Nigerian Journal of Technology (NIJOTECH)
Vol. 39, No. 4, October 2020, pp. 1190 – 1197

Print ISSN: 0331-8443, Electronic ISSN: 2467-8821
www.nijotech.com
http://dx.doi.org/10.4314/njt.v39i4.27

OPTIMIZATION OF RBF-SVM HYPERPARAMETERS USING GENETIC ALGORITHM FOR FACE RECOGNITION

Y. Ibrahim[1,*], E. Okafor[2] and B. Yahaya[3]
[1, 2, 3] DEPT. OF COMPUTER ENGINEERING, AHMADU BELLO UNIVERSITY, ZARIA, KADUNA STATE, NIGERIA
Email addresses: [1] *yibrahim@abu.edu.ng*, [2] *eokafor@abu.edu.ng*, [3] *byahaya@abu.edu.ng*

ABSTRACT
Manual grid-search tuning of machine learning hyperparameters is very time-consuming. Hence, to curb this problem, we propose the use of a genetic algorithm (GA) for the selection of optimal radial-basis-function based support vector machine (RBF-SVM) hyperparameters; regularization parameter C and cost-factor γ. The resulting optimal parameters were used during the training of face recognition models. To train the models, we independently extracted features from the ORL face image dataset using local binary patterns (handcrafted) and deep learning architectures (pretrained variants of VGGNet). The resulting features were passed as input to either linear-SVM or optimized RBF-SVM. The results show that the models from optimized RBF-SVM combined with deep learning or hand-crafted features yielded performances that surpass models obtained from Linear-SVM combined with the aforementioned features in most of the data splits. The study demonstrated that it is profitable to optimize the hyperparameters of an SVM to obtain the best classification performance.

Keywords: *Face Recognition, Feature Extraction, Local Binary Patterns, Transfer Learning, Genetic Algorithm and Support Vector Machines.*

1. INTRODUCTION

The human face carries a lot of sensitive information about a person's identity and emotion. Face recognition is hence a challenging and interesting problem as it finds application in several areas such as banking, law enforcement, access control, person identification, etc. The typical recognition pipelines involved image acquisition, feature extraction, and image classification. The broad focus of this study is directed towards comparing the performances of different computer vision feature extraction techniques combined with heuristic or non-heuristic based supervised learning algorithms. Several computer vision methods were combined with supervised learning algorithm to perform face recognition tasks.

A previous study has shown that kernel-based support vector machine (SVM) has proven to be effective in many pattern recognition and classification applications such as image classification [1], face recognition [2], and liveness detection [3]. Traditional computer vision techniques have been inspired by hand engineered features that involve a lot of tedious feature engineering and are not transferrable to a variant of another domain. Some examples of hand-crafted features include; Local Binary Patterns (LBP) [4], Histogram of Oriented Gradients (HoG) [5], and Scale-Invariant Feature Transform (SIFT) [6]. Related work as reported in [7] investigated traditional and non-traditional computer-vision techniques for extracting features from facial images, then the resulting features were passed as input to conventional classifiers such as K-Nearest Neighbor (KNN), Artificial Neural Networks (ANN) or SVM. In recent times, several advances and evolution in artificial intelligence have led to the emergence of deep learning techniques [1] which have obtained

* Corresponding author: +234 – 703 – 762 – 7128

successful performance on different computer-vision challenges [8-13].

Some research papers have explored deep learning for face recognition with major emphasis on the use of softmax or sigmoid for computing the probability prediction of a target class [13-15]. The choice of deep learning techniques is motivated due to its ability to be transferred and co-adapt to a variant of another domain based on a special paradigm called transfer learning [16]. In other words, transfer learning involves using pre-trained models that have already been trained to perform specific tasks (on a large dataset), whose weights are initialized on a different domain while aiding to fine-tune the network and for learning novel filters. The resulting learned filters are used to extract another set of discriminative features for a new task.

Contribution: In this paper, we propose the use of pretrained weights from two convolutional neural network architectures; VGG16 (imagenet-vgg-verydeep-16) and CNN-F (imagenet-vgg-f) to extract spatial features from a set of facial images. The pretrained CNNs and the local binary pattern were treated as an arbitrary feature extractors. The resulting features were separately passed as input to two variants of an SVM; linear SVM and optimized RBF-SVM. The optimization of the RBF-SVM hyperparameters was actualized using the Genetic Algorithm. The end goal is to train each of the recognition models to perform face recognition. The study demonstrated that it is important to optimize the hyper-parameters of supervised learning algorithms to obtain the best classification, generalization, and improved discriminatory potential.

