Defence Science Journal, Vol. 72, No. 6, November 2022, pp. 824-835, DOI : 10.14429/dsj.72.17599

Terrain Specific Real-Time Pixelated Camouflage Texture Generation and its Impact Assessment

Sachi Choudhary[$,*] and Rashmi Sharma[#]

[$]*University of Petroleum & Energy Studies, Dehradun, Uttarakhand – 248 007, India*
[#]*Shri Vaishnav Institute of Computer Applications, Indore, Madhya Pradesh – 453 111, India*
[*]*E-mail: sachi.choudhary1617@gmail.com*

ABSTRACT

"Camouflage" is a natural or nature identical phenomenon where the sensory route of vision is delayed to avoid visual detection. Reducing detection capability and hiding in the background environment is critical for Army vehicles, equipment, and soldiers. This research aims to implement a process that will generate digital camouflage patterns specific to the terrain. The adapted digital pattern helps an object blend symmetrically into the background environment. Pixelated textures combine macro and micro designs that blend with ambient shrubs, trees, branches, and shadows. The technique presented in this paper consists of the following main modules: terrain classification model, pixelated camouflage texture generation, and texture evaluation. Experiments have been conducted to detect camouflage objects in the scene to evaluate the performance of the resultant camouflage texture generated for a natural environment. Photo simulation and saliency maps for hidden object detection have been used to evaluate the effectiveness of generated textures for a given terrain.

Keywords: Digital camouflage; Pixelated camouflage; Terrain classification; Scene classification; Color clustering; K-means clustering; Photo simulation; Saliency map

1. INTRODUCTION

Camouflage is the way to match the surroundings by adapting to the natural uniformity in color, shape, or texture. Militaries use camouflage to hide the presence and position from enemy.[1,2] Reducing and hiding identities in background environments is a major of concern for Army vehicles, equipment, and soldiers.[3]

Traditional camouflage patterns used by the armed forces depend on the designer's experience and include irregular spot shapes and stripes. It may also have poor and unmatched color combinations with the area in which it is used. The ability of the human eye to detect any object around it depends on the size and shape of its boundaries and the light reflection from that object. There is a need for a disruptive digital pattern that can blend the edges of the object and have good concealment in the surrounding environment.[4]

The actual conditions in which the military is operating must be known and they need to camouflage themselves for defence and better hiding. To blend in with the environment, the defence industry needs an intelligence system that can classify the battlefield terrain before creating a texture for camouflaging. The model proposed in this paper generates terrain-specific pixelated camouflage textures that can blend any object into that environment.

There are eight distinct parts in this article. The first section discussed the usage of camouflage in the military and proposed the research needs. The second section provides an overview of the relevant literature from past studies. Sections 3, 4, and 5 detail the research methods and sub-module implementations. The created camouflage textures are tested, and the findings are presented in Section 6. The article is summarized in Sections 7 and 8, which provide the conclusion and suggestions for future research.

2. BACKGROUND AND PRELIMINARIES

There has historically been a necessity for camouflage in combat. It is claimed that ancient combat involved camouflage strategies, and since the 19th century, their designs have changed dramatically with improvements in their accuracy and performance in modern military[5]. Therefore, researchers are interested in developing an adaptive camouflage system that can generate color and pattern depending on the surroundings.

Many researchers over the years have contributed their findings to produce effective camouflage patterns. P. Bian, *et al.* (2010)[6] presented a novel approach to produce a digital pattern. Which uses the fuzzy c-means clustering to extract primary colors from the background image and morphological operations to generate the disruptive patterns. M. Friškovec, *et al.* (2010)[7] reviewed different aspects of designing patterns to camouflage military personnel and objects. The aim was to review various theoretical aspects such as the human visual system, fractal geometry, shape and size of camouflage texture to create a process for the generation of the camouflage texture for the urban environment. J. Yu and Z.Y. Hu (2012)[8] used a simple approach to generate the texture pattern using color quantization to extract the main colors. After determining the size of the mosaic block, the blocks were filled with

Received : 21 October 2021, Revised : 23 August 2022
Accepted : 17 October 2022, Online published : 6 December 2022

extracted colors to create disruptive patterns. N. Pezeshkian,[9] *et al.*gave color and texture synthesis based method to produce a camouflage texture similar to the input image. H.F. Yang,[10-11] et al.proposed the method based on the k-means clustering algorithm and background color similarity. F. Xue,[2] *et al*. proposed a method that iteratively overlaps spot pattern templates to create camouflage textures. X. Wei,[12] *et al.* proposed a Convolutional Neural Network (CNN) based model. The primary features from the image are extracted by CNN, and then mapped with the style patterns similar to the surrounding environment. Q. Jia,[13] *et al.* worked on the process of generating digital camouflage patterns by using spots of different shapes and sizes. H. Xiao,[14] *et al.* proposed a Deep Learning (DL) based model for the generation of adaptive optical camouflage patterns. The model was used for the recognition of the target, and a texture synthesis-based deepfill method was used to generate the synthetic image. X. Yang,[15] *et al.* researched the extraction of spot shape from the image and the generation of camouflage spots using a Generative Adversarial Network (GAN) model. X. Yang,[16] *et al.* used an Adversarial Autoencoder Network (AAN) to produce digital camouflage patterns.

Over the years afterward, several attempts have been made to enhance camouflage's effectiveness. Researchers in this area have proposed various methods based on image processing, ML, and DL for generating camouflaged textures. Camouflaging anything in the environment serves primarily to hide it by making it look like its surroundings. Hence, it is essential to incorporate numerous environmental aspects into the camouflage pattern.[13,17-18] It has been found in several studies that color values and intensities, as well as ambient texture, are the major determinants of the key design principles. Aside from these, factors such as light, object features, and environmental context all affect the range of human vision.[19] Therefore, the literature recommends including all the abovementioned parameters for optimal concealment.

3. PROPOSED MODEL

In this research work, a novel pixelated camouflage texture generation method is proposed to achieve the following objectives:

1. The pattern should resemble the environment of the background in terms of color, brightness, and texture.
2. The camouflage texture for a terrain should work for any distance.

