Indonesian Journal of Electrical Engineering and Computer Science
Vol. 31, No. 3, September 2023, pp. 1551~1558
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i3.pp1551-1558

Image classification using machine learning

Debani Prasad Mishra[1], Sanhita Mishra[2], Smrutisikha Jena[1], Surender Reddy Salkuti[3]
[1]Department of Electrical and Electronics Engineering, International Institute of Information Technology Bhubaneswar, Bhubaneswar, India
[2]Department of Electrical Engineering, Kalinga Institute of Industrial Technology Deemed to be University, Odisha, India
[3]Department of Railroad and Electrical Engineering, Woosong University, Daejeon, Republic of Korea

Article Info

Article history:

Received Oct 21, 2022
Revised Apr 26, 2023
Accepted May 6, 2023

Keywords:

Image classification
JavaScript
Machine learning
Python flask
Support vector machine

ABSTRACT

The objective of this paper is to implement different tools available in machine learning/artificial intelligence to classify faces and identify different features, highlights, and correlations or similarities between different celebrity faces which can apply in everyday security purposes to identity virtually if the authorized personnel is using certain access or not. The material present in this paper is a literature review of a machine learning model developed by the students. This is a classical problem of machine learning executed using a support vector machine. Images are separated based on sub-images. Each sub-image has been classified into a responsive class by an artificial neural network. The website then fetches the data from the back end and classifies the image into the corresponding personal.

Corresponding Author:

Surender Reddy Salkuti
Department of Railroad and Electrical Engineering, Woosong University
17-2, Jayang-Dong, Dong-Gu, Daejeon-34606, Republic of Korea
Email: surender@wsu.ac.kr

1. INTRODUCTION

Machine learning (ML) by its definition is a field of computer science that was evolved from pattern recognition. It helps to learn and make algorithms of code that learns from and makes predictions based on the input data sets. Image classification is a method of visual processing that distinguishes between several categories of objectives based on image attributes. In pattern recognition and computer vision, it is commonly utilized. The support vector machine (SVM) is a new ML method that is based on statistical learning theory and has a solid mathematical foundation. It is based on the structural risk minimization model [1]. Various deep learning techniques for recognizing various images are described in [2]. The time spent at some commonly used image processing operations using OpenCV's built-in central processing unit (CPU) and OpenCV's built-in graphics processing unit (GPU) functions which are written with compute unified device architecture (CUDA) support is presented in [3]. A representation for checking the video structure alongside the device and improving it further using artificial intelligence (AI) methods such as SVM and artificial neural network (ANN) is proposed in [4]. Both techniques are used in tandem to train and test the classification to obtain results that are appropriate for the investigation and, as a result, proven to be successful.

In the factual context, ML is characterized as the utilization of man-made brainpower where accessible information is utilized through algorithms to process or help the preparing of measurable information. While ML involves concepts of computerization, it requires human direction [5], [6]. AI involves an undeniable degree of speculation altogether to get a framework that performs well on yet inconspicuous information instances. ML is a moderately new control inside computer science that gives an assortment of information

investigation procedures [7], [8]. Some of these strategies depend on grounded factual techniques (for example calculated regression and SVM), while numerous others are not [9]. Most measurable methods follow the worldview of deciding a specific probabilistic model that best depicts noticed information among a class of related models [10], [11]. Also, most AI strategies are intended to discover models that best-fit information (for example they take care of certain enhancement issues), then again, actually these AI models are no longer limited to probabilistic ones. AI could possibly give a more extensive class of more adaptable elective examination techniques more qualified to present-day wellsprings of information [12]. It is basic for offices and organizations to oversee gigantic heaps of information and all the while keeping up the security of that very information from frauds. Agencies and companies must manage huge loads of data and simultaneously maintain the security of that very data from imposters [13]–[15]. To manage, analyze and sort without much manual human labor, ML gives people vent of opportunity by developing required models as per the problem statement and solving the issue as quickly as possible.

