International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-1, January 2022
https://doi.org/10.55524/ijircst.2022.10.1.12
Article ID IJIRD-1157, Pages 73-78
www.ijircst.org

Mitigation and Identification of Camouflage Attack Over Computer Vision Applications

Chandra Sekhar Sanaboina[1], and Sree Bharathi Garikiparti[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Kakinada, India.
[2] PG Student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Kakinada, India.

Correspondence should be addressed to Chandra Sekhar Sanaboina; chandrasekhar.s@jntucek.ac.in

ABSTRACT- Computer vision technologies are now commonly used in real-time image and video recognition applications using deep neural networks. Scaling or Mitigation is the basic input pre-processing feature in these implementations. Image scaling methods are designed to maintain visual elements before and after scaling, and are widely utilized in a variety of visual and image processing applications. Content disguising attacks may be carried out by taking advantage of the picture scaling mechanism, which causes the machine's extracted content to be drastically different from the input before scaling. In this project the actual input image with dimension Am×n is resized to an attacked image dimension Om1×n1, which will mislead the classifier as the input is attacked image. To illustrate the threats from such camouflage attacks, some computer vision applications taken as targeted victims that includes multiple image classification applications based on popular deep learning frameworks like OpenCV, pillow. This attack is not detected while checking with the YOLOV4 demo. To defend against such attacks, in this project using Dhash i.e., difference hash concept. The hash the value of the actual image and the resized are same when using the Dhash. If the image is camouflaged the means the content is modified so the Dhash value of the image also changed if the image is not camouflaged the then the hash value of the actual image and resized images are same. By using the Dhash concept detecting the image is camouflaged or not.

KEYWORDS: Camouflage attack, Image Scaling, Computer Vision, Dhash

I. INTRODUCTION

As a branch of artificial intelligence (AI), computer vision is the ability of computers and systems to analyse digital photos, videos and other visual inputs in order to extract meaningful information and take action or make suggestions. Computer vision is a lot like human eyesight, except that humans have an advantage since they've been around longer. Computer vision relies heavily on data. It repeatedly analyses data in order to identify differences and, eventually, pictures. Deep learning and a convolutional neural network are used to accomplish this (CNN). Computer vision technologies are now commonly used in real-time image and video recognition applications.

Deep Neural Networks are widely used in Computer vision applications. Now a day's various pre-trained Deep Neural Network models are available for developers to implement the computer vision applications. Most of these models have fixed input layers. The input sizes of the mostly used Deep Neural Networks are shown in Table1. In this Deep Neural Networks image scaling processing is used to handle the images from the different sources which lead to the down sampling the image called as Scaling. In many visual and image processing applications, picture scaling methods are utilized to ensure that the visual characteristics are preserved before and after scaling. Perform a camouflage assault by altering the original material using the picture scaling process. Camouflage picture refers to the foreground and background being mixed together in this context, which is one of the most important areas in machine vision. This study discusses a potential security issue with picture scaling methods, namely the camouflage attack.

Computer vision is a field of science that enables computers or devices to recognize different objects like humans, animals. The applications of computer vision applications are increasing day by day in all fields like engineering, surveillances, health care, military and so on. For this computer need to be train to detect objects, poses etc. based on the requirements. Deep Neural Networks (DNN) models widely used in computer vision applications. Deep Neural Networks (DNN) has multiple hidden layers between input and output layers. To perform detections Deep Neural Networks, need to process huge number of images from different sources. To handle the images from the different resources Deep Neural Networks (DNN) uses the scaling functions before sending to the input layers. General image processing during the image classification as shown in the fig. 1 which is carried out before sending the image as input to Deep Neural Network models.

