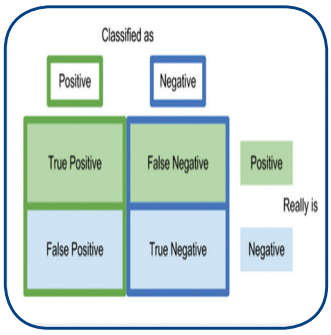
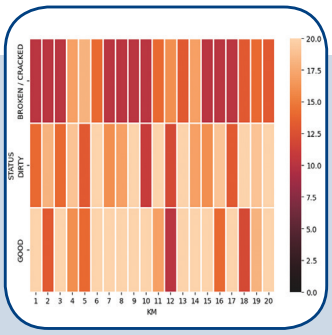
Volume 16, Number 3, 2025 ISSN 1847-6996

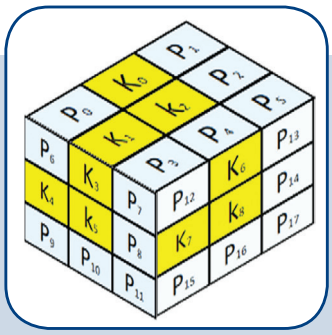
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY **OSIJEK**

IJECES International Journal of Electrical and Computer Engineering Systems

International Journal of Electrical and Computer Engineering Systems







ISSN: 1847-6996

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Croatia

Osijek, Croatia | Volume 16, Number 3, 2025 | Pages 195 - 263

The International Journal of Electrical and Computer Engineering Systems is published with the financial support of the Ministry of Science and Education of the Republic of Croatia

CONTACT

International Journal of Electrical and Computer Engineering Systems (IJECES)

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b, 31000 Osijek, Croatia
Phone: +38531224600, Fax: +38531224605
e-mail: ijeces@ferit.hr

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.
Giro account: 2390001 - 1100016777, Croatian Postal Bank

Journal is referred in:

- Scopus
- Web of Science Core Collection (Emerging Sources Citation Index - ESCI)
- Google Scholar
- CiteFactor
- Genamics
- Hrčak
- Ulrichweb
- Reaxys
- Embase
- Engineering Village

Bibliographic Information
Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003
Published: quarterly
Circulation: 300

IJECES online
https://ijeces.ferit.hr

TABLE OF CONTENTS

Deep Learning-Based Approach for Disease Stage Classification of Sunflower Leaf

Original Scientific Paper

Rupali Sarode*
Thadomal Shahani Engineering College, Computer Engineering Department,
Off Linking Road, Bandra(west), India
rupali.patil@thadomal.org

Arti Deshpande
Thadomal Shahani Engineering College, Computer Engineering Department,
Off Linking Road, Bandra(west), India
arti.deshpande@thadomal.org

*Corresponding author

Abstract *– Accurate disease severity evaluation is crucial for managing the disease and yield loss. The classification of disease stages is essential for the estimation of disease severity. It takes extensive time for cultivators and botanical researchers to meticulously examine each leaf image and identify the disease stage to assess the severity of the disease at the field scale. Extracting the damaged leaf area is also achievable with image segmentation, although there are drawbacks such as threshold selection and lack of grayscale difference. Thus, deep learning has produced recent breakthroughs in various fields, such as high-resolution image synthesis, recognition, and categorization of images. In this work, the disease stages of two diseases (Alternaria leaf blight and Powdery Mildew) are classified using sunflower leaf images taken from sunflower farms in India (Marathwada State) during the Rabi season. With the help of botanists, images are labeled as three disease stage classes and one healthy stage as ground truth. A series of deep convolutional neural networks (Visual Geometry Group models with 16 and 19 neurons, respectively) with transfer learning and fine-tuning approach is trained, validated, and tested using stratified k-fold values four and five. The findings indicate that VGG16, with k-fold=5, gives the highest testing accuracy, which is 90.25%, with fine-tuning for Alternaria Leaf Blight. For VGG19 with kfold=5, the highest testing accuracy is 86.89% with fine-tuning for Powdery Mildew. Additionally, confidence interval calculation shows smaller intervals of 3% and 4% with a significance level of 95% for the VGG16 and VGG19 models, respectively.*

Keywords: *Convolution neural network, transfer learning, fine-tuning, multiclass classification, Alternaria leaf blight, Powdery Mildew*

Received: August 24, 2024; Received in revised form: October 27, 2024; Accepted: November 12, 2024

1. INTRODUCTION

The sunflower plant serves several important functions. It recycles nutrients and organic matter from the soil through its roots, produces oil from its seeds, provides feed for animals, acts as green manure when its leaves are used, and also produces flowers and honey [1]. In various regions of India, diseases like Powdery Mildew, Alternaria leaf blight, and Downy Mildew have impacted sunflower plants [2]. Growers can make informed decisions at the field level to protect plants using early disease prediction and forecasting. Predicting and forecasting diseases will be essential for safeguarding plants, enabling prompt action to prevent crop loss and enhance oilseed yield and production. This research will develop technologies to enhance the nutritional and medicinal benefits of sunflower oilseed crops, which are widely used as functional food [3]. Sunflower has various nutrients which are beneficial for humans as well as for animal health. It is rich in essential nutrients, including protein, fiber, unsaturated fats, copper, zinc, selenium, iron, and vitamins, especially vitamin E. Sunflower seed meal is commonly used as animal and pet feed due to its high content of sulfuric amino acids. It can also be used as a salted or roasted snack or used as cooking oil. [4] Sunflower seed production Vol. in India was 544 in the fiscal year 2013. It decreased to 228 in the fiscal year 2022, then rose to 250 in the fiscal year 2023, and finally increased to 279 in the same year. [5] In contemporary times, CNN is often used in agricultural research due to its powerful image feature processing capabilities. The most common applications of deep learning include plant and crop classification, which aids in robotic harvesting, pest control, yield forecasting, and disaster monitoring. At the field scale, manual diagnosis of plant diseases and

their severity is time-consuming. [6] Deep learning has greatly enhanced computer vision by enabling computers to analyze, understand, and make intelligent decisions based on visual data.

Computer vision advances through the development of convolutional neural networks and deep learning algorithms [7]. Harm to the plant and its growth is based on the extent of the disease. Growers assess disease severity on a field scale by observing individual plant leaves. This manual process is expensive and takes a lot of time. As a result, by employing various new deep learning technologies, this lengthy procedure could become automated, allowing growers to make decisions earlier and, in less time, to protect plants.

2. RECENT STUDIES:

Numerous researchers have focused on agriculture to assist growers in identifying and classifying various crop diseases, as well as in diagnosing and forecasting them. Field observations can often be slow and inefficient. Deep learning approaches have been applied by multiple researchers to automate disease assessment for rice, apple, corn, and cotton crops. Zahraa Al Sahili et al. [8] provided an AgriNet dataset of 1.6 lakh images recorded from 19 distinct places and grouped into 423 plant types and illness classifications. Five ImageNet architectures were used to categorize plant species, diseases, pests, and weeds, including VGG16, VGG19, Inception-V3, InceptionResNet-V2, and Xception pre-trained networks. The VGG19 model provided the highest accuracy of 94%, whereas the InceptionV3 model achieved the minimum accuracy of 87%. To further enhance the AgriNet project, the authors suggested performing more complex data augmentation on its datasets. However, due to the limited number of agricultural public datasets, the research community is recommended to convert the data set's private information to publicly available at any moment.

Srinivasa Rao Dammavalam et al. [9] recommended employing Deep Learning models such as VGG16 and VGG19 to classify leaf images. From the Kaggle data sets [10], 14 crop leaf images were used and classified as healthy and infected images. Both models gave better results than others, such as ResNet50, DenseNet121, ResNet50V2, MobileNet, MobileNetV2, etc. The study's authors strongly advocate for including unique leaf disease classifications for each crop in future research.

Le Yang et al. [11] suggested a model for identifying corn weeds using SE-VGG16, which was evaluated on an image dataset [12] of corn seedlings and weeds. Using a Canon PowerShot SX600 HS camera, 6000 photos from the dataset were taken. These photos include one corn seedling and four weed categories—bluegrass, Chenopodium album, Cirsium setosum, and sedge—each with 1200 photos. Squeeze-and-Excitation mechanism is used with the VGG16 model to concentrate an important part of images and it gave superior weed identification results than VGG16. To support the complete agricultural production process and increase agrarian efficiency in production, deep learning combined with agricultural output will be applied to other fields in the future.

Prabira Kumar Sethy et al. [13] used 11 CNN models with fine-tuning approaches and deep feature plus support vector machine (SVM). Four types of rice leaf diseases—bacterial blight, blast, brown spot, and tungro—were identified using 5932 on-site photos gathered from agricultural sites in Odisha, India's Sambalpur and Bargarh districts. To perform feature extraction, the mentioned deep learning models were applied: InceptionResNetV2, GoogleNet, AlexNet, VGG16, InceptionV3, VGG19, ResNet18, ResNet50, ResNet101, DenseNet201, and XceptionNet. A Support Vector Machine is used to classify the extracted features. The transfer learning approach was used again to identify the four diseases of rice leaf. Finally, the outcomes of transfer learning and feature extraction were assessed.

Ghazanfar Latif et.al [14] proposed detecting and classifying six distinct diseases: healthy, narrow brown spot, leaf scald, leaf blast, brown spot, and bacterial leaf blight using deep CNN transfer learning with the VGG19 model. The image dataset [15] used has 2167 photos classified based on the six diseases listed. The final evaluation uses accuracy, precision, and the F1 measure. This article proposed that the future scope is to acquire field-scale images using drone technology in conjunction with IoT technology and a deep learning approach in real-world circumstances.

3. DISEASE SEVERITY ESTIMATION:

Guan Wang et al. [16] mentioned that the degree of damage caused by a plant disease is determined by its severity. Usually, experienced professionals use visual inspection of plant tNo. to rate the extent of plant diseases. To estimate disease severity, based on disease extent, classes made as Healthy, Initial, Middle, and Last with the help of a botanist. Deep learning networks are trained and tested using such class-wise images and evaluated further to classify disease stages.

3.1 DATASET INFORMATION:

This article considers two sunflower leaf diseases: Alternaria Leaf Blight and Powdery Mildew. The images used are collected from India. A total of 378 sunflower leaf images were considered for this study. These images belong to two diseases, Alternaria leaf blight, and Powdery Mildew, with their disease stage, and 105 are healthy images. A smartphone camera with 64 Megapixels was used to take images of the Rabi season. With the aid of plant researchers, the disease's extent was determined and divided into three stages: the initial stage, which ranged from 0 to 30%. The middle stage ranged from 31% to 60%, and the last stage ranged from 61% to 100%. The following table shows the count of images for each disease and its stage.

Table 1. Disease Stage count per disease

Sr. No	Disease Name	Disease Grade and Stage	Count of Original images
1.	Powdery Mildew	1-Stage1	64
		2-Stage2	84
		3-Stage3	38
2.	Alternaria Leaf Blight	1-Stage1	34
		2-Stage2	33
		3-Stage3	20
3	Disease Free	Healthy-Stage0	105
	Total		378

3.2. DISEASE INFORMATION

1. Alternaria Leaf Blight:

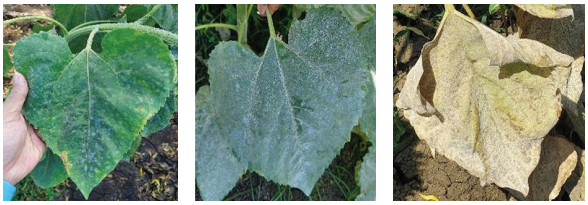
The disease affects the stems, sepals, petals, and leaves, causing brown spots. The disease stage is shown in Fig 1 a), b), and c). The leaves have dark brown spots with a golden circle around them and a pale edge. Later on, the spots become larger and take on an irregular shape with concentric rings. Leaf fall and dryness are caused by bigger, uneven lesions formed when multiple spots come together [17].

(a) (b) (c)

Fig.1. Brown spots of Alternaria Leaf Blight, a) Stage1, b) Stage2, c) Stage3

2. Powdery Mildew:

On the leaves, the disease causes white, powdery growths. The disease stage is shown in Fig 2 a), b), and c). A white to grey mildew occurs on the upper surface of older leaves. As the plant ages, areas of white mildew can be seen with black pinhead-sized leaves. The impacted leaves lose color, curl, become chlorotic, and eventually die. [17]



(a) (b) (c)

Fig. 2. White powdery growth on Sunflower Leaves, a) Stage1, b) Stage2, c) Stage3

4. CONCEPTS AND METHODOLOGY USED

4.1. DATA PREPROCESSING

The ImageDataGenerator class from Kera's library was used to apply data augmentation, increasing the sample size. This procedure comprised a rotation range of 40, a shear and zoom factor of 0.2, and a brightness range of 0.5 to 1.5 with a horizontal flip. The desired image size is (224,224), and a total of images, including training and testing, were generated after the dataset augmentation [18].

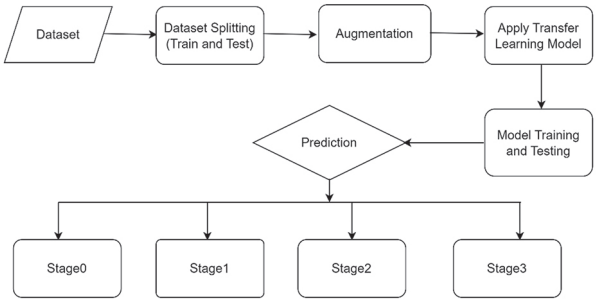


Fig. 3. Methodological block diagram

Table 2. Sample Size after Augmentation

Stage	Number of images per class after Augmentation	
	Alternaria Leaf Blight	Powdery Mildew
Healthy-Stage0	610	610
1-Stage1	685	641
2-Stage2	672	732
3-Stage3	598	767
Total	2565	2750

4.2. DEEP LEARNING PROPOSAL:

1. Convolution Neural Network:

The foundation of Artificial Intelligence is Deep Learning. Deep Learning technologies have multiple uses in agriculture, including disease detection, disease identification and categorization, and disease severity assessment. The Convolution Neural Network, a component of Deep Learning architecture, is a multilayered, hierarchical network that functions nonlinearly and resembles the human brain. Convolution neural networks (CNN) have so far exhibited the greatest power in image classification.[19].

2. Model Regeneration:

The deep learning model's potential is to acquire knowledge from a hierarchical representation of features effortlessly. The first layers of CNN-extracted features are always generic, while features at later layers are increasingly specialized. Hence, to perform classification based on required interest, the model needs to be regenerated by adding a new classifier as per the interest, and finally, the model needs to be fine-tuned based on three approaches:

1. Train the entire model learning from scratch.
2. Train some layers and freeze the other layers—The network's weights can be controlled by keeping more layers frozen for small datasets and training more layers for large datasets.

3. Freeze layers of the convolution model—Keep the convolution layer base model as it is and use its output to give to the classifier.

In the first two approaches, the learning rate must be carefully selected (usually smaller) to avoid knowledge loss. In the last approach, the pre-trained model can be used as it is and based on extracted features to classify required interest [20]. For this approach, the Adam optimizer is used with a learning rate of 0.0001.

3. Fine-tuning with Transfer Learning:

The transfer Learning process is based on the third approach, as mentioned above. In this process, first, select the pre-trained networks VGG [21] and InceptionV3 [22], which are available to use on Keras [23].

The second is to classify the required problem based on the size similarity matrix, as shown below in Fig 3 [20]. This size similarity matrix shows how the size of the dataset is related to the model's fine-tuning. Hence, there is a proper mapping based on the number of dataset images and the number of layers to be kept, not frozen. Finally, the third is to select and try various values of fine-tuning parameters based on the dataset size and pre-trained model dataset. So, for disease stage estimation, from Fig. 4, orange circled mapping is used to incorporate transfer learning with fine-tuning parameter of value=2; as per class, there are around 600 images.

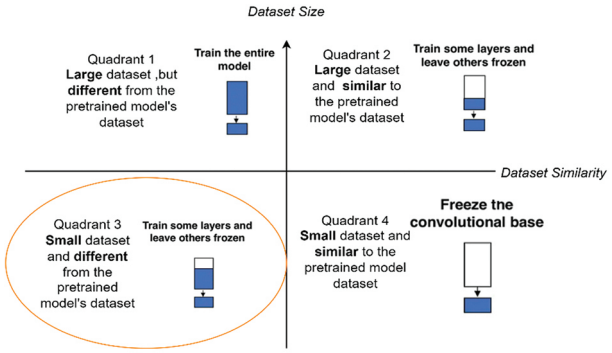


Fig. 4. Size-similarity matrix & decision map for fine-tuning pre-trained models

4. VGG16 and VGG19:

We employed two architectures, VGG16 and VGG19, for the fine-grained disease stage classification problem with minimal training data. Each architecture uses transfer learning by fine-tuning the higher layers of a pre-trained deep network to classify the disease stage.

The VGG architecture consists of three fully linked layers, a SoftMax activation function at the end, and Conv-1 Layer with 64 filters, Conv-2 with 128 filters, Conv-3 with 256 filters, and Conv 4 and Conv 5 with 512 filters. A max-pooling layer of a 2x2-pixel window with a stride of 2 follows each filter of size 3 × 3, a Rectified Linear Units (ReLu) activation, and all layers are followed by a dropout layer with a dropout ratio of 20% followed by the last convolutional layer. The final fully connected layer produces four outputs, one for each of the four classes. The SoftMax layer uses these outputs to determine the probability output. For both models, the input shape is 224 ×224 × 3, and the kernel size is 3 x 3 pixels. Fig 5 and 6 show the model architecture. The VGG16 model has 16 convolution layers, and we trained the model by freezing the first 12 layers for the VGG19 model, which has 19 convolution layers. So, both models are retrained by freezing 12 and 15 layers to obtain precise disease stage classification. (Refer to Table 3).

Table 3. Details of fine-tuning used

Model	Frozen layer	Trainable Layer
VGG16	12	4
VGG19	15	4

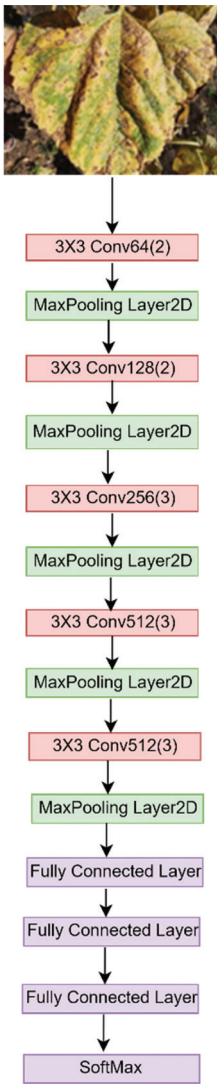


Fig. 5. VGG16 architecture

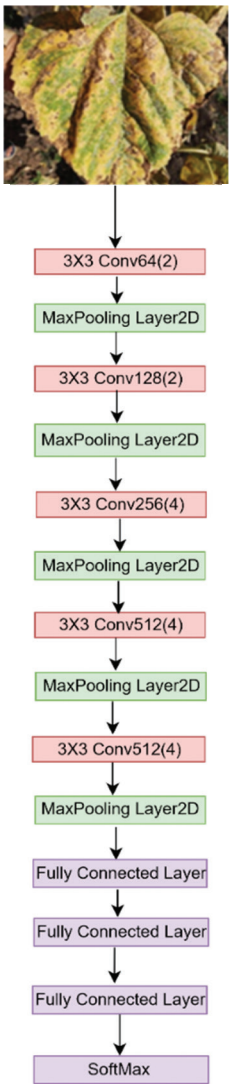


Fig. 6. VGG19 architecture

5. EXPERIMENTAL PREPARATION AND ASSESSMENT

This work uses a dataset of sunflower images of two diseases. Google Colab is used to train, validate, and test phase of both models. As shown in Table 2, each class has uneven samples per class. Hence, this uneven sample distribution makes the model learn slowly, re-

sulting in a biased model. Thus, to mitigate this, stratified K-fold cross-validation is employed. The samples are organized into K strata to provide nonoverlapping sets. The first strata from each class are then combined into the first fold, the second strata from each class into the second fold, and so on, to generate the stratified folds. By replicating the dataset's initial groupings, folds are created. After that, one-fold is used as the test set, and the remaining K-1 folds for training in each K-Fold Cross-Validation process iteration (Refer to Figs. 7 and 8).

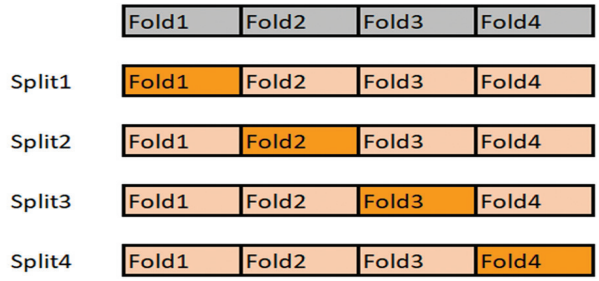


Fig. 7. K4 Cross Validation

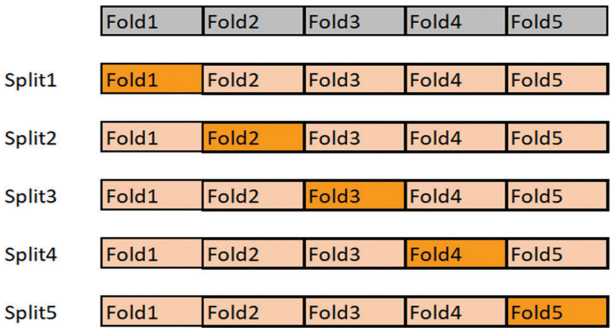


Fig. 8. K5 Cross Validation

Well-known Python machine-learning toolkit Scikit-Learn natively facilitates stratified K-Fold Cross-Validation. To reduce data balancing, stratified K-fold cross-validation ensures that samples from each class are used for testing and training the model. Using K=4 and 5 values, image samples are arranged into K strata in this work to construct non-overlapping sets. There will be 2360 images distributed across four classes since 590 image samples are kept for each class to have even samples. There are 236 images for testing and 2124 for training because the split is 10% for testing and 90% for training. Since this work uses K4 and K5 cross-validation, there are 531 and 425 images per fold, respectively. The training dataset was used to train the transfer learning model with a batch size 16. and the test dataset is used to assess it. Both models trained till epoch10 with fine-tuning =2 and without fine-tuning. TensorFlow, Kera's, and Sklearn Python libraries are combined in the models. The hyperparameters for both models are defined in Table 4. The network weights were iteratively adjusted using the Adam optimizer (with a learning rate of 0.0001) to underrate the loss function during the model construction based on training data [24]. A sparse categorical cross-entropy function with metric accuracy is utilized as the loss function because this task requires multiclass image classification. The loss function calculates the difference between the input label and the predicted result.