This paper aims at solving a few key features of security and classification. Using the methods like wavelet transform (WT) and ML kits [16], the proposed model can differentiate between objects and personalities by observing and analyzing the various nuances and intricate details of facial features and object's shape and size to correctly classify them [17]. In this period of time, where everything is getting online, this work can further be developed and integrated into security cameras and webcams to identify the legitimacy of the person on the other side by determining whether that person using some filter/photoshop or he/she is originally present there.

2. RESEARCH METHOD

An image might contain several features like animals, humans, landscapes, and street art. An image might also contain multiple faces, also the face can be obstructed and not clear due to several factors like lens blur, lighting issues, and another person's face blocking them. So, to classify successfully, this work has been divided into three blocks [18], [19]. The user frontend website, pre-processing of images and testing, backend server to fetch the data and give the required output. Figure 1 shows the flowchart of the main basic components of our project. The first step in the pre-processing pipeline is to detect faces from an image. Once the face is detected, we will detect eyes [20], if two eyes are detected then only we keep that image otherwise discard it. Images with closed eyes, side faces with one eye visible are also rejected.

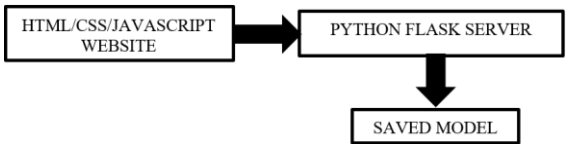

Figure 1. Flow chart of the proposed approach

Photos of the concerned celebrities were taken from google using the fatkun extension provided by Google Chrome. Images of five celebrities were collected and put inside respective folders inside the work folder [21]–[23]. About five hundred images of each personality were collected for analysis which was further sorted and cleaned using various techniques and they are discussed next.

Matplotlib and OpenCV are used to read and edit the required image. Cascade classifier in OpenCV (Haar cascade) is used to detect and crop the face in the given picture [24]. The cropped images are then stored in a folder. Cropped images are then manually cleaned for any random issues like photos of wrong personalities, and wrongly cropped images [25], [26]. Most of the time was consumed in manual removal of these photos which is essential for the best training of the model and so it can properly detect the concerned faces and correlate how similar the face are to each other [27], [28]. Figure 2 shows the flowchart of life of the images from raw images given as an input by the user to how it is transformed into a digitally useful picture.

The WT technique was used to convert images into signals that can be understood by the program. After transforming the images they are stacked one above the other for comparison [29], [30]. All the images are stored inside a dictionary file using shutil. The WT method is further discussed in detail in the results and discussion section.

SVM is a supervised ML model that uses a classification algorithm into a two-group classification problem. This model is associated with learning algorithms that analyze data for classification and regression analysis. A training model is a dataset used to train the SVM model which was created using OpenCV, Haar

cascade and scikit learn [31]–[33]. It is used to run the input data through our model to correlate the processed output against sample output. In this work, 60% of the data is used for training and 40% is used for testing. Gradually, the number of images used for training is decreased and testing is increased for improved efficiency [34]–[36]. Figure 3 is the code written for testing and training the model. Scikit learn is used to code the algorithm which will be used to train our model [37]. After training the model is saved as a .pkl file. Pickle is created by pickle, a Python module that enables objects to be serialized to files on disk and deserialize back into a program at run time.

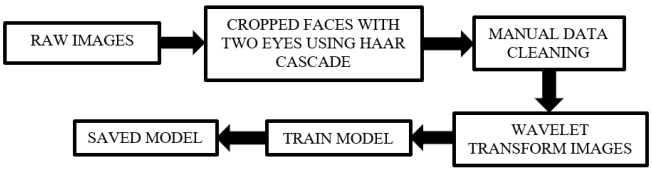

Figure 2. Steps followed to clean our dataset of pictures

```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC(kernel = 'rbf', C = 10))])
pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)
```

```
0.9047619047619048
```