Paper Outline: The remainder of this paper is organized as follows: Section 2 discusses the two broad computer vision feature extraction techniques. Section 3 describes the examined SVM with consideration of heuristic optimization or not. A brief description of the used dataset was provided in Section 4. The simulation results are presented and discussed in Section 5. The conclusion and possible areas for future works were highlighted in Section 6.

2. FEATURE EXTRACTION

This section entails a detailed discussion about two main kinds of feature extraction methods; handcrafted and deep learning features.

2.1 Handcrafted Features

The handcrafted feature is a traditional or classical approach for extracting useful vectorial or representation from several images within a dataset lexicon. The approach to image extraction relies on tedious feature engineering and is not transferable to a variant of another domain. We examined one example of handcrafted features which is described below.

2.1.1 Local Binary Patterns Features

A local binary pattern (LBP) is a handcrafted feature descriptor often used in image representation applications. It is a simple texture operator which effectively characterizes texture in local neighborhood. Small image regions (typically of size 3x3) are used to encode the image pixels where neighborhoods pixels are thresholded against the center pixels. Let g_c represent the intensity value of the center pixel and g_p for the neighborhood pixels (where p = 0, 1, …, 7), then the pixels are thresholded according to (1).

$$s(g_p - g_c) = \begin{cases} 1, if\ g_p \geq g_c \\ 0, \quad otherwise \end{cases} \quad (1)$$

An 8-digit binary number is hence obtained for each pixel which is then converted to decimal. The LBP code is thereafter computed according to (2).

$$LBP = \sum_{P=0}^{7} S(g_p - g_c) \times 2^p \quad (2)$$

Where p is the number of neighborhood pixels from the centre (p=8 for a 3x3 window). The image descriptor is therefore the frequency of occurrence of all the patterns obtained over the entire image. The histogram representation of an entire image ($W \times H\ pixels$) whose LBP for all pixels has been computed can be constructed using (3).

$$H = \sum_{I=0}^{W-1} \sum_{J=0}^{H-1} f(LBP(i,j), p), p \in [0, N], \quad (3)$$

$$f(x,y) = \begin{cases} 1, & if\ x = y \\ 0, otherwise \end{cases} \quad (4)$$

Where N is the LBP pattern's maximal value.

2.2 Deep Learning Features:

As opposed to extracting features using handcrafted features, we adopted the use of transfer learning [16] in the context of deep CNNs. Transfer learning involves utilizing knowledge (pretrained weights) gained from one problem and extending the same to a new but related domain [16]. For example, the

knowledge derived from recognizing cars and ships can be used to recognize bikes. In our approach, we visualized the set of facial images I as a function encoded to map the images to a vector. Based on the motivation from previous works [17-19], we utilized the vector activities of the penultimate layer of existing pretrained weights from deep CNN earlier trained on ImageNet data [20]; for extracting facial features that are passed to supervised learning techniques for performing classification task. Training a deep learning architecture using pretrained weights presents the advantages; increases classification accuracy and learning speed while aiding to avoid overfitting problems due to the dropout regularization. In our current study, we used the pretrained weights of the VGG16 variant of the VGGNet [21] and CNN-F [17] as the backbone for feature extraction before passing the effective features to the supervised learning algorithms. The descriptions of the aforementioned deep learning architectures are explained below.

2.2.1 CNN-F

The CNN-F also called the imagenet-vgg-f was utilized for extracting facial features and was adjudged by [22] to be the best when compared with CNN-M, and CNN-S. The CNN architecture details of the networks were presented in Table I. The CNN-F is architecturally similar to the popular AlexNet [1].

Table 1: Architectures of CNN-F [23]

Architecture Layers	Details of Layers
conv1	64x11x11, st. 4, pad 0, LRN, x2 pool
conv2	256x5x5, st. 1, pad2, LRN, x2 pool
conv3	256x3x3, st. 1, pad 1
conv4	256x3x3, st. 1, pad 1
conv5	256x3x3, st. 1, pad 1, x2 pool
fc6	4096, dropout
fc7	4096, dropout
fc8	1000, softmax

Note that all the face images were resized accordingly as CNN-F strictly accepts input image of size 224×224×3. The CNN consist of 8 learnable layers: 5 convolutional layers (conv1-conv5) and 3 fully-connected layers (fc6-fc8). The pixel stride of 4 in conv1 ensures speedier processing [17]. Local response normalization (LRN) was applied to conv1 and conv2. We modified the latest fully connected layer (fc8) to contain 40 output nodes which represents the number of classes as described in ORL dataset lexicons.

