Forensic face image generation and recognition

Adithya VIKRAM, Anusha CHANDRASEKARAN, S. K. SRINITHYEE, R. PRIYADHARSINI

Department of Computer Science and Engineering,

Sri Sivasubramaniya Nadar College of Engineering,

Chennai, Tamil Nadu, India

priyadharsinir@ssn.edu.in

Abstract: One of the toughest Heterogeneous face recognition scenarios, involving the comparison of face images residing in different modalities, is face photo-sketch recognition. In forensic science, face sketch will be used for identification of criminals based on the description provided by the eyewitness. With the latest technologies available, the face sketches are obsolete, and they do not accurately support recognition and identification of criminals. In this work, we propose a system that generates the image of the suspect using the novel Transparent Latent Generative adversarial network (TL-GAN) is proposed. The generated image can be compared with images from the police database, using the DeepFace facial recognition system. The performance of the system has been compared with various CNN models and proved to be performing well.

Keywords: GAN, DeepFace, Deep Learning, Face Generation, Face Recognition.

1. Introduction

As the population increases, it is no surprise that crime rates also increase proportionally. But along with crime, the number of witnesses also increase. Hand-drawn face sketches are still considered the most reliable way to identify a criminal based on witness statements. But hand drawing face sketches is a time-consuming process that requires more than a few extra steps to be used for face recognition, as comparing a sketch to an image is no easy task. Thus, a system where a random face image is generated by a PG-GAN model is suggested, which allows further changes to be made to the face with the help of buttons corresponding to a feature which increment or decrement the prevalence of that feature. The UI is easy to use and doesn't require any previous knowledge to make use of. This allows as much as possible to be extracted from witness statements. A novel model, TL-GAN, is used for the modification of faces. Once satisfied with the modifications, the image is directly compared with existing criminal databases in a one-to-many pattern to find the closest match to the target. Algorithms from the DeepFace framework are used for the face recognition process.

2. Literature survey

An automatic generation of faces using a newly designed image generation model was explored by many researchers (Zhang et al., 2019). This model utilizes Deep Convolutional Neural Network and Deep Transpose Convolutional Neural Network as key components of the GAN. This proposed model was trained using the CelebA dataset from The Chinese University of Hong Kong. The notable drawbacks of the proposed methods had very long training times and large amounts of data. The possibility of using Transparent Latent Space GAN to produce fine-tunable face images that can be used to aid Law Enforcement and Forensics was explored (Jalan et al., 2020). The proposed method utilizes StyleGAN to generate images and Transparent Latent-Space GAN for tuning features. It was trained using CelebA dataset, FFHQ (Flickr Faces High Quality) Dataset. The disadvantage of this approach was that it produced images that had noticeable distortion and background wrapping. It produced a lot of noise in the output image. To bridge the gap between the publicly available face recognition system and the state-of-the-art system, an Openface face recognition library was presented (Amos et al., 2016). Facial recognition process by leveraging transfer learning of a pre-trained neural network has been done (Khan et al., 2019). They spun up a self-made database with the help of MATLAB and various other tools. The key features of their methodology include Haar Cascade for face detection and CNN AlexNet for face recognition. The drawback of this method was that a large database was required. Face verification and

https://doi.org/10.33436/v32i3y202207

identification of a specific face within a set of available faces in the database has been done (Heidari & Fouladi-Ghaleh, 2020). The dataset used was Labeled Faces in the Wild (LFW) dataset. The paper proposed a method to use Convolutional Neural Network and pre-trained-VGG-16 in order to extract features in Siamese architecture. It however has a slow speed of convergence and requires extensive computation resources. A facial recognition model with small-scale training samples and limited computing power was proposed by (Zhigi, 2021). The model was trained on CASIA WebFace dataset. The paper uses a methodology called Microface which is a modified VGGNET algorithm that uses VGG-16 network. This wasn't chosen as a method to implement and enhance because it has a lower accuracy rate than DeepFace and DeepID2. Matching of query face image with the template face image using AT and T dataset was done by Tun et. al. (2019). The methodology proposed was however novel and more of a combination of two existing models. It uses hybrid methods of the ZCA feature extraction and artificial neural network, which loses the spatial information while processing. In the work by (Ahdid et al., 2017) two feature extraction methods were used. Euclidean distance and Geodesic distance are computed for facial feature point detection. FaceNet, a deep neural network, was proposed to map the face images in the Euclidean soace which corresponds to the measure of face similarity (Schroff et al., 2015). The alignment and representation step in the face recognition process was replaced by 3D face modelling along with piecewise affine transformation and a nine-layer deep neural network (Taigman et al., 2014; Wang et al., 2021).