Figure 3. Code for testing and training

Figure 4 shows the code written in Python language using visual studio code as the text editor. This code is written to locally host the backend server of the project in localhost5000:classify image\. Util file is used to write all the functionalities of the backend server and is imported in the server.py for the server to access all the data. Flask is a micro web framework [38], [39]. Here it is used to fetch the backend data from util. Util file is used to write all the functionalities of the backend server and is imported in the server.py for the server to access all the data [40].

```
from flask import Flask, request, jsonify
import util

app = Flask(__name__)

@app.route('/classify_image', methods=['GET', 'POST'])
def classify_image():
    image_data = request.form['image_data']

    response = jsonify(util.classify_image(image_data))

    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

if __name__ == "__main__":
    print("Starting Python Flask Server For Sports Celebrity Image Classification")
    util.load_saved_artifacts()
    app.run(port=5000)
```

Figure 4. Code for Python flask

Base64-image.de website was used to convert the jpeg files into a unique string to be read by the model for testing purposes. Base64 is an encoding algorithm that can even convert Chinese characters, emoji, and even images into a "readable" string, which can be saved or transferred anywhere. Figure 5 contains the code to convert a certain image into a string. This even can decode the string and point us to the actual picture. A function is to be coded to convert the class name i.e names of the players into the corresponding number they are associated to i.e 0-4. This is done so that the computer can easily identify the picture and relate it to its corresponding name.

```
def get_cv2_image_from_base64_string(b64str):

    encoded_data = b64str.split(',')[1]
    nparr = np.frombuffer(base64.b64decode(encoded_data), np.uint8)
    img = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
    return img
```

Figure 5. Code to read the base64 file and return the cv2 image

Table 1 presents the player names and their serial numbers. Lionel Messi is assigned 0, Maria Sharapova is assigned 1, Roger Federer is assigned 2, Serena Williams is given 3 and Virat Kohli is assigned to 4. Once the image has been matched to the desired number that represents a certain celebrity the number then again has to be converted back to a name that represents that particular personality.

Table 1. Player names and their serial numbers

Player name	Serial number
Lionel Messi	0
Maria Sharapova	1
Roger Federer	2
Serena Williams	3
Virat Kohli	4

Figure 6 contains the logic for converting the assigned serial number to the corresponding celebrity. After classifying and converting the images to the name it is displayed on the website where the user gets the result. HTLM, JavaScript, and cascading style sheets (CSS) styling has been used to create the webpage that is encountered by the user. A collection of the photos and names of the celebrities that our model can detect was also shown as a static image on the top of the website.

```
def load_saved_artifacts():
    print("loading saved artifacts...start")
    global __class_name_to_number
    global __class_number_to_name

    with open("./artifacts/class_dictionary.json", "r") as f:
        __class_name_to_number = json.load(f)
        __class_number_to_name = {v:k for k,v in __class_name_to_number.items()}

    global __model
    if __model is None:
        with open('./artifacts/saved_model.pkl', 'rb') as f:
            __model = joblib.load(f)
    print("loading saved artifacts...done")
```

Figure 6. Converting a serial number to a name

A drop-off area for the image that has to be identified is made available. Any random image can be dropped off on the spot for our model to classify it and provide a probability score table to see who it is and if

not being able to classify it will show how similar it is to the personalities we have in the model. The highest score gotten by the celebrity will be displayed as the final result. The model does not discard any classification rather it just compares how similar the given photos are to the personalities present in the model.

3. RESULTS AND DISCUSSION

This section discusses how wavelet transforms happen and what is the output of this looks and works in a real practical scenario and see how similar a picture is to a given random input photo. Wavelet transformation convolutes the input signal with a scaled wavelet. Figure 7 shows the result of an image after wavelet transformation. After transforming the images, they are stacked one above the other for comparison. Images in the cropped folder can be used for model training. We will use these raw images along with wavelet transformed images to train our classifier. 60% of the images are used to train the model and the remaining 40% is used to test the reliability of the model formed.

