses two input images and two output images. As a result, a standard loss function might not work. To improve the architecture's efficiency, we provide a novel, tuned loss function. The embedding loss and the extraction loss must be computed. When an embedding network is used to create a StegoFace from a cover image, the embedding loss is determined. However, the extracted secret image is compared to the original secret image in order to determine the extraction loss. When the loss associated with embedding and the loss associated with extracting are added together, we get the total loss. Let the hidden picture I be generated by the embedding network, and the reconstructed cover image I serve as examples. Let h be the hidden image, and h' be the hidden image after being processed by the extraction network. The loss function needs to be adjusted so that it encourages the model to maximise its learning potential. In back-propagation, the model receives input about its performance at the end of each training epoch in the form of a loss. Equation 1 describes the embedding network loss, Lemb, and Equation 2 describes the extraction network loss, Lext.

$$L_{emb} = |i - i'|$$
$$L_{ext} = |h - h'|$$
$$L = L_{emb} + \alpha * L_{ext} = |i - i'| + \alpha * |h - h'|$$

Fig. 9 Loss Function Formula

Where α is the fitting error, which has been set to 0.3. In the first set of studies, we tried out a range of values from 0.3 to 0.6 to 0.9. Loss increases as grows; 0.3 was the best loss value observed. The loss function from the embedding network is sent back to it, and the overall loss is sent back to the extraction network so that the latter can try to recover as much of the original secret image as possible while minimising distortion..

2. Result Analysis

This web software uses a number of cutting-edge technologies to make it simple for businesses to verify the identification documents of their employees from a distance. The widespread manipulation and falsification of national identity cards is a problem that this technology has the potential to address.

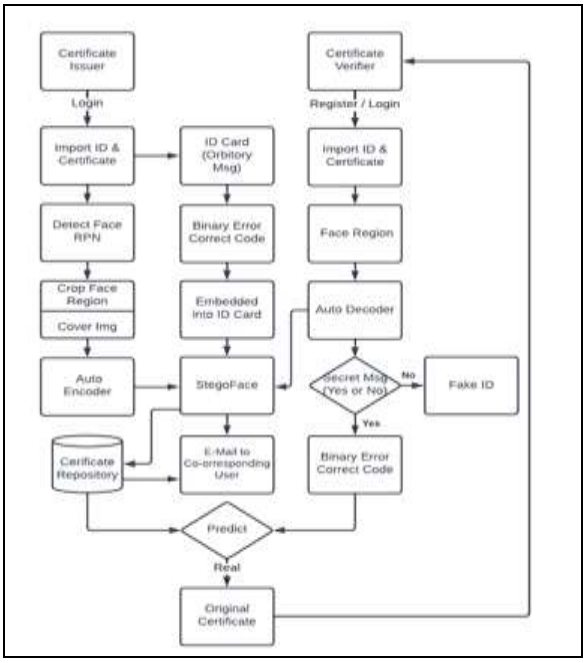

Fig. 10 Data Flow Diagram of Proposed System

TABLE I Registration Table

Field	**Type**	**Constraint**
id	int(11)	Primary Key
name	varchar(20)	Not Null
mobile	bigint(20)	Not Null
email	varchar(40)	Not Null
uname	varchar(20)	Unique
pass	varchar(20)	Not Null

Fig. 11 Face Detection using RNN

Fig. 12 Facial Image Cropping

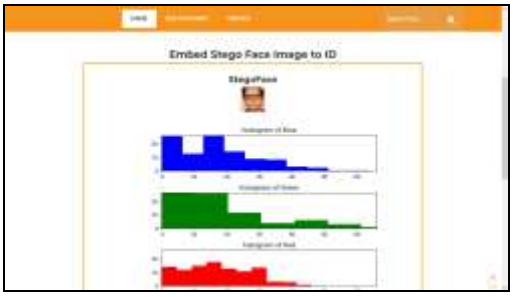

Fig. 13 Embedding Facial Image into ID

Fig. 14 Verifier Login

Fig. 15 Verifying using BECC Algorithm

Fig. 16 Final Result

3. Conclusion

This white paper is centred on concealment the security-encoded information found in ID and MRTD documents while still allowing for portrait integrity checking. Given this, we provide StegoFace, the first practical steganography solution tailored specifically for the face photos typically found on IDs and MRTDs. StegoFace is a full-stack network that uses a deep convolutional autoencoder to create encoded images with concealed secret messages in face portraits and a deep convolutional autodecoder to decipher those messages. increase. Even if the image was printed and subsequently scanned, the message is still an encoded image. StegoFace surpasses current practises by enabling the use of photographs in their original settings, independent of their origin.