Achieving the distinctive camouflage texture of the terrain requires the identification of the dominant colors and different shapes of specific terrain. According to retired US Army Lieutenant Timothy R. O'Neill, larger speckle patterns work best for longer ranges, and vice versa. Close-up small patches mimic natural patterns at the scale of leaves on a tree, but clusters of squares create a macro texture that blends in with branches, trees, and shadows. Literature also says that the patterns of different shapes and sizes help the texture to mimic the environment in a better way.[2,13] In addition, pixilation of the texture will help to avoid its observation not only from a close distance but also from a longer distance.

Figure 1 shows the proposed methodology with following sub-modules:

1. Terrain Classification
2. Camouflage texture generation
3. Texture evaluation

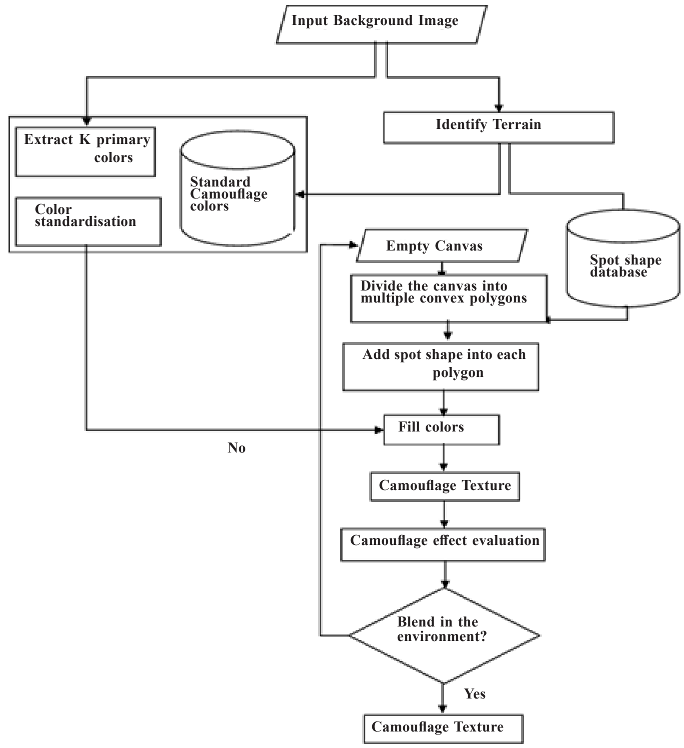


Figure 1. Camouflage texture generation for a specific environment.

4. TERRAIN CLASSIFICATION USING CNN

Terrain should be an important consideration when deciding on a camouflage texture or pattern for the armed forces or equipment where it is being deployed. This section classifies the terrain into three classes - forest, snow, and desert - for which the camouflage texture is needed.

4.1. Terrain Dataset Formation

Following are the steps followed while creating the terrain dataset.

4.1.1 Downloading Images

It aims to develop a dataset of specific images of the terrain where the defence force operates. It has been used to train ML systems to visually understand different terrains. The images have been downloaded randomly from various online search engines (Bing Images, Google Images, etc.).

4.1.2 Pre-processing of Images

Noise removal[20] and standardizing the dimensionality of all images in the dataset require pre-processing to make images more conducive to processing. All images in this dataset have been then resized to 512x512 pixels.

4.1.3 Labelling Image Class

Semantic classes of terrain are defined according to their appearance and human intelligence. The initial version of the dataset has three categories designated as "snow field," "forest area," and "desert land."

4.1.4 Image Augmentation

Image augmentation is a technique used to artificially enlarge the size of the training dataset by generating transformed versions of images in a dataset. Training the neural network model on a large dataset will yield more efficient models. In this paper, augmentation has been done using some image transformation techniques like rotation, scaling, and flipping.[21]

4.2 TerrainCNN Model

The Convolutional Neural Network (CNN) is a concept widely used in computer vision applications like classification, detection, and recognition, etc.[21-22] The architecture of TerrainCNN[39] consists of three convolution layers (Conv2D), an average pooling layer attached to each convolution layer, followed by a flattening layer and two dense layers. It was trained for 12 epochs on the NVIDIA GeForce GTX 1650.

Description of layers are as follows:

1. All convolutional layers are having 3x3 kernel-sized filters and a ReLU activation function. ReLU is used in hidden layers because it trains faster than tanh and sigmoid activation functions. Equation represents the ReLU activation function:[23]

$$f(x) = \max(0, x) \tag{1}$$

2. All average pooling layers are having 2x2 pool size. It down samples the input values by calculating the average value of its input.
3. Flatten layer flattens the dimension volume 62x62x64 to 1×246016.

Table 1 shows the details of the training and test segmentation of the images for each class of terrain.

go down to a stable point. It can be observed that the plot of validation loss has a small gap with the training loss plot.

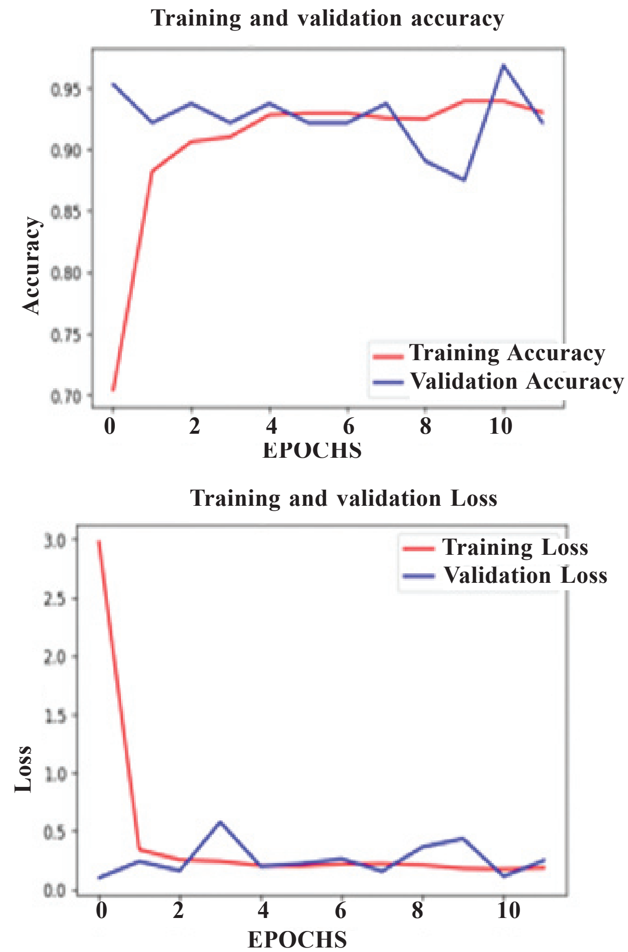


Figure 2. Performance of TerrainCNN model on terrain dataset.