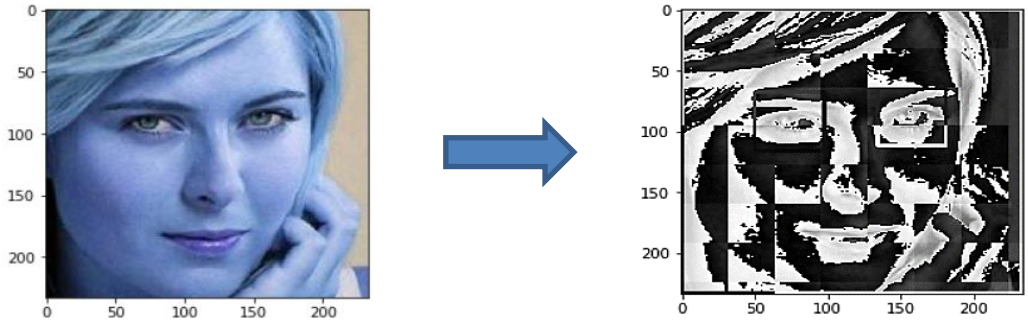

Figure 7. Result of an image after wavelet transformation

Figure 8 shows the front end of the website with the celebrity personalities that our model can classify. When the user drops a random image for classification classify image function is run to convert to wavelet, rescale and classify the image into corresponding celebrity personality and show the probability of similarity between that image and other personalities. A probability score table is also been coded in the software and after classification, it is automatically shown on the result page how similar the given picture is to the faces of other celebrities. A picture of Virat Kohli was uploaded to the website and after classifying to photo as a picture of Virat Kohli the following comparison table was shown in the picture. The probability score table is also implemented to provide the user with the statistics of how similar the uploaded image is to all the celebrities on our website.

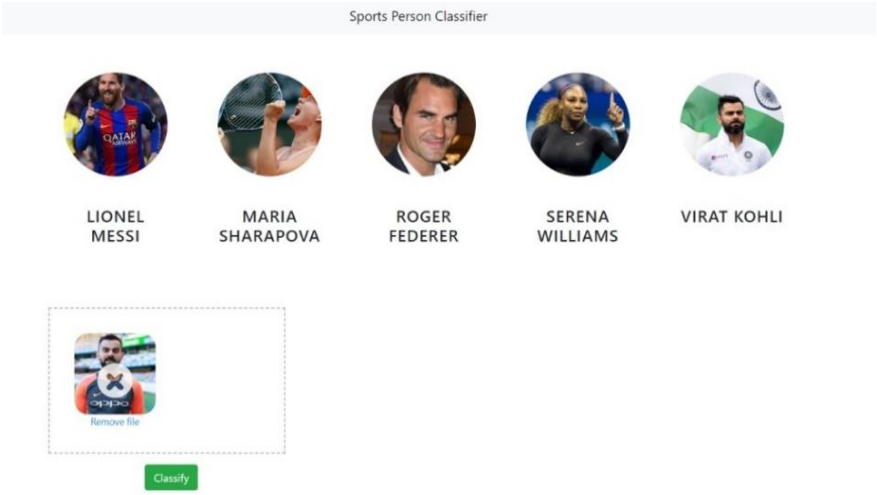

Figure 8. The frontend of the website

Figure 9 shows the result page and the probability table which compares and gives us the final answer and it shows how similar it is to other celebrities in the model. Table 2 presents the representation of the result table on the result page of the website and shows the probability table which contains the probability score that defines how accurately (in %) the user input picture is similar to the provided set of celebrities. Different parameters and methods were also used to determine the best score possible. When similar inputs were given to the SVM model it gave us a score of 90.4%, random forest (RF) gave a score of 59.52% and logistic regression (LR) resulted at 92.8%. But when a cross-fold validation data set was given as input SVM gave a result of 84.1%, RF gave a result of 70.6% and results were 80.9%. Hence SVM was considered to be the best among the three for this research.