Table. 4. Hyperparameters used

Metrics	Metrics Value
Batch size	16
Optimizer	Adam
Learning rate	0.0001
Criterion	Categorical cross Entropy Loss

6. EXPERIMENTAL RESULTS

6.1. ALTERNARIA LEAF BLIGHT

Fig. 9 shows both model's training and testing accuracies per fold without fine-tuning for Alternaria Leaf Blight. The testing accuracy of VGG16 on the holdout dataset is greater than VGG19 after K5 cross-validation.

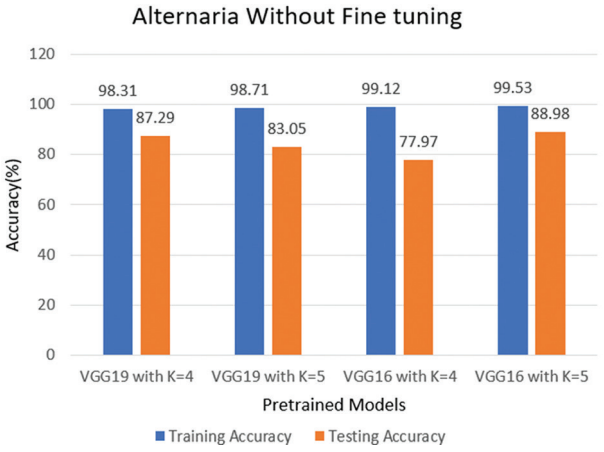


Fig. 9. Training and Testing accuracies for Alternaria Leaf Blight (without fine tuning)

The confusion matrix for the VGG16 model on the hold-out test set is presented in Table 5. In the initial stage, the model achieves a classification accuracy of 100% for K=5. The accuracy rates for the healthy and last stages are 93.22%, while the middle stage has a lower accuracy of 74.57%, making it more susceptible to misclassification.

Table 5. Confusion Matrix for Alternaria Leaf Blight for VGG16 at K5 cross-validation (without Fine-tuning)

		Predicted			
		Stage 0	**Stage1**	**Stage2**	**Stage3**
Ground Truth	**Stage 0**	**55**	1	0	3
	Stage1	0	**59**	0	0
	Stage2	0	9	**44**	6
	Stage3	1	3	0	**55**

Table 6 presents the confusion matrix for the VGG19 model tested on the hold-out dataset. All initial stages are accurately classified for K=4. The accuracy for the healthy stage is 91.52%, while the last stage achieves an accuracy of 86.44%. However, the middle stage has a lower accuracy of 71.18%, indicating it was frequently misclassified.

Table 6. Confusion Matrix for Alternaria Leaf Blight for VGG19 at K4 cross-validation without fine-tuning)

		Predicted			
		Stage 0	Stage1	Stage2	Stage3
Ground Truth	Stage 0	**54**	0	0	5
	Stage1	0	**59**	0	0
	Stage2	0	9	**42**	8
	Stage3	1	5	2	**51**

Fig. 10 shows training and testing accuracies per fold with fine-tuning for Alternaria Leaf Blight. The testing accuracy for VGG16 and VGG19 on the holdout dataset is improved with fine tuning=2

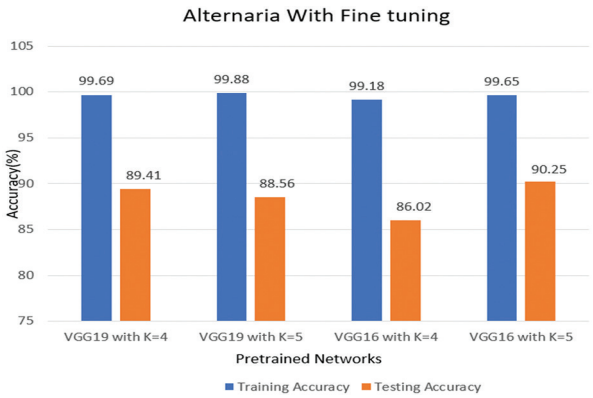


Fig 10. Training and Testing accuracies for Alternaria Leaf Blight with fine-tuning

The confusion matrix of the VGG16 model on the hold-out test set is shown in Table 7. The accuracy of the healthy stage is 86.44%, and the accuracy of the initial and last stages is 96.61%. The middle stage is not classified correctly, with an accuracy of 76.27%.

Table 7. Confusion Matrix for Alternaria Leaf Blight for VGG16 at K5 cross-validation (with fine-tuning)

		Predicted			
		Stage 0	Stage1	Stage2	Stage3
Ground Truth	Stage 0	**54**	0	0	5
	Stage1	0	**59**	0	0
	Stage2	0	9	**42**	8
	Stage3	1	5	2	**51**

The confusion matrix for the VGG19 model with K4 cross-validation on the hold-out data set is shown in Table 8. The Accuracy of the healthy stage is 72.88%. The initial stage is correctly classified at K=4. The Accuracies for the middle and last stages are 94.91% and 89.83% respectively.

Table 8. Confusion Matrix for Alternaria Leaf Blight for VGG19 at K4 cross-validation (with fine-tuning)

		Predicted			
		Stage 0	Stage1	Stage2	Stage3
Ground Truth	Stage 0	**43**	0	3	13
	Stage1	0	**59**	0	0
	Stage2	0	0	**56**	3
	Stage3	0	5	1	**53**

By the VGG19 model, stage 1 is more correctly classified than VGG16.

6.2. POWDERY MILDEW:

Both model's training and Testing accuracy plots (K4 and K5 cross-validation without fine-tuning) are shown in Fig 11. The plots show that the VGG19 model has the same accuracy for both K4 and K5 cross-validation. Also, VGG16 accuracy is increased to 74.15% after K5 cross-validation.

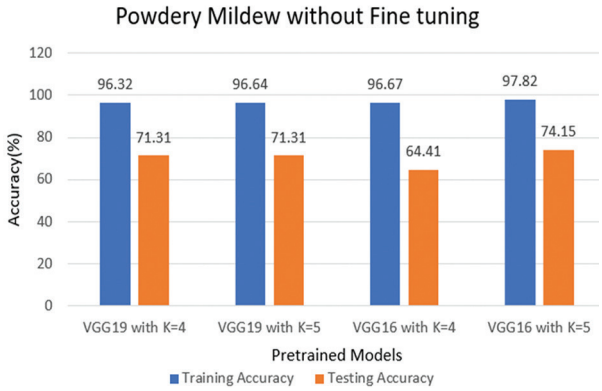


Fig 11. Training and Testing accuracies for Powdery Mildew (without fine tuning)

The confusion matrix of the VGG16 model on the hold-out test set is shown in Table 9. Accuracies of healthy and initial stage accuracies are 67.79% and 67.79%, respectively, and also the accuracy of the last stage is 96.61%, and the middle stage is 100% classified.

Table 9. Confusion Matrix for Powdery Mildew for VGG16 at K5 cross-validation (without fine-tuning)

		Predicted			
		Stage 0	Stage1	Stage2	Stage3
Ground Truth	Stage 0	**40**	0	18	1
	Stage1	0	**40**	19	0
	Stage2	0	0	**59**	0
	Stage3	0	0	2	**57**

Table 10 shows the confusion matrix for the VGG19 model on the hold-out data set. The Accuracy of the healthy stage is 91.52%. The initial stage is classified with an accuracy of 67.79%. The Accuracies for the middle and last stages are 98.30% and 79.66%, respectively.

Table 10. Confusion Matrix for Powdery Mildew for VGG19 at K5 cross-validation (without Fine-tuning)

		Predicted			
		Stage 0	Stage1	Stage2	Stage3
Ground Truth	Stage 0	**54**	0	3	4
	Stage1	5	**40**	15	1
	Stage2	0	0	**58**	3
	Stage3	1	0	13	**47**

VGG19 outperformed disease stage classification for the initial and middle stages with accuracy of 91.52% and 98.30% than VGG16.

From Fig. 12, we can observe that both model's testing accuracy is improved with fine tuning of 2 than without model fine-tuning (Refer Fig. 11). Also, after K5 cross-validation, the accuracy of both models is enhanced.

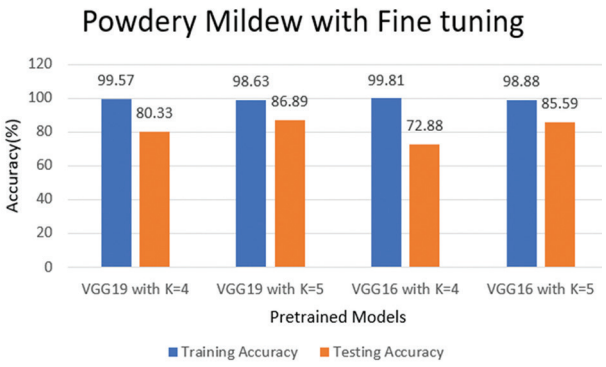


Fig 12. Training and Testing accuracies for Powdery Mildew (with fine tuning)

Table 11 shows the confusion matrix for the VGG16 model on the hold-out data set. The Accuracy of the healthy stage and last stage is 91.52%. The initial and middle stages are classified with an accuracy of 62.71% and 96.61%, respectively.

Table 11. Confusion Matrix for Powdery Mildew for VGG16 at K5 cross-validation (with Fine-tuning)

		Predicted			
		Stage 0	**Stage1**	**Stage2**	**Stage3**
Ground Truth	**Stage 0**	**54**	1	2	2
	Stage1	1	**37**	21	0
	Stage2	1	0	**57**	1
	Stage3	1	0	4	**54**

Table 12 shows the confusion matrix for the VGG19 model on the hold-out data set at K=5. The accuracy of the healthy and initial stages is 89.83% and 81.35%, whereas the middle and last stages are classified as 94.91% and 93.22%, respectively.

Table 12. Confusion Matrix for Powdery Mildew for VGG19 at K4 Cross-validation (with Fine-tuning)

		Predicted			
		Stage 0	**Stage1**	**Stage2**	**Stage3**
Ground Truth	**Stage 0**	**53**	2	2	4
	Stage1	3	**48**	9	1
	Stage2	0	1	**56**	4
	Stage3	0	0	6	**55**

Accuracies obtained by models are statistically witnessed by calculating confidence intervals. The confidence interval tells us how much precise the estimate values are calculated. This work estimates testing accuracy for each K fold with and without an acceptable tuning approach. This calculation shows accuracy is likely to come in which range. Gaussian distribution [25][26] of proportion helps to calculate the interval's radius. As this work is based on multiclass image classification, the radius of the interval [26] is calculated as per equation1 shown below:

$$interval = z * sqrt((accuracy * (1 - accuracy)) / n) \quad (1)$$

Where the interval is the radius of the confidence interval, accuracy is the estimate(testing accuracy of the model is used) that is to be witnessed, z is the number of standard deviations from the Gaussian distribution. z value is used as 1.96 with a 95% significance level for calculating the confidence interval, and n is the total number of samples. In this case, n is 236, which is the size of the out dataset. Table 13 shows the calculated confidence interval values with a range of testing accuracy.

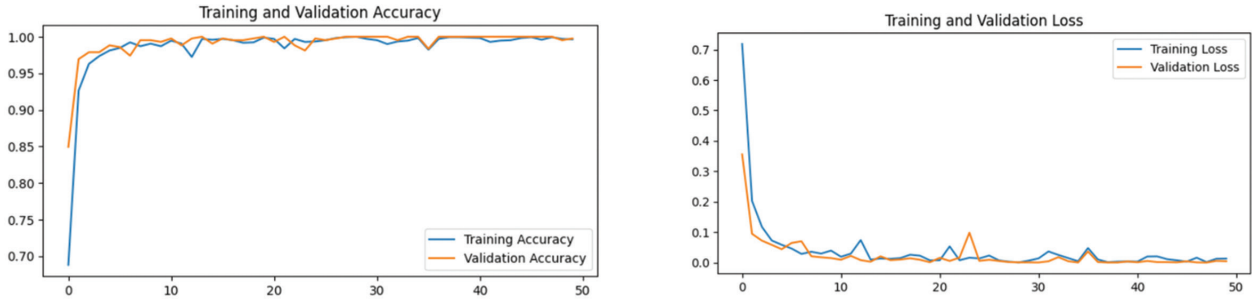
Table 13. Calculation of Confidence Intervals

Model	Kfold	Accuracy (%)	Calculated Confidence Interval at 95% Significance Level	Range
Alternaria Leaf Blight Without Fine Tuning				
VGG16	K=4	77.97	5%	72% to 82%
	K=5	88.98	4%	84% to 92%
VGG19	K=4	83.05	4%	79% to 87%
	K=5	87.29	4%	83% to 91%
Alternaria Leaf Blight with fine-tuning				
VGG16	K=4	86.02	4%	82% to 90%
	K=5	**90.25**	**3%**	**87% to 93%**
VGG19	K=4	88.56	4%	84% to 92%
	K=5	89.41	4%	85% to 93%
Powdery Mildew Without Fine Tuning				
VGG16	K=4	64.41	6%	58% to 70%
	K=5	74.15	5%	69% to 79%
VGG19	K=4	71.31	5%	66% to 76%
	K=5	71.31	5%	66% to 76%
Powdery Mildew with Fine Tuning				
VGG16	K=4	72.88	5%	67% to 77%
	K=5	85.59	4%	81% to 89%
VGG19	K=4	80.33	5%	75% to 85%
	K=5	**86.89**	**4%**	**82% to 90%**

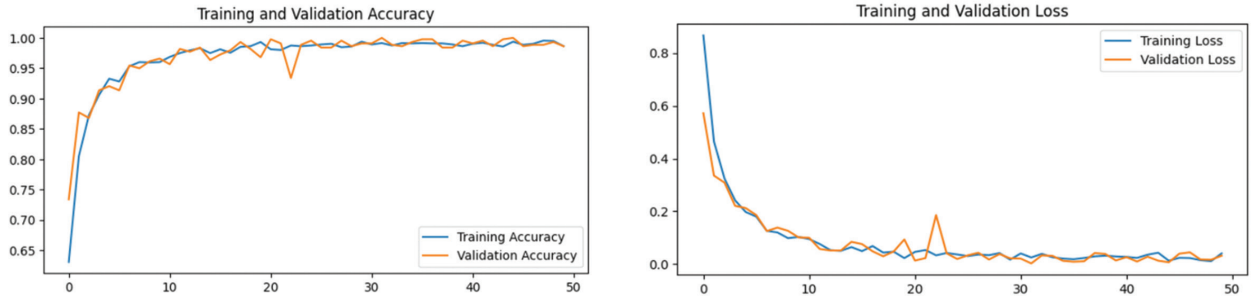
From the above table, a claim can be made that both models with K5 cross-validation by fine-tuning approach give smaller confidence intervals shown with yellow color highlighted form, which indicates the testing accuracy is more precise.

In this work, VGG16 and VGG19 models with K-fold (4 and 5) cross-validation with fine-tuning and without fine-tuning are applied to both diseases. But the confidence interval shows that VGG16 (K5 cross-validation) with finetuning = 2 for Alternaria leaf blight and VGG19(K5 cross-validation) with fine tuning=2 for Powdery Mildew giving smaller confidence interval and hence model accuracy 90.25% and 86.89% is claimed to be suitable for selected sample size respectively. Thus, Fig13(a) shows the accuracy and loss curves for models with precise testing accuracy. Models are trained till epoch 10. On the x-axis, 50 values are shown, which is epoch 10 per iteration, and the y-axis shows accuracy. Loss and accuracy curves are closer to each other. Fig

13(b) has higher training loss for initial iterations, and later, it decreases and flattens at the later iteration. Similarly, for both models' validation loss is also higher at the start of iterations, and later, it decreases. This indicates that the Powdery Mildew VGG19 model with K5 cross-validation, fine-tuned with 2 layers, and the Alternaria Leaf Blight VGG16 model with K5 cross-validation, fine-tuned with 2 layers, are good fit models.



(a)



(b)

Fig. 13. a) Accuracy and Loss curves for VGG16 after K5 cross validation(Alternaria Leaf Blight) b) Accuracy and Loss curves for VGG19 after K5 cross validation(Powdery Mildew)

7. CONCLUSION

This study recommends a deep learning approach with fine tuning for performing disease stage classification of Alternaria leaf blight and Powdery Mildew diseases. This work creates a path for calculating plant disease severity. Both pre-trained models are trained with and without a fine-tuning approach based on training samples. Stratified K-fold validation is applied as samples per class are not uniformly distributed. The results showed that VGG16 and VGG19 both performed well after fine-tuning the parameter set to 2. VGG16 and VGG19 models gave more precise testing accuracy at 90.25% and 86.89% for Alternaria Leaf Blight and Powdery mildew, respectively, with small confidence intervals after K5 cross-validation. Both models have undergone K4 and K5 cross-validation; the result shows that with K5 cross-validation, the confidence interval is more minor. Hence K5 cross-validation is best suited for VGG16 and VGG19 models, demonstrating that the deep learning approach is the most favorable technique for disease stage classification based on plant disease severity estimation.

In future work, more image samples at different stages of both diseases will be collected with hyperspectral imaging with more powerful sensors, improving the model's performance by training with minutiae details of affected leaves. It can give better testing on unseen data. Also, the framework can be proposed to capture field intensity using advanced drone cameras.

8. ACKNOWLEDGEMENT:

This research could not have been completed without the invaluable support of Dr. R.D. Prasad, who serves as the Principal Scientist of Plant Pathology at ICAR-Indian Institute of Oilseeds Research. I extend my sincere appreciation to Dr. Maharudra Ghodke, who previously led the Latur Oilseed Research Centre, as well as to Santosh Waghmare and Sangita Aradwad, who hold the positions of senior and junior plant pathologists, respectively.

9. REFERENCES:

[1] Y. devi Puraikalan, M. Scott, "Sunflower Seeds (Helianthus Annuus) and Health Benefits: A Review", Recent Progress in Nutrition, Vol. 3, No. 3, 2023, pp. 1-5.

[2] R. Shivani, "Diseases of Sunflower: Necrosis, Leaf Blight, Mildew and Other Diseases", https://www.biologydiscussion.com/ (accessed: 2023)

[3] S. Guo, Y. Ge, K. Na Jom, "A review of phytochemistry, metabolite changes, and medicinal uses of the common sunflower seed and sprouts (Helianthus annuus L.)", Chemistry Central Journal, Vol. 11, 2017, p. 95.

[4] M. Alagawany, M. R. Farag, M. E. Abd El-Hack, K. Dhama, "The practical application of sunflower meal in poultry nutrition", Advances in Animal and Veterinary Sciences, Vol. 3, 2015, pp. 634-648.

[5] "India: sunflower seeds production Vol. 2023 | Statista", https://www.statista.com/statistics/769814/india-sunflower-seeds-production-Vol./ (accessed: 2023)

[6] J. Chai, H. Zeng, A. Li, W. T. E. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios", Machine Learning with Applications, Vol. 6, 2021.

[7] M. A. Hajam, T. Arif, A. M. Ud Din Khanday, M. Neshat, "An Effective Ensemble Convolutional Learning Model with Fine-Tuning for Medicinal Plant Leaf Identification", Information, Vol. 14, No. 11, 2023, p. 618.

[8] Z. Al Sahili, M. Awad, "The power of transfer learning in agricultural applications: AgriNet", Frontiers in Plant Science, Vol. 13, 2022.

[9] S. R. Dammavalam et al. "Leaf Image Classification with the Aid of Transfer Learning: A Deep Learning Approach", Current Chinese Computer Science, Vol. 1, 2020.

[10] S. Bhattarai, "New Plant Diseases Dataset (2019)", https://www.kaggle.com/vipoooool/new-plant-diseases-dataset (accessed: 2024)

[11] L. Yang, S. Xu, X.Y. Yu, H. B. Long, H. Zhang, Y. W. Zhu, "A new model based on improved VGG16 for corn weed identification", Frontiers in Plant Science, Vol. 14, 2023, p. 1205151.

[12] P. Lameski, "weed-datasets (2020)", https://gitee.com/Monster7/weed-datase/tree/master/ (accessed: 2024)

[13] P. K. Sethy, N. K. Barpanda, A. K. Rath, S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine", Computers and Electronics in Agriculture, Vol. 175, 2020.

[14] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, Z. A. Kazimi, "Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model", Plants, Vol. 11, No. 17, 2022, p. 2230.

[15] A. Fiqri, "Rice Leafs 5 diseases (2021)", Kaggle, https://www.kaggle.com/datasets/adefiqri12/rice-leafsv3 (accessed: 2024)

[16] G. Wang, Y. Sun, J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning", Computational Intelligence and Neuroscience, Vol. 2017, 2017.

[17] R. Venkatachalam, K. Ilamurugu, "Sunflower: Index: Diseases of Sunflower", Development e-Courses for BSc Agriculture, Tamilnadu Agriculture University, http://www.eagri.org/, (accessed: 2024)

[18] Keras, "ImageDataGenerator | TensorFlow API Version", https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator (accessed: 2023)

[19] V. S. Magomadov, "Deep learning and its role in smart agriculture", Journal of Physics: Conference Series, Vol. 1399, No. 4, 2019.

[20] P. Marcelino, "Transfer learning from pre-trained models | Size-Similarity matrix and decision map for fine-tuning pre-trained models", https://towardsdatascience.com/transfer-learning-from-pre-trained-models (accessed: 2024)

[21] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556, 2014.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 2818–2826.

[23] Keras, "Keras 3 API documentation / Keras Applications", https://keras.io/api/applications/ (accessed: 2023)

[24] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization", Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 2015.

[25] A. Hazra, "Using the confidence interval confidently", Journal of Thoracic Disease, Vol. 9, No. 10, 2017, pp. 4124-4129.