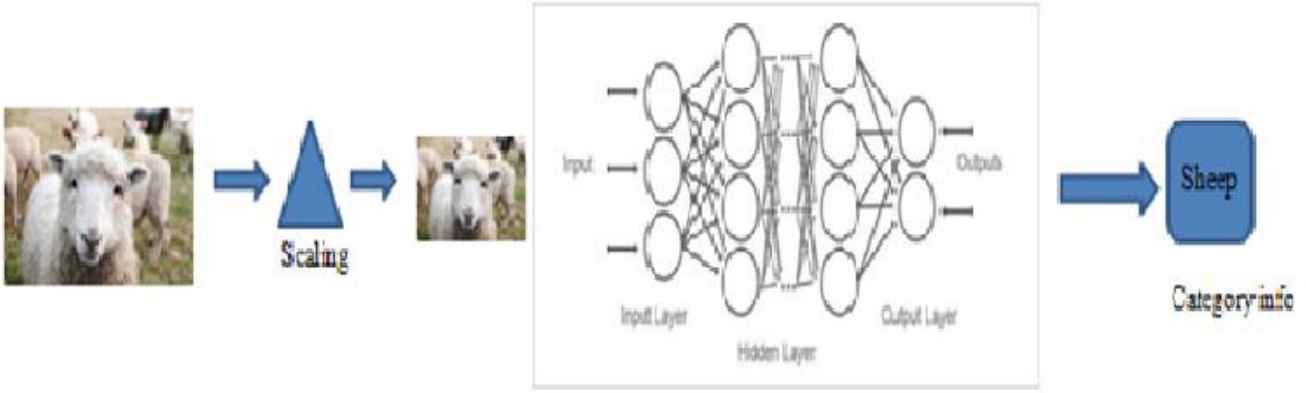

Figure 1: Image processing during the image classification

A. IMAGE SCALING

Images may be scaled by altering their dimensions. If you want to reduce the size of a picture, you may use the inverse operations of scaling down and scaling up. Processing a picture begins with scaling. Downscaling is required if the picture received from an external source is bigger than the present model size; upscaling is required if the image received is smaller. When resizing a picture, image scaling algorithms work to keep the image's aesthetic characteristics intact. Fig 2 shows the concept of image scaling.

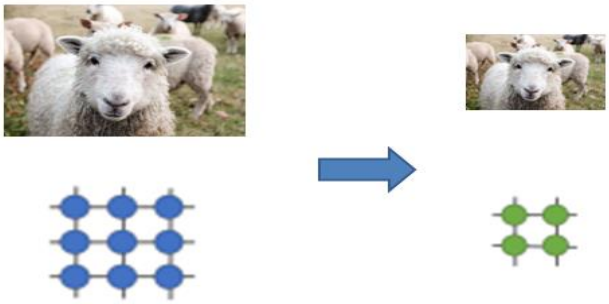

Figure 2: The concept of image scaling

B. INTERPOLATION

Each "missing point" in a scaling process is interpolated to provide an approximate value. P in the output picture is constructed using the pixels of Q11, Q12, Q21 and Q22 in the original image shown in fig. 3.

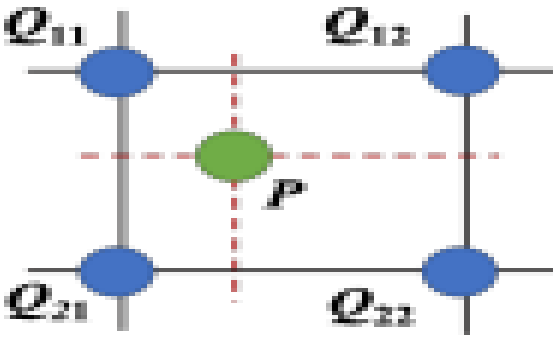

Figure 3: Selecting P using interpolation in scaling

The various interpolation techniques employ a scaling algorithm to determine which neighbour pixels should be used to generate the final image's pixels. The bilinear and bicubic algorithms, as well as the closest neighbour technique, are the most often used interpolation methods.