Table 1. Terrain dataset description

Classes	Training test	Test set	Total images
Forest rea	779	195	974
Desert land	646	162	808
Snow area	766	192	958
Total	2191	549	2740

Training set: 80 %
Testing set: 20 % of total images in each category

4.3 Performance of TerrainCNN

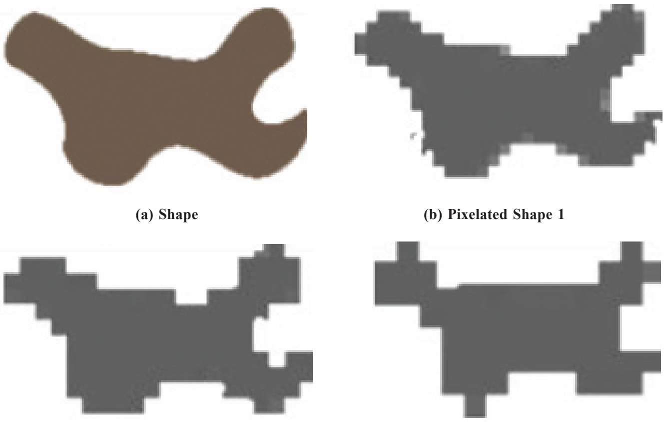
TerrainCNN[39] performs very well on the terrain dataset. The training accuracy and loss of the model are 93.03 % and 18.55 % respectively. The validation accuracy and loss are 92.12 % and 24.92 % respectively, as shown in Fig. 2. The performance curve for training accuracy shows a good increase in initial epochs, which shows the network is learning fast. After that, the training accuracy goes flat, which shows that not too many epochs are required to train the model. The validation accuracy plot has ups and downs due to the diverse set of images in the dataset. A good fit can be seen in the plot of learning curves as the plots of training loss and validation loss

5. PIXELATED CAMOUFLAGE TEXTURE GENERATION

After classifying the background image with its specific terrain, the next step is to generate a texture that can resemble the background environment. This section includes the requirements and process for generating the camouflage texture.

5.1 Shapes Dataset

Another dataset of typical shapes for different terrains has been created of different sized templates extracted using morphological processing. In addition, the shapes have been converted and stored in different sizes to enhance the dataset and obtain texture diversity. The shapes were pre-processed and then pixelated to obtain the distinct boundaries in different pixel dimensions, as described in Fig. 3(b), (c) & (d).



(a) Shape (b) Pixelated Shape 1 (c) Pixelated Shape 2 (d) Pixelated Shape 3

Figure 3. Sample shapes and their pixelated forms.

5.1.1 Morphological operations

Morphology is the scientific study of form and structure. In morphological mathematics, images being analyzed are treated as sets of points, and set theory is applied to the operations. It can detect logical relationships between pixels and extract geometric features by choosing the appropriate structural shape as the probe. The grayscale image is selected for boundary extraction, and the threshold value used for segmentation is [0-255]. Pixels with a value greater than the set threshold are assigned a value of 255, otherwise 0. All pixels having a value of 255 are replaced by 1 to obtain the threshold pixel. Boundary pixels are pixels that have at least one 0 in their eight neighbors.[24]

Let E^N is the set of all the points $p = (x_1, x_2, x_3, ..., x_N)$ in N-dimensional Euclidean space. Here, A is the image as an array and B is the kernel used for erosion. A and B are two sets that belongs to E^N having elements $a = (a_1, a_2, a_3, ..., a_N)$ and $b = (b_1, b_2, b_3, ..., b_N)$, respectively.

Equation was used for Boundary Extraction, and equation for Erosion.

$$\partial A = A - (A\theta_b B) \quad (2)$$

$$A\theta_b B = \{x \in E^N \mid x + b \in A, \forall b \in B\} \quad (3)$$

5.2 Color Clustering using K-means

Colors in a camouflage design should be as distinguishable as possible, with brightness being the primary factor. Choosing colors from the surrounding environment image might be a better decision. Intuitively, visible light is described by its intensity and its hue, or chroma. The RGB (Red, Green, Blue) model only provides primary color weights and no chroma or brightness information. Unlike RGB, HSV (H for Hue, S for Saturation, and V for Value) separates image intensity from color data.

In this paper, K-means clustering is used to divide n colors into k clusters by assigning them to the nearest cluster. The proposed methodology check for a match with military standard colors after extracting dominant colors from the image. This step involves eliminating black, blue, and other colors that do not belong to specific terrain, as these can capture human visual attention very quickly. It also calculates the Euclidean distance between the extracted and the military standard colors of the respective terrain and operates on these colors to blend thoroughly with the surrounding.[25] Figure 4 shows the process of extraction and standardization of the dominant colors. Eqn. (4) can calculate the proportion of each primary color in a background image. The ratio P(c) of a color (c) is the number of pixels (n_k) that belong to a cluster k out of the total pixels (N) in the image.

$$P(c) = \frac{n_k}{N} \times 100 \quad (4)$$

Table 2. Standard colors for specific terrain

Terrain	Standard colors
Dense forest land	Tan, Brown (different shades), Green (different shades), Khakhi
Forest/woodland	Tan, Brown, Green,
Snow field	White, Dark green, brown, Sky blue
Dessert land	Tan, Brown (different shades), Khaki, Dark Tan or Tan, Brown (different shades), Light Grey, Dark Tan

Many observations have been made on images of camouflaged army personnel, uniforms, tanks, and weapons available on the Internet to make some assumptions on military standard colors. The conclusions drawn are presented in Table 2 and have been used to finalize the colors for the generation of the camouflage texture. The following are the other general observations that also help in finalizing the colors:

1. Black is not a color created by nature; usually, it is the effect of the shade of trees, hills, mounds, etc. Therefore, if there are darker shades of black in the list of primary colors, they will be removed.
2. If the image of the area is taken as the front view, the colors of the sky and clouds will be on the list of dominating colors. Hence, blue and white colors for a desert land and forest areas will not be considered. In the case of the snow regions, it has been seen that white snow has an effect of the color of the sky. So, some shades of blue can be considered for the snowfield.
3. If any color in the primary color list has a larger difference in HSV values than the standard colors (Reference Table 2), that color will be removed from the final list. The Euclidean distance (Eqn.) can be used to calculate the difference in the HSV of the extracted primary color and the corresponding standard color.[13]

$$Dist(C,S) = \sqrt{(Hc - Hs)^2 + (Sc - Ss)^2 + (Vc - Vs)^2} \quad (5)$$

5.3 Pattern Generation

The objective behind the use of camouflage texture is to reduce the visibility in the specific terrain for which it is intended.[3] So, the generated texture should:

1. resemble the maximum terrain features
2. be similar and perceived as the terrain itself

Here, the part of concern should be the parameter that

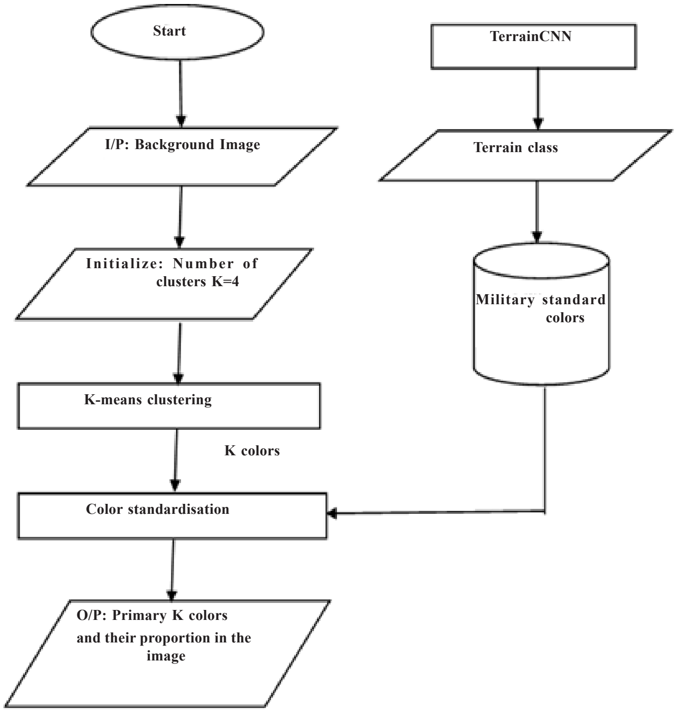


Figure 4. Primary colors extraction from the image.

is prominently considered for camouflage object detection. Although there are many factors, the most prominent are color, size, and shape. Human visual perception is also explored here to capture outlines and other regular aspects of the object to be hidden. In a cluttered scene, the texture and other background features are not homogeneous. So, shapes have been distributed at random. Although distributing shapes and colors to meet the canvas's perimeter can be considered an optimization problem[2].

Steps to create the camouflage pattern are:

1. Take an empty rectangular canvas (WxH) and divide it into multiple subsections of different sizes.
2. Create the list of shapes and their parameters that can fit into subsections.
3. Fill the extracted shapes into the canvas.
4. Fill standardized colors into different shapes.

The pseudocode shapeExtraction() shows the step-by-step process for the selection of shapes, and textureGenerate() shows the process for generating the final camouflage texture for the input background image.

Calculating the perimeter of any shape using Eqn.

$$P = \sum_{i=1}^{N} \text{len}(\text{side}_i) \quad (6)$$

Pseudocode: textureGenerate()

```
Input: Empty canvas of the dimension (WxH)and colors[]
Output: Textured Canvas
textureGenerate()  // Function use to generate textures
Begin
Step 1:   Canvas(W,H)        // Creation of an empty canvas
Step 2:  subSectionList = Add(subSection,edge_coordinates) // Divide the canvas into
sub sections of different dimensions
Step 3: perimeterList = addPerimeter(subSection,calcPerimeter(subSection)) // calculate
perimeter of each sub section
Step 4: shapes = shapeExtraction(Terrain,W,H) //Extract shapes from shape dataset
          sort(shapes)              //sorting of shapes on the basis of perimeters
Step 5: For i in subSectionList
if (shapePerimeter <= perimeterListi)   //Spread multiple shapes into the
subsection i
  {
    shapeList = addShape(shapes,subSection) //add shapes into subsections of canvas
    Canvas = append(shapeList)
}
Step 6: For i in shapeList
          If (shapeListi.color == None) && (shapeListi.neighbor.color! = Color)    // Select
color and its proportion
            fillColor(shapeListi,Color)          // spread color
        else
            Repeat step 6
Step 7: Return textured canvas
End
```

Pseudocode: shapeExtraction()

```
shapeExtraction()  //Function use to extract shape
Input:  Terrain,  Shape_dataset(S),w,h   //Terrain image, dataset of shapes,
 width & height

Begin
Step 1: Select S // Select shape dataset S for specific terrain
Step 2:
  P = calcPerimeter(w,h)    //calculate perimeter and assign to P
   Shape_list = []          // initially Shape list is empty
  Edge_list = []            //Initially Edge list is empty
Step 3:   Select RS ∈ D              //Select a random shape from dataset
          S = random_shape(Shape_dataset)  //Assign randomly chosen shape to S
          SP = calcPerimeter(W_S, H_S) //calculated perimeter of shape assign to SP
         If (P > SP)
        {
  Add_shape(Shape_list, (S, SP))
  Edge_list(S, list(E))
         P-= SP
        }
Repeat Step 3
Step 4:   Return (Shape_list, Edge_list)  //Return the list of shapes with edges
End
```

6. EXPERIMENTATION, RESULTS, AND DISCUSSION

The experiments and findings of all the modules of the proposed work are covered in this section.