[26] J. Brownlee, "Confidence Intervals for Machine Learning | Interval for classification accuracy", https://machinelearningmastery.com/confidence-intervals-for-machine-learning/ (accessed: 2024)

Optimized Weed Image Classification via Parallel Convolutional Neural Networks Integrating an Excess Green Index Channel

Original Scientific Paper

Seyed Abdollah Vaghefi
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
P144621@siswa.ukm.edu.my

Mohd Marzuki Mustafa
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
marzuki@ukm.edu.my

Mohd Faisal Ibrahim*
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
faisal.ibrahim@ukm.edu.my

Seri Mastura Mustaza
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
seri.mastura@ukm.edu.my

Mohd Hairi Mohd Zaman
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
hairizaman@ukm.edu.my

Mohd Asyraf Zulkifley
Universiti Kebangsaan Malaysia,
Faculty of Engineering and Built Environment,
Selangor, Malaysia
asyraf.zulkifley@ukm.edu.my

*Corresponding author

Abstract *– Weed management is an essential operational task to ensure the excellent health of crops or trees. The emergence of machine vision enables convolutional neural networks (CNNs) to classify weed types automatically, which can subsequently be used for a weed management strategy. A dominant approach to implement CNN-based weed classification is to train a network with RGB images as input either by adopting a transfer learning approach or a custom network. However, such an approach limits the process of incorporating prior knowledge as a significant feature of the network to improve the classification accuracy. This work proposes a novel network based on parallel convolutional neural networks (P-CNN), leveraging the excess green index (ExG) channel as an additional input to the RGB image channels. We argue that using the ExG channel can capture the greenness feature of weeds from the visible light spectrum, an important feature in many vegetation images such as leaves or green plants. The results show that the proposed P-CNN combining ResNet50 and a custom CNN obtains a Top-1 accuracy of 97.2% on a public weed dataset called DeepWeeds compared to the baseline ResNet50 alone with only 95.7%. The results show the significant contribution of domain-specific knowledge of green indexes in improving the classification performance of weed images. This enhancement could transform real-world weed management by enabling highly precise detection by allowing the classifier to focus intensively on differentiating green color features between leaves with nearly identical morphology.*

Keywords: *weed classification; deep learning; convolutional neural network; machine vision*

Received: July 16, 2024; Received in revised form: October 27, 2024; Accepted: November 8, 2024

1. INTRODUCTION

Weed management is a crucial task in agriculture. It is required to minimize the effect of weed growth on crop production [1]. Weeds compete with crops, consuming nutrients, sunlight, and other growth factors. Weed management is also one of the costliest maintenance operations in various plantations. Weeds are classified as unwanted plants that can affect the productivity of vegetation trees. Popular weed management approaches include chemical, biological, mechanical, and cultural controls [2].

Digital image processing of weeds is an essential tool for automatic weed management control and modern

precision agriculture practices. The result from image processing can be used to analyze weed occurrence detection and weed species classification. In chemical-based weed control, the process of spraying herbicides throughout the fields is commonly utilized worldwide [3]. In such a case, weed identification using image processing can be manipulated to determine the types of herbicides that must be sprayed according to the weed species.

Weed image analysis requires special attention due to unique challenges, including a wide range of species types, wide distribution, different leaf shapes and sizes, and various texture features. Different weed growth stages could also make it difficult to detect the species of the weeds [4]. Weed classification also intrinsically faces a challenging problem because of the monotonous green color on the weed's surface.

Research interest in weed image detection and classification has increased significantly in the past few years due to the need for automatic weed management and the advancement of supported digital technologies. Various computer vision methods used in recent works to detect weed from images have been extensively reviewed [5]. There are two main categories: 1) traditional image processing combining feature extraction and conventional machine learning, and 2) deep learning with ample data training.

Conventional machine learning approaches typically require small image samples, short training time, and low computational power requirements. Images are pre-processed to extract and enhance distinct features. Due to such low requirements, an algorithm of conventional machine learning can be easily implemented as an embedded system for real-time image processing and analysis. However, this class of image processing approaches suffers from low accuracy and is prone to misclassification errors due to changes in a natural environment such as ambient light.

On the other hand, deep learning gains its attractiveness in various image processing domains due to the algorithms' capability to provide an end-to-end detection paradigm and to achieve highly significant accuracy for real-world applications. Another advantage of using deep learning over conventional machine learning is the automatic feature extraction mechanism that can be learned through backpropagation. This capability comes with the requirements of having large image samples, longer training time, and high computational power, specifically graphic processing unit (GPU).

However, training a deep learning model can be challenging when dealing with a small sample size. A model trained with small datasets exposes the problem of underfitting and overfitting due to bias and variance in the dataset. One of the well-accepted solutions to the small dataset problem is to use the transfer learning approach. The transfer learning approach is a method that takes some or all parts of a pre-trained network that has been trained on large datasets of other images and re-trained some parts of the network with a desired and typically small dataset. Although the approach works, it tends to miss extracting essential features that could be domain-specific main features.

Combining the method used by conventional machine learning and deep learning can demonstrate a potentially innovative approach to leverage the strength of both machine learning categories [6]. Specifically, combining two methods for feature extraction, namely handcrafted features and learned features, could yield benefits, including improved accuracy, reduced training time, and enhanced robustness. Furthermore, it allows the designers to control the known feature to be utilized as one of the inputs by the classifier to make decisions. In the case of weed classification, the green color feature can play a significant role in differentiating classes of weed types. This is also applied to most vegetation-related problems such as leaf type, plant disease type, or crop growth stage. Enhancing this feature manually can ensure that the classifier block is considering the processed input to make decisions while allowing other unclear features to be automatically learned by deep networks.

This work proposes a parallel convolutional neural network (P-CNN) incorporating excess green information in the network. The proposed network adopts transfer learning deep CNN combined with another customized CNN network featuring handcrafted excess green feature input to improve weed classification accuracy. An analysis of the performance of the proposed methods is presented based on a pre-trained CNN network, namely ResNet50.

The contribution of this work is two-fold. First, the proposed P-CNN classifier suggests a method to merge conventional machine learning and deep learning mechanisms by utilizing handcrafted feature extraction for a known vital feature and transfer learning to exploit a trained network. Second, the work explores the benefit of the excess green feature in improving the classification task for weed images in particular and the potential to be further applied to various greenish vegetation images.

The article is organized as follows: The next section presents related works by other researchers. Then, the proposed approach taken in this work to perform weed classification is discussed in detail. The results and discussion section presents the experimental results and performance of the P-CNN classifier. Finally, the conclusion section summarizes the findings.

2. RELATED WORKS

This section investigates the relevant literature on leveraging deep learning-based weed classification techniques with various feature extraction techniques. Several studies have shown that deep learning yields superior results to traditional machine learning [7]. Weed image classification with conventional machine

learning yields low and inconsistent accuracy [8-9]. For supervised deep learning, generating datasets is both labor-intensive and time-consuming. Reducing the workload involved in data acquisition and annotation presents a significant challenge in deep learning research [10]. Therefore, various works have explored ways to optimize the performance of deep learning-based classifiers. There are three main categories for optimizing deep learning-based weed classifiers: 1) transfer learning approach, 2) modification of neural network structure, and 3) addition of feature vectors.

The transfer learning approach is one of the most common approaches among deep learning designers. The approach is used when the available dataset is considerably small, depending on the problem at hand. Transfer learning empowers the reusable of some parameters from an existing model trained with other domains to the desired classification domain. This approach can be seen in various works involving weed classification. For example, a semantic segmentation based on SegNet is proposed in [11] to differentiate images of rice seedlings, backgrounds, and weeds. Transfer learning of a pre-trained VGG16 network as the encoder of SegNet was applied to save the training time, while a decoder and a softmax classifier were retrained with 224 images of rice seedlings and weeds.

In another work by [12], transfer learning played a crucial role, enabling the extensive training of 24 deep learning models for Saffron crops and weeds classification. Leveraging a dataset of 291 images depicting standard weed classes around Saffron crops and selection of Xception as the final model, the study highlighted the superior classification by making the last 20 layers in the middle flow and exit flow of Xception trainable. The transfer learning approach is also applied to some other weed classification works, as in [13, 14]. The current limitation of the transfer learning approach for weed classification is that the base model, including feature extraction layers, is made non-trainable. No new feature vectors are extracted, limiting the model's ability to adapt to new data and capture unique characteristics of different weed classes. Consequently, the performance may suffer, mainly when dealing with domain dissimilarity.

Another way to enhance classifier performance is by modifying neural network structures. This approach entails replacing, adding, or removing certain layers within the network, thus optimizing the structure for improved results. A work by [15] performed a study on real-time categorization of weed severity, employing 275 images of five prevalent weeds near lettuce crops. The work utilized a multimodal YOLOV7-L model, attaining a 97.5% mAP@0.5. The approach incorporated a simplified model and a novel ELAN-B3 feature extraction layer, facilitating real-time processing in 4 to 13 milliseconds. The viability of such an approach was primarily dependent upon augmented photos to enhance the sample size.

A two-stage encoder–decoder architecture is investigated by [16] for pixel-level classification and differentiation between crops and weeds, utilizing 1,920 images from tobacco and sesame datasets with RGB channels. The W-shaped CNN attained 90% and 94% accuracy on the tobacco and sesame datasets, respectively, surpassing the performance of UNet and SegNet semantic classifiers. Nonetheless, the network's extensive parameters necessitate substantial training resources, and it has not been sufficiently evaluated on crops exhibiting colors akin to weeds.

A study in [17] introduces a graph-based deep learning framework named Graph Weeds Net (GWN). The classifier, utilizing recurrent neural networks (RNN), notably ResNet50 and DenseNet202, was trained to discriminate patterns in graph vertices that represent image sub-patches formed from various scales, ranging from local to global contexts. On another hand, a work by [18] created a lightweight real-time weed classifier for embedded systems, employing 40,000 photos obtained from UAVs. The preprocessing included bounding box filtering and color-indexed segmentation, utilizing ResNet18 to attain 94% accuracy. The model size was refined from 32-bit to 16-bit, facilitating real-time detection at 2.2 frames per second. Nonetheless, the performance deteriorated subsequent to resizing. Overall, the primary disadvantage of structural modification is the heuristic method employed to substitute appropriate layers within the network, rendering the procedure a trial-and-error strategy.

Last but not least, the incorporation of feature vectors constitutes the third strategy for improving the performance of deep learning. This method achieves the greatest degree of flexibility since it enables the preprocessing and enhancement of input images prior to their introduction into the deep neural network. In [19], text-based descriptors were employed to classify 4,232 images from the TomatoWeeds dataset. The study utilized text-based descriptors as input for ResNet50, encompassing additional features of image-to-text projection, morphological characteristics, and habitat descriptions. Transfer learning was implemented. Nevertheless, the outcome is suboptimal due to the constraints of a limited and unbalanced dataset. Another work by [20] integrated grey-level characteristics with RGB features and presented the hybridized whale and sea lion algorithm as an optimizer for CNNs. Employing a crop/weed field dataset for weed detection in soybean cultivation, this work attained 92% accuracy. However, due to the dual-phase data preprocessing, the method necessitated substantial CPU resources and was deficient in real-time analytical capabilities.

In [21], the work employed multispectral image decomposition and feature vector methodologies utilizing Wavelet and CapsNet on 2,000 images from the Madurai LISS IV dataset, encompassing five weed classifications. Their methodology utilizing multispectral sub-bands and a Deep Denoising Auto-Encoder

(DDAE) achieved an accuracy of 96.75%, surpassing traditional CNNs such as AlexNet, VGG, ResNet, and Inception. Despite its excellent accuracy, the intricate data preprocessing and feature vector production presented obstacles for real-time application.

Another work by [22] concentrated on classifying corn crops, narrow-leaf weeds, and broadleaf weeds by connected component analysis (CCA) to extract regions of interest. Utilizing 15,000 cornfield images captured under natural conditions, the work implemented VGG16, VGG19, and Xception models, attaining an accuracy of 97%. These CNN models surpassed SVM utilizing LBP feature extraction, although no real-time detection processing was documented.

The integration of deep learning for vegetable detection with color index-based segmentation was proposed by [23] to extract weed features across 12 maize, sunflower, and potato classes. Employing the novel CentreNet model and color index-based segmentation, the work attained a 95.3% F1 score. The effort enhanced the color index equation using a genetic algorithm but encountered sluggish sequential processing in the recognition of vegetables and weeds.

The performance of VGG16, ResNet50 and Xception models is compared in [14] on the classification of 12 weed types in maize, sunflower, and potatoes farms. The work introduced a semi-automatic approach for weed labeling utilizing the Excess Green–Red Index threshold, applied to 93,000 images across 12 weed categories. Utilizing VGG16, ResNet50, and Xception, this work attained 98% accuracy through transfer learning with two fully connected top layers. Nevertheless, the methodology necessitated an extensive dataset for optimal training.

Table 1 compares and summarizes all works related to machine learning algorithms in weed classification applications. The analysis highlighted in this section emphasizes the limitations of existing research methodologies. This article concatenates all three optimization strategies to enhance weed classification performance by integrating transfer learning, a modified neural network topology, and the incorporation of feature vectors.

3. PROPOSED METHOD

The proposed method for weed image classification follows a workflow as depicted in Fig.1. The workflow can be divided into three stages. The first stage explains the processes taken for data acquisition and processing steps. The second stage describes model training steps in which the structure of the proposed parallel CNN network will be explained in detail. Finally, model testing steps are performed to analyze and validate the performance of the trained classifier. Note that k-fold cross-validation is applied in this work to avoid any bias. Thus, the model training and testing steps are repeated a few times to get the overall model performance.

3.1. DATA ACQUISITION AND PROCESSING PHASE

This work uses a weed image dataset from a publicly available source called DeepWeeds [24]. The dataset contains 17,509 images of weed species commonly found across northern Australia. The dataset provides weed images from eight locations in a natural rangeland environment. The rangeland environment presents unique challenges for classifying weeds under uneven terrains, complex backgrounds, and difficulty in differentiating weeds from native plants.

There are nine weed classes identified within the dataset namely 1) Chinee apple (*Ziziphus mauritiana*), 2) Lantana (*Lantana camara*), 3) Parkinsonia (*Parkinsonia aculeata*), 4) Parthenium (*Parthenium hysterophorus*), 5) Prickly acacia (*Vachellia nilotica*), 6) Rubber vine (*Cryptostegia grandiflora*), 7) Siam weed (*Chromolaena odorata*), 8) Snake weed (*Stachytarpheta spp.*), and 9) Negative (indicates non-weed class). Fig. 2 shows a few samples of weed images in the DeepWeeds dataset.

The size of the images fed into the proposed classification model is 224 x 224 pixels, so each image I_{RGB} is the size of $R^{224x224}$. Each weed class contains at least 1000 images. Meanwhile, the negative class concatenates all images with no weed, accumulating around 8690 images. The resampling procedure based on the k-fold cross-validation technique is used to evaluate the trained CNN model. Expressly, number of folds, k=5 is set. The dataset is split into training, validation, and testing sets with a ratio of 60:20:20.

3.2. MODEL TRAINING PHASE

The overall process of model training steps is discussed further in this section. This phase starts with the design of the proposed parallel CNNs model, including the input selection, learning paradigm, and classifier layer configuration. Then, the architecture of all main networks used in the proposed P-CNN model is presented, including pre-trained CNNs and a custom CNN. Thirdly, the generation of excess green images using the excess green feature extractor as one input type to the model is explained.

The Model Design

In the proposed P-CNN model, the classifier receives two inputs, namely RGB image I_{RGB} and its corresponding excess green image I_{ExG}. The I_{RGB} size is $224 \times 224 \times 3$ indicating 224 pixels height (h_{rgb}), 224 pixels weight (w_{rgb}) and three-color channels (d_{rgb}). IExG has a size of $224 \times 224 \times 1$ implying 224 pixels for both height and weight (h_{exg} and w_{exg}) and expands only one gray-color channel (d_{exg}) as the second input to the classifier. Both inputs are fed into two different convolutional network blocks of the proposed parallel CNN classifier. These blocks act as an automated feature extractor that learns important image features the classifier block requires. As illustrated, these blocks are organized paral-

lel to each other so that both blocks can be processed simultaneously.

A pre-trained CNN block gets I_{RGB} input. Transfer learning extracts I_{RGB} features using well-trained network information. All network weights are untrainable. ResNet50 [25] is chosen in this study by maintaining all layers except the top layers for classifier block. ResNet50 was chosen for its deep layers and unique residual convolutional layers, which achieve one of the highest classification accuracies in diverse applications. DeepWeeds reference dataset uses ResNet50 as the basis and best model. Consequently, this study aims to demonstrate how the extra green feature enhances performance without the need for new parameters while utilizing the same model.

Authors	Contribution	Datasets	Input Parameters	Classifiers	Results	Advantages	Disadvantages
Hu et al. (2024)	Real-time deep learning classifier for weed severity classification	275 images of 5 common weeds around lettuce crops	RGB image	Multimodule YOLOV7-L	97.5% mAP@0.5	Lightweight model and novel ELAN-B3 feature extraction module. Real-time processing 4-13ms	Small datasets and highly depends on augmented images
Makarian et al. (2024)	Deep learning for Saffron crops and weeds classification	291 images of common weed classes around Saffron crops	RGB image	Xception (the best model)	100% F1-score	Evaluate 24 deep learning models. Apply transfer learning	Manual hyperparameter tuning
Belissent et al. (2024)	Deep learning model leveraging text-based descriptors for tomato weeds classification	4,232 images of 4 weed classes in TomatoWeeds dataset	Text embeds with image-to-text projection, morphological and habitat descriptions	ResNet50	77.8% accuracy.	Embed text-based descriptors. Deploy transfer learning. Zero-shot learning for unseen weed classes	Limitation of performance due to a small, unbalanced dataset
Moldvai et al. (2024)	Conventional feature-based classifiers of vegetation weeds	3,000 images from public dataset of corn, lettuce and radish weeds.	Weed area, hull area, and solidity	SVM, RF, KNN, ANN, NB, GBM	59% to 94% accuracy in various classifier	Extraction of various features such as shape descriptors and color histograms	Small dataset for verification
Martins et al. (2024)	Feature-based classifiers of broadleaf weeds in narrowleaf crops	126 points for pasture area and 89 points for sorghum area.	Soil, terrain conditions, color and spatial information	Random Forest	84% (pasture) and 74% (sorghum) accuracy	Geo-referenced map for groundtruth. Exploit terrain and soil variables as parameters	No parameter correlation analysis
Moazzam et al. (2023)	Deep learning model for tobacco and sesame crop weeds	1,920 images of tobacco dataset and sesame dataset	RGB image	W-shaped CNN	90% - 94% accuracy	Two stage encoder–decoder structures for pixel-level classification	Large network with large parameters to be trained
Panda et al. (2023)	Deep learning model for soybean crop weeds classification	Crop/weed field image dataset and weed detection in soybean crops	GLCM, GLRM and RGB features	Customized CNN with HW–SLA optimizer	92% accuracy	Incorporate RGB and grey-level features. Introduce hybridized HW–SLA algorithm as CNN optimizer	Two-stage data pre-processing increases time and computational power
Rajakani & Kavitha (2023)	Deep learning model with multispectral image decomposition	2,000 images from Madurai LISS IV with 5 weed classes.	Multispectral sub-bands	Deep Denoising Auto-Encoder (DDAE)	96.75% accuracy	Multispectral image decomposition and feature vector using Wavelet and CapsNet	Complex data pre-processing and feature vector generation
Garibaldi-Márquez et al. (2022)	Deep learning classifier for narrow-leaf weeds, and broadleaf weeds.	15,000 cornfield images	Texture features	VGG16, VGG19 and Xception	97% accuracy	Deploy connected component analysis (CCA) for region of interest extraction	No real-time detection processing
Hu et al. (2021)	Multi-scale detection via graph vertices	DeepWeeds	RGB Image	Graph Weeds Net (GWN)	98.1% accuracy	Semi-supervised learning approach	Complex parameters network
de-Camargo et al. (2021)	Deep learning model for real-time weed classifier	40,000 images from UAV view	Bounding box filtering and color indexed segmentation	ResNet18	94% accuracy	Model size reduction from 32-bit to 16-bit. Real-time detection with 2.2 frames per second.	Acceptable result degradation after resizing
Jin et al. (2021)	Deep learning and color index segmentation classifier	12 classes of maize, sunflower, and potato weeds	RGB Image	CentreNet and Color index-based segmentation	95.3% F1 score	Optimized color index equation with Genetic algorithm	Slow sequential process of vegetable and weed detection
Peteinatos et al. (2020)	Deep learning classifier with semi-automatic image labeling	93,000 images of 12 weed classes	RGB Image	VGG16, ResNet50 and Xception	98% accuracy	Semi-automatic method for weed labeling using Excess Green–Red Index threshold	Large dataset required

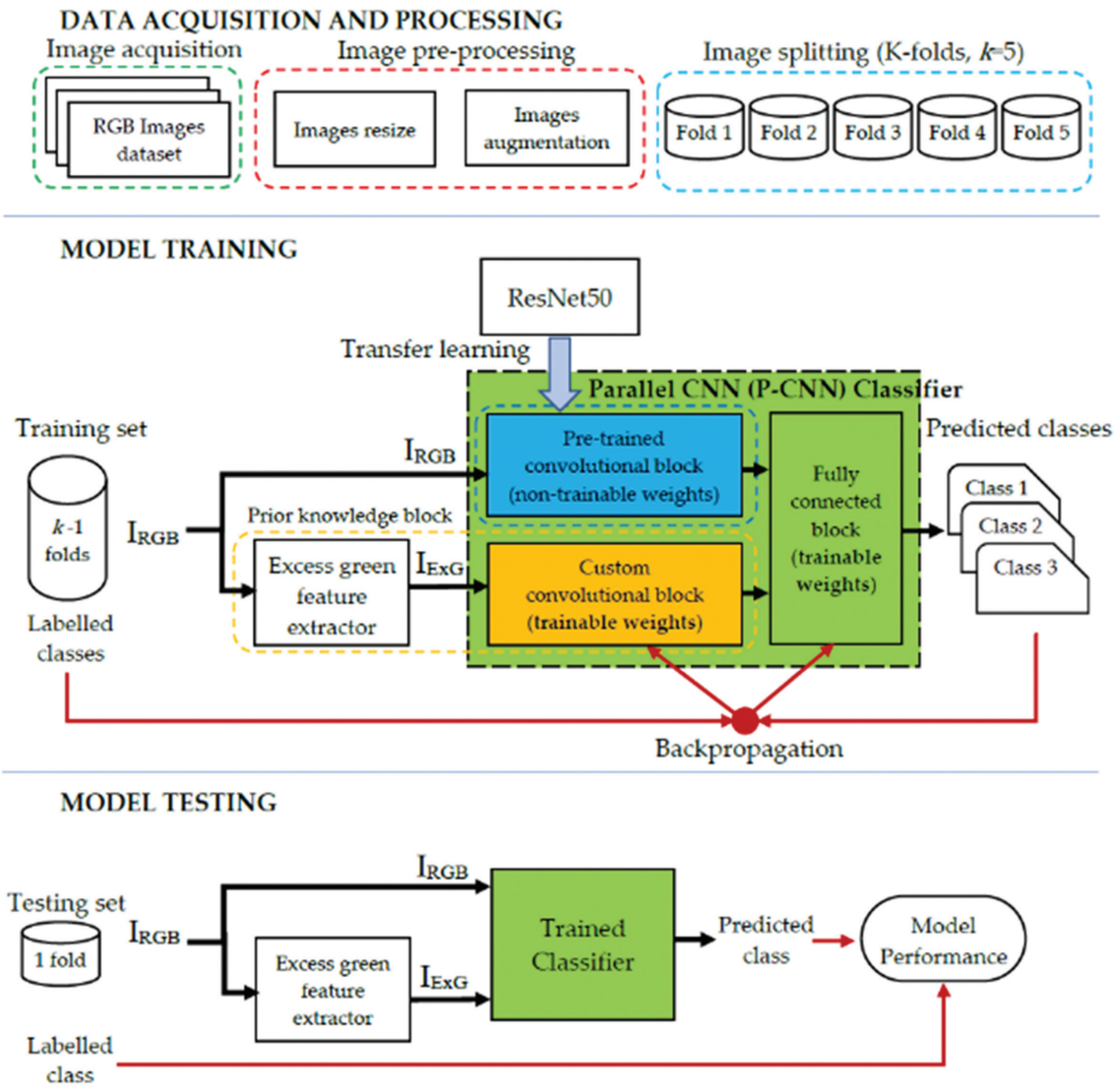


Fig. 1. Overall workflow diagram for the proposed weed classification methodology

Fig. 2. Samples of RGB images of different weed classes in DeepWeeds dataset

On the other hand, an IExG input is processed via a custom trainable convolutional network. The intuition behind this is that very little pre-trained CNN model is available for a grayscale or a one-channel input image. In addition, this block is the one that is responsible for integrating prior domain-specific knowledge that may vary in the form of a range of image formats, sizes, or depths. Thus, any chosen convolutional network must be trained to find the best-configured weights and biases with the acquired labeled data. In this work, an excess green image generator is used to generate IExG from its corresponding IRGB before running this convolutional block. Hereafter, the proposed custom network for one channel IExG images is called ExGNet.