C. Nearest Neighbour Algorithm

Pattern recognition methods such as Nearest Neighbor are non-parametric and may be used for classification or regression. Finding a sample in training data that is similar to the upcoming sample is a straightforward process. This just replaces a single pixel (the one closest to the input) with a new one. Categorical data, such as land use or slope, works well with Nearest Neighbor. To prevent overfitting the training data, it is employed here. All the values that are entered into the grid are precisely the same. An input grid cell's closest cell centre is used to compute a value for the output cell. It is possible to utilize Nearest Neighbor on continuously streaming data, although the output may be blocky.

D. Bilinear Algorithm

Using a linear interpolation in both directions is known as a "bi-linear" interpolation method. A weighted average is calculated by comparing each of the four closest neighbors. The bilinear interpolation is used to pick the pixel that will be used to create the pixel for scaling the picture in the bilinear process. The weighted average of the four closest cell centers is used in bilinear interpolation. The closer an input cell centre is to the output cell centre, the greater the effect its value has on the output cell value. To put it another way, while the output value may deviate from the closest input, it will always fall within the same range of possibilities as the input value. Bilinear is not recommended for categorical data since the values might vary. For continuous data, such as elevation and raw slope, it should be used instead.

E. Bicubic Algorithm

A bicubic interpolation is one in which a linear interpolation is applied in two directions simultaneously. A weighted average is calculated by taking into account the 8 closest neighbors. The bilinear interpolation is used by bicubic algorithms to pick the pixel that will be used to scale the picture. Cubic Convolution finds the value by fitting a smooth curve between the points in the 16 closest cell centers to the output. When this is done, not only does it alter input values but it may also result in a value that is beyond its expected range of values (imagine a sink or a

peak occurring on a surface). Aside from categorical data, this approach is best used for smoothing continuous data.

F. Concept of Camouflage Attack

One possible cause of this attack is the downscaling of input pictures, which is what happens when the scaling function reduces the original image's size. Consider the following picture as an example of the downscaling phenomena. This is an example of how a picture gets downsized from 4 by 4 pixels to two-by-two pixels. Our content disguise attacks produce images that an attacker wants to locate in order to carry out their plan of action. The attack image's source image, or the picture that the attacker intends to use as the assault image. The machine's real output picture and the attacker's desired output image are known as the "output image" and the "target image," respectively. Figure 4 illustrates the concept of a source picture, a target image, and an attack image. Spreading a picture with offensive material is easy thanks to the source image. The target picture, on the other hand, is a regular image that may be accepted by moderators. A constant and certain distance measurements are needed for this purpose, and the output picture would not be revealed by human eyes if these distance measures are used in conjunction with a constant that trades off dissimilarity of source and target images. Even while it isn't readily apparent to the naked eye, this might mislead moderators or go undetected.

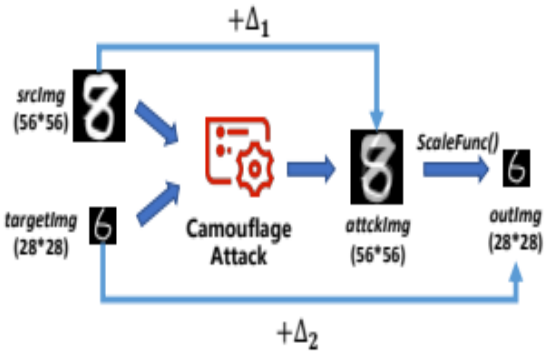

Figure 4: Automatic attack image crafting

G. Content based on distance matrices

This work examined the impact of a camouflage assault on photographs using three distinct distance matrices L0 Attack, L2 Attack, L∞ Attack

In the L0 attack distance matrices tries to reduce the number of different pixels between two images. In the L2 Attack evaluates a global distance. It tries to constrain the image "average" distortion level and L∞ Attack represents the upper bound of the pixel difference.