3. Proposed system

This system consists of two modules: The Face Image Generation Module and the Face Image Recognition Module. The Face Image Generation Module generates a face and allows changes of facial features while the Face Image Recognition Module matches the face with existing criminal databases. The TL-GAN is used for generation and modification of images, and DeepFace framework's various algorithms is used for facial recognition.

A. System Flow

The application generates a random face image, which can be modified according to the features of the target based on witness statements, with the help of a simple user interface. The generated image is closely matched with existing criminal databases and outputs the image which is most likely the target along with the similarity measure.

B. Data Set

The dataset used in this work for face recognition is CelebFaces Attributes Dataset (CelebA) which has more than 200K celebrity images with different face attributes.

C. Face Image Generation Module

Generative Adversarial Network (GAN), or GANs, is a deep-learning-based generative model that is used for Image Generation in this module. Figure 1 shows the architecture of a GAN. GAN consists of two important entities - the Generator Model and the Discriminant Model. The Generator Model captures the distribution of sample data and generates a sample similar to the training data with noise, following distributions, like Gaussian distribution. The Discriminant Model is a classifier that estimates the probability of deriving the samples from the training data. It differentiates the images taken from training data and those generated by the generator model. If the sample is taken from the training data, the probability is higher. The best-known extension of GAN is the Progressive Growing GAN (PGGAN), which was developed by NVIDIA. It produces realistic images that are almost identical to original faces of people who exist in real life.

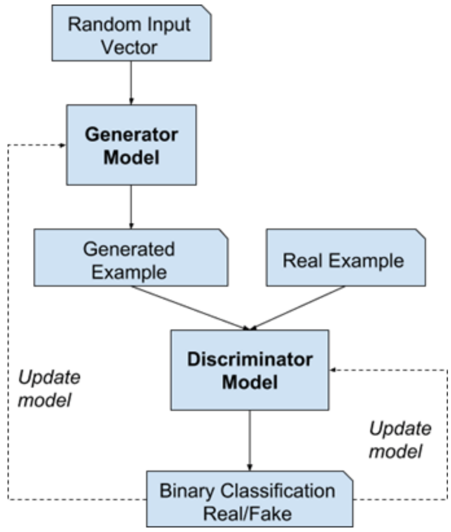


Figure 1. GAN architecture flow diagram

The Transparent Latent-space GAN (TL-GAN), is a single network, that can work with one or multiple features. It achieves this by making the noise in the latent space transparent. All the features of a 1024 px * 1024 px image generated by the PG-GAN are determined solely by a 512-dimensional noise vector given as input in the latent space. The face generation process can be controlled by the understanding of latent space. Latent space will be generally densely populated with interpolation between neighbourhood points in the space leading to a smooth transition of corresponding images and their features. Thus, it is possible to find directions in the latent space that correspond to certain specific features that are relevant (i.e., customized features according to the statements of the witness). The face generation process can be controlled by using the unit vectors of various directions as the feature axes.

D. Face Image Recognition Module

A link is built between the latent vector and the feature labels using supervised learning methods trained on paired (latent vector, feature labels) data to find the feature axes. In this model, the Mobilenet Convolutional neural network trained on the CelebA dataset is used as the featured extractor. The next step is to perform regression between the latent vectors given as input and the extracted features. Here, the Generalized Linear Model (GLM) is used as the regressor. After this step, the regression slope becomes the feature axes. Finally, by moving the points in the feature axes, the face image can be modified with respect to that feature.