Next, outputs from both CNN blocks are combined and fed into a fully connected layer block that functions as the classifier layer. This block is constructed with dense layers with trainable weights and biases to form input, hidden, and output nodes. The number of nodes for the input layer is equal to the combination of output size from both the pre-trained network and the trainable network blocks. The output layer is designed

to have the nodes equivalent to the number of weed classes in the database. Each node represents a weed class that is activated based on an activation function. In this work, the softmax activation function caters to multi-class classification by providing a probability value for each class. The softmax activation function can be calculated using equation (1),

$$softmax(y_i) = \frac{exp^{y_i}}{\Sigma_{j=1}^{N} y_j} \quad (1)$$

where y_i is the value of the output node of class i and N is the total number of classes.

The Network Architecture

The proposed model architecture is depicted in Fig. 3. This figure and Table 2 represent the proposed P-CNN network combining the canonical ResNet50 and the ExGNet for the DeepWeeds dataset classification.

ResNet50 is a residual learning framework to overcome the problem of accuracy degradation of deeper network layers. In many deeper networks, training errors are supposed to converge.

However, it is common to observe that the learning process runs the other way around causing the accuracy to become saturated and drops rapidly. ResNet50 incorporates residual functions to solve this degradation problem. A shortcut connection is added to the feedforward neural networks. The method has eased the training process of deeper layers to achieve better accuracy substantially.

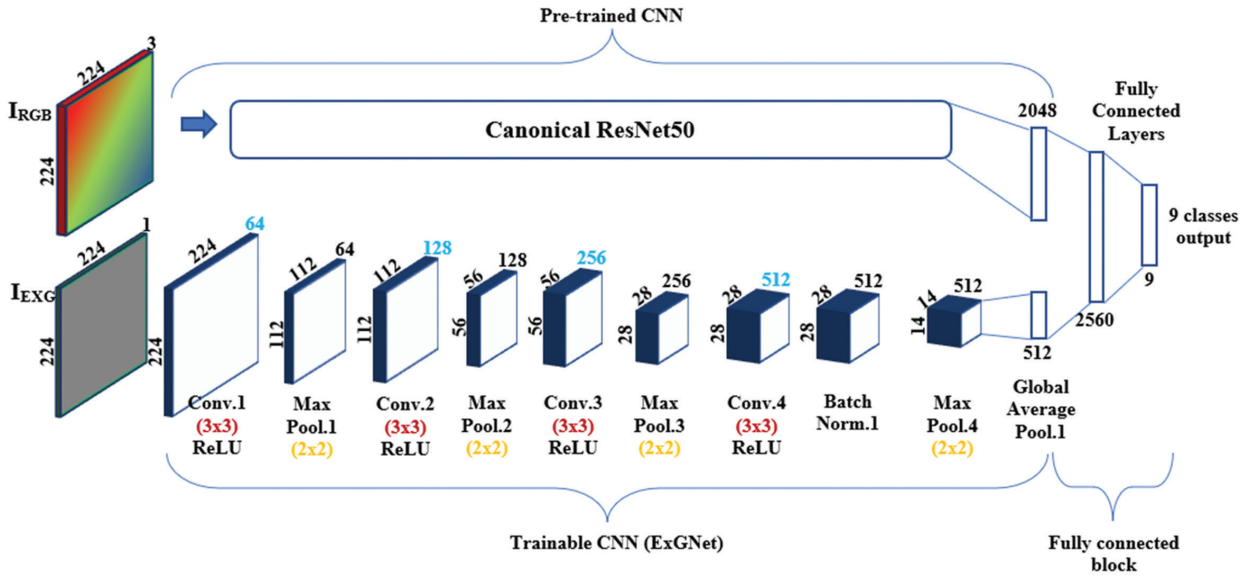


Fig. 3. A parallel CNN (P-CNN) model combining the ResNet50 and the ExGNet for the DeepWeeds dataset

Table 2. Model structure of ExGNet for DeepWeeds dataset

Layer	Filter	Kernel size	Pool size	Stride	Padding	Activation function
Conv.1	64	3x3	-	1	Same	ReLU
Max Pool.1	-	-	2x2	None	Valid	-
Conv.2	128	3x3	-	1	Same	ReLU
Max Pool.2	-	-	2x2	None	Valid	-
Conv.3	256	3x3	-	1	Same	ReLU
Max Pool.3	-	-	2x2	None	Valid	-
Conv.4	512	3x3	-	1	Same	ReLU
Batch Norm.1	-	-		-	-	-
Max Pool.4	-	-	2x2	-	Valid	-
Global Avg. Pool.1	-	-	-	-	-	-

The main idea of the proposed architecture is to fully utilize available resources via integrating transfer learning, prior domain-specific knowledge, and a limited labeled dataset.

The parameters of the pre-trained ResNet50 blocks are preserved in the architecture via a transfer learning approach. The powerful transfer learning approach makes those networks highly reusable for RGB image applications. The networks have been extensively trained using thousands of RGB images, such as from the ImageNet dataset, and thus, are capable of extracting low-level image features like lines and edges and high-level image features, such as object shapes, as the network layers go deeper.

For the trainable ExGNet, the network is built from a sequence of convolutional layers, max-pooling layers, a batch normalization layer, and a global average pooling layer. The first convolutional layer (Conv.1) uses 64 kernels (d_{conv} = 64) to produce a feature map of size 224 × 224 × 64. Each kernel has the size of 3 x 3 (k_{conv} = 3 × 3). Standard convolution with scalar multiplication operation and stride one and same padding is used in this work. Such standard convolution has the computational cost as in (2).

$$Cost_{conv} = h_i \text{ x } w_i \text{ x } d_i \text{ x } d_{conv} \text{ x } k_{conv} \quad (2)$$

where h_i, w_i and di are the height, width and depth of an input, respectively.

A Rectified Linear Unit (ReLU) is adopted as the activation function after the convolutional layer to introduce a non-linearity function to the network. A pooling layer (Max Pool.1) is added after the Conv.1 layer. The pooling layer is introduced to down-sample the feature maps, reducing the number of parameters to be learned. Max pooling type is performed such that the maximum element of any feature map region covered by a filter with 2 x 2 pool size (k_{pool} = 2 × 2) is selected. The same convolutional and max pooling layers block are repeated three times (Conv.2, Max Pool.2, Conv.3, Max Pool.3) for the network to learn more complex features. Another convolutional layer (Conv.4) is added, followed by a batch normalization layer.

The loss function is used as the guide for the back-propagation algorithm to fine-tune all trainable parameters. The loss function calculates prediction errors by comparing the models and labeled outputs. In this work, categorical cross entropy, $Loss_{catx}$, is used as the loss function as in (3),

$$Loss_{catx} = \sum_{j=1}^{N} \hat{y}_j \cdot \log(y_j) \quad (3)$$

where $\hat{y}_j$ is the target value of class j. Here, one-hot encoded labeled output is established from the available dataset for all outputs. Table 2 shows the configurations of hyperparameters used in this work.

Table 2. Hyperparameters setting

Hyperparameter	Filter
Optimizer	Adam V2
Learning rate	0.0001
Epoch	100

Excess Green Image Generator

An excess green image I_{ExG} can be generated from the excess green feature extractor block in Fig. 1. This block acts as the medium to extract greenness index information from an RGB image IRGB. Greenness identification is vital for many vegetation and crop identification by focusing on the green color spectrum and reducing the effect of red and blue color spectra.

Various visible spectral-index methods are available, such as the excess green index, the vegetation index, the excess green minus excess red index, and the green leaf index. However, this work selects the excess green index due to its capability to distinguish green plants with its background effectively and outperforms other greenness indices in terms of greenness identification performance [26].

The excess green index can be calculated for each pixel of an RGB image using equation (4),

$$ExG=2g-r-b \quad (4)$$

where, g, r and b are the chromatic green, red and blue colors defined by equation (5).

$$g = \frac{G}{(R^* + G^* + B^*)} \quad r = \frac{R}{(R^* + G^* + B^*)} \quad b = \frac{B}{(R^* + G^* + B^*)} \quad (5)$$

where G, R and B are the pixel values of an RGB image, while G^*, R^* and B^* are the maximum pixel values of an RGB image.

3.3. MODEL TESTING PHASE

Every trained model's performance can be validated with data outside the training dataset. This work chooses the cross-validation approach rather than the single 'training-testing' split approach to avoid bias and reduce variance. Each dataset is equally divided into several folds, k, with the amount of data in every fold almost close to each other. k=5 is used in this work, which means the proposed classifier model was trained 5 times, with each training using k-1 or four folds as the training dataset, alternating each remaining fold as the testing dataset once. The model performance is measured based on all performance indices' mean and standard deviation.

The main performance index is accuracy. A model's accuracy can be calculated with equation (6). Accuracy gives the overall rate of correct predictions over all tested cases.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

where TP, TN, FP and FN are true positive, true negative, false positive and false negative cases, respectively, acquired from a confusion matrix. For further statistical analysis, three more performance indices are calculated based on precision, recall and F1-score indices. Equations of (7), (8) and (9) show the calculation for all three indices, respectively.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN} \quad (9)$$

Precision measures the percentage of correct positive predictions from all positive predictions. Recall gives a percentage of correct positive predictions over all positive cases. F1-score considers the trade-off between precision and recall.

4. RESULTS AND DISCUSSION

The results obtained from both the training and testing phases of the proposed P-CNN classifier model are presented in this section. The hardware utilized for this experiment comprises an AMD Ryzen 7 4800HS processor, 16GB of RAM, and a Nvidia GeForce GTX 1660 Ti graphics card.

The proposed parallel CNN model was implemented on the Deep-Weeds dataset, with nine weed classes and 17,509 images. In this experiment, the ResNet50 network in the pre-trained convolutional block was used to support the complexity of the classification problem. Transfer learning was made from the network trained in [24], with the last two layers acting as the classification function removed. Then, the custom convolutional block, ExGNet, and the fully connected block, as described in section 3, were combined with the ResNet50 network.

Fig. 4 shows the learning process of the proposed model across all five cross-validated training and validation folds of the DeepWeeds dataset. The acceleration stage occurred for the first 5 epochs, the optimization stage between epochs 6 and 49, and the short plateau stage from epoch 50 onward. All training sessions stopped early before reaching the maximum epoch as no further improvement can be seen in the validation accuracy. The training time was around 91 minutes. In comparison, it took 13 hours to train a single ResNet50 model without transfer learning in [25].

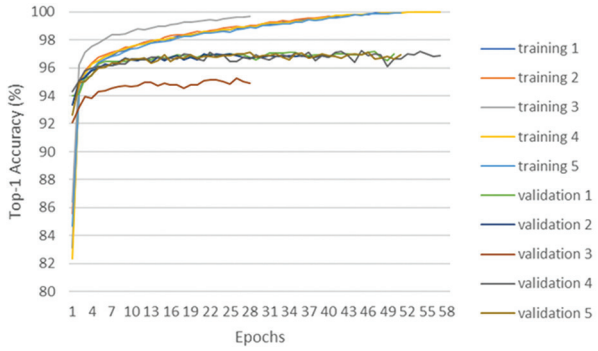


Fig. 4. Graph of learning performance in terms of training dataset accuracies (training 1,2,3,4,5) and validation dataset accuracies (validation 1,2,3,4,5) for all 5 cross validated folds of the DeepWeed dataset

Moving to the testing phase, Table 3 contains the confusion matrix of the average prediction results expressed as percentages across all five cross-validated testing folds for nine weed classes. Negatives and Parkinsonia classes have the two highest prediction accuracy, with 98.6% and 98.3%, respectively. In contrast, Chinee apple and Snake weed classes show the two lowest prediction accuracies with 91.6% and 93.1%, respectively.

This is due to a relatively large misclassification between these two classes compared to other classes. This confusion is contributed by certain lighting conditions that make the leaf material of both weed classes look intensely similar. However, the misclassification errors of the Chinee apple image as Snake weed image at 2.4%, and 3.1% vice-versa, are lower than the errors produced by the original ResNet50 model. The proposed P-CNN can increase the accuracy of the Chinee apple class from 88.5% in [25] to 91.6%. The same goes for the Snake weed class, which has improved from 88.8% to 93.1%. Overall weighted accuracy for all classes of the proposed model is 97.2%.

Table 3. Confusion matrix between actual and predicted weed classes of DeepWeeds dataset for all 5 cross validated testing folds. The weighted class accuracy is expressed as percentages.

		Predicted								
		Chinee Apple	Lantana	Parkinson	Parthenium	Prickly Acacia	Rubber Vine	Siam Weed	Snake Weed	Negatives
Actual	Chinee Apple	**91.6**	0.5	0.0	0.8	0.0	0.2	0.1	2.4	4.4
	Lantana	0.6	**96.7**	0.0	0.1	0.0	0.2	0.0	0.5	2.0
	Parkinson	0.0	0.0	**98.3**	0.2	0.7	0.0	0.0	0.0	0.9
	Parthenium	0.1	0.0	0.1	**97.1**	0.9	0.1	0.0	0.0	1.8
	Prickly Acacia	0.1	0.0	0.6	0.9	**95.8**	0.0	0.0	0.0	2.6
	Rubber Vine	0.5	0.2	0.0	0.2	0.1	**95.7**	0.0	0.4	2.9
	Siam Weed	0.0	0.0	0.0	0.0	0.0	0.0	**97.7**	0.0	2.3
	Snake Weed	3.1	0.6	0.0	0.2	0.0	0.0	0.0	**93.1**	3.0
	Negatives	0.3	0.2	0.1	0.1	0.3	0.1	0.2	0.2	**98.5**

This is due to a relatively large misclassification between these two classes compared to other classes. This confusion is contributed by certain lighting conditions that make the leaf material of both weed classes look intensely similar. However, the misclassification errors of the Chinee apple image as Snake weed image at 2.4%, and 3.1% vice-versa, are lower than the errors produced by the original ResNet50 model. The proposed P-CNN can increase the accuracy of the Chinee apple class from 88.5% in [25] to 91.6%. The same goes for the Snake weed class, which has improved from 88.8% to 93.1%. Overall weighted accuracy for all classes of the proposed model is 97.2%.

Statistical analysis with precision, recall, and F1-score performance indices was conducted and tabulated in Table 4. The highest precision achieved is 98.8% by the Parkinsonia class, while Chinee Apple shows the lowest precision with 93.9%. 7 out of 9 classes have precision above 95%. The highest and the lowest recall is 98.5% and 91.6% recorded by Negatives and Chinee Apple classes, respectively. Again, all classes have recall above 95% except for the Chinee Apple and Snake Weed classes. Combining precision and recall, the highest F1-score is attained by the Parkinsonia class with 98.5%. In contrast, the Chinee apple class carries the least performed class with an F1-score of 92.7%.

Finally, the performance improvement of the proposed parallel CNN (P-CNN) model was observed against the

performance reported by the ResNet50 model in [12]. Table 5 compares accuracy, precision, and false positive rate (FPR) indices for all classes in the DeepWeeds dataset between both models. P-CNN achieved better accuracy than ResNet50 in all weed classes. Chinee Apple and Snake Weed classes show the highest improvement, at 4.3% and 3.1%, respectively.

Table 4. The average precision, recall and F1-score for all 5 cross validated testing folds of the DeepWeeds dataset. All values are expressed as percentages

Class	Precision	Recall	F1-score
Chinee Apple	93.9	91.6	92.7
Lantana	97.0	96.7	96.8
Parkinson	98.8	98.3	98.5
Parthenium	96.6	97.1	96.8
Prickly Acacia	95.6	95.8	95.7
Rubber Vine	98.5	95.7	97.1
Siam Weed	97.9	97.7	97.8
Snake Weed	94.7	93.1	93.9
Negatives	97.7	98.5	98.1

Table 5. Comparison of accuracy, precision and false positive rate (FPR) performance between the proposed model (DP-CNN) and the conventional ResNet50 model (ResNet50). The bolded texts indicate the improvement of 1% and more

	Accuracy		Precision		FPR	
Class	**DP-CNN**	**ResNet50**	**DP-CNN**	**ResNet50**	**DP-CNN**	**ResNet50**
Chinee Apple	91.6	88.5	93.9	91.0	0.42	0.61
Lantana	96.7	95.0	97.0	91.7	0.19	0.55
Parkinsonia	98.3	97.2	98.8	97.9	0.07	0.13
Parthenium	97.1	95.8	96.6	96.7	0.21	0.21
Prickly Acacia	95.8	95.5	95.6	93.0	0.29	0.46
Rubber Vine	95.7	92.5	98.5	99.1	0.09	0.05
Siam Weed	97.7	96.5	97.9	97.2	0.13	0.18
Snake Weed	93.1	88.8	94.7	90.9	0.32	0.55
Negatives	98.5	97.6	97.7	96.7	2.50	3.59
Average	**97.2**	95.7	**97.2**	95.7	**1.40**	2.04

For the precision and FPR indices, P-CNN outperformed ResNet50 in all classes except the Parthenium and Rubber Vine classes. The precision and FPR results of the Parthenium class are on par with those of both models. Meanwhile, the Rubber vine class has a very minimum performance reduction of 0.6% and 0.04% for precision and FPR, respectively. Interestingly, the weighted average FPR of P-CNN has significant error improvement, where it recorded only a 1.40% error rate compared to Res-Net50 with a 2.04% error rate. This, in turn, could be beneficial for weed control and management. For example, smaller FPR can save the cost of herbicide application by minimizing cases with herbicide and weed type mismatches. Overall, the results show the dominance of the proposed P-CNN model over the ResNet50 model.

The capability of weed classification models to accurately identify weed types is essential for several reasons. Robust identification of weed classes can assist in managing effective weed control strategies, especially in determining the correct type and amount of chemical sprayer, mechanical weed removal, or other weed management techniques. Weed image classification can also aid in analyzing invasive weed types that displace native crops and disrupt ecological balance via preventive maintenance actions.

The applicability of the excess green index (ExG) to the weed image classification can be seen from the performance improvement of various classification indices. The results show that the features extracted from ExG are important in vegetation classification, where greenness information plays a vital role in distinguishing patterns of different weed types.

The total time required to compute the excess green index and execute a prediction using parallel CNN is around 200ms. The size of the P-CNN model is approximately 210MB. In contrast, a solitary ResNet50 required approximately 180ms when utilizing the ordinary TensorFlow package. The ResNet50 model's size is 283.6MB, encompassing its original fully connected layers. The data indicates that the incorporation of ExG-Net has minimal effect on the time and sizing performance of the classifier.

The suggested method utilizing ExG index extraction has effectively distinguished various weeds exhibiting similar greenness patterns; however, it is limited in enhancing other parameters, such as lighting circumstances, which are less correlated with green color. Exploration of a parallel network utilizing other established feature vectors that have a high correlation with a desired factor is feasible. Moreover, the ExGNet architecture is subject to additional optimization. This is justifiable when the dimensions of the structure and the execution duration must be minimized for certain applications, such as embedded systems.

In terms of the proposed model's applicability, future research should concentrate on integrating the proposed model into an embedded system for in-situ industrial applications. For example, the proposed model can be employed for an automated herbicide sprayer to eliminate weeds. The selection of the suitable herbicide can occur in real-time with accurate classification of weed types. This method provides significant economic benefits by minimizing herbicide usage, resulting in cost reductions for farmers. Furthermore, it reduces the environmental impact of pesticides, fostering sustainable agriculture methods. By precisely targeting weeds, it also aids in maintaining crop health and productivity, so further aiding the agricultural sector.

5. CONCLUSION

The proposed parallel convolutional neural network (P-CNN) has surpassed a state-of-the-art network in reducing the classification error of weed types. A public dataset of weed images has been utilized to assess the P-CNN. The P-CNN achieved an average accuracy of 97.2% on the DeepWeeds dataset, compared to the standard ResNet50 model's accuracy of 95.7%. The total error rate varies between 1.5% and 8.4%. The P-CNN surpasses ResNet50 across all nine categories. The experimental results indicate that using green excess index information can substantially enhance classification accuracy while preserving the requirement for rapid computer processing. The suggested network demonstrates significant progress towards reaching a near-zero error rate in weed classification, warranting further investigation to attain a substantial and acceptable degree of accuracy. The proposed model can be implemented into an embedded system for in-situ industrial applications in future research. A weed-killing automatic herbicide sprayer can use the model to perform herbicide selection in real time.

6. ACKNOWLEDGEMENT

The authors would like to thank Universiti Kebangsaan Malaysia for the financial support under the grant GUP-2023-071, TAP-K016268, and the Department of Electrical, Electronic and Systems Engineering UKM for the technical supports.

7. REFERENCES

[1] M. Dilipkumar, T. S. Chuah, S. S. Goh, I. Sahid. "Weed management issues, challenges, and opportunities in Malaysia", Crop Protection, Vol. 134, 2020.

[2] K. Ruzlan, M. Hamdani, "Integrated weed management programs at oil palm plantation - a survey", International Journal of Agriculture, Forestry and Plantation, Vol. 11, 2021, pp. 32-38.

[3] D. Loddo, J. S. McElroy, V. Giannini, "Problems and perspectives in weed management", Italian Journal of Agronomy, Vol. 16, No. 4, 2021.

[4] A. M. Hasan, F. Sohel, D. Diepeveen, H. Laga, M. G. Jones, "A survey of deep learning techniques for weed detection from images", Computers and Electronics in Agriculture, Vol. 184, 2021.

[5] Z. Wu, Y. Chen, B. Zhao, X. Kang, Y. Ding, "Review of weed detection methods based on computer vision", Sensors, Vol. 21, No. 11, 2021.

[6] P. Castro, G. Fortuna, P. Silva, A. G. C. Bianchi, G. Moreira, E. Luz, "Merging Traditional Feature Extraction and Deep Learning for Enhanced Hop Variety Classification: A Comparative Study Using the UFOP-HVD Dataset", Proceedings of the 12th Brazilian Conference, Belo Horizonte, Brazil, 25-29 September 2023.