H. Dhash

Dhash is a Python library that generates a "difference hash" for a given image – a perceptual hash. Fig 1.6 shows the Concept of Dhash

- This converts the image to grayscale.
- It Downsize it to a 9x9 thumbnail.
- Produce a 64-bit "row hash": a 1 bit means the pixel intensity is increasing in the x direction, 0 means it's decreasing.
- Do the same to produce a 64-bit "column hash" in the y direction.
- It Combine the two values to produce the final 128-bit hash value.

II. REVIEW OF LITERATURE

- "Universal physical camouflage attacks on object detectors" [1] by L. Huang et al. includes research on physical adversarial attacks on real-world object detectors and a proposed adversarial pattern for successfully attacking all instances in the same object category, known as a "Universal Physical Camouflage Attack. " (UPC). When the classifier and regressor use this UPC, they end up with the wrong results.
- Hiding physical-world assaults using natural styles in the work "Adversarial camouflage: Hiding attacks with natural styles" by R. Duan and colleagues X. Ma and Y. Wang, J. Bailey, and AK Qin and Y. Yang [2] AdvCam was presented as a method for designing and concealing adversarial samples in real-world forms that seem authentic to human observers. Deep learning algorithms can't identify private information if you employ this technique. Other computer vision problems, such as object detection and segmentation, should be investigated using AdvCam in the future.
- Flash-based face liveness detection has been proposed by P. K. Chan and colleagues in their study "Face Liveness Detection versus 2D Spoofing Attack," [3]. In personal identification systems, face-recognition technology has been extensively used, although some of these algorithms are susceptible to being misidentified. As a result of our tests, we found that our Face Liveness detection method is effective against 2D Spoofing attacks.
- "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," by N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami [4] It has been shown that defensive distillation may improve DNN generalizability and robustness when used in conjunction with deep learning algorithms that are subject to adversarial sampling assaults. An adversarial sample-creation success rate of less than 0.5 percent may be achieved using this method on the MNIST data set.
- On "Towards Evaluating the Robustness of Neural Network," N.Carlini and D. Wagner propose a "Defensive distillation approach" that increases the robustness of an arbitrary neural network and reduces the adversarial attacks that occur on neural networks, making it difficult to use them in security-critical areas. In addition, we demonstrate that adversarial assaults may be used to assess the performance of proposed defences by proposing a strong attack that defeats defensive distillation.
- An adversarial behaviour in deep learning systems was described by N. Papernot, P. Mcdaniel, S. Jha, M Fredrikson, Z. Celik, and AA Swami in this publication "The constraints of deep learning," [6]. Along with analysing DNN adversaries' intents and capabilities, a new class of algorithms for creating adversarial samples was introduced that uses forward derivatives to compute them. Analyzing DNN outputs using adversarial saliency maps, a threat actor that is familiar with the

architecture may find out exactly which input attributes have the biggest impact on DNN outputs.

- When D. Moreira et al. published "Image Provenance Analysis at Scale," [7] they explored how large-scale provenance systems can be feasible by examining the problems with images' provenance and proposing a way for solving them. Analyze real-world Provenance using the Reddit dataset. While our suggested technique produces almost 0.8 correct findings, we still need to work on fully-automatic picture provenance analysis as a next step.
- The authors of "Invisible adversarial attack against deep neural networks: An adaptive penalization approach" [8] proposed an invisible adversarial attack for synthesising adversarial examples that are visually indistinguishable from the benign images while mounting a valid attack against DNNs and proposed the JNDp metric for better simulating the perceptual similarity and adaptively sets penalty for adversarial perceptual. By performing Lp-norm regularised with a unique spatial constraint, these assaults are able to disperse disruption in the benign picture according to human sensitivity to a local stimulus.
- Provably Secure Camouflaging Strategy for IC Protection [9] by M. Li et al. combined the two approaches low-overhead cell library and AND-tree structure to achieve exponentially rising security levels at the expense of linearly growing overhead. [9]. Measures of de-camouflaging complexity were used to demonstrate that the active learning technique is equivalent to an existing de-camouflaging strategy. The SAT-based assault is well protected by this architecture, which has a little impact on performance.
- "Computer vision, camouflage breakdown, and countershading," [10] by A. Tankus and Y. Yeshurun Introduces the Darg operator for recognising three-dimensional smooth convex objects in 3D. A comparison with an edge-based approach (radial symmetry) showed that it is especially effective at breaking camouflage, and there seem to be camouflages built expressly against it. Its neural network implementation is likewise quite simple.