DeepFace is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for Python. The module is primarily based on Keras and Tensorflow. Experiments by researchers show that human beings have 97.53% accuracy on facial recognition tasks. This model has been found to beat this accuracy. In this application, DeepFace will be used to perform a one-to-many face image comparison between the generated image and the existing criminal databases. The Face Recognition Pipeline is discussed below:

Face Detection:

Face Detection is the first step in comparing faces. Various libraries and frameworks provide solutions for this process with varying levels of success in different sections, with some being able to mis-detect even non-facial objects such as tie or badges as faces. Overall, SSD and MTCNN give comparatively robust results.

Face Alignment and normalization:

Face Alignment is an early stage of the modern face recognition pipeline (Schroff et al., 2015). Google declared that Face Alignment increases the accuracy of its face recognition model, FaceNet, from 98.87% to 99.63%. This is almost a 1% accuracy improvement. Dlib can find 68 different facial landmark points including chin and jaw line, eyebrows, nose, eyes and lips, as shown in Figure 2. The exact facial area based on those landmark points can be extracted beyond rough face detection. This will increase the accuracy of face recognition models dramatically because noise will be discarded.

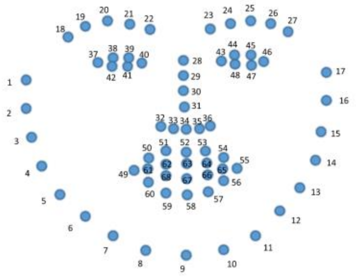


Figure 2. 68 high point landmark on a face

4. Results

A. Face Image Generation

The system's face generation module initially generates a random face image. The module consists of a user interface which the user can interact with. The user can decide to generate another random face image using a button in the user interface. Once the user is satisfied with the base face image, he/she can utilize the feature axes buttons, which are labelled GUI buttons, that provide a method to increment and decrement the values for the corresponding features axes for that particular facial feature to fine tune the face image. An increment in the values results in the facial feature becoming more prominent and vice-the-versa with the decrement button. The system has the ability to fine-tune 21 different facial features, some of which include nose, mouth, smile, hair colours, beard, skin tone etc. Once the user is satisfied with a particular feature, he/she must use the GUI button to lock the value in the feature axis. This is to prevent any unintentional changes in the value due to the correlational side effects from modifying other features. An example of this would be, when the value on the age feature axes is incremented, the value of hairline/bald feature axes will also be modified as a correlational side-effect. Some sample face images generated by the module are shown in Figure 3. below:

Figure 3. Face Images generated by our module

The value in the latent space vector can be moved along several example feature axes (gender, age, etc) which helps to smoothly morph the image between male ←→ female, young←→old, etc. An example of this is shown below in Figure 4.

Figure 4. Moving along the gender feature axis

B. Face Image Recognition

Once the face from the face image generation module is finalized, the image is exported and then processed by the face image recognition module. This image is compared with the faces in the database in a one-to-many recognition pattern. All the images below the set distance metric threshold will be found and the closest among those will be displayed as the match along with the similarity measure. An example of a target image and the corresponding match is shown below in Figure 5.

Figure 5. Target & matched face image

5. Performance analysis

A. Face Image Generation

In order to determine the subjective performance of the face image generation module, a survey-based approach was taken. Two images were shared with the survey participants, one of which is an actual human face image, and the other was a computer-generated face image, who's resemblance was the closest possible match to the target image (former image). The survey had 50 participants and the findings from the opinion polls are as follows:

1) The first question required the participants to score the similarity between the two given faces. The scores can range between 1 to 5.

Findings: The conclusion was that 50% of the participants felt that the images were strongly similar (score of 4 - 5) and 80% of them gave a score of 3 and above. This helps to indicate that the image produced by the module is extremely similar to an actual image of a human face, despite being synthetically generated (Figure 6).