6.1 Texture Generation

The experiments were carried out on the "Terrain" dataset using the approach proposed in this paper. Figure 5 depicts some of the experiment's findings. The camouflage textures were created from the input images of snow, desert, and forest terrains. Some of the input images for the camouflage pattern are shown in Fig. 5(a). Next, column (b) shows the output of the TerrainCNN classification model. The model predicts the input image's terrain class. Column (c) depicts the bars of the dominant colors in the input image and the colors after standardization for texture generation. The extracted color's HSV values are maintained in the output texture to match the background image's lighting condition. It enhances the final texture's effectiveness. The resultant camouflage textures are shown in column (d). It can be observed that these are combinations of different shapes and sizes that make them less noticeable from different distances. The patterns are not regular and the color distribution is reasonable. The textures are applied to the objects (human and horse stencils) for effectiveness assessment (see column (e)).

6.2 Effectiveness Evaluation

The following experiments have been performed to check how effectively the resultant camouflage texture blends the object into the environment.

6.2.1 Photo Simulation

Human vision has an incomparable ability to detect and identify artifacts. The main role of the human eye is to focus our vision on objects of interest. Initially, neurons adapt to primary visual characteristics like size, position, color, orientation, and speed of eye movement.[33] This is the traditional but effective way of evaluating camouflage performance. Some techniques based on the human visual system (HSV) are field observation, photo simulation,[34] and calculation of the observer's eye motion.[35]

In this experiment, photo simulation was used to detect hidden objects in the scene. Fifteen people participated in the study, in which no one had any eye disease, visual loss, and color blindness. Setup: Observers were given a personal computer with a digital imagery system.[19] The process and set of examples were explained in advance. The experiments were conducted with the following assumptions and constraints about the input image:

1. It may or may not contain camouflaged object

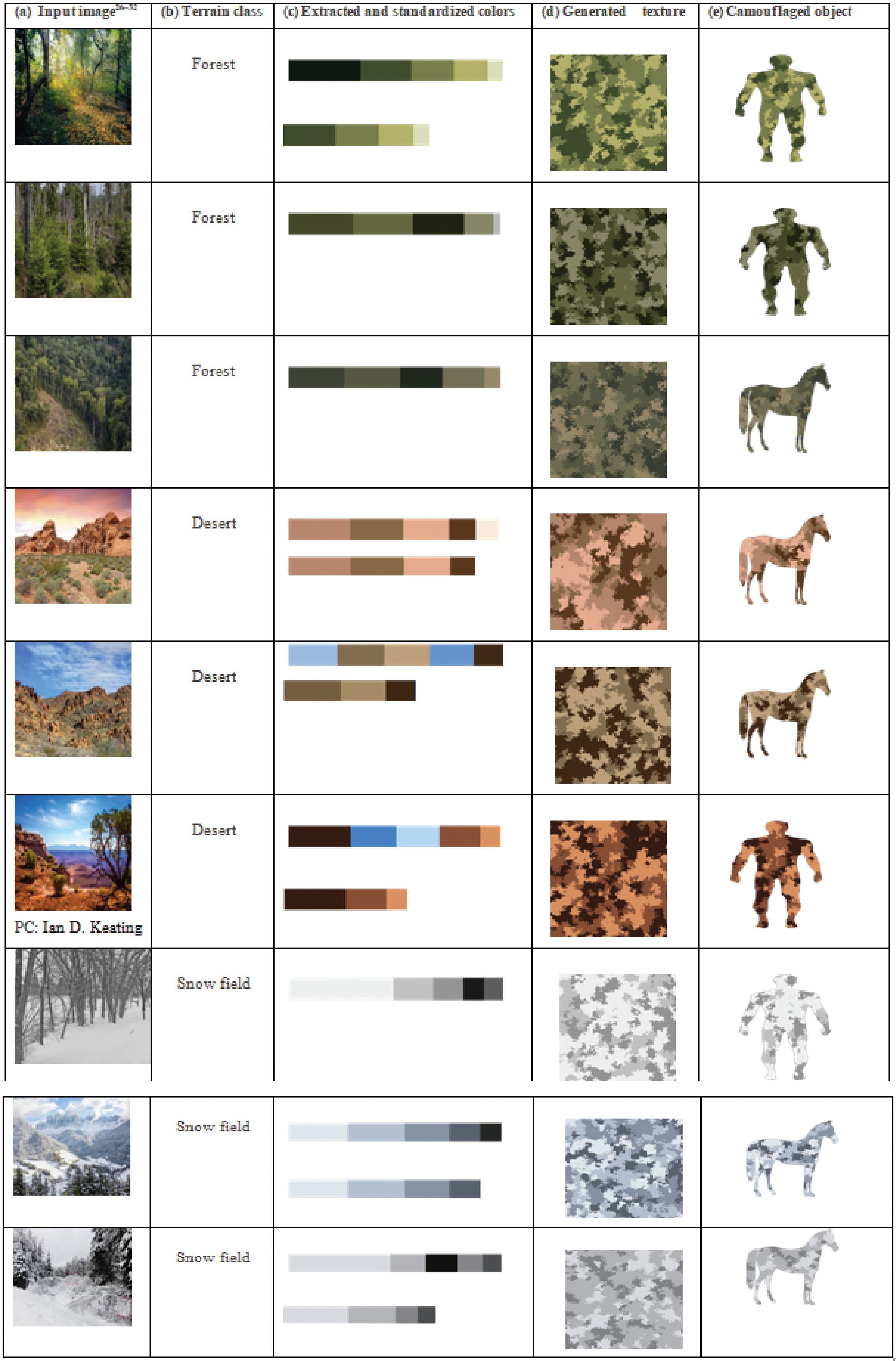


Figure 5. Experimental results.