III. REQUIREMENTS AND PROCEDURE

A. Requirements

OpenCV, Pillow, TensorFlow, and Numpy were utilised in the implementation. A collection of computer vision-related operations, OpenCV is primarily intended for real-time computer vision. OpenCV may be used for a variety of tasks, including image processing and deep learning. Reading a picture, scaling an image, extracting the distinct colour channels of the image, and manipulating these colour channels are some of the fundamental tasks that may be performed with this library. For image editing, python's Pillow library is a must-have. You may use Pillow, a free and open-source library for Python programming, to effortlessly produce and edit digital pictures. Tensor Flow is a Python library developed and distributed by Google for rapid numerical computation. There are wrapper libraries that ease the process of creating Deep Learning models built on top of Tensor Flow, which is a foundation library. A general-purpose array-processing software, Numpy, may be found here. Multidimensional array objects and tools for interacting with them are both included in the package.

Procedure:

Multiple modules were used in this project. The L0 attack is implemented in the first module, which is done by reducing the number of distinct pixels between two pictures. It keeps the amount of changed pixels under control, but it may also result in certain pixels that have been considerably disturbed. The L2 attack is implemented in the second module. This L2 calculates the distance across the globe. An attempt is made to limit the average distortion level of the picture by employing a differential equation. The L attack, which indicates the upper limit of the pixel difference, is implemented in the third module. The largest difference rather than the overall difference is the emphasis of the L measure. Resize techniques, estimation of a conversion matrix, an info way to show the picture meta data and some image loading and image saving methods were developed in the other modules while working on the above three assaults. The hash value of photos is calculated in the final module using the Dhash idea, which takes into account the image's rows and columns.

IV Proposed System

In the project taking the target image, source image and attack image as mandatory inputs and also taking the resize, interpolation, penalty and norm distance matrices as optional values. Based on the norm distance the output image is produced. We need to compare the source image and scaled image hash values get from the Dhash method. The hash the value of the actual image and the resized are same when using the Dhash. If the image is camouflaged the means the content is modified so the Dhash value of the image also changed if the image is not camouflaged the then the hash value of the actual image and resized images are same. By using the Dhash concept detecting the image is camouflaged or not.