2.2.2 VGG-16

VGGNet [24] is a powerful network that has a convolutional size of 3x3, a stride of 1x1, and a pooling window of 2x2. It has two successful structures namely VGG-16 (with 13 convolutional and 3 fully connected layers) and VGG-19 (with 16 convolutional and 3 fully connected layers) [24]. The used VGG-16 (*vgg_veydeep_16)* won the ILSVRC-2014 challenge. The pretrained models of VGG16 and VGG-19 were originally trained on ImageNet dataset which contains 1000 classes [20]. These models were fine-tuned for training instances of the neural network system to suit a 40-class multi-classification task, by extracting the deep transfer CNN features from the ORL face dataset. The pretrained networks used in this paper are available in the link. We employed a MATLAB deep learning framework known as MatConvNet [25], an open-source implementation for the aforementioned CNN architectures. Note that the handcrafted and deep learning features were passed to two different supervised learning algorithms as explained in the next section.

3. SUPPORT VECTOR MACHINE (SVM)

In this section, we described two forms of SVM based on their consideration of heuristic optimization of hyper parameters or not.

3.1 Linear SVM

We adopted the SVM classifier as reported in [26] because it presents good empirical performance. For the experiments, we used the popular LIBSVM library [27] which has an implementation of SVM with different kernel methods or non-kernel based. The SVM can be formulated using hypothetic expressions detailed as follows. Given the extracted LBP and deep learning features from the image samples and their corresponding labels defined as;

$(x_i, y_i), i = 1, \dots, l, x_i \in R^n, y_i = \{1,2, \dots ,40\}$

then LIBSVM solves the optimization problem given by:

$$\min_{w,b,\xi} \ \frac{1}{2} \mathrm{w}^T \mathrm{w} + C \sum_{i=1}^{n} \xi_i \tag{5}$$

Subject to:

$$y_i(w^T\phi(x_i)+b) \geq 1-\xi_i,\ \xi_i \geq 0 \quad (6)$$

The quantity ξ denotes the slack variable, $C>0$ is the regularization parameter which penalizes the error margin between the support vectors of the hyper-planes, and the variable ϕ accounts for the kernel function.

3.2 Radial Basis Function Kernel Based SVM

A challenging task in setting up an SVM model is the selection of appropriate kernel function and hyper-parameter values [28]. This is because optimal choice of these parameters aids the learning algorithm to obtain good classification accuracy results [28, 29] and presents effective generalization ability [30]. This paper considered comparing the performance of training and testing the respective extracted features on linear SVM (i.e. without kernel) and GA-optimized Gaussian (or RBF) based SVM to determine which classification framework generalizes better. The Gaussian (RBF) kernel is mathematically evaluated as:

$$\emptyset(x,x') = exp\left(\frac{-\|x-x'\|^2}{2\sigma^2}\right) \quad (7)$$

$$\emptyset(x,x') = \exp(-\gamma\|x-x'\|^2) \quad (8)$$

From (7); given that the cost factor is defined as; $\gamma = \frac{1}{2\sigma^2}$, where γ denotes variance of Gaussian bell around the support vectors called the cost factor. Therefore, for an SVM to be trained using the Gaussian kernel equation depicted in (8), the values of C and γ need to be properly tuned to obtain accurate results. Traditional methods often involve the use of a grid search to find good parameter combinations for the SVM which is tedious and time-consuming [31]. However, to get a near optimal C and γ values; this study trained a genetic algorithm evolutionary-based optimization technique with an expectation to improve the classification performance of the supervised learning technique.