[7] J. Jiao, Y. Zang, C. Chen, "Key Technologies of Intelligent Weeding for Vegetables: A Review", Agriculture, Vol. 14, No. 8, 2024.

[8] L. Moldvai, P. Á. Mesterházi, G. Teschner, A. Nyéki, "Weed Detection and Classification with Computer Vision Using a Limited Image Dataset", Applied Sciences, Vol. 14, No. 11, 2024.

[9] C. L. Martins, A. L. G. Oliveira, I. A. da-Cunha, H. Oldoni, J. C. Pereira, L. R. do-Amaral, "Classification of the Occurrence of Broadleaf Weeds in Narrow-Leaf Crops", Engenharia Agricola, Vol. 44, 2024.

[10] W. Hu, S.O. Wane, J. Zhu, D. Li, Q. Zhang, X. Bie, Y. Lan, "Review of deep learning-based weed identification in crop fields", International Journal of Agricultural and Biological Engineering, Vol. 16, No. 4, 2023.

[11] X. Ma, X. Deng, L. Qi, Y. Jiang, H. Li, Y. Wang, X. Xing, "Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields", PLOS ONE, Vol. 14, No. 4, 2019.

[12] H. Makarian, S. I. Saedi, "Automated classification of saffron and broadleaf weeds of Flixweed and Hoary Cress using deep learning and color images", Crop Protection, Vol. 183, 2024.

[13] Y. Xu, Y. Zhai, B. Zhao, Y. Jiao, S. Kong, Y. Zhou, Z. Gao, "Weed recognition for depthwise separable network based on transfer learning", Intelligent Automation & Soft Computing, Vol. 27, No. 3, 2021, pp. 669-682.

[14] G. G. Peteinatos, P. Reichel, J. Karouta, D. Ujar, R. Gerhards, "Weed identification in maize, sunflower, and potatoes with the aid of convolutional neural networks", Remote Sensing, Vol. 12, No. 24, 2020.

[15] R. Hu, W.H. Su, J. L. Li, Y. Peng, "Real-time lettuce-weed localization and weed severity classification based on lightweight YOLO convolutional neural networks for intelligent intra-row weed control", Computers and Electronics in Agriculture, Vol. 226, 2024.

[16] S. I. Moazzam, T. Nawaz, W. S. Qureshi, U. S. Khan, M.I. Tiwana, "A W-shaped convolutional network for robust crop and weed classification in agriculture", Precision Agriculture, Vol. 24, No. 5, 2023, pp. 2002-2018.

[17] K. Hu, G. Coleman, S. Zeng, Z. Wang, M. Walsh, "Graph weeds net: A graph-based deep learning method for weed recognition", Computers and Electronics in Agriculture, Vol. 174, 2021.

[18] T. de-Camargo, M. Schirrmann, N. Landwehr, K.H. Dammer, M. Pflanz, "Optimized deep learning model as a basis for fast UAV mapping of weed species in winter wheat crops", Remote Sensing, Vol. 13 No. 9, 2021.

[19] N. Belissent, J. M. Peña, G. A. Mesías-Ruiz, J. Shawe-Taylor, M. Pérez-Ortiz, "Transfer and zero-shot learning for scalable weed detection and classification in UAV images", Knowledge-Based Systems, Vol. 292, 2024.

[20] B. Panda, M. K. Mishra, B. S. P. Mishra, A. K. Tiwari, "Optimized Convolutional Neural Network for Robust Crop/Weed Classification", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 37, No. 4, 2023.

[21] M. Rajakani, R. J. Kavitha, "Invasive weed optimization with deep transfer learning for multispectral image classification model", Multimedia Tools and Applications, Vol. 83, No. 15, 2024, pp. 45519-45534.

[22] F. Garibaldi-Márquez, G. Flores, D. A. Mercado-Ravell, A. Ramírez-Pedraza, L. M. Valentín-Coronado, "Weed Classification from Natural Corn Field-Multi-Plant Images Based on Shallow and Deep Learning", Sensors, Vol. 22, No. 8, 2022.

[23] X. Jin, J. Che, Y. Chen, "Weed identification using deep learning and image processing in vegetable plantation", IEEE Access, Vol. 9, 2021, pp. 10940-10950.

[24] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, "Deepweeds: A multiclass weed species image dataset for deep learning", Scientific Reports, Vol. 9, No. 1, 2019.

[25] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 770-778.

[26] A. R. Larrinaga, L. Brotons, "Greenness indices from a low-cost UAV imagery as tools for monitoring post-fire forest recovery", Drones, Vol. 3, No. 1, 2019.

A Deep Learning Framework with Optimizations for Facial Expression and Emotion Recognition from Videos

Original Scientific Paper

Ranjit Kumar Nukathati*
Department of Computer Science and Engineering, JNTUA,
Anantapur, Andhra Pradesh, India
e-mail: ranjitnukathati@gmail.com

Uday Bhaskar Nagella
Government College (A),
Anantapur, Andhra Pradesh, India
e-mail: udaynagella@gmail.com

AP Siva Kumar
Department of Computer Science and Engineering, JNTUA,
Anantapur, Andhra Pradesh, India
e-mail: sivakumar.ap@gmail.com

*Corresponding author

Abstract *– Human emotion recognition has many real-time applications in healthcare and psychology domains. Due to the widespread usage of smartphones, large volumes of video content are being produced. A video can have both audio and video frames in the form of images. With the advancements in Artificial Intelligence (AI), there has been significant improvement in the development of computer vision applications.Accuracy in recognizing human emotions from given audio-visual content is a very challenging problem. However, with the improvements in deep learning techniques,analyzing audio-visual content towards emotion recognition is possible. The existing deep learning methods focused on audio content or video frames for emotion recognition. An integrated approach consisting of audio and video frames in a single framework is needed to leverage efficiency. This paper proposes a deep learning framework with specific optimizations for facial expression and emotion recognition from videos. We proposed an algorithm, Learning Human Emotion Recognition (LbHER), which exploits hybrid deep learning models that could process audio and video frames toward emotion recognition. Our empirical study with a benchmark dataset, IEMOCAP, has revealed that the proposed framework and the underlying algorithm could leverage state-of-the-art human emotion recognition. Our experimental results showed that the proposed algorithm outperformed many existing models with the highest average accuracy of 94.66%. Our framework can be integrated into existing computer vision applications to recognize emotions from videos automatically.*

Keywords: *Emotion Recognition, Spatial Expression Analysis, Deep Learning, Artificial Intelligence, Hyperparameter Tuning*

Received: June 11, 2024; Received in revised form: August 23, 2024; Accepted: August 30, 2024

1. INTRODUCTION

Human emotion recognition is the skill of interpersonal relationships. Therefore, it has a vital role in day-to-day communications. Humanscan intuitively understand communication through text, audio, and facial expressions. The ability to recognize emotions depends on the level of perception. With the advancements in Artificial Intelligence (AI), there has been an increased number of computer vision applications used to solve problems in the real world. With the help of deep learning techniques, solutions are being provided for various issues. Automatic recognition of human emotions is one of the challenging problems researchers have considered. However, recognizing emotions accurately is a problematic phenomenon. Many researchers contributed to building deep learning models meant for emotion recognition.

Profiled sentiment analysis is made possible by profound learning advancements. The state of the art in AI facial expression recognition is reviewed in this study

[1]. Provided a wealth of user-generated material for studying emotions, it is made more accessiblefor recognizing emotions. A 72% accurate approach based on facial expressions is suggested for automated video subtitle annotation [2]. LLEC is a model used for emotion cognition using entropy and similarity models on unlabeled data. Experiments show that enhanced LLEC significantly increases emotion recognition accuracy [3]. Deep reinforcement learning and algorithm optimization are among the tasks that lie ahead. Emotion identification is enhanced by this end-to-end method without the need for human feature engineering. Upcoming research will refine deep learning models for EEG-based emotion identification and enhance cross-subject categorization [4]. It was observed from the literature that most of the existing works considered textual content or audio or video. There is a need for an integrated approach that considers audio and video information processing to efficiently recognize human emotions. The contributions in this paper are as follows.

1. We proposed a deep learning framework with specific optimizations for facial expression and emotion recognition from videos.
2. We proposed a Learning-based Human Emotion Recognition (LbHER) algorithm that exploits hybrid deep learning models that could process audio and video frames towards emotion recognition.
3. Our empirical study, which used a prototype and the IEMOCAP benchmark dataset, has revealed the significance of our hybrid deep learning methodology.

The remainder of the paper is structured as follows—section 2 reviews prior worksabout human emotion recognition using deep learning models. Section 3 presents our methodology for efficiently detecting human emotions using hybrid deep learning models. Section 4 presents the results of our empirical study. Section 5 discusses the research findings in this paper and provides the study's limitations. Section 6 concludes our research work, besides giving directions for future research.

2. RELATED WORK

Human emotion recognition is an important research area that has attracted many researchers across the globe. Zhang *et al.* [1] profiled sentiment analysis is made possible by profound learning advancements. The state of the art in AI facial expression recognition is reviewed in this study. Villegas-Ch *et al.* [2] provided a wealth of user-generated material for the study of emotions. A 72% accurate approach based on facial expressions is suggested for automated video subtitle annotation. Casado *et al.* [3] presented LLEC for emotion cognition using entropy and similarity models on unlabeled data. Experiments show that enhanced LLEC dramatically increases the accuracy of emotion recognition. Deep reinforcement learning and algorithm optimization are among the tasks that lie ahead. Hassouneh *et al.* [4] used deep CNN for EEG emotional feature learning; the proposed technique outperforms conventional classifiers on the DEAP dataset by 3.58%. Emotion identification is enhanced by this end-to-end method without the need for human feature engineering. Upcoming research will refine deep learning models for EEG-based emotion identification and enhance cross-subject categorization. Pise *et al.* [5] recognized emotions thanks to recent developments in information fusion and machine learning, especially when using EEG signals for accurate emotion detection. Building higher-dimensional emotion models and enhancing the techniques for classifying emotion-related datasets are examples of future studies. Patel *et al.* [6] evaluated student gestures to identify emotions and provide instantaneous feedback to enhance instruction. However, identifying nuanced emotions and dealing with skewed data present hurdles for AI. Bazgir *et al.* [7] harmed by depression. It's critical to discover early. A new technique that shows promise for depression screening uses face recordings to extract physiological information. Anbarjafari *et al.* [8] used facial landmarks and EEG data; research focuses on real-time emotion identification for physically disabled people and children with autism. Diamantini *et al.* [9] addressed the absence of online learning. The accuracy of the suggested deep learning model for e-learning's face emotion identification is enormous.

Revina *et al.* [10] used EEG data, an emotion detection system was created, broken down into frequency bands, and then categorized using SVM with 91.3% accuracy— Cimtay *et al.* [11] with compound emotions like happily-disgusted, affective computing improvements in emotion recognition. The 50-category iCV-MEFED dataset aids research. Kumar *et al.* [12] examined the phases, functionality, databases, and applications of FER approaches. Social communication relies heavily on Face Expression Recognition (FER).

Zulfiqar *et al.* [13], with 81.2% accuracy on LUMED-2 and 91.5% accuracy on DEAP datasets, multimodal emotion recognition incorporates facial expressions, EEG, and GSR. Topic and Russo [14], automated face identification has become increasingly important with the growth of picture databases. The methods, difficulties, and applications are covered in this overview. Chen *et al.* [15], with uses like biometric authentication and video monitoring, say that facial recognition is becoming increasingly important. The recognition accuracy of a CNN-based system is 98.76%. Song *et al.* [16] Because of noise, interpreting EEG signals for emotions is complex. Deep learning, HOLO-FM, and TOPO-FM improve the recognition of datasets. The approach could help the realms of medicine and authentication. Future objectives are to enhance cross-validation and add more features.

Khan *et al.* [17] presented a method for recognizing emotions called the Multi-Modal Physiological Emotion Database (MPED). A new A-LSTM technique en-

hances emotion identification feature extraction. The database is open to the public for use in research. Chen *et al.* [18] presented face recognition smart glasses that help with security by providing a 98% detection rate to identify offenders using Haar-like characteristics. With immense accuracy, it uses Convolutional Neural Networks (CNN) for facial recognition. Parui *et al.* [19] proposed an approach that combines linear reconstruction with FRI theory to represent pictures as piecewise smooth functions for single-image super-resolution. It is superior to current techniques. Zheng *et al.* [20] suggested that the Emotion Meter achieves 85.11% accuracy in multimodal emotion identification by integrating EEG and ocular movements. EEG is best at cheerful, fearful eye movements.

Qing *et al.* [21] presented a machine-learning approach to interpretable emotion identification from EEG data. Emotional activation curves utilizing entropy and correlation coefficients are suggested to improve emotion identification accuracy. Gandhi *et al.* [22] depend heavily on sentiment analysis (SA) with both AI and NLP. Sentiment identification in text and videos is improved by Multimodal Sentiment Analysis (MSA), which is investigated in eleven fusion categories employing machine learning and deep learning. Sarkar and Etemad [23], by acquiring representations through pretext tasks, self-supervised deep multi-task learning improves ECG-based emotion identification and achieves state-of-the-art performance across datasets. Ayata *et al.* [24] presented a framework for music selection that enhances the functionality of current systems by utilizing wearable sensors to identify user moods. He *et al.* [25] presented a novel NIR-VIS facial image generation method that streamlines the procedure and raises the accuracy of HFR.

Feng *et al.* [26] explained an automated technique that uses EDA signals to categorize children's emotions. The dataset includes one hundred children's recordings with annotations for acceptance, boredom, and joy. Comparatively speaking, time-frequency analysis with CMorlet wavelets enhances SVM classifier performance. Upcoming projects will focus on real-time processing and growing datasets to capture more diverse emotional patterns. Tolosana *et al.* [27] discussed the rise of lifelike fake content and looked at DeepFakes, datasets, facial modification methods, and benchmarks. Real-world detection poses obstacles, which has led to research on generalization and fusion methods. Hazarika *et al.* [28] suggested employing pre-trained dialogue models to facilitate transfer learning for emotion identification in discussions. Experiments demonstrate enhanced robustness and performance—Baltrusaitis*et al.* [29] integrated data from several senses, known as multimodal machine learning. In classifying the field's obstacles, this poll highlights the promise of co-learning. Davison *et al.* [30] presented a 3D HOG technique for micro-expression detection verified on the SAMM and CASME II datasets. It focuses on 26 FACS-based areas and performs better than current techniques. Future efforts will focus on enhancing sensitivity and quickness. It was observed from the literature that most of the existing works considered textual content or audio or video. There is a need for an integrated approach that considers audio and video information processing to recognize human emotions efficiently.

3. PROPOSED FRAMEWORK

The section presents the proposed methodology, including a deep learning-based framework, preprocessing approaches, proposed algorithm, and evaluation methodology.

3.1. PROBLEM DEFINITION

If any given test video is provided, developing a deep learning-based framework that exploits audio and video frames with a hybrid deep learning approach toward automatic recognition of human emotions is a challenging problem.

3.2. OUR FRAMEWORK

As shown in Fig. 1, we proposed a deep learning-based framework for human emotion recognition. The framework is based on supervised learning, exploiting hybrid learning models for emotion recognition from given video content. The given data set is subjected to pre-processing, which includes a specific methodology, as illustrated in Section 3.3 and Section 3.4. After completion of pre-processing, the dataset is divided into training and test sets of 80% and 20%, respectively. The hybrid deep learning model proposed in this paper is trained with the training data. The model is persisted for future reuse and incremental learning or transfer learning. The saved model is loaded, and test samples will be subjected to emotion recognition. The experimental results are then compared with the ground truth to evaluate the proposed framework.

The proposed framework is based on a hybrid deep learning approach considering multiple modalities while processing the video content. As illustrated in Fig. 2, the proposed framework exploits audio content and video frames from the given input video. From the audio content, an audio spectrogram is generated. The video frames and the audio spectrogram are used to train the hybrid deep learning model. The pre-trained hybrid deep learning model performs emotion recognition from a given test video. Since it is a supervised learning process, it includes training and testing phases. In the training phase, the hybrid deep learning model is trained with 80% of training data to gain the required knowledge. Any given test video is subjected to an emotion recognition process in the testing phase by considering both audio content and video frames. Eventually, the proposed framework can classify emotions into happy, sad, angry, and neutral.

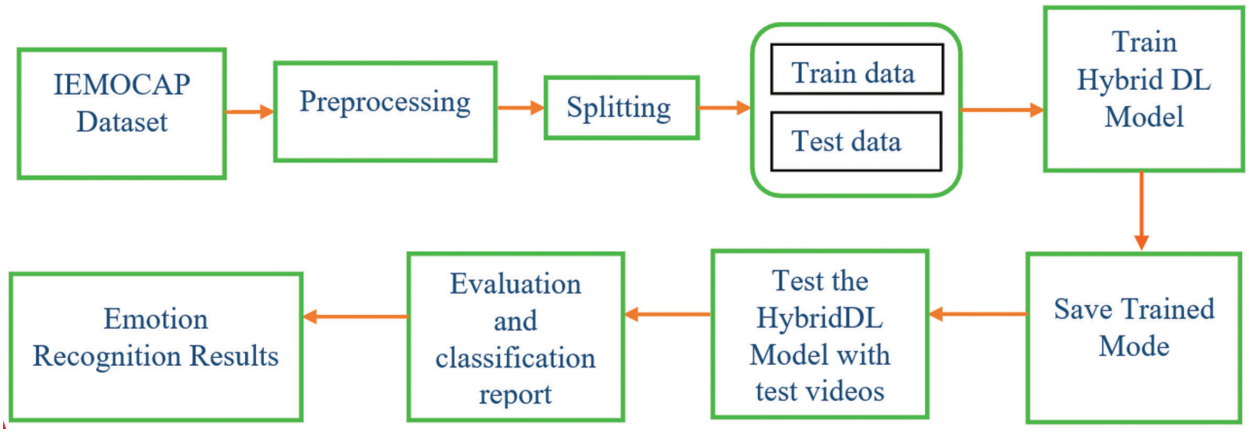


Fig. 1. The proposed deep learning-based framework

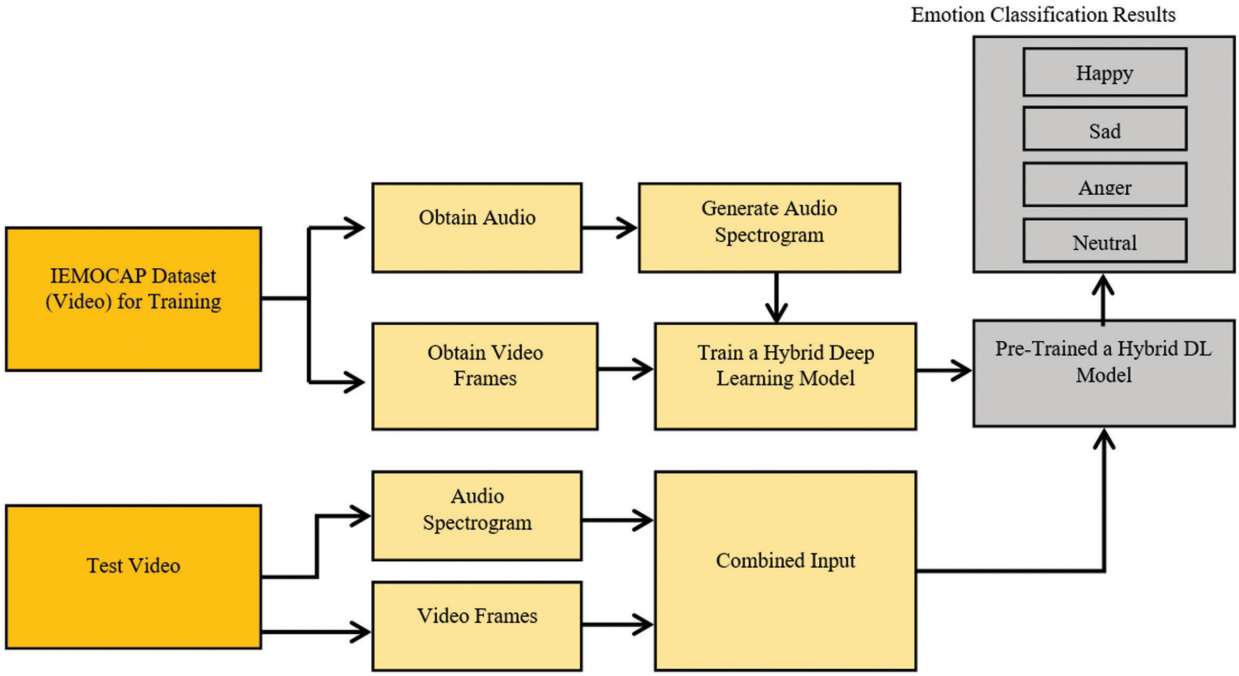


Fig. 2. Illustrates the training and testing phases involved in the proposed framework

A benchmark dataset, IEMOCAP, contains labeled data meant for supervised learning. The dataset is divided into a labeled training set and an unlabeled test set. The labeled data is used to train the hybrid deep learning model, while the unlabeled data is used to test the proposed hybrid deep learning model's performance.

3.3. PREPROCESSING AUDIO DATA

The IEMOCAP data corpus comprises audio wave files of varying durations, each labeled with the honest emotion for the relevant time segment. IEMOCAP generates its audio waves at a 22 KHz sample rate. Using the librosa1 Python library and a 44KHz sample rate, the audio spectrogram is recovered from the WAV file. 44 KHz sample rate was chosen because, according to the Nyquist-Shannon sampling theorem, the sampling frequency must be at least twice the signal frequency to recover the signal correctly. 20Hz to 20KHz is the frequency range of the audio signal. Therefore, the most widely utilized sampling rate is 44 KHz. The spectrograms were created in two parts: the initial duration of the speech or emotion and divided into three-second halves. Data segmentation was also carried out both with and without noise removal.

We used a bandpass filter ranging from 1Hz to 30KHz to eliminate the ambient noise. Work in [31] is also followed by denoising, or noise cleaning, of the input audio stream for data augmentation. To ensure consistency in noise level and frequency relative to other signal sections, noise is injected into sentence utterances lasting less than three seconds. The original audio signal was distorted when noise was introduced with a signal-to-noise ratio (SNR) of 1 throughout the signal duration. Zero padding was also experimented with to have a 3-second time scale. After that, the signal is denoised. We then denoise the resultant signal. To improve prediction accuracy for each emotion, denoising

enhances the visibility of the input audio signal's frequency, time scale, and amplitude components. Audio spectrograms are constructed using the same color bar intensity scale (+/- 60dB) to preserve the spectrum analysis's consistency that spans various emotional states. Normalization of data is comparable to this. The signal with the accurate information remains with a high power intensity or signal amplitude after denoising. The power level of some places in the spectrogram is lower than that of the actual signal of interest. In contrast, some signal strength is seen across the time scale, which, throughout the time scale, a signal intensity, that is, noise, is seen. The resulting spectrogram pictures have a pixel size of 200x300.