2. It may contain more than one camouflaged object

Each observer's job was to keep track of the time they spent detecting the camouflaged object (s) and its location. To calculate the effectiveness, the average hit rate, average detection time, and difficulty level were used as the major criteria. The hit rate is the percentage of times an observer finds the correct object [Eqn. (7)]. In Eqn. (8), the detection time is the time it takes for the observer to successfully locate the disguised object in the scene. Observers were asked to score the texture's complexity on a scale of 1 to 5, with 1 being easy and 5 being very tough. More effective camouflage material has a lower hit rate and longer detection time[35].

$$\text{Hit rate (\%)} = \frac{\text{True Positive Events}}{\text{Number of Observer}} \times 100 \qquad (7)$$

$$\text{Average detection time} = \frac{\sum \text{detection time}}{\text{Number of observers}} \qquad (8)$$

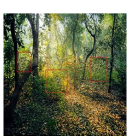
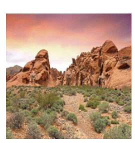
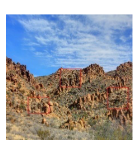
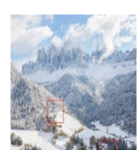
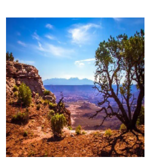
Sample observations based on experimental results have been compiled and presented in Table 3. The average hit rate in forest and desert areas is lower than in snow regions. Similarly, the average detection time for a camouflaged target in a forest or desert area is faster than in a snow field. However, the results of both assessment measures for the snow area are also impressive. The overall result of the texture difficulty ratings for the various terrains is high. The effectiveness of camouflage textures for a specific terrain can be evaluated with performance metrics alone. However, in the real environment, the effect would have something more helpful, such as a proper hiding place for the object, which would help in indistinguishability. Figure 6 (a), (b), and (c) present hit rate, average detection time and difficulty rating versus camouflage texture for different terrains respectively. The graph contains the analysis using the average value, harmonic mean, and geometric mean of detection time taken and the rating given by different observers. In statistics, the Harmonic Mean (HM) and the geometric mean (GM) are both types of average. The HM is the reciprocal of the mean of the reciprocals of the numbered data, while GM is the mean that shows the central tendency of a data set of numbers. HM can be calculated by finding the product of the data values in the set. It can be observed that the results are good and effective.

6.2.2. Saliency Map

Each pixel in an image has its saliency. Creating a saliency map of an image allows you to focus on the primary characteristics of the image[36]. It is a 2-dimensional scalar map used to represent visual saliency regardless of the features of the main area. Evaluation of camouflage texture using a saliency map is based on mechanisms of the human visual attention system. This method compares the foreground object to the background scene [37].

Saliency maps are extensively used in object detection systems and have been used in research to find concealed objects as well. The saliency maps have been generated in this research project using region contrast (RC) based [38] and Histogram Contrast (HC) based method[37-38]. To evaluate an image's saliency, the HC-based technique calculates the difference between a local region and the full image. The resultant saliency map comprises contrast aspects including color, intensity, and orientation. The object being disguised has a texture comparable to the environment since camouflage is designed to mimic the surroundings. As scanning the full image would be necessary for camouflage object detection technologies, experiments have also been conducted utilizing saliency maps using the RC approach. It produces reliable and useful saliency maps. This method uniformly exposes the entire input region to draw attention to the salient features.

Table 3. Example of results of human observation on photo simulation for camouflaged object detection

Forest area				Desert land				Snow field			
Scene with camouflaged object(s)	Avg. Hit rate	Avg. detection time (second)	Difficulty	Scene with camouflaged object(s)	Avg. Hit rate	Avg. detection time (second)	Difficulty	Scene with camouflaged object(s)	Avg. Hit rate	Avg. detection time (second)	Difficulty
	57%	14.514	4.2		40 %	19.23	4.2		79 %	8.2	3.4
	55%	21.871	4.8		45 %	15.624	4.4		67 %	13.935	3.7
	55%	11.453	3.9		64 %	13.162	4.4		40 %	20.636	5

The resultant saliency maps are shown in Fig. 7. Columns (a) and (b) show photos with concealed objects (s). It can be seen how well those objects blend into the image; b) it reveals the location of the hidden object in the scene; and c) it contains the image's saliency map using the RC approach. To compute the color contrast of each region, the image is first segmented into N regions (R_1, R_2,..., R_N). This method's saliency is the weighted total of each region's contrast with all others. Column (d) shows the HC saliency map. This model compares all pixels or regions in the image. They are then added up. The intensity and highlighted regions of the saliency maps are unable to segment the disguised object as shown in Fig. 7 (c) and (d). It indicates that the camouflaged objects match the background features so well that it is hard to detect.

Therefore, the predominant colors in the background image, their HSV values and proportions, shapes of different sizes, and pixilation of texture have been considered as key design parameters. Average hit rate, detection time, and difficulty level in locating the target object in the background scene have been considered performance measures. In addition to performance metrics, saliency maps have also been used to highlight camouflaged objects in the image. The performance of the camouflage texture generated for a specific scene demonstrates effective object concealment and blending into the surrounding environment. The generated patterns can be used in uniforms and to paint equipment. This work can also be used to obtain camouflage patterns in real time for a particular terrain.

Figure 6. Results of performance measures on photo simulation.