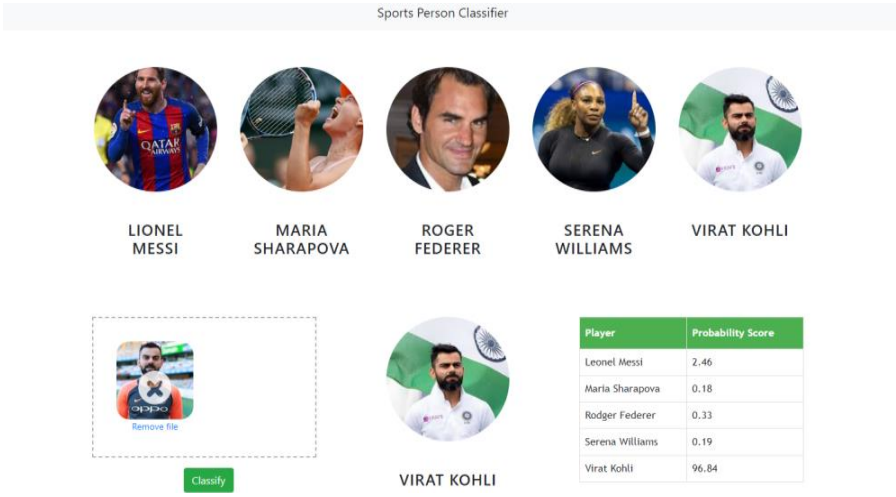

Figure 9. Frontend of the website after the result

Table 2. Probability table

Player	Probability score
Lionel Messi	2.46
Maria Sharapova	0.18
Roger Federer	0.33
Serena Williams	0.19
Virat Kohli	96.84

4. CONCLUSIONS

This paper was made keeping ML and its various tools as the base OpenCV, Haar cascade, and various other tools of this branch were used to make a classification work that was successfully able to differentiate among various personalities it was trained on. Various other technology like WT were also used to convert the images into a computer-readable format. It taught us how different pictures and sounds also can be represented by a string of characters, numbers, and alphabets. Using basic HTML, JavaScript and CSS the website was created. This paper's application is highly useable in the real world where our phones and various security cameras can use this method to identify unauthorized people.

ACKNOWLEDGMENTS

This research work was funded by "Woosong University's Academic Research Funding – 2023".

REFERENCES

[1] J. Wu, "Efficient HIK SVM learning for image classification," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4442–4453, Oct. 2012, doi: 10.1109/tip.2012.2207392.
[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, doi: 10.1109/cvpr.2016.90.

[4] D. P. Mishra and P. Ray, "Fault detection, location and classification of a transmission line," *Neural Computing and Applications*, vol. 30, no. 5, pp. 1377–1424, Dec. 2017, doi: 10.1007/s00521-017-3295-y.

[5] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, "Face detection and recognition using OpenCV," *In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Oct. 2019, doi: 10.1109/icccis48478.2019.8974493.

[6] E.-J. Lim, S.-Y. Ahn, Y.-M. Park, and W. Choi, "Distributed deep learning framework based on shared memory for fast deep neural network training," *In 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2018, doi: 10.1109/ictc.2018.8539420.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Jun. 2014, doi: 10.1109/cvpr.2014.223.

[8] Y. Li *et al.*, "A network-centric hardware/algorithm co-design to accelerate distributed training of deep neural networks," *In 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 2018, doi: 10.1109/micro.2018.00023.

[9] B. Ay, G. Aydin, Z. Koyun, and M. Demir, "A visual similarity recommendation system using generative adversarial networks," *In 2019 international conference on deep learning and machine learning in emerging applications (Deep-ML)*, Aug. 2019, doi: 10.1109/deep-ml.2019.00017.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, Mar. 2015, doi: 10.1109/tpami.2014.2345390.

[11] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014, doi: 10.1109/tkde.2013.39.