The cheerful emotion count is noticeably low, as can be seen. So, to get the final figure of 1600, we repeated the joyful data. The emotion count for fury was likewise replicated. The number of data points for the sad and neutral emotions was lowered to 1600 for each. The model is trained using a total of 6400 photos. Equilibrium data is essential for practical model training. To validate the model, 400 pictures representing each emotion are employed. The training set never contains the photos used for validation. Before realizing that including axis and scale may harm prediction accuracy, we began using audio spectrograms with xy axis and colorbar scale. Rotation and cropping of the input audio spectrograms were done to see an improvement in class accuracy. Every picture was reduced to 200x300 pixels and cropped by 10 pixels at the top. This cropping is done to mimic a little shift in emotion frequency. Likewise, a +/-10-degree rotation was applied to every picture. This rotation modifies the time scale in addition to simulating frequency changes. The rotation was done to a minor degree of 10 degrees because augmenting data that affects the temporal scale is not preferable. After cropping and rotation, 19200 total data points are included in the training set. Using both the original and data-augmented pictures for comparison, the model was trained independently. Images were not horizontally flipped since doing so would cause the timeframe to be flipped and simulate someone speaking backward, reducing the accuracy of the model's predictions.

Executing separate model training on an audio spectrogram with a complete duration of not only three seconds was necessary. The entire time length spectrogram was substituted for the provided 3-second audio spectrogram, preserving the data needed for balance. Approximately one hundred audio spectrograms were visually analyzed. The highest frequency found in all of these spectrograms was found to be around 8 KHz. This indicates that about 60% of the spectrogram picture is blue and contains no emotional information. Every supplied audio spectrogram was scaled to 200 by 300 pixels after being 60% chopped from the top. If the frequency range is known beforehand, creating spectrograms with a defined frequency scale would be the best action.

3.4. PREPROCESSING VIDEO DATA

We also performed video data pre-processing because part of our study involves creating a video model to evaluate where forecast accuracy of emotion identification might be improved. We first divided each video file into sentences using the same method as the audio files to handle the video data. This ensured that the video file we searched matched the specified audio spectrogram. Next, from each video avi file, we retrieved 20 pictures every 3 seconds, which matched the 3-second audio spectrum. Since the film has two performers, the frames were cropped to the appropriate left or right to only include the actor whose emotion was being captured. After that, we further cropped the video frames to hide the actor's head and face. The video frames have a final resolution of 60 x 100. One drawback of the dataset is that because the performers are not speaking directly to the camera in the film, it is impossible to see their entire facial expressions when it comes to a particular emotion. It was discovered that the computer was using more than 12GB of RAM to extract audio spectrograms and video frames. Computer crashes resulted from this. Each audio and video file was analyzed separately in batches to retrieve the data.

3.5. PROPOSED HYBRID DEEP LEARNING METHOD

We proposed a hybrid deep learning model comprising CNN+RNNto process audio content and enhanced 3D CNN to process video frames. Our cross-entropy loss, expressed in Eq. 1,trains the model.

$$L_{\text{cross entropy}} = \frac{1}{N}\sum_{n=1}^{N} -\log\left(\frac{exp(x_c^n)}{\sum_j exp(x_j)}\right) \quad (1)$$

N denotes the total amount of data in the dataset, x_c^n the accurate class score of the n-th data point, and, x_j the class score of the jth input data points out of the n-th data. Because the cross entropy loss will only be minimal when the true class's score for a given data point is noticeably higher than the scores of all other classes, minimizing the loss will compel our model to learn the emotion-related features from the audio spectrogram.

As seen in Fig. 3, our hybrid model is a two-stream network made up of two sub-networks, which was inspired by the work of [32]. We have selected to employ the best-performing audio model we have ever built, CNN + RNN since the first sub-network is the audio model. As the CNN+RNN model illustrates, it dumps the original output layer to get high-level properties of audio spectrograms. The second sub-network is the video model, which consists of two fully connected layers, three 3D max-pooling layers, and four 3D convolutional layers (3DCNN). Ultimately, the last layer of both sub-networks is joined together, succeeded by one displaying the output layer. Semi-supervised and supervised training are the two approaches we

use to train this model. We first pre-train our model for the semi-supervised training strategy using video frames and audio spectrograms from the same and distinct videos. As a result, the model is compelled to discover how a video's visual and aural components correlate. Three different kinds of inputs are used in the pre-training process: positive, where the audio spectrogram and video frames are from the same video; complicated negative, where the audio spectrogram and video frames are from different videos with different emotions; and super hard harmful, where the audio spectrogram and video frames are from other videos with the same emotion. Contrastive loss, as expressed in Eq. 2, is the loss function we employ in pre-training.

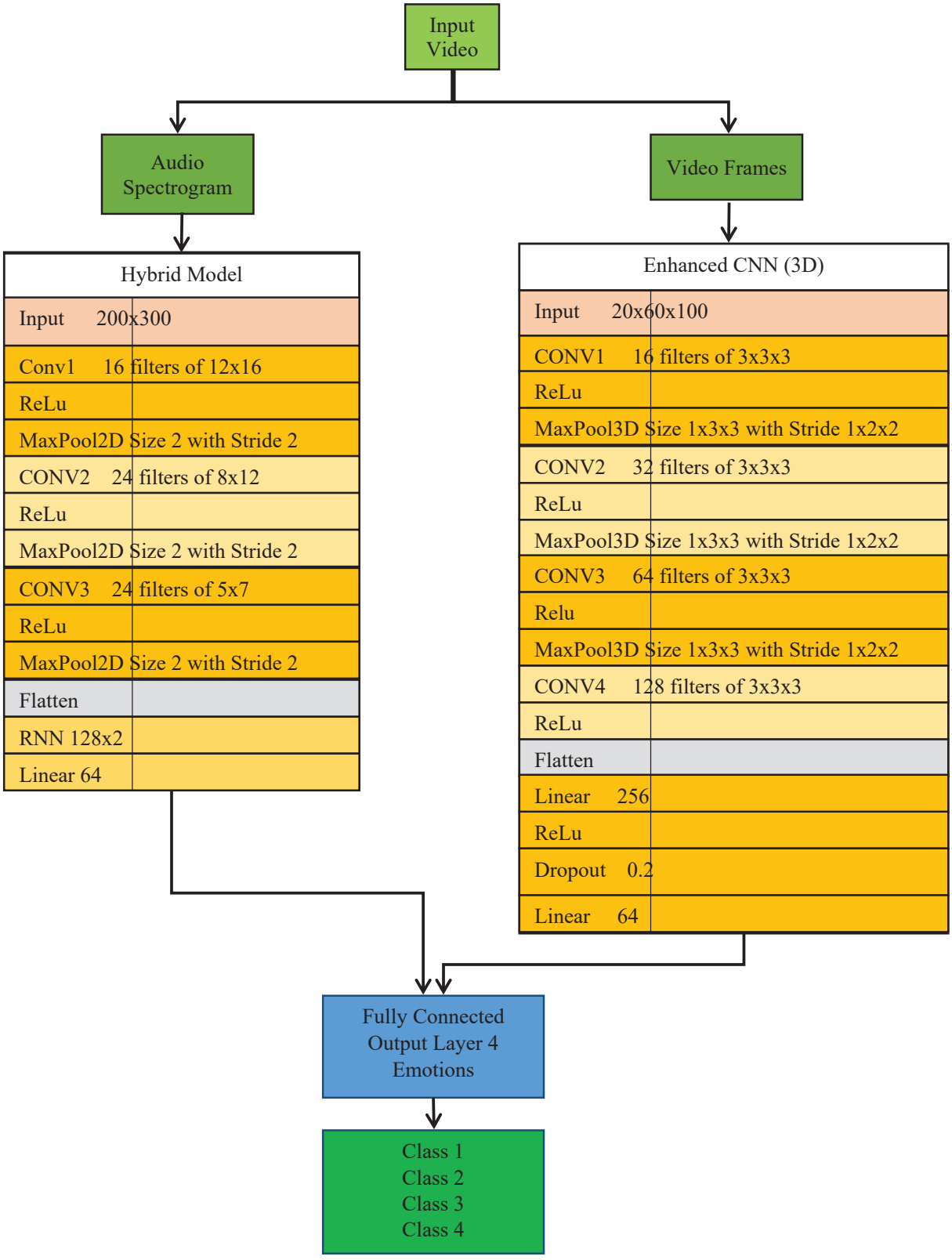


Fig. 3. Proposed hybrid deep learning model for emotion recognition

$$L_{\text{contrastive loss}} = \frac{1}{N}\sum_{n=1}^{N} L_1^n + L_2^n \quad (2)$$

Where

$$L_1^n = (y^n)\|f_v(v^n) - f_a(a^n)\|_2^2$$

$$L_2^n = (1 - y^n)max(\eta - \|f_v(v^n) - f_a(a^n)\|_2, 0)^2$$

The number N denotes the number of data points in the dataset; the variables represent the video and audio sub-networks f_v, f_a; if the video frames and audio spectrogram are from the same movie, y^n is one; if not, it is zero. The margin hyperparameter is denoted by η. When the audio spectrogram and video frames are from the same video, $\|f_v(v^n) - f_a(a^n)\|_2$ should be tiny; otherwise, it should be significant. Thus, the audio and video models are driven to output identical values when their inputs are from the same video and highly different values not reducing the contrastive loss. This enables the model to understand how the same video's audio and visual components relate. Supervised learning is performed on the pre-trained model following pre-training. The output is the anticipated emotion, with the input being an audio spectrogram and video frames from a video.

3.6. PROPOSED ALGORITHM

We proposed an algorithm, Learning Human Emotion Recognition (LbHER), which exploits hybrid deep learning models that could process audio and video frames toward emotion recognition.

Algorithm: Learning based Human Emotion Recognition (LbHER)

Input: IEMOCAP dataset D

Output: Emotion classification results R, performance statistics P

1. Begin
2. D'←Preprocess()
3. ($T1$, $T2$)←SplitData(D')

Training Phase

4. audios←getAudio($T1$)
5. spectrograms←GenerateSpectrogram(audios)
6. videoFrames←getVideoFrames($T1$)
7. Configure hybrid DL model m (as in Fig. 3)
8. Compile m
9. m'←TrainModel(spectrograms, videoFrames)
10. Save m'

Testing Phase

11. audios←getAudio($T2$)
12. spectrograms←GenerateSpectrogram(audios)
13. videoFrames←getVideoFrames($T2$)
14. Load m'
15. R←RecognizeEmotions (spectrograms, videoFrames, m')
16. P←Evaluate(R, ground truth)
17. Display R
18. Display P
19. End

Algorithm 1. Learning-based Human Emotion Recognition (LbHER)

As presented in Algorithm 1, it takes IEMOCAP dataset as input and performs human emotion recognition. The algorithm is designed to process audio-visual data to classify human emotions and provide performance statistics. The algorithm starts with preprocessing the IEMOCAP dataset (D), which is then split into training (T1) and testing (T2) sets. During the training phase, audio data from T1 is extracted and converted into spectrograms, while video frames are also obtained. A hybrid Deep Learning (DL) model (m) is configured, compiled, and trained using spectrograms and video frames. The trained model is then saved for later use. In the testing phase, audio from T2 is converted into spectrograms, and video frames are extracted. The saved model (m') is loaded, and emotions are recognized using the test data and the model. The emotion recognition performance is evaluated against ground truth data, and the results (R) and the performance statistics (P) are displayed. The algorithm follows a structured approach to emotion recognition, leveraging audio and video data to train a hybrid DL model, which is then used to classify emotions in new data. The output includes both the emotion classification results and performance statistics, indicating the effectiveness of the LbHER algorithm in recognizing human emotions from audio-visual cues.

3.7. PERFORMANCE EVALUATION

Since we used a learning-based approach, metrics derived from the confusion matrix, shown in Fig. 4, evaluate our methodology.

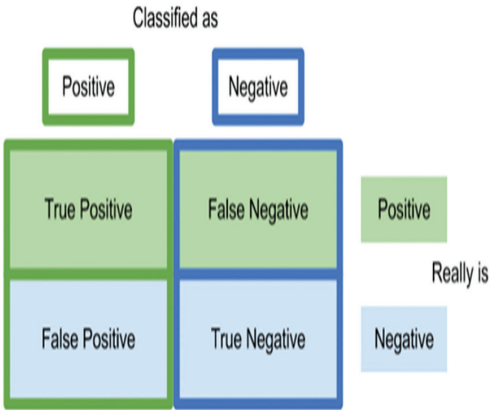


Fig. 4. Confusion matrix

Our method's predicted labels are compared with the ground truth based on the confusion matrix to arrive at performance statistics. Eq. 3 to Eq. 6 express metrics used in the performance evaluation.

$$\text{Precision (p)} = \text{TP}/(\text{TP+FP}) \quad (3)$$

$$\text{Recall (r)} = \text{TP}/(\text{TP+FN}) \quad (4)$$

$$\text{F1-score} = 2*((\text{p}* \text{r}))/((\text{p+r})) \quad (5)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

The measures used for performance evaluation result in a value that lies between 0 and 1. These metrics are widely used in machine learning research.

4. EXPERIMENTAL RESULTS

This section presents the experimental results of the proposed hybrid deep learning model, which automatically recognizes human emotions from a given video. The proposed approach's novelty is that it considers both audio content and video frames from the given input video.

Table 1. Class labels on the corresponding description

Class Label	Description
0	Happy
1	Anger
2	Sad
3	Neutral

The hybrid deep learning model exploits a benchmark dataset named IEMOCAP for getting trained towards automatic detection and classification of emotion. The proposed model performs multi-class classification. The class labels and the description are given in Table 1.

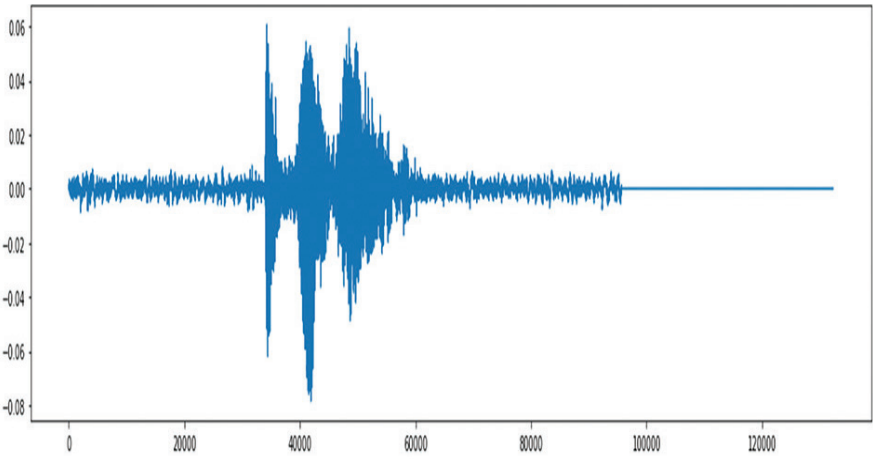


Fig. 5. Original audio content

Fig. 5. presents the content of the original audio file associated with the given test video.

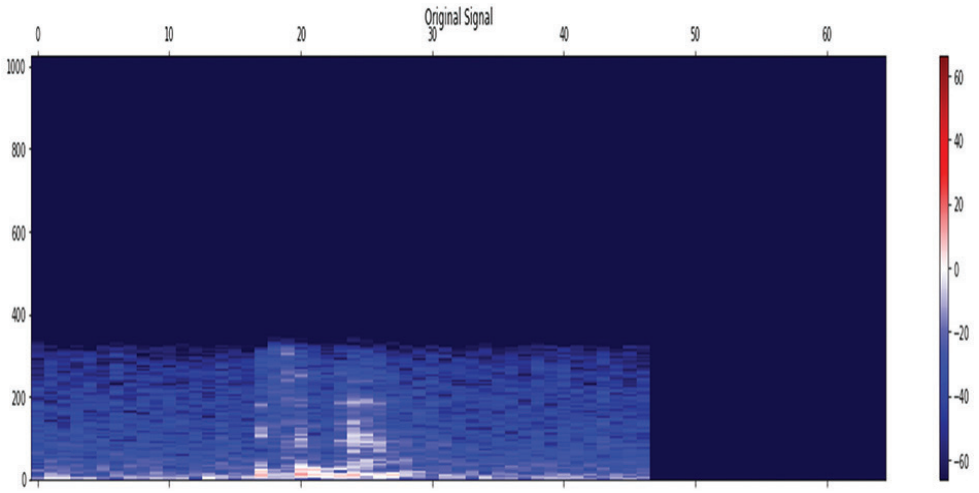


Fig. 6. Shows the spectrogram of the original audio file

As presented in Fig. 6, the given original audio file is converted to a spectrogram because the proposed hybrid deep learning model needs spectrogram input.

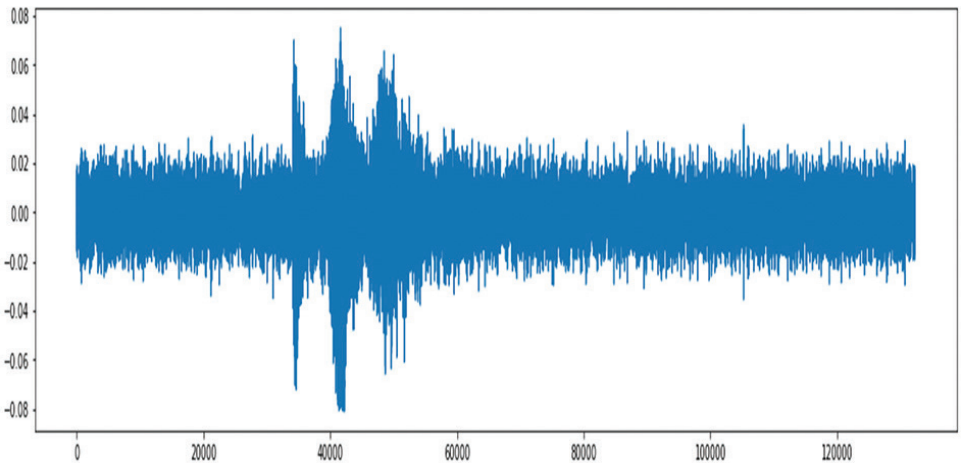


Fig. 7. Shows noise signal

As presented in Fig. 7, the noise signal associated with the given audio content is provided with visualization.

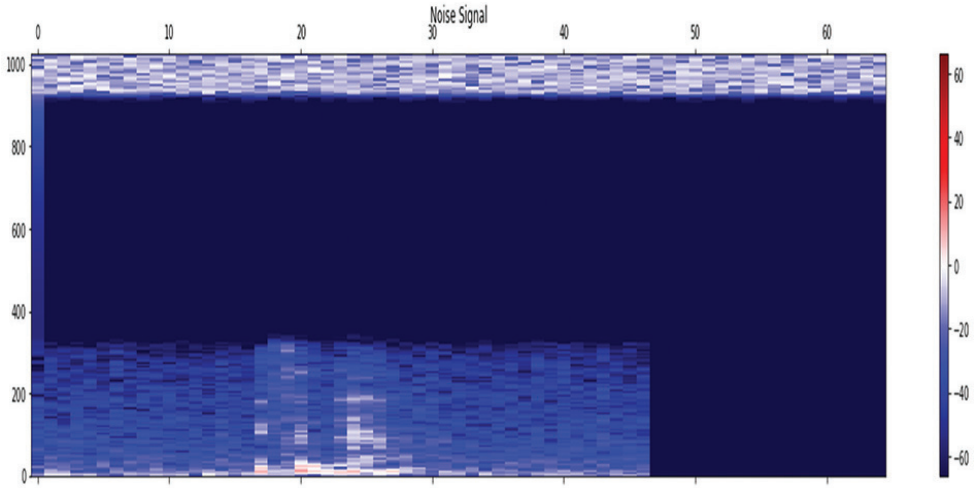


Fig. 8. Spectrogram of the audio with noise

The spectrogram of audio with noise is provided, as visualized in Fig. 8, to understand the data distribution in the form of the spectrogram.

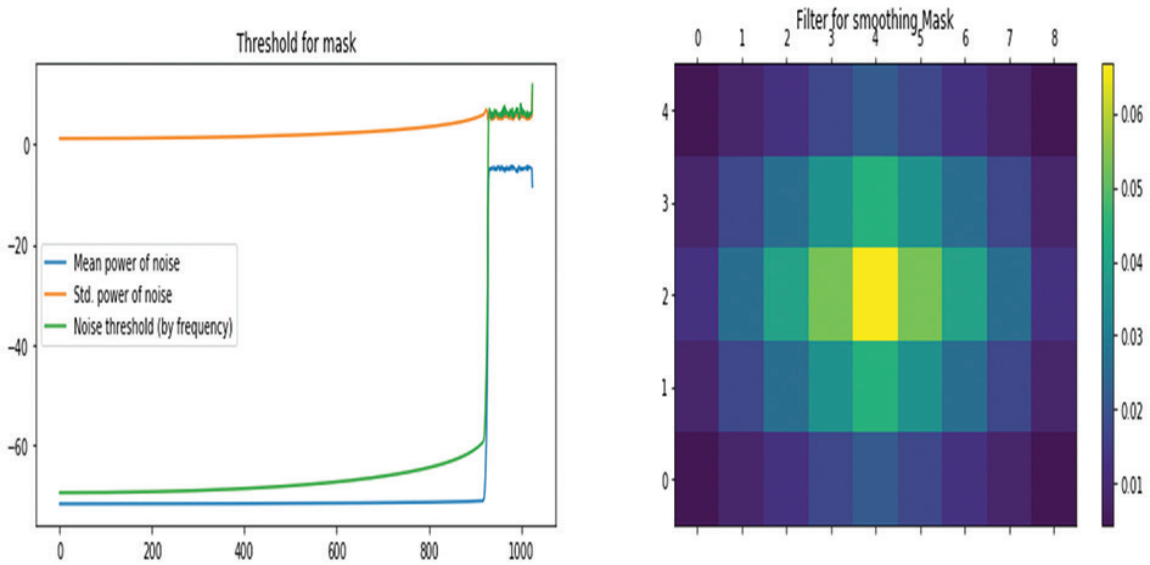


Fig. 9. Illustrates threshold for mask (left) and filter for smoothing mask (right)

As presented in Fig. 9, the threshold for the mask is provided in terms of mean power of noise, standard power of noise, and a noise threshold by frequency besides the filter for smoothing the mask.