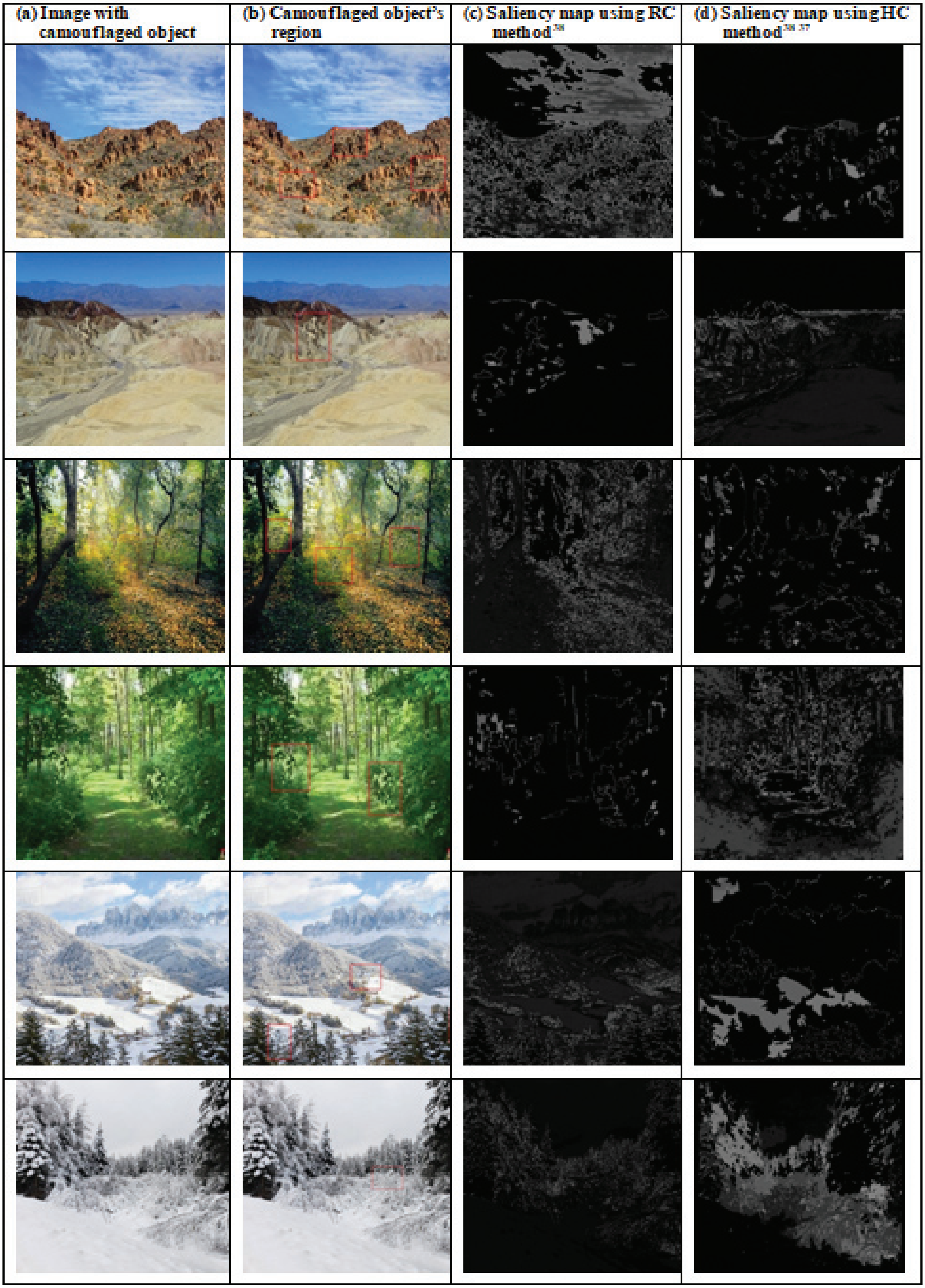
7. CONCLUSION

The design of a camouflage pattern for a specific terrain is still an essential requirement for the military globally. Here, efforts are made toward generating a pixelated camouflage texture for a specific scene. The ability to detect an object in the environment depends on the difference in texture, color, brightness, and distance from which it is being observed.

8. FUTURE SCOPE

In the future, other types of terrain can be added to the terrain dataset, such as cities, oceans, deep seas, and dry mountains. Further tests can be done on the natural environmental consequences of the method proposed in this paper. Further studies can also be done on improving the texture and performance with other advanced assessment parameters.

Figure 7. Camouflage effect evaluation using saliency map.



REFERENCES

1. Talas, L.; Baddeley, R.J. & Cuthill, I.C. Cultural evolution of military camouflage. *Philosophical Trans.of the Royal Society, B: Bio. Sci.*, 2017, **372**(1724). doi:10.1098/rstb.2016.0351
2. Xue, F.; Xu, S.; Luo, Y.T. & Jia, W. Design of digital camouflage by recursive overlapping of pattern templates. *Neurocomput.*, 2016, **172**, 262-270. doi:10.1016/j.neucom.2014.12.108
3. Camouflage, Concealment, and Decoys. Marine Corps Reference Publication, Department of the Army, Washington. ATTP 3-34.39 (FM 20-3)/MCRP 3-17.6A, November 2010.
4. Price, N.; Green, S.; Troscianko, J.; Tregenza, T. & Stevens, M. Background matching and disruptive coloration as habitat-specific strategies for camouflage. *Sci. Reports*, 2019, **9**(1), 1-10. doi:10.1038/s41598-019-44349-2
5. Pezeshkian, N. & Neff, J.D. Adaptive electronic camouflage using texture synthesis. *In* Proceedings of SPIE 8387, Unmanned Systems Technology XIV SPIE, **838707**, 2012. doi:10.1117/12.920507
6. Bian, P.; Jin, Y. & Zhang, N.R. Fuzzy c-means clustering based digital camouflage pattern design and its evaluation. *In* Proceedings of IEEE 10th International Conference on Signal Processing, *ICSP*, 2010, 1017-1020. doi:10.1109/ICOSP.2010.5655856
7. Friškovec, M. & Gabrijelčič, H. Development of a procedure for camouflage pattern design, *FIBRES & TEXTILES in Eastern Europe*. 2010, **18**, 68–76.
8. Yu, J. & Hu, Z.Y. Generation Method of Digital Disruptive Pattern. *Adv. Eng. Forum*. 2012, **6**(7), 1103-1107. doi:10.4028/www.scientific.net/aef.6-7.1103
9. Pezeshkian, N. & Neff, J.D. Adaptive electronic camouflage using texture synthesis. *In* Proceedings of SPIE 8387, Unmanned Systems Technology XIV SPIE, **838707**, 2012. doi:10.1117/12.920507
10. Yang, H.F. & Yin, J.P. An adaptive digital camouflage scheme using visual perception and K-mean clustering. *In* Proceedings of advanced materials research, 2013, **834,** 1091-1094. doi:10.4028/www.scientific.net/AMR.834-836.1091
11. Yang, H. & Yin, J. A digital camouflage generation algorithm using color similarity. *Int. J. Multimedia and Ubiquitous Eng.*, 2015, **10**(6), 159-164. doi:10.14257/ijmue.2015.10.6.16
12. Wei, X.; Wang, K.; Li, G. & Kim, H. An Automatic design of camouflage patterns based on CNNs. *In* Proceedings of ACM International Conference Proceeding Series, Association for Computing Machinery, 2020, 257-260. doi:10.1145/3404555.3404637
13. Jia, Q.; Xu, W.D.; Hu, J.H.; Liu, J.; Yang, X. & Zhu, L.Y. Design and evaluation of digital camouflage pattern by spot combination. *Multimedia Tools and Appl.*, 2020, **79**(29-30), 22047-22064. doi:10.1007/s11042-020-09002-5
14. Xiao, H.; Qu, Z.; Lv, M.; Jiang, Y.; Wang, C. & Qin, R. Fast self-adaptive digital camouflage design method based on deep learning. *Appl. Sci. (Switzerland)*, 2020, **10**(15). doi:10.3390/APP10155284
15. Yang, X.; Xu, W. dong; Jia, Q. *et al*. Research on extraction and reproduction of deformation camouflage spot based on generative adversarial network model. *Def. Technol.*, 2020, **16**(3), 555-563. doi:10.1016/j.dt.2019.06.021
16. Yang, X.; Xu, W.D.; Jia, Q. & Li, L. Research on digital camouflage pattern generation algorithm based on adversarial autoencoder network. *Int. J. of Pattern Recognit. and Artif. Intell.*, 2020, **34**(6). doi:10.1142/S0218001420500172
17. Yang, H. & Yin, J. A digital camouflage generation algorithm using color similarity. *Int. J. Multimedia and Ubiquitous Eng.*, 2015, **10**(6), 159-164. doi:10.14257/ijmue.2015.10.6.16
18. Liu, Z. yang; Ding, F.; Xu, Y. & Han, X. Background dominant colors extraction method based on color image quick fuzzy c-means clustering algorithm. *Def. Technol.*, 2021, **17**(5),1782-1790. doi:10.1016/j.dt.2020.10.002
19. Miao, C. & Shaohui, T. An extraction method for digital camouflage texture based on human visual perception and isoperimetric theory. *In* Proceedings of 2nd International Conference on Image, Vision and Computing*, ICIVC,* 2017, **3**, 158-162. doi:10.1109/ICIVC.2017.7984538
20. Choudhary, S. & Ram, B.P. Survey Paper on Different Approaches for Noise Level Estimation and Denoising of an Image. *Int. J. of Sci. and Res.*, 2014, **3**(4), 618-622.
21. Elgendy, M. Deep Learning for Vision Systems. Manning Publications, 2020.
22. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S. & Lew, M.S. Deep learning for visual understanding: A review. *Neurocomput.*, 2016, **187**, 27-48. doi:10.1016/J.NEUCOM.2015.09.116
23. Ding, B., Qian, H. & Zhou, J. Activation functions and their characteristics in deep neural networks. *In* Proceedings of the 30th Chinese Control and Decision Conference, CCDC, 2018, 1836-1841. doi:10.1109/CCDC.2018.8407425
24. Shih, F.Y. Image processing and pattern recognition: Fundamentals and techniques. Wiley, IEEE Press, United States of America, 2010. doi:10.1002/9780470590416
25. Dugas, A.; Zupkofska, K.J.; DiChiara, A. & Kramer, F.M. Universal Camouflage for the Future Warrior. *In* Proceedings for the Army Science Conference (24th) Orlando, Florida, 2004. http://www.dtic.mil/docs/citations/ADA433081
26. Free Forest Background. https://eskipaper.com/free-forest-background.html. Accessed August 17, 2022.
27. Mint Images. Winter landscape with valley, forest and snow-capped mountains. Accessed August 17, 2022. https://dissolve.com/stock-photo/Winter-landscape-valley-forest-snow-capped-royalty-free-image/101-D1024-94-367