4. References

C. A. Sari, G. Ardiansyah, and E. H. Rachmawanto, ``An improved security and message capacity using AES and Huffman coding on image steganography," Telkomnika, vol. 17, no. 5, pp. 2400_2409, 2019.

N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," IEEE Security Privacy, vol. 1, no. 3, pp. 32–44, May 2003.

Mielikainen, "LSB matching revisited," IEEE Signal Process. Lett., vol. 13, no. 5, pp. 285–287, May 2006.

V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," IEEE Trans. Inf. Forensics Security, vol. 11, no. 2, pp. 221–234, Feb. 2016.

V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS), Dec. 2012, pp. 234–239.

T. Pevn`y, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in Proc. Int. Workshop Inf. Hiding. Berlin, Germany: Springer, 2010, pp. 161–177.

B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 4206–4210.

Arora, M. P. Singh, P. Thakral, and N. Jarwal, ``Image steganography using enhanced LSB substitution technique," in Proc. 4th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC), 2016, pp. 386_389.

R. Kaur, A. Thakur, H. S. Saini, and R. Kumar, ``Enhanced steganographic method preserving base quality of information using LSB, parity and spread spectrum technique," in Proc. 5th Int. Conf. Adv. Comput. Commun. Technol., Feb. 2015, pp. 148_152.

Kumar and K. K. KM, ``Enhanced LSB algorithm for stegano communication," J. Web Develop. Web Designing, vol. 1, no. 3, 2016.

N. Kaur and A. Kaur, ``Art of steganography," Int. J. Adv. Trends Comput. App. (IJATCA), vol. 4, no. 2, pp. 30_33. Feb. 2017.

Ali, L. E. George, A. A. Zaidan, and M. R. Mokhtar, ``High capacity, transparent and secure audio steganography model based on fractal coding and chaotic map in temporal domain," Multimedia Tools Appl., vol. 77, no. 23, pp. 31487_31516, Jun. 2018.

H. Ali, M. R. Mokhtar, and L. E. George, ``Enhancing the hiding capacity of audio steganography based on block mapping," J. Theor. Appl. Inf. Technol., vol. 95, no. 7, pp. 1441_1448, Apr. 2017.

A. Alsabhany, F. Ridzuan, and A. H. Azni, ``The adaptive multilevel phase coding method in audio steganography," IEEE Access, vol. 7, pp. 129291_129306, 2019.

M. H. N. Azam, F. Ridzuan, M. N. S. M. Sayuti, and A. A. Alsabhany, ``Balancing the trade-off between capacity and imperceptibility for least significant bit audio steganography method: A new parameter," in Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS), Nov. 2019, pp. 48_53.

R. Chakraborty and A. Roy, ``Audio steganography_A review," Int. J. Trend Res. Develop., vol. 6, no. 3, pp. 144_149, Jun. 2019. [Online]. Available: http://www.ijtrd.com

P. Bhitre and M. R. Sayankar, ``A review on audio and video based steganography for data hiding," IJSRSET, vol. 4, no. 1, pp. 2394_4099, 2018.

Bhowal, ``Multilevel steganography to improve secret communication," in Digital Image and Video Watermarking and Steganography. London, U.K.: IntechOpen, 2019.

Singh, N., & Gupta, P. (2016). ``A Review of Steganography in Digital Images". International Journal of Computer Applications, 148(7), 17-20.

Madhu, K., Kumar, A., & Kumar, S. (2018). ``Face Recognition Using Steganography Technique". International Journal of Engineering and Technology, 7(3), 292-295.

Naveen Kumar, B. R., Padmavathi, G., & Girish, K. (2017). ``Face Recognition and Counterfeit Detection Using Steganography". International Journal of Pure and Applied Mathematics, 116(11), 97-104.

Agarwal, A., Agarwal, M., & Agarwal, M. (2020). ``An Improved Technique for Image

Steganography and its Application in National Identity Cards". International Journal of Advanced Science and Technology, 29(6), 6456-6463.