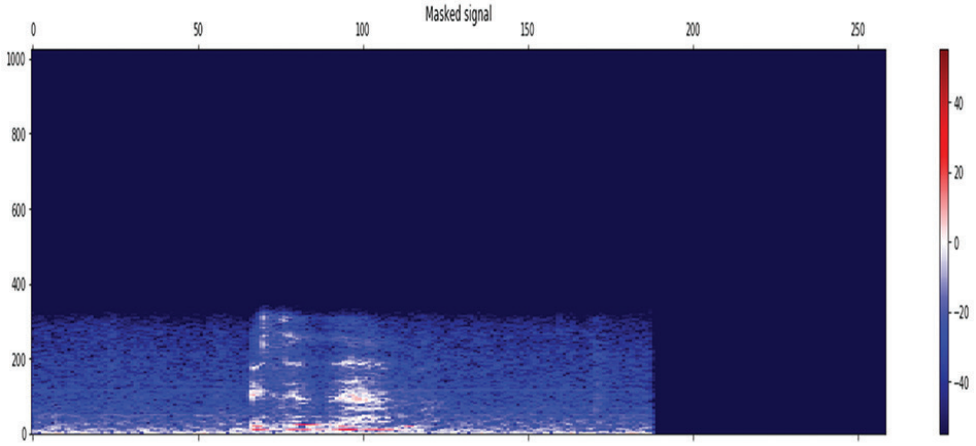


Fig. 10. Shows masked signal in terms of spectrogram

As presented in Fig. 10, the spectrogram of the masked signal is provided with visualization. It is the resultant spectrogram of the masked signal.

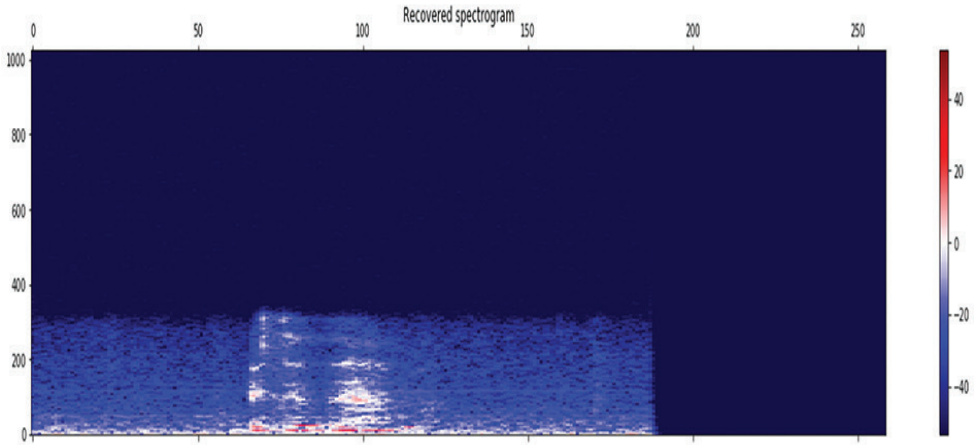


Fig. 11. Shows recovered spectrogram

As presented in Fig. 11, the recovered spectrogram of the masked signal is provided with visualization.

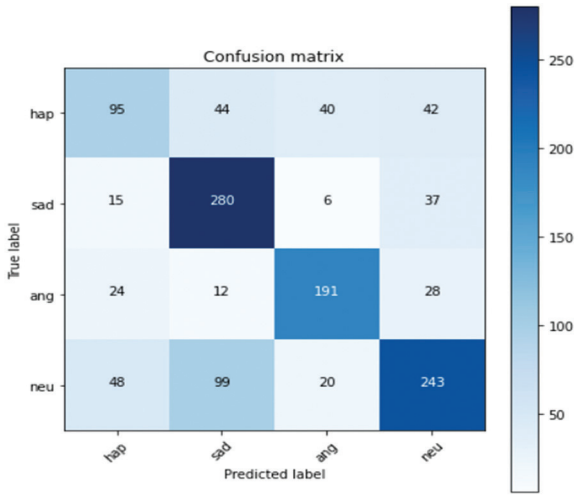


Fig. 12. shows the resultant confusion matrix reflecting the performance of the proposed model.

As presented in Fig. 12, the output of the proposed hybrid model for human emotion recognition is presented as a confusion matrix for different classes. It is the result of multiclass classification from which the model's accuracy is computed.

Table 2. Performance comparison

Model	Accuracy (%)		
	90-10 (Train-Test)	80-20 (Train-Test)	70-30 (Train-Test)
CNN	0.8942	0.8895	0.8833
CNN+LSTM	0.9174	0.9124	0.906
CNN+RNN	0.9341	0.9293	0.9228
CNN+RNN+3DCNN (Proposed)	0.952	0.9472	0.9406

Table 2 shows that the proposed model's performance is compared to many state-of-the-art deep learning models.

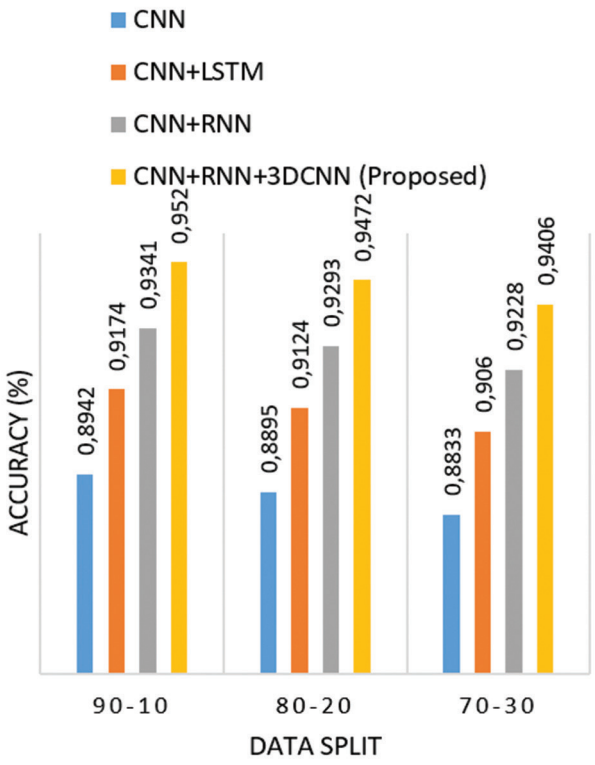


Fig. 13. Shows performance comparison among different models

As presented in Fig. 13, it is understood that different deep learning models showed varied levels of performance in human emotion recognition from a given video. The baseline CNN model, CNN and LSTM hybrid model, and CNN and RNN hybrid model are the existing models compared with the proposed hybrid deep learning model, which comprises CNN, RNN, and enhanced CNN model. CNN model exhibited 88.90% average accuracy, CNN + LSTM 91.19% accuracy, and CNN + RNN hybrid 92.87%, while the proposed hybrid deep learning model exhibited the highest average accuracy with 94.66%. The results show that the proposed hybrid model could perform better than existing and baseline CNN models.

5. DISCUSSION

Human emotion recognition has its utility in many real-world applications. Particularly in healthcare and psychology domains, it is essential to understand emotions to make well-informed decisions. Deep learning models are widely used in computer vision applications to perform various tasks. Since the proposed framework aims to recognize emotions from a given video, deep-learning models are preferred in this paper. However, from the empirical study, it was understood that deep learning models like CNN could help extract features from the input video but lack ability in multi-class classification. To overcome this problem, we proposed a hybrid deep learning model that exploits CNN, RNN, and enhanced CNN in this paper. The framework is designed in such a way that it makes use of both audio content and video frames from the given input. The CNN + RNN combination is used to extract features from audio content, while enhanced CNN is used to extract features from video frames. A fully connected layer is used to perform multi-class classification. The empirical study shows that the proposed hybrid deep learning model could outperform many existing deep learning models in human emotion recognition. However, the proposed methodology has certain limitations, as discussed in Section 5.1.

5.1. LIMITATIONS

In this paper, the proposal framework has certain limitations. The framework comprises a hybrid deep learning model trained with a particular dataset. It is essential to use diversity in data to have generalized findings. This limitation must be overcome with diversified datasets to train the proposed hybrid deep learning model. Another significant limitation is that the proposed hybrid model needs further optimization with improved hyperparameter tuning strategies. The proposed framework can also be enhanced by introducing Generative Adversarial Network (GAN) architecture suitable for human emotion recognition.

6. CONCLUSION AND FUTURE WORK

We proposed a deep learning framework with specific optimizations for facial expression and emotion recognition from videos. We proposed an algorithm, Learning Human Emotion Recognition (LbHER), which exploits hybrid deep learning models that could process audio and video frames toward emotion recognition. Our empirical study with a benchmark dataset, IEMOCAP, has revealed that the proposed framework and the underlying algorithm could leverage state-of-the-art human emotion recognition. Our experimental results showed that the proposed algorithm outperformed many existing models with the highest average accuracy of 94.66%. In the future, we intend to improve our deep learning framework with a Generative Adversarial Network (GAN) architecture. Another direction for future work is to investigate and evaluate our deep learning framework with multiple diversified datasets to generalize the findings.

7. REFERENCES

[1] J. Zhang, Z. Yin, P. Chen, S. Nichele, "Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review", Information Fusion, Vol. 59, 2020, pp. 103-126.

[2] W. E. Villegas-Ch, J. García-Ortiz, S. Sánchez-Viteri, "Identification of emotions from facial gestures in

a teaching environment with the use of machine learning techniques", IEEE Access, Vol. 11, 2023, pp. 38010-38022.

[3] C. Á. Casado, M. L. Cañellas, M. B. López, "Depression recognition using remote photoplethysmography from facial videos", IEEE Transactions on Affective Computing, Vol. 14, No. 4, 2023, pp. 3305-3316.

[4] A. Hassouneh, A. M. Mutawa, M. Murugappan, "Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods", Informatics in Medicine Unlocked, Vol. 20, 2020, pp. 1-9.

[5] A. Pise, H. Vadapalli, I. Sanders, "Facial emotion recognition using temporal relational network: an application to E-learning", Multimedia Tools and Applications, Vol. 81, No. 19, 2022, pp. 26633-26653.

[6] K. Patel, D. Mehta, C. Mistry, R. Gupta, S. Tanwar, N. Kumar, M. Alazab, "Facial sentiment analysis using AI techniques: state-of-the-art, taxonomies, and challenges", IEEE Access, Vol. 8, 2020, pp. 90495-90519.

[7] O. Bazgir, Z. Mohammadi, S. A. H. Habibi, "Emotion recognition with machine learning using EEG signals", Proceedings of the 25th national and 3rd International Iranian Conference on Biomedical Engineering, Qom, Iran, 29-30 November 2018, pp. 1-5.

[8] G. Anbarjafari, J. Guo, Z. Lei, J. Wan, E. Avots, N. Hajarolasvadi, B. Knyazev, A.Kuharenko, "Dominant and Complementary Emotion Recognition From Still Images of Faces", IEEE Access, Vol. 6, pp. 26391–26403.

[9] C. Diamantini, A. Mircoli, D. Potena, E. Storti, "Automatic annotation of corpora for emotion recognition through facial expressions analysis", Proceedings of the 25th International Conference on Pattern Recognition, Milan, Italy, 10-15 January 2021, pp. 5650-5657.

[10] I. M. Revina, W. R. S. Emmanuel, "A survey on human face expression recognition techniques", Journal of King Saud University-Computer and Information Sciences, Vol. 33, No. 6, 2021, pp. 619-628.

[11] Y. Cimtay, E. Ekmekcioglu, S. Caglar-Ozhan, "Cross-subject multimodal emotion recognition based on hybrid fusion", IEEE Access, Vol. 8, 2020, pp. 168865-168878.

[12] A. Kumar, A. Kaur, M. Kumar, "Face detection techniques: a review", Artificial Intelligence Review, Vol. 52, pp. 927-948.

[13] M. Zulfiqar, F. Syed, M. J. Khan, K. Khurshid, "Deep face recognition for biometric authentication", Proceedings of the International Conference on Electrical, Communication, and Computer Engineering, Swat, Pakistan, 24-25 July 2019, pp. 1-6.

[14] A. Topic, M. Russo, "Emotion recognition based on EEG feature maps through deep learning network", Engineering Science and Technology, an International Journal, Vol. 24, No. 6, 2021, pp. 1442-1454.

[15] M. Chen, Y. Hao, " Label-less learning for emotion cognition", IEEE Transactions on Neural Networks and Learning Systems, Vol. 31, No. 7, 2019, pp. 2430-2440.

[16] T. Song, W. Zheng, C. Lu, Y Zong, X. Zhang, Z. Cui, "MPED: A multi-modal physiological emotion database for discrete emotion recognition", IEEE Access, Vol. 7, 2019, pp. 12177-12191.

[17] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah, S. U. Ali, "Facial recognition using convolutional neural networks and implementation on smart glasses", Proceedings of the International Conference on Information Science and Communication Technology, Karachi, Pakistan, 9-10 March 2019, pp. 1-6.

[18] J. X. Chen, P. W. Zhang, Z. J. Mao, Y. F. Huang, D. M. Jiang, Y. N. Zhang, "Accurate EEG-based emotion recognition on combined features using deep convolutional neural networks", IEEE Access, Vol. 7, 2019, pp. 44317-44328.

[19] S. Parui, A. K. R. Bajiya, D. Samanta, N. Chakravorty, "Emotion recognition from EEG signal using XGBoost algorithm", Proceedings of the IEEE 16th India Council International Conference, Rajkot, India, 13-15 December 2019, pp. 1-4.

[20] W. L. Zheng, W. Liu, Y. Lu, B. L. Lu, A.Cichocki, "Emotionmeter: A multimodal framework for recogniz-

ing human emotions", IEEE Transactions on Cybernetics, Vol. 49, No. 3, 2018, pp. 1110-1122.

[21] C. Qing, R. Qiao, X. Xu, Y. Cheng, "Interpretable emotion recognition using EEG signals", IEEE Access, Vol. 7, 2019, pp. 94160-94170.

[22] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions", Information Fusion, Vol. 91, 2023, pp. 424-444.

[23] P. Sarkar, A. Etemad, "Self-supervised ECG representation learning for emotion recognition", IEEE Transactions on Affective Computing, Vol. 13, No. 3, 2020, pp. 1541-1554.

[24] D. Ayata, Y. Yaslan, M. E. Kamasak, "Emotion based music recommendation system using wearable physiological sensors", IEEE Transactions on Consumer Electronics, Vol. 64, No. 2, 2018, pp. 196-203.

[25] R. He, J. Cao, L. Song, Z. Sun, T. Tan, "Adversarial cross-spectral face completion for NIR-VIS face recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, No. 5, 2019, pp. 1025-1037.

[26] H. Feng, H. M. Golshan, M. H. Mahoor, "A wavelet-based approach to emotion classification using EDA signals", Expert Systems with Applications, Vol. 112, 2018, pp. 77-86.

[27] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection", Information Fusion, Vol. 64, 2020, pp.131-148.

[28] D. Hazarika, S. Poria, R. Zimmermann, R. Mihalcea, "Conversational transfer learning for emotion recognition", Information Fusion, Vol. 65, 2021, pp. 1-12.

[29] T. Baltrušaitis, C. Ahuja, L. P. Morency, "Multimodal machine learning: A survey and taxonomy", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 41, No. 2, 2018, pp. 423-443.

[30] A. Davison, W. Merghani, C. Lansley, C. C. Ng, M. H. Yap, "Objective micro-facial movement detection using facs-based regions and baseline evaluation", Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition, Xi'an, China, 15-19 May 2018, pp. 642-649.

[31] D. Amodei et al. "Deep speech 2: End-to-end speech recognition in english and mandarin", Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19-24 June 2016, pp. 173-182.

[32] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, J. Dawson, "3D convolutional neural networks for cross audio-visual matching recognition", IEEE Access, Vol. 5, 2017, pp. 22081-22091.

Classification of Road Scenes Based on Heterogeneous Features and Machine Learning

Original Scientific Paper

Sanjay P. Pande
Yeshwantrao Chavan College of Engineering,
Department of Computer Technology,
Hingna, Nagpur, Maharashtra, India
sanjaypande2001@gmail.com

Poonam T. Agarkar*
Ramdeobaba University,
School of Computer Science and Engineering
Katol Raod, Nagpur, Maharashtra, India
agarkarp@rknec.edu

Sarika Khandelwal
G H Raisoni College of Engineering,
Department of Computer Science and Engineering
Digdoh Hills, Nagpur, Maharashtra, India
sarikakhandelwal@gmail.com

Rajani D. Singh
Ballarpur Institute of Technology,
Department of Master of Computer Application
Ballarpur, Chandrapur, Maharashtra, India
rajanidsingh@gmail.com

Pratik R. Hajare
Mansarovar Global University,
Department of Electrical and Electronics Engineering
Raison Road, Bhopal, Madhya Pradesh, India
pratikhajare8@gmail.com

Prashant R. Patil
Smt. Radhikatai Pandav College of Engineering,
Department of Management Studies
Umrer Road, Nagpur, Maharashtra, India
patilnagpur@gmail.com

*Corresponding author

Abstract *– There is a rapid advancement in Artificial intelligence (AI) and Machine Learning (ML) that has extensively improved the object detection capabilities of smart vehicles today. Convolutional Neural Networks (CNNs) based on small, medium, and large networks have made significant contributions to in-vehicle navigation. Simultaneously, achieving higher level accuracies and faster response in autonomous vehicles is still a challenge and needs special care and attention and must be addressed for human safety. Hence, this article proposes a heterogeneous features-based machine learning framework to distinguish road scenes. The model incorporates object-based, image-based, and diverse conventional features from the road scene images generated from four distinct datasets. Object-based features are acquired using the YOLOv5m model and modified VGG19 networks, whereas image-based features are extracted using the modified VGG19 network. Conventional features are added to the object-based and blind features by applying a variety of descriptors that include Matched filters, Wavelets, Gray Level Occurrence Matrix (GLCM), Linear Binary Pattern (LBP), and Histogram of Gaussian (HOG). The descriptors are used to extract fine and course features to enhance the capabilities of the classifier. Experiments show that the proposed road scene classification framework performed better in classifying two scene categories, including crosswalks, parking, roads under bridges/tunnels, and highways achieving an average classification accuracy of 97.62% and the highest of 99.85% between crosswalks and Parking. A marginal improvement of approximately 1% is seen when all four categories were considered for evaluation using a multiclass SVM compared to other competing models.*

Keywords: *Artificial intelligence, Machine Learning, smart vehicles, CNN, object-based, image-based, diverse conventional features, YOLOv5m, and VGG19.*

Received: July 27, 2024; Received in revised form: December 16, 2024; Accepted: December 23, 2024

1. INTRODUCTION

Safer autonomous vehicles work on algorithms based on computer vision that can distinguish certain scenarios and accurately predict labels. Related scenes consist of several details and are infinite. Varying image classification achievements are significant and include a wide range of image classes [1, 2]. Remarkable results have been obtained on the ImageNet dataset using convolutional neural networks and frequent improvements are suggested by many researchers [3]. However, further initiatives are needed in scene categorization to improve visual perception in autonomous driving. Work introduced in [4] considered 2.5 million images for training with 205 categories of worldwide places that included outdoor scenes. This was based

on scene-centric Places and used CNN for recognition/classification. The work was extended in [5] for 365 categories from 2.1 million images. Object-centric feature-based training is easier than scene-level tasks due to their varied scene diversities and possible scene combinations.

Several techniques infer scene categories based on object detection related to certain scenes. Semantic information was also used to guide a mobile robot [6, 7] in the indoor environment for high-level navigation. Thus, semantic mapping within a scene image is widely used for navigating vehicles with better accuracy. It uses semantic information that includes parking lots, objects beside roads, buildings, towers, sidewalks, and constructions to represent rural or urban road scenarios. However, rural conditions are different and are concerned with the abundance of plants, trees, and several vegetation. Open broad roads and heavy vehicles are dominant over highways which are different from the crowded streets in rural and urban areas. Scene categories are prominently defined based on weather and light conditions [8], stationary objects in the scene [9], special traffic scene categories (An example to quote - Square with Street lights), and intersections. Related datasets are not provided with labels, however, they include an overall description and attributes for objects in the scene [10]. Scene videos are typically summarized on a clip basis, since they are not annotated frame by frame, making the labeling process more time-consuming and costly.

In this article, we have collected road scene images from different source datasets and segregated them using a machine-learning tool. The work comprises heterogeneous features that are extracted from the scene images which assist the classifier in distinguishing two classes with better accuracy. The quality features used as input to the support vector machine are based on semantic information about the scene, objects in the scene, and conventional attributes. Pre-trained networks are used to extract the object-based and image-based features whereas diverse descriptors are used to lift different details of the scene images. The scene images are manually selected and labeled [11]. We believe that the proposed scene classification network can be used for offline and real-time applications including driver assistance and mapping semantically autonomous datasets.

Despite several deep-learning and machine-learning models, AI is still not capable of annotating and distinguishing the RS environment autonomously, without human intervention. Numerous experiments were conducted considering the good road conditions and weather, but more recent experiments include real road and weather conditions. Adverse weather conditions such as rain, fog, thick pollution, and snow are still to be evaluated properly for self-driven cars. The resolution of the LiDAR cameras has been enhanced to a great extent and is not the image quality that matters, relating to object recognition and classification. Many findings infer that autonomous robots are no longer a question, but when and how they would be launched in human society. The only question is how safely they will drive on the real roads, irrespective of the geographical structures and conditions.

This emphasizes a critical need for reliable detection of the scene objects using efficient techniques, mathematical modeling, and simulations that can exactly represent reality and converge at the best performance parameters and architectures to adapt to variations in the surroundings. Various contextual factors are required to be considered to improve the generalization capability and confident predictions for the road scene classification, from where the images were acquired. Vision-based perceptual systems are greatly influenced by contextual factors such as geographical locations, weather conditions, and illuminations, geographical or artificial processes.

Findings reported that recent state-of-the-art techniques incorporated deep learning for object detection and scene understanding in the scene images, and there is a broader scope for additional improvements. Still, the performance of CNNs is yet to be investigated under realistic conditions, that when and under what fatal conditions it will cease to operate and can pose a great threat to precious human life in self-driven circumstances.

The weaknesses and deficiencies found after surveying most of the recent and persistent studies are:

1. The inability to detect and classify large objects in the scenes.
2. False detections for small objects.
3. Lack of generalization ability due to changes in weather conditions.
4. Lack of scene content representation, and
5. Complexity of time and computation.

Therefore, there is always room for improvement in distinguishing scenes based on their contents to assist Automated Vehicles. The present research aims to design an intelligent scene classification framework with higher accuracy and lower complexity irrespective of the object dimensions, weather conditions, and uneven illuminations.

The paper claims the following contributions:

1. There is an extraction of object-based features using the VGG19 pre-trained network after detecting the objects with the YOLOV5 network and resizing them to a predetermined dimension.

2. Handcrafted low- and high-level features on gray-scale images are also extracted to improve the disparities among the classes and improve classification. The features include wavelet-based features, local binary pattern-based features, gray-level co-occurrence

matrix features, histograms of Gaussian features, and matched filter coefficients.