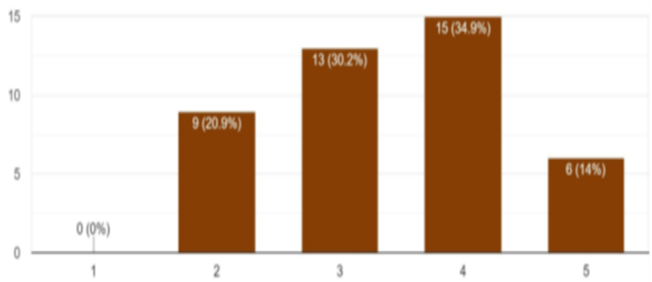


Figure 6. Bar chart representing the number of responses (Y-axis) for each rating (X-axis)

2) The second question required the participants to point out the features of similarity between the two images.

Findings: Out of the 10 true features of similarity, 73% of the participants were able to state 7 features correctly (Figure 7). This assets to the fact that the module proposed can effectively produce human-like organic modifications to existing face image. 48.8% of the participants of the survey were convinced that the two images were biologically related. This serves to prove that the module proposed succeeds in presenting images that are extremely close to the image that the witness has in her/his mind, who is obviously an actual person.

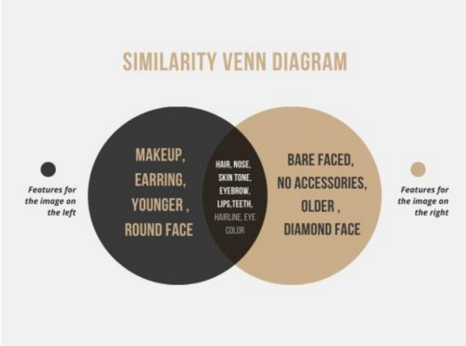


Figure 7. Venn Diagram for feature similarity

3) The third question raised the issue of there being a computer-generated image.

Findings: 61.3% of the participants were able to identify the image produced by the module to be computer generated. Most of the remaining participants felt that both the images were organic and not synthetically generated (Figure 8 and 9).

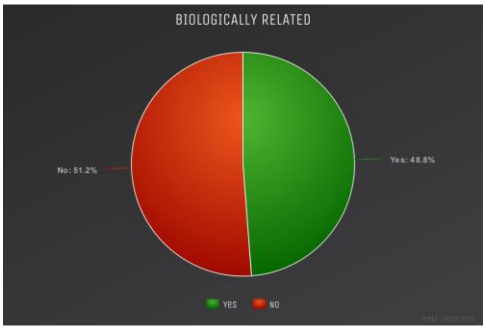


Figure 8. Pie chart representing responses for biological relations

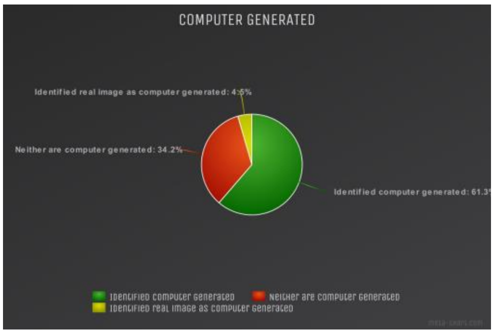


Figure 9. Pie chart representing responses for possibility of computer-generated images

B. Face Image Recognition

To analyse the performance of face recognition module, some state-of-the-art methods present in DeepFace framework are considered. FaceNet (Google), DeepFace (Facebook), VGGFace (Oxford), and OpenFace (CMU) have been considered for comparison. Among these 4 models, FaceNet gave the best result. In general, FaceNet has given better results than the other 3 models, in this case and from the present testing. FaceNet is considered to be a state-of-art model developed by Google. It is based on the inception layer. FaceNet uses inception modules in blocks to reduce the number of trainable parameters. This model takes RGB images of 160×160 and generates an embedding of size 128 for an image. Different distance metrics are used to measure the similarity between two given images.

The three metrics used are:

Euclidean Distance: The Euclidean distance can be used to calculate the distance between any two points in two-dimensional space, and also to measure the absolute distance between points in N-dimensional space. For face recognition, smaller values indicate more similar faces. The feature points representing facial features were calculated according to the Haar feature values, the image to be detected is processed by a function to obtain a 128- dimensional face feature vector. This constitutes the condition for face similarity calculation under Euclidean distance. Supposing the face feature of the image to be detected is $A = (x_1, x_2, , , x_{128})$, training sample face features are $B = (y_1, y_2, , , y_{128})$, the Euclidean distance calculation formula (1) is as follows:

$$AB = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_{128} - y_{128})^2} \tag{1}$$

The value of AB is the Euclidean distance between two points in 128 dimensions.