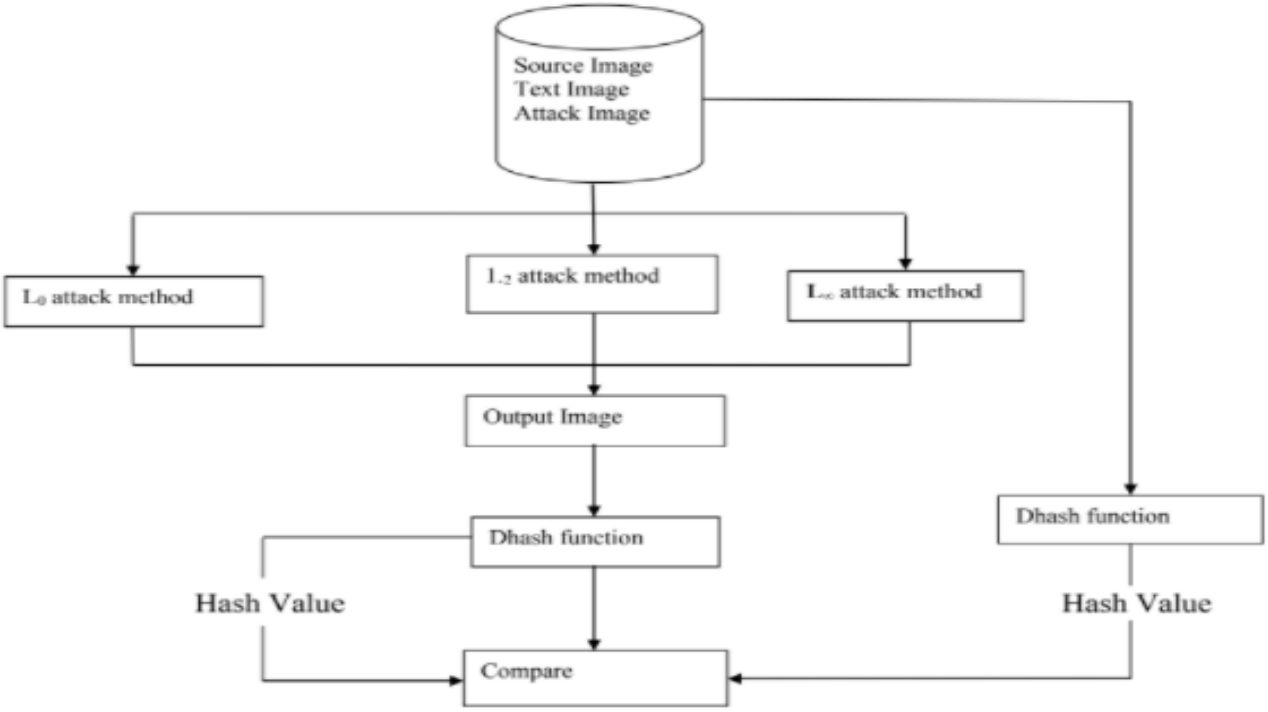

Figure 5: Dhash concept detecting the image is camouflaged or not

IV. RESULTS AND COMPARISON

To evaluate the accuracy and comparison of attack consider the time taken to perform and accuracy of the attack. The L∞ attack image strives to keep the pixel difference between the source and attack picture to a minimum. L2 attack uses the differentiable algorithm in scaling the image, which is easy to implement, but large data modifications are taking place L0 preserves the pixels.

Table 1: Comparison of attacks accuracy, time cost and no of bits changed

Attack	Frame Works	Output Image	Time Cost	Dhash no of different bits
L0	OpenCV		41.64	7
	Pillow		42.05	5
L2	OpenCV		24.86	31
	Pillow		20.96	13
L∞	OpenCV		77.38	1
	Pillow		59.04	1

The time taken to perform the attacks with the frameworks OpenCV and Pillow are represented in the graph as showed in the figure. When comparing these L2 attack perform in both and accuracy is efficient when compared with L0 and L∞ attacks.