28. Gauthier, R. Snow forest winter landscape. https://www.publicdomainpictures.net/en/view-image.php?image=286910&picture=snow-forest-winter-landscape. Accessed August 17, 2022.
29. Misra, R. Forest background 3. https://www.publicdomainpictures.net/en/view-image.php?image=152924&picture=forest-background-3. Accessed August 17, 2022.
30. Free Images : landscape. Accessed August 17, 2022. https://pxhere.com/en/photo/1192027
31. Free Images : landscape, nature. Accessed August 17, 2022. https://pxhere.com/en/photo/968901
32. Free Images : landscape, sea. Accessed August 17, 2022. https://pxhere.com/en/photo/272010
33. Itti, L, & Koch, C. Computational modelling of visual attention. *Nature Rev. Neuroscience*, 2001, **2**(3), 194-203. doi:10.1038/35058500
34. Arseneau, Lise E.E.J. A Guidebook for the Analysis of Camouflage Data, Department of National Defence, Canada, 2003.
35. Lin, C.J.; Chang, C.C. & Lee, Y.H. Evaluating camouflage design using eye movement data. *Applied Ergonomics*, 2014, **45**(3), 714-723. doi:10.1016/j.apergo.2013.09.012
36. Itti, L.; Koch, C. & Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998, **20**(11), 1254-1259. doi:10.1109/34.730558
37. Feng, X.; Guoying, C.; Richang, H. & Jing, G. Camouflage texture evaluation using a saliency map. *Multimedia Systems*, 2015, **21**(2), 169-175. doi:10.1007/s00530-014-0368-y
38. Cheng, M.M.; Mitra, N.J.; Huang, X.; Torr, P.H.S. & Hu, S.M. Global contrast based salient region detection. *IEEE Trans. on Pattern Analysis and Mach. Intelligence*. 2015, **37**(3), 569-582. doi:10.1109/TPAMI.2014.2345401
39. Choudhary, S. & Sharma, R. Perceived dominating colors optimization for camouflage texture generation & its effect evaluation. *In* Proceedings of 4th International Conference on Recent Trends in Computer Science and Technology, ICRTCST, 2022, 334-339. doi:10.1109/ICRTCST54752.2022.9782048

CONTRIBUTORS

Ms Sachi Choudhary earned a Bachelor of Engineering in 2010 and a Master of Technology in 2013 in computer science and engineering, both from the Chhattisgarh Swami Vivekanand Technical University in Bhilai, (C.G.), India. She is a Computer Science PhD candidate at India's University of Petroleum and Energy Studies in Dehradun and also currently working as an Assistant Professor (SS) at the School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. She has many research publications, conference papers, and published patent applications in her name. Her current research interests include computer vision, image processing, machine learning, and big data analytics.

Her involvement in the present study included not only an examination of the relevant literature but also the actual implementation and testing of results.

Dr Rashmi Sharma is an Associate Professor at Shri Vaishnav Institute of Computer Applications, Shri Vaishnav Vidhyapeeth Vishwavidyalaya in Indore, M.P., India. She has contributed to over thirty international conferences, journals, and edited volumes. Also, three of her patents have been made public. Real-time systems, distributed and parallel computing, algorithms, the internet of things, and image processing are all areas of study that she is passionate about.

She has contributed to the current study by doing the literature review, identifying research gaps, and analysing experiments.