Euclidean L2: One way to normalize the vector is to apply some normalization to scale the vector at a length of 1 i.e., a unit norm. There are different ways to define "length" such as l1 or l2 normalization. If the l2-normalization is used the "unit norm" essentially means that if each element in the vector is squared and summed, it will be equal to 1.

So given a matrix X, where the rows represent samples and the columns represent features of the sample, L2-normalization can be applied to normalize each row to a unit norm. After this normalization, Euclidean distance is used as a similarity metric

Cosine Similarity: Consider two vectors of features, x and y; Cosine similarity measures the similarity using the cosine of the angle between two vectors in a multidimensional space. It is given by (2):

$$\text{similarity}(x, y) = \cos(\theta) = \frac{x \cdot y}{|x||y|} \tag{2}$$

where $|x|$ is the Euclidean norm of vector $x = (x_1, x_2, \ldots, x_p)$, defined as $\sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$. Conceptually, it is the length of the vector. Similarly, |y| is the Euclidean norm of vector y. A cosine value of 0 means that the two vectors are at 90 degrees of each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors.

Pair Wise Test Results: Testing involves comparing the faces in the data set in a pair wise manner and setting a distance threshold for each model and corresponding distance metric It is considered a match only when the distance falls below the acceptable threshold. The results are shown in Figure 10 and Table 1.

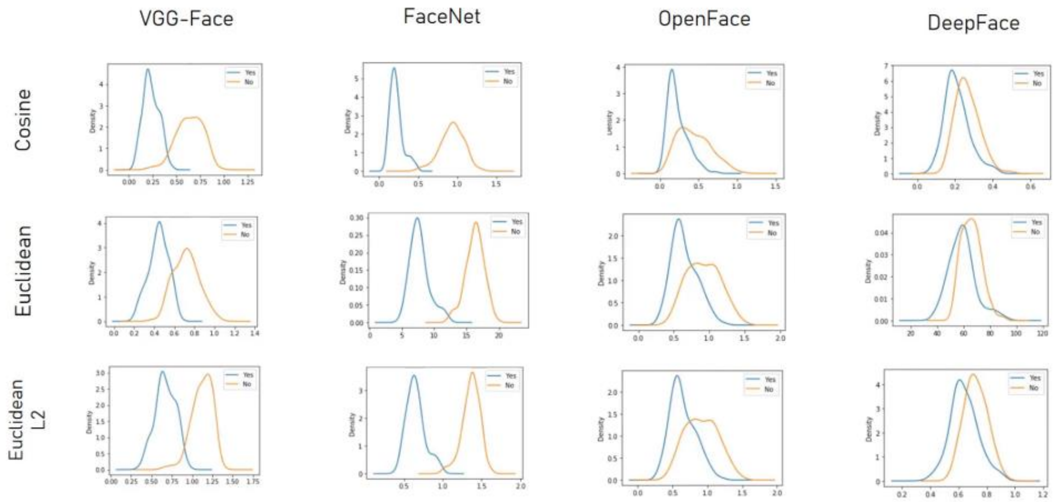


Figure 10. Discretization of positive and negative classes for different models

Table 1. Models and distance metrics

Models	Measures	Cosine	Euclidean	Euclidean L2
VGGFace	Threshold	0.31	0.47	0.79
	Accuracy	89.28	81.42	89.28
	Precision	97.41	97.82	97.41
	Recall	80.71	64.28	80.71
	F1	88.28	77.58	88.28
FaceNet	Threshold	0.40	11.26	0.90
	Accuracy	98.21	98.57	98.21
	Precision	100	100	100
	Recall	96.42	97.14	96.42
	F1	98.18	98.55	98.18
OpenFace	Threshold	0.11	0.47	0.47
	Accuracy	57.85	57.85	57.85
	Precision	95.83	95.83	95.83
	Recall	16.42	16.42	16.42
	F1	28.04	28.04	28.04
DeepFace	Threshold	0.13	42.21	0.51
	Accuracy	54.64	52.50	54.64
	Precision	100	100	100
	Recall	9.28	5.00	9.28
	F1	16.99	9.52	16.99