3.2.1 Parameter Optimization using GA:

We employed the use of a binary-coding scheme to determine the chromosomes which consist of two main hyper-parameters; C and γ. To find the best hyper-parameter values for the RBF-SVM, GA optimizer was trained based on 5-fold cross-validation to determine the optimal parameter pair that reveals the best fitness performance. The GA optimizes the objective (fitness) function by minimizing the mean squared error (*MSE*) and maximizing the classification accuracy of the SVM. In the GA tuning process, for every C and γ value pair selected in each generation, the data is divided into 5 folds, using 4 distinct folds for training and the remainder for testing. The best results for each of the 5 cases are then recorded as the best pair for that particular generation (i.e. evaluating the RBF-SVM accuracy using these parameters). After exhausting the entire generation the C and γ pair that yielded the least MSE (highest classification accuracy) was treated as the optimal parameters. The GA based RBF-SVM fitness curve is shown in Figure 1. The chromosomes were generated based on a user customized GA whose parameters were arbitrarily set to; population size of 20, generation number of 50, C parameter was bounded in the range [0 100], and γ was constrained in the bound [0 0.020]. The C and γ chromosomes fitness values are computed and the extracted features from the computer vision methods are used for training an SVM classifier based on 5-fold cross-validation to obtain the accuracy. Genetic processes of selection (using stochastic universal sampling), single-point crossover, and uniform mutation continue with succeeding generations until convergence is achieved. A summary of the face recognition system pipeline is shown in Figure 2.

4. DATASET

In our experiment, we employed the ORL dataset. The dataset contains a total of 400 facial images of size 112x92. It has 40 classes (or persons) with each class containing 10 images. The images have varying facial illuminations, expressions, and different pose representations as shown in figure 3. The split distributions for the training and testing sets used during the experiments were arranged in the following split ratio; 50%:50%, 60%:40%, and 70%:30%.

5. SIMULATION RESULTS & DISCUSSIONS

We conducted experiments to evaluate the performance of the proposed method on the ORL face datasets.

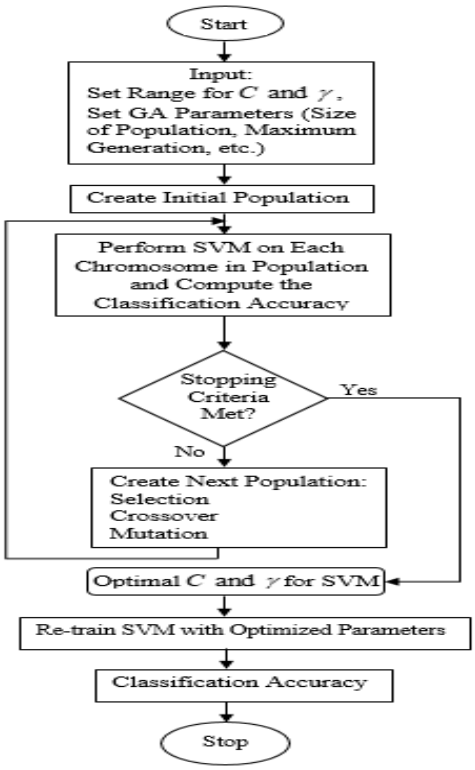

Figure 1: Flowchart of RBF-SVM system optimized by Genetic Algorithm

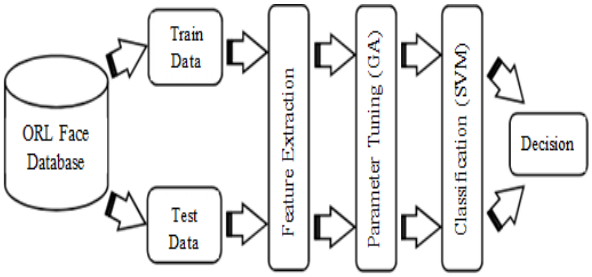

Figure 2: A summary of the face recognition pipeline

To evaluate the performance of the SVM classifier, we employed the use of performance metrics namely accuracy, sensitivity, and specificity whose formulae are computed and expressed in (9) - (11).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (9)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (10)$$

$$Specificity = \frac{TN}{TN + FP} \quad (11)$$

Where the variables TP, TN, FN, and FP represent the true positive, true negative, false negative and false positive respectively. In our experiments, we evaluated the test performance of the extracted features (LBP and CNN features) that were passed as input to the Linear SVM (i.e. no kernel) and optimizing RBF-SVM hyperparameters using genetic algorithm (GA-RBF-SVM). The convergence plot for the GA optimization based on LBP, CNN-F and VGG16 based features are shown in Figures 4 – 6. A summary of the optimal C and γ values for these methods are summarized in Table 2. These value pairs provide the best fitness (lowest cross validation error in terms of mean squared error) for the entire GA computation.