[12] H. Cao and S. Lian, "Internet forum texual information filtering mechanism for emergency based on Bi-LSTM neural network," *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2020, pp. 2051-2054, doi: 10.1109/ITAIC49862.2020.9339174.

[13] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[14] D. Fisch, E. Kalkowski, and B. Sick, "Knowledge fusion for probabilistic generative classifiers with data mining applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 652–666, Mar. 2014, doi: 10.1109/tkde.2013.20.

[15] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014, doi: 10.1109/tpami.2014.2321376.

[16] N. Jin and D. Liu, "Wavelet basis function neural networks for sequential learning," *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 523–528, Mar. 2008, doi: 10.1109/tnn.2007.911749.

[17] J. P. Barddal, H. M. Gomes, and F. Enembreck, "A survey on feature drift adaptation," *In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2015, doi: 10.1109/ictai.2015.150.

[18] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput & Applic*, vol. 28, pp. 3941–3951, 2017, doi: 10.1007/s00521-016-2294-8.

[19] J. Ma *et al.*, "Applications of computer vision for assessing quality of agri-food products: a review of recent research advances," *Critical reviews in food science and nutrition*, vol. 56, no. 1, pp. 113-127, 2016, doi: 10.1080/10408398.2013.873885.

[20] S. Sharma and R. Mehra, "Conventional machine learning and deep learning approach for multi-classification of breast cancer histopathology images-a comparative insight," *Journal of digital imaging*, vol. 33, no. 3, pp. 632-654, 2020 doi: 10.1007/s10278-019-00307-y.

[21] M. Panda, D. P. Mishra, S. M. Patro, and S. R. Salkuti, "Prediction of diabetes disease using machine learning algorithms," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, p. 284, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp284-290.

[22] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[23] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, Jun. 2014, doi: 10.1109/cvpr.2014.241.

[24] R. J. Hassan and A. M. Abdulazeez, "Deep learning convolutional neural network for face recognition: a review," *International Journal of Science and Business*, vol. 5, no. 2, pp. 114-127, 2021.

[25] W. Xiong, L. Zhou, L. Yue, L. Li, and S. Wang, "An enhanced binarization framework for degraded historical document images," *EURASIP Journal on Image and Video Processing*, vol. 2021, no. 13, 2021, doi: 10.1186/s13640-021-00556-4.

[26] R. Premalatha and P. Dhanalakshmi, "Enhancement and segmentation of medical images through pythagorean fuzzy sets-an innovative approach," *Neural Computing and Applications*, vol. 34, no. 4, pp. 11553–11569, Mar. 2022, doi: 10.1007/s00521-022-07043-5.

[27] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, Mar. 2014, doi: 10.1016/j.inffus.2013.04.006.

[28] Y. Lin *et al.*, "Large-scale image classification: fast feature extraction and SVM training," *In CVPR 2011*, Jun. 2011, doi: 10.1109/cvpr.2011.5995477.

[29] M. A. Chandra and S. S. Bedi, "Survey on {SVM} and their application in image classification," *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, Jan. 2018, doi: 10.1007/s41870-017-0080-1.

[30] O. W. Herng, A. S. A. Nasir, O. B. Chin, and E. S. M. M. Tan, "Harumanis mango leaves image segmentation on RGB and HSV Colour spaces using fast k-means clustering," *Journal of Physics: Conference Series*, 2107. no. 1, doi: 10.1088/1742-6596/2107/1/012068.

[31] A. A. Khaleel, "Multi-spectral images classification based on intelligent water drops algorithm," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 1, p. 116, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp116-125.

[32] D. Chudasama, T. Patel, S. Joshi, G. I. Prajapati, "Image segmentation using morphological operations," *International Journal of Computer Applications*, vol. 117, no. 18, 2015.