6. Conclusion and future works

The proposed system generates an image of a face, allows modifications of features based on witness statements and matches the face with existing criminal databases to find the closest match. The computer-generated image has about 97.2% resemblance to actual human face and the features can be made prominent in order to match the requirements of the witness. The whole application is end-to-end automated.

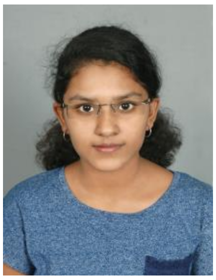
As a part of future work, the intension is to expand the model to generate faces of different races. The enhancement of the user journey is desired so that the GUI can be more user-friendly. In the interest of time, the application could not be implemented using local criminal databases. However, its performance in real time is desired to be put to the test.

REFERENCES

1. Ahdid, R., Safi, S. & Manaut, B. (2017). Euclidean and Geodesic Distance Between a Facial Feature Points in Two-dimensional Face Recognition System. *The International Arab Journal of Information Technology*, 14(4), A Special Issue, 565–571.
2. Amos, B., Ludwiczuk, B. & Satyanarayanan, M. (2016). *OpenFace: A general-purpose face recognition library with mobile applications*. Technical report, CMU-CS-16-118, CMU School of Computer Science.
3. Heidari, M. & Fouladi-Ghaleh, K. (2020). Using Siamese Networks with Transfer Learning for Face Recognition on Small-Samples Datasets. *International Conference on Machine Vision and Image Processing (MVIP)*, pp. 1-4.
4. Jalan, H. J., Maurya, G., Corda, C., Dsouza, S. & Panchal, D. (2020). Suspect Face Generation. *International Conference on Communication System, Computing and IT Applications (CSCITA)*, pp. 73-78.

5. Khan, S., Ahmed, E., Javed, M. H., Shah, S. A. & Ali, S. U. (2019). Transfer Learning of a Neural Network Using Deep Learning to Perform Face Recognition. *International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1-5.
6. Schroff, F., Kalenichenko, D. & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823.
7. Taigman, Y., Yang, M., Ranzato M. & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701-1708.
8. Tun, N. M., Gavrilov, A. I. & Paing, P. P. (2019). Human Face Recognition Using Combination of ZCA Feature Extraction method and Deep Neural Network. *International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 1-5.
9. Wang, M. & Weihong, D. (2021). DeepFace Recognition: A Survey. *Neurocomputing*, 429, 215-244.
10. Zhang, T., Tian, W., Zheng, T., Li, Z., Du, X. & Li, F. (2019). Realistic Face Image Generation Based on Generative Adversarial Network. *International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 303-306.
11. Zhiqi, Y. (2021). Face recognition based on improved VGGNET convolutional neural network. *IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2530- 2533.

Adithya VIKRAM obtained his Bachelor's Degree in Computer Science and Engineering in 2022, at Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India. He is currently working as a Software Engineer at PayPal. His areas of research interests includes: Digital Image Processing and Computer Vision.



Anusha CHANDRASEKARAN obtained her Bachelor's Degree in Computer Science and Engineering in 2022, at Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India. She is currently working as a Technology Analyst at Citibank India. Her areas of research interests includes: Digital Image Processing and Software Engineering.

S. K. SRINITHYEE obtained her Bachelor's Degree in Computer Science and Engineering in 2022, at Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu. She is currently a Software Engineer at Walmart Global Tech India. Her areas of research interest includes: image processing and Computer Vision.

R. PRIYADHARSINI is working as an Associate Professor in the Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu. She received her Ph.D. in Information and Communication Engineering from Anna University. Her areas of research interest includes: underwater acoustic image processing, Computer Vision, and Data Analytics.