Figure 3: Samples of the ORL dataset showing different facial pose representation

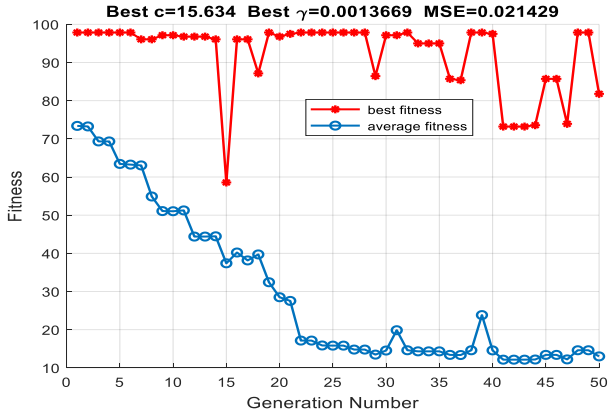

Figure 4:GA fitness curve of RBF-SVM on LBP features

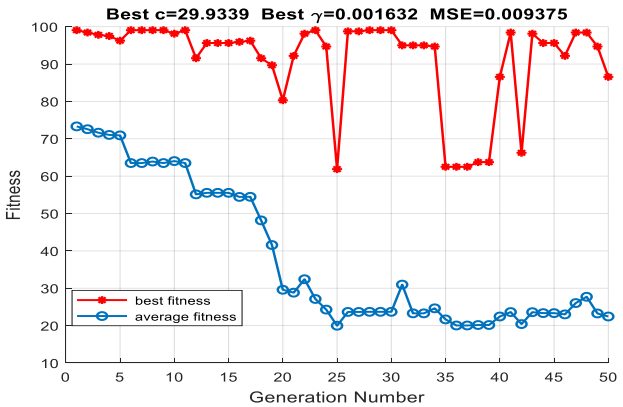

Figure 5: GA fitness curve of RBF-SVM on CNN-F features

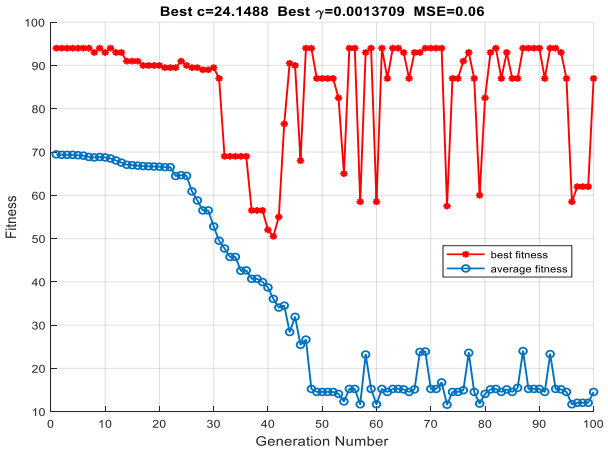

Figure 6: GA fitness curve of RBF-SVM on VGG16 features

These optimal values were passed to the RBF-SVM as the required parameters and then the same pipeline shown in Figure 3 are followed to actualize the recognition system. The preliminary evaluation of the supervised learning algorithms were based on 5-fold cross-validation based on a central goal to determine the RBF-SVM optimal hyperparameters. Based on the desired hyperparameters, we experimented on one final experimental run per method for each of the data splits. The examined supervised learning methods on both handcrafted and deep learning features yielded 100% classification accuracy on the training data. Hence to determine the best method, it is important to determine which of the models generalizes best on the testing set.

A summary of the results obtained in context of the performance metrics (accuracy, sensitivity, and specificity) on the different test partitions are reported in Tables 3-5.

From Tables 3-5, it can be observed that the classification accuracies for all three features were improved using the optimized GA-RBF-SVM pipeline. The sensitivity of the classifier or true positive rate (TPR) which tells how often the correct decision is obtained for an actual positive class also tends to be stable for most of the models (except CNN-F (40% testing instance) which slightly deteriorated). The specificity or true negative rate (TNR) which tells how often the correct decision is obtained when True value is negative also improved for most of the cases based on the data splits considered. From these results, it can be concluded that metaheuristic optimization algorithms such as GA can be used reliably to tune hyperparameters of RBF based SVM to improve the average classification accuracies of face recognition applications.