[33] T. Thaipanich, B. Oh, P.-H. Wu, D. Xu, and C.-C. Kuo, "Improved image denoising with adaptive nonlocal means ({ANL}-means) algorithm," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2623–2630, Nov. 2010, doi: 10.1109/tce.2010.5681149.

[34] S. A. Ahmed, S. Dey, and K. K. Sarma, "Image texture classification using artificial neural network (ANN)," *In 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, pp. 1-4. IEEE*, Mar. 2011, doi: 10.1109/ncetacs.2011.5751383.

[35] L. Wu and M. Li, "Applying the CG-logistic regression method to predict the customer churn problem," *In 2018 5th International Conference on Industrial Economics System and Industrial Security Engineering (IEIS)*, Aug. 2018, doi: 10.1109/ieis.2018.8597855.

[36] J. Fan, W. Xie, and T. Ge, "Automatic gray image coloring method based on convolutional network," *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/5273698.

[37] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, Feb. 2017, doi: 10.1109/tgrs.2016.2612821.
[38] S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, "A survey of image classification methods and techniques," *In 2014 International conference on control, instrumentation, communication and computational technologies (ICCICCT)*, Jul. 2014, doi: 10.1109/iccicct.2014.6993023.
[39] P. Ray and D. P. Mishra, "Support vector machine based fault classification and location of a long transmission line," *Engineering Science and Technology, an International Journal*, vol. 19, no. 3, pp. 1368–1380, Sep. 2016, doi: 10.1016/j.jestch.2016.04.001.
[40] H. Liu *et al.*, "Unsupervised classification of polarimetric SAR images integrating color features," *In 2014 IEEE Geoscience and Remote Sensing Symposium*, Jul. 2014, doi: 10.1109/igarss.2014.6947048.

BIOGRAPHIES OF AUTHORS

Debani Prasad Mishra received the B.Tech. in Electrical Engineering from the Biju Patnaik University of Technology, Odisha, India, in 2006 and the M.Tech. in power systems from IIT, Delhi, India in 2010. He has been awarded the Ph.D. degree in power systems from Veer Surendra Sai University of Technology, Odisha, India, in 2019. He is currently serving as Assistant Professor in the Department of Electrical Engineering, International Institute of Information Technology Bhubaneswar, Odisha. His research interests include soft computing techniques application in power system, signal processing and power quality. He can be contacted at email: debani@iiit-bh.ac.in.

Sanhita Mishra received the B.Tech. in Electrical Engineering from the Biju Patnaik University of Technology, Odisha, India, in 2009 and the M.Tech. in power systems from, Veer Surendra Sai University of Technology, Odisha, India in 2011. She has been awarded a Ph.D. degree in power systems from KIIT Deemed to be University, Odisha, India, in 2021. She is currently serving as Assistant Professor in the Department of Electrical Engineering, KIIT Deemed to be University Bhubaneswar, Odisha. Her research interests include soft Computing techniques application in power systems, signal processing, and power quality. She can be contacted at email: sanhita.mishrafel@kiit.ac.in.

Smrutisikha Jena received the B.Tech. in Electrical Engineering from the Biju Patnaik University of Technology, Odisha, India, in 2019, the M.Tech. in Electrical Engineering-power system specialization from Veer Surendra Sai University of Technology, Odisha, India, in 2021 and continuing Ph.D. degree in IIIT Bhubaneswar since 2021. Her research interests include machine learning, deep learning and signal processing in power system domain. She can be contacted at email: c121012@iiit-bh.ac.in.

Surender Reddy Salkuti received the Ph.D. degree in Electrical Engineering from the Indian Institute of Technology, New Delhi, India, in 2013. He was a Postdoctoral Researcher with Howard University, Washington, DC, USA, from 2013 to 2014. He is currently an Associate Professor with the Department of Railroad and Electrical Engineering, Woosong University, Daejeon, South Korea. His current research interests include market clearing, including renewable energy sources, demand response, smart grid development with the integration of wind and solar photovoltaic energy sources. He can be contacted at email: surender@wsu.ac.kr.