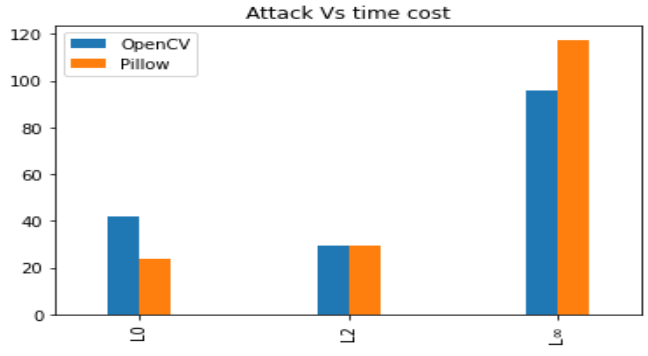

Figure 6: Comparison of time over attacks

V. CONCLUSION AND FUTURE WORK

The scaling camouflage attack was described in this project as a content disguising attack against computer vision applications. In this project, the assault is carried out at the scaling step, which occurs during the data pre-processing stage. The contents of the camouflage are extracted and passed to neural networks, resulting in a tailored deceptive effect. Three attack methodologies based on L0-, L2-, and L- norm distance metrics were implemented and GPU acceleration was used. The three attacks were implemented using the image libraries OpenCV and pillow, and when comparing the time cost, the pillow library takes less time when scaling the picture. When compared to accuracy, the L2 attack accuracy is good while camouflaging the source image's information. To fight against camouflage assaults, the Dhash idea is used, which offers the same hash values for pictures even when they are shrunk and greyscaled. So, we can tell whether the material is hidden. Object detection technologies such as YOLO do not identify these disguised assault pictures. Camouflaged items must be detected in future development. Camouflage detection is a difficult challenge in computer vision applications. Camouflage object modelling and characterisation are evolving rapidly, necessitating the development of new classifiers and statistical models to recognise objects in camouflage photos.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] L. Huang et al., "Universal physical camouflage attacks on object detectors," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. 2, pp. 717–726, 2020, doi: 10.1109/CVPR42600.2020.00080.

[2] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 997–1005, 2020, doi: 10.1109/CVPR42600.2020.00108.

[3] P. P. K. Chan et al., "Face Liveness Detection Using a Flash Against 2D Spoofing Attack," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 2, pp. 521–534, 2018, doi: 10.1109/TIFS.2017.2758748.

[4] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," Proc. - 2016 IEEE Symp. Secur. Privacy, SP 2016, pp. 582–597, 2016, doi: 10.1109/SP.2016.41.

[5] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," Proc. - IEEE Symp. Secur. Priv., pp. 39–57, 2017, doi: 10.1109/SP.2017.49.

[6] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," Proc. - 2016 IEEE Eur. Symp. Secur. Privacy, EURO S P 2016, pp. 372–387, 2016, doi: 10.1109/EuroSP.2016.36.

[7] D. Moreira et al., "Image Provenance Analysis at Scale," IEEE Trans. Image Process., vol. 27, no. 12, pp. 6109–6123, 2018, doi: 10.1109/TIP.2018.2865674.

[8] Z. Wang, M. Song, S. Zheng, Z. Zhang, Y. Song, and Q. Wang, "Invisible adversarial attack against deep neural networks: An adaptive penalization approach," IEEE Trans. Dependable Secur. Comput., vol. PP, no. c, pp. 1–15, 2020, doi: 10.1109/TDSC.2019.2929047.

[9] M. Li et al., "Provably Secure Camouflaging Strategy for IC Protection," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. 38, no. 8, pp. 1399–1412, 2019, doi: 10.1109/TCAD.2017.2750088.

[10] A. Tankus and Y. Yeshurun, "Computer vision, camouflage breaking and countershading," Philos. Trans. R. Soc. B Biol. Sci., vol. 364, no. 1516, pp. 529–536, 2009, doi: 10.1098/rstb.2008.0211.

ABOUT THE AUTHORS

Dr. Chandra Sekhar Sanaboina is currently working as an Assistant Professor in the Department of Computer Science and Engineering from University College of Engineering Kakinada, JNTUK University Kakinada. He obtained Bachelors (B. Tech) degree in Electronics and Computer Science Engineering from Koneru Lakshmaiah College of Engineering in the year 2005. Later obtained Masters (M. Tech) degree in Computer Science and Engineering from Vellore Institute of Technology in the year 2008. Complete Doctorate of Philosophy (Ph. D) in the area of Internet of Things from JNTUK in the year 2020. He had over 11 years of teaching experience and around 8 years of research experience. His areas of interest include Wireless Sensor Networks, Internet of Things, Machine Learning, Data Science, Cyber Physical Systems and Artificial Intelligence.

Sree Bharathi Garikiparti, PG Student, Department of Computer Science and Engineering Jawaharlal Nehru Technological University Kakinada, Kakinada, India