Table 2: Summary of the optimized SVM parameters

Method	Optimized C	Optimized γ
LBP-GA-RBF-SVM	15.634	0.0013669
VGG16-GA-RBF-SVM	24.1488	0.0013709
CNNF-GA-RBF-SVM	29.9339	0.001632

Table 3: Performance results of the various methods on a 50% test data

Method	Accuracy	sensitivity	specificity
LBP-Linear SVM	96.00	1.0000	0.9594
LBP-GA-RBF-SVM	96.50	1.0000	0.9641
VGG16-Linear-SVM	95.50	0.8000	0.9590
VGG16-GA-RBF-SVM	97.50	1.0000	0.9744
CNNF-Linear-SVM	97.00	1.0000	0.9692
CNNF-GA-RBF-SVM	99.50	1.0000	0.9948

Table 4: Performance results of the various methods on a 40% test data

Method	accuracy	sensitivity	specificity
LBP-Linear SVM	96.88	1.0000	0.9682
LBP-GA-RBF-SVM	97.50	1.0000	0.9744
VGG16-Linear-SVM	96.88	1.0000	0.9679
VGG16-GA-RBF-SVM	98.12	1.0000	0.9808
CNNF-Linear-SVM	98.12	1.0000	0.9810
CNNF-GA-RBF-SVM	98.75	0.6667	0.9936

Table 5: Performance results of the various methods on a 30% test data

Method	accuracy	sensitivity	specificity
LBP-Linear SVM	99.17	1.0000	0.9915
LBP-GA-RBF-SVM	99.17	1.0000	0.9915
VGG16-Linear-SVM	96.67	1.0000	0.9652
VGG16-GA-RBF-SVM	99.17	1.0000	0.9914
CNNF-Linear-SVM	97.50	1.0000	0.9744
CNNF-GA-RBF-SVM	98.33	1.0000	0.9828

5. CONCLUSIONS

The use of a genetic algorithm (GA) to optimize the hyperparameters of an RBF based support vector machine (RBF-SVM) classifier for face recognition application is developed in this paper. Features based on two paradigms were extracted; handcrafted features (using local binary patterns) and deep transfer feature based on pre-trained convolutional neural networks. The performance evaluation of these methods on the ORL dataset showed that using a meta-heuristic optimization algorithm such as GA to tune the parameters of an RBF based Support Vector machine can enhance the classification performance of face recognition

pipelines than recognition models built using Linear SVM on most instances of the different split distribution. Possible areas for future research include; investigate other feature extraction techniques combined with the optimization scheme presented in this work. The explored recognition pipeline can also be extended to other image classification applications. Lastly, it would be interesting to explore other forms of deep learning architectures with more depth to access the peak of performance improvement

6. REFERENCES

[1] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "Imagenet classification with deep convolutional neural networks", *in Proceedings of the International Conference on Neural information processing systems*, Lake Tahoe, NV, USA, 3-6 December 2012; pp. 1097-1105.

[2] Rahim, M. A., Azam, M. S., Hossain, N. and Islam, M. R. "Face recognition using local binary patterns (LBP)", *Global Journal of Computer Science and Technology*, Vol. 13, No 4-F, 2013.

[3] Ibrahim, Y., Mu'azu, M. B., Adedokun, E. A. and Sha'aban, Y. A. "A performance analysis of logistic regression and support vector machine classifiers for spoof fingerprint detection," *in 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, 2017, pp. 1-5.

[4] Ojala T., Pietikainen M., Maenpaa T. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis*, Vol. 24, no. 7, pp. 971-987, 2002

[5] Dalal N. and Triggs B., "Histograms of oriented gradients for human detection," *in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, vol. 1, pp. 886-893: IEEE.

[6] Lowe D. G. "Object recognition from local scale-invariant features," *in Proceedings of the seventh IEEE international conference on computer vision*, 1999, vol. 2, pp. 1150-1157.

[7] Nanni, L., Ghidoni, S., and Brahnam, S. "Handcrafted vs. non-handcrafted features for computer vision classification", *Pattern Recognition,* Vol. 71, pp. 158-172, 2017.

[8] He K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. "Mask r-cnn". *In Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[9] Suman C., Kumar, P., Saha, S. and Bhattacharyya, P., 2019, December. "Gender, Age and Dialect Recognition Using Tweets in a Deep Learning Framework-Notebook for FIRE 2019". *In Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings*. CEUR-WS. org, Kolkata, India, December (pp. 12-15).

[10] He K., Zhang, X., Ren, S. and Sun, J., 2016. "Deep residual learning for image recognition". *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[11] Haider K. Z., Malik, K.R., Khalid, S., Nawaz, T. and Jabbar, S., 2019. "Deepgender: real-time gender classification using deep learning for smartphones". *Journal of Real-Time Image Processing*, 16(1), pp.15-29.

[12] Okafor E., Akut J., Ibrahim Y., Kunya A. B., and Jibril Y., (2020). "The Detection of face gender using single-shot multi-box detector", *International Journal of Information Processing and Communication (IJIPC)*, Vol 8 (1), on pp. 123-135.

[13] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C. Y. and Berg A. C., (2016), October. "SSD: Single shot multibox detector", *In European conference on computer vision* (pp. 21-37). Springer, Cham.

[14] Serra X. and Castán J. (2017). "Face recognition using Deep Learning," *Catalonia: Polytechnic University of Catalonia*, Vol. 78.

[15] Sun Y., Liang D., Wang X. and Tang X., "Deepid3: Face recognition with very deep neural networks," *arXiv preprint: 11502.00873,* 2015.

[16] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. "Decaf: A deep convolutional activation feature for generic visual recognition," *Proceedings of the 31st International conference on machine learning*, PMLR 32(1), 2014, pp. 647-655.

[17] Chatfield, K., Simonyan, K., Vedaldi, A. and Zisserman, A. "Return of the devil in the details: Delving deep into convolutional nets", *CoRR,* abs/1405.3531, 2014.

[18] Dorian, C., Lefort R., Bonnel, J., Zarader, J. L. and Adam, O., "Bi-class classification of humpback whale sound units against complex background noise with Deep Convolution Neural Network,"(2017), *arXiv preprint: 1703.10887*.

[19] Zeiler, M. D. and Fergus, R. "Visualizing and understanding convolutional networks," *European conference on computer vision*, 2014, pp. 818-833: Springer.

[20] Deng J., Dong W., Socher R., Li L.-J., Li K., and Fei-Fei L., "Imagenet: A large-scale hierarchical image database," *in 2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248-255.

[21] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," 2014.

[22] Dorian, C., Lefort, R., Bonnel, J., Zarader, J.-L. and Adam, O. "Bi-class classification of humpback whale sound units against complex background noise with Deep Convolution Neural Network," *arXiv preprint arXiv*:1703.10887, 2017.

[23] Chatfield, K., Simonyan, K., Vedaldi, A. and Zisserman, A. "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv*:1405.3531, 2014.

[24] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint: 1409.1556*, 2014.

[25] Vedaldi, A., Lux, M., and Bertini, M., "MatConvNet: CNNs are also for MATLAB users," *ACM SIGMultimedia Records.* Vol. 10, no. 1, pp. 9, 2018.

[26] Vapnik, V., Golowich, S. E., and Smola, A. J., "Support vector method for function approximation, regression estimation and signal processing," *in Advances in neural information processing systems*, 1997, pp. 281-287.

[27] Chang, C.-C. and Lin, C.-J. "LIBSVM: A library for support vector machines", *ACM transactions on Intelligent Systems and Technology.* Vol. 2, no. 3, pp. 1-27, 2011.

[28] Syarif I., Prugel-Bennett A., and Wills G., "SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, (2016) Vol. 14, no. 4, pp. 1502-1509.

[29] Ardjani, F., Sadouni, K. and Benyettou, M. "Optimization of SVM multiclass by particle swarm (PSO-SVM)," *in 2nd International Workshop on Database Technology and Applications (DBTA)*, 2010, pp. 1-4: IEEE.

[30] Xiao T., Ren D., Lei S., Zhang J. and X. Liu, "Based on grid-search and pso parameter optimization for support vector machine," *Proceedings of the 11th IEEE World Congress on Intelligent Control and Automation (WCICA)*, 2014, pp. 1529-1533.

[31] Li X. and Kong J., "Application of GA-SVM method with parameter optimization for landslide development prediction," *Natural Hazards and Earth System Sciences*, vol. 14, no. 3, p. 525, 2014.