3. Blind features (Image-Based features) using VGG19 from the color scene images are extracted from the last fully connected layer of the VGG19 network to obtain the depth level information of the images.

4. The analysis based on the experiments displayed that the proposed Machine learning framework employing a diverse set of features can classify road scenes with higher accuracy.

The remaining paper is framed as follows: Work carried out by different researchers is summarized in the forthcoming section and our proposed scene classification framework is elaborated in the preceding section after the literature review. The last section concludes by discussing the experimental results and avenues of future studies after analyzing the results obtained through our proposed framework.

2. RELATED WORK

The objects associated with the road scene images need accurate detection and classification for precise decisions to assist the driver in taking different actions along the road. Nowadays, for a better 3D perspective, object identification has taken its place as a subdomain in computer vision tasks [12]. The objective is to provide safety, save lives, minimize accidents, and make transportation reliable, and efficient [13, 14]. A variety of techniques are found in the literature for detecting objects in images relative to several applications. Specific objects for specific applications are now a sub-problem of the generalized recognition task. It includes attribute and name assignments for specific objects [15]. The most crucial and challenging part is dealing with 3D objects for autonomous vehicle driving using an optical navigation system. Several sensors are mounted to provide road scene details to the navigation module. In the end, a classifier system is used to collect information and guide the vehicle along a derivable region [16]. Udacity recognized multiple transportation means in the scene by employing HOG features and classified them using various classifier networks. They primarily used the GTI (Grupo de Tratamiento de Imágenes, Madrid, Spain) and the KITTI (Karlsruhe Institute of Technology, Karlsruhe, Germany and Toyota Technological Institute, Nagoya, Japan) benchmark datasets. They obtained superior results using the logistic regression module [17] as compared to SVM and decision trees.

The BDD100K dataset was constructed using several images making it large and comprehensive including a variety of objects acquired in diverse weather situations, places, and times, with occlusions and a wide range of intensity conditions. The YOLO model constructed using the Deep CNN is based on learned features extensively used to detect objects in the real-time environment in videos and images. The work proposed in [12] used YOLOv3 and YOLOv4 models on the BDD100K dataset and obtained significant results improving the detection rate. The authors replaced Leaky RLU with advanced activation functions (MISH and SWISH) and further improved the detection accuracy over the Leaky RLU [12].

Objects of different dimensions (small, medium, and large) were detected in [14] using a single–shot multibox detector (SSD), faster region-based CNN (RCNN), and algorithms present in PyTorch. Experiments were carried out on the BDD100K dataset images. Further research included the KITTI dataset where the performance was measured using average precision for detecting 3D objects in the scene images. The outputs were significantly enhanced by dividing the 3D objects into easy, moderate, and difficult levels. The last level included classifying objects in foggy environments for autonomous vehicles [18].

Object detection in the dark (night) was better in [19] using YOLOv3, Aggregate view object detection, and PointPillars. The techniques resulted in better average precision over the KITTI dataset than others. PointPillars performed the best over objects at night, however, it failed to detect objects in rainy conditions [19]. Sparse LiDAR Stereo Fusion Networks were incorporated in [18] to improve object detection in foggy weather (Multi-fog environment – KITTI) [18]. A combination of YOLOv3 and Darknet-53 was used for detecting and classifying various objects [20]. The work suggested in [21-22] used CNN to convert semantic details from sensory data in the images on the road to recognize cycle riders, pedestrians, vehicles, etc. A novel approach was proposed to process ambulance sounds from a long distance to determine the direction of the emergency vehicle [23].

3D object detection and classification on real and synthetic samples was implemented in [24]. Most studies are compared using the average precision as the evaluation measure. The weaknesses and deficiencies found after surveying most of the recent and persistent studies [25-33] are time complexity, inability to detect and classify large objects in the scenes, lack of generalization ability due to changes in weather conditions, and false detection for small objects. Therefore there is always room for improvement in distinguishing scenes based on their contents to assist autonomous vehicles. The present research aims to design an intelligent scene classification framework with higher accuracy and lower complexity irrespective of the object dimensions, weather conditions, and uneven illuminations.

3. MATERIALS AND METHOD

The authors of this work collected the road scene images from four different sources. The objective was to consider the worst possible scenario for scene classification to assist automated vehicles. The manually separated scene images possess complexity related to multiclass, poor illumination on account of different weather conditions and dimensions. The familiar data-

sets that were used to generate a custom dataset for this work include the LabelMe [34], KITTI [35], BDD100K [36], and Places365 [37] datasets. The crucial parameter to list out the complexity of scene images is their poor imperceptibility. The significant class was still undistinguished even with visual perceptivity. The challenge was to detect the relevant objects in the scene that were not easily detectable. The authors customized the dataset that contained a sum of 2725 scene images from the four benchmark datasets and included road scene images with highway roads (HR), vehicle parking lots (VPL), crosswalks (CRW), and roads under bridges/tunnels (RB/T). Fig. 1 shows road scene images from all four classes.

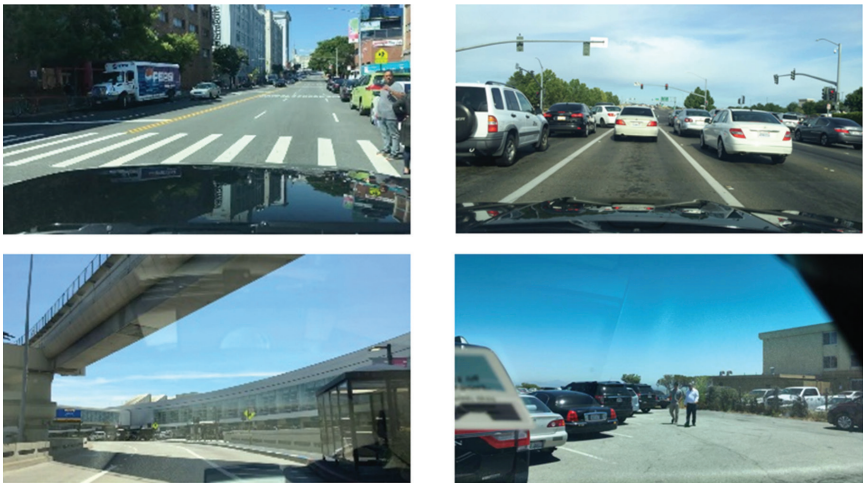


Fig. 1. Road scene images. From top left to bottom right - A crosswalk on the street, vehicles along a highway, a road under an overpass, and vehicles parked at a parking lot

The generated dataset consists of an equal number of images (700 each) for classes HR, VPL, and CRW, while RB/T included 625 images thus resulting in an unbalanced dataset. Fig. 2(a-d) below shows some examples that are difficult to distinguish since the significant objects are either missing or captured during the night due to which significant objects are not clear. Fig. 2(a) shows a highway without vehicles, Fig. 2(b) depicts vehicles parked at night, Fig. 2(c) is an underpass that is not clear, and Fig. 2(d) has a partial crosswalk covered due to vehicles.

(a)

(b)

(c)

(d)

Fig. 2. Sample road scene images from each category. (a) A deserted Highway. (b) A dimmed roadside parking at night. (c) A non-significant under-tunnel road. (d) An Occluded crosswalk

The classification framework is based on the fact that road scenes can be categorized with higher accuracy when the details in the scene images are extracted carefully to represent the road scene correctly. Researchers have suggested that the features extracted from the scene should carry details regarding the objects in the scene, the global or overall characteristics of the scene (image-level), and the fine or local details in the scene images (conventional/handcrafted). Due to the high resemblance among the variety of road scenes, sufficient discriminative information from the scene images is required to properly distinguish scene classes. Redundant information would certainly mislead the classifier, thus increasing the possibility of false detection. Therefore, the proposed scene classification framework is based on an efficient integrated feature-based machine learning approach. Diverse features including overall, fine, and object-based features are integrated using modified pre-trained networks and handcrafted or conventional descriptors. The overall or global features and the object-based features are extracted from the scene images using a modified pre-trained network VGG19 whereas the fine patch-based or window-based features are acquired using eight different descriptors.

3.1. OBJECT-ORIENTED FEATURES

– All the scene images are resized to 256x256 and the scene objects are detected using the YOLOV5m pre-trained network. The capabilities of the YOLOV5m network to identify 80 different objects are utilized to recognize objects in the scene images. Fig. 3 shows an example of object detection on a road scene using the YOLOv5m network. The identified objects include bicycles, cars, persons, and benches. Due to the varying dimensions of objects in the scene, the objects were priory detected from the scene image and then resized to a dimension for further feature extraction. The objective was to consider the contribution of every single object either small or large in dimension from the image. Experimental analysis displayed that every detected object from the scene should be resized to 32x32 so that their contribution is guaranteed. The strength of the feature vector was made dependent on the number of objects detected in the scene. The resized object was subjected to the feature extraction to a modified VGG19 network. The last layer of the VGG19 network was replaced with 1024 and 512 fully connected layers. Thus, for every single object a feature vector of 512 was obtained. The feature vectors obtained from different objects of a single image were then added to determine the strength of each element of the feature vector. The summation also mitigates the presence of zero values in the feature vector.

3.2. SCENE-LEVEL FEATURES

These features are directly obtained from the scene image. The original color image of size 256x256 is subjected to the pre-trained network (modified VGG19) and features are extracted from the image. For each image, a feature vector of 512 lengths was obtained and appended to the object-level feature vector.

Fig. 3. YOLOV5m object detection

3.3. CONVENTIONAL FEATURES

Local features from the scene images (reduced to half-dimension) were extracted using various global and local descriptors. The descriptors include matched filters (kernel-based and orientation-based), wavelets (coarse features using 6 wavelets and fine features using the haar wavelet), linear binary pattern (using 3x3 and 5x5 window), histogram of Gaussian, and Gray level co-occurrence matrix. Kernel-based matched filters [38], haar-based wavelet features [39], and LBP features [40] were used for extreme details whereas orientation-based match filters, wavelet-based (['bior3.1', 'bior3.5', 'bior3.7', 'db3', 'sym3', 'haar']), GLCM [41] and HOG [42] were used to contribute in terms of slight or medium details. The words extreme and medium correspond to details acquired from the image. The former represents details with more elements as compared to the latter one. The total number of feature elements corresponding to all the descriptors was 2310. The following Fig. 4 shows the variety of features and their dimensions.

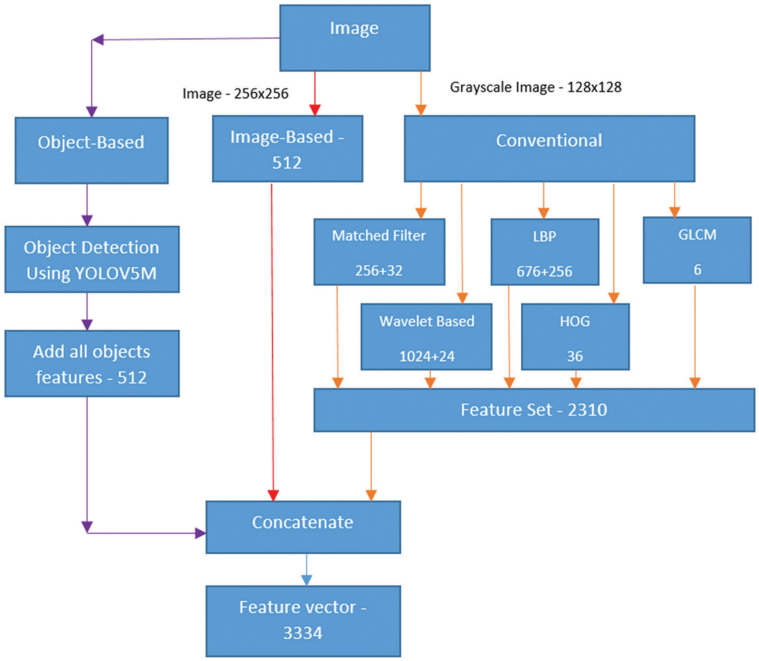


Fig. 4. Object-based, Image-based, and Conventional features with dimensions

Additional details for the conventional/handcrafted features can be found in [43]. These quality features improve the ability of the classifier network or a machine learning classifier. After all the dataset images are completely used for the feature extraction process, the features are normalized using the Max-normalization to fit the values in the range [0 1]. The normalization process was carried over the individual feature column while the missing values in the columns were substituted using the column mean. Fig. 4 shows objects detected by the YOLOv5m network from a street image and Fig. 5 (b-c) depicts the segmented objects from the crosswalk class image shown in Fig. 5(a).

The detailed scene classification framework is shown in Fig. 6. The global FEM uses the VGG19 network without the top layer directly on the input image resized to 256x256. The number of features extracted using the VGG19 network is 512. The blind features thus extracted depend on the scene information and ability of the

network. This is to ensure that regions not belonging to the objects detected using the local FEM contribute to the feature set. The only problems with such features are too many missing values which depend on the quality of the image. Even though the local features are considered using the two-stage deep network framework using the YOLOV5 and the VGG19 networks, the resizing stage for the detected objects may suffer from information loss. A size of 32x32 is considered to uplift the fine features concerning small objects but objects greater than 32x32 would suffer data loss. Therefore, we added fine and coarse features to the local and global features to improve the classification accuracy.

A total of 2310 HF are extracted using various feature descriptors which include wavelet-based, matched filter-based, LBP-based texture, GLCM-based, and the HoG features. The classifier (SVM) is used to learn the representation and predict the sample class for the assessment sample.

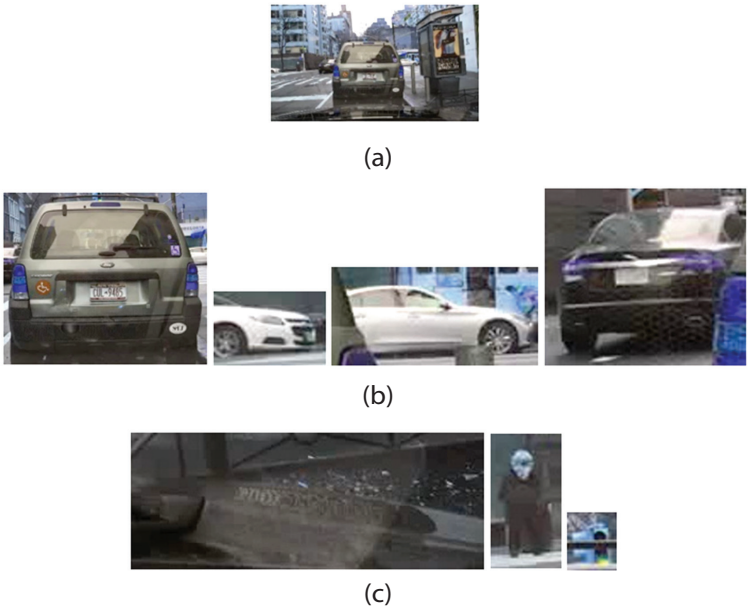


Fig. 5. Object detection using the YOLOV5m network from a single road scene image shown in (a) - Crosswalk Class image). (b)- Objects located in the image by YOLOV5m) Vehicles were detected at the scene. (c) - Other detected objects: keyboard, person and traffic light) Other detected objects include a keyboard, a person, and a traffic light.

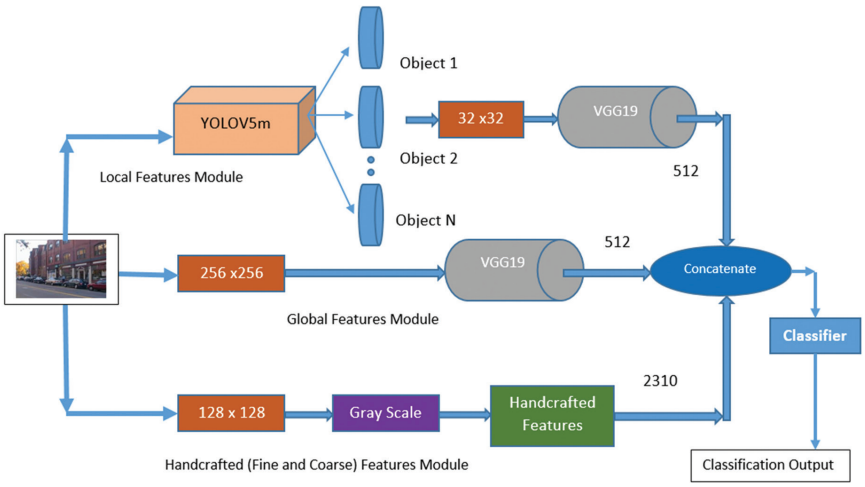


Fig. 6. The framework for the scene classification system

4. RESULTS

The performance metric that is used to evaluate the performance of the proposed road scene framework is classification accuracy. It is simply computed by calculating the ratio of scene samples correctly classified to total samples. It is expressed by the following expression (1):

$$\text{Accuracy} = \frac{\text{Number of scenes correctly classified}}{\textit{Total test samples}} \quad (1)$$

Heterogeneous features of all the 2275 images belonging to four different classes were extracted and stored in a CSV file. The proposed binary scene classification framework was developed in Python 3.9 on

Spyder 5, Windows 11 Environment, i5 Processor, 16 GB RAM, and 512 GB SSD. The feature samples were partitioned in the ratio of 80:20% for training, and testing randomly from the available set.

The training and testing sequence was iterated 10 times to note the classification accuracy between any two classes or categories at any instant. Support vector machine (SVM) with a 'Gaussian' kernel was used to train the feature samples and the maximum accuracy for any two classes was considered. Table 1 below shows the results obtained in terms of classification accuracy between two different categories considered for this research work. The heterogeneous features using different descriptors are uplifted and stored. Finally, the features are split as shown in Fig. 7 for training and testing. The SVM is trained on the training set to learn the scene representation and classify the test samples.

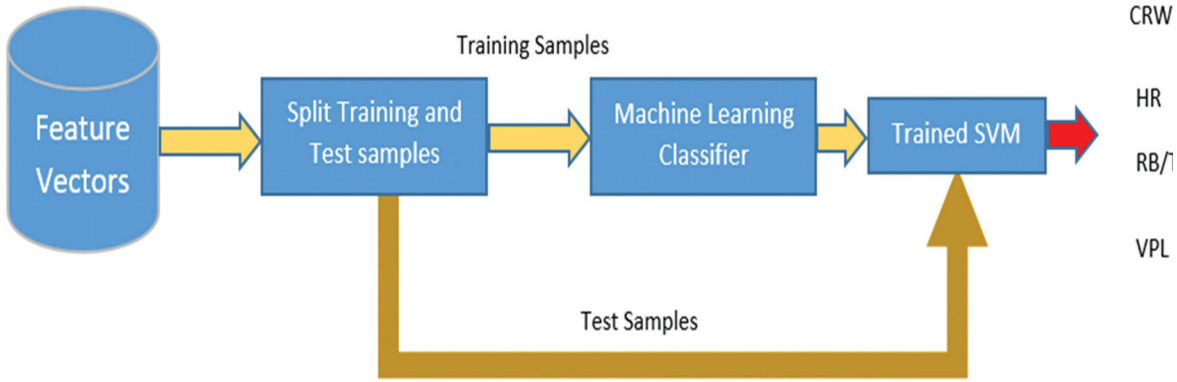


Fig. 7. The SVM-based classification model

Table 1. Classification Accuracy relating to Two Class performance

Class 1	Class 2	Accuracy-Training	Accuracy- Test
CRW	HR	100%	99.22%
CRW	RB/T	100%	93.87%
CRW	VPL	100%	99.85%
HR	RB/T	100%	97.54%
HR	VPL	100%	99.51%
RB/T	VPL	100%	95.74%

As seen from Table 1, the SVM was successful in training the samples with 100% accuracy. However, the test samples were not classified up to the 100% mark. The reason behind the low accuracy particularly in the case of CRW-RB/T, HR-RB/T, and RB/T-VPL is due to the multiple classes' existence in a single road scene image. Fig. 8 and Fig. 9 show some examples from the scene images. Fig. 8 below shows a crosswalk along with an underpass, a crosswalk below a tunnel, and a misleading crosswalk under a tunnel. Similarly, Fig. 9 shows vehicles parked beside a crosswalk, an unseen crosswalk under a tunnel, and far away parking with a crosswalk. Such images when falling under test samples would probably increase the chances of false detection. The research work uses the YOLOv5m network in its standard form and no transfer learning approach has been carried out to train the existing network for scene-based objects. The YOLOv5m network trained on the ImageNet dataset does not include several objects that are associated with road scene images. Therefore, significant objects are not detected by the network from the scenes which also amounts to the reason for low accuracy while detecting scene images. Also, no pre-processing is carried out to eliminate the uneven illumination effects caused by street-side lightning and vehicle lights. Several scene images were acquired during night, rain, and fog which need special attention.

(a) (b) (c)

Fig. 8. Multi-category scene samples.
(a) Crosswalk and an overpass. (b) Crosswalk under a tunnel and (c) Misleading crosswalk under tunnel.

(a) (b) (c)

Fig. 9. Closely correlated objects (Ambiguous) classes.
(a) Partial Crosswalk with parking (b) Unseen crosswalk under a tunnel and (c) Multiclass scene.

Similar work was suggested in [44] that used the contextual semantic relationship between the objects of the scene to classify the indoor and the outdoor scenes. The authors used visual attention regions marked with context-based saliency and deep CNN. They selected scene images from four different datasets including the MIT67, UIUC-Sports, LabelMe, and the Scene 15 dataset. They obtained a maximum classification accuracy of 97.70% over the LabelMe dataset and the lowest over the MIT67 dataset with 72.37%. However, the authors worked to differentiate indoor and outdoor scenes, the work carried out in this work is to classify different street scenarios which is more complex.

We have gone through several research papers to compare our results based on two-class road scenes however we found two research papers [27, 44] that included the categories considered in this work. Work introduced by Jianjun Ni et al. [27] particularly was oriented toward classifying the scenes into five different categories including crosswalks, gas stations, parking lots, highways, and streets using a deep network. While work by Jing Shi et al. [44] classified indoor and outdoor scenes utilizing a deep network. For comparison, we used a multi-class SVM and subjected all scene images for training and testing using the same ratio. Table 2 shows the total samples that were considered for the evaluation purpose.

Table 2. Number of sample images considered in each category for scene classification

Class	Class	Number of images
0	CRW	700
1	HR	700
2	RB/T	625
3	VPL	700

Although the work introduced by Jianjun Ni et al. and Jing Shi et al. is not comparable with our proposed work, we tried to obtain a better picture regarding the framework that considered diverse features extracted from the scene for multi-class configuration using SVM. The work differs in classes that were considered for the research. The competing models used deep learning networks for classifying the scenes, the proposed work utilizes a machine learning classifier. Table 3 shows the comparison between the competing models and results obtained through our proposed framework.

Table 3. Comparative test results based on Average Accuracy

Method	Categories	Classes	Average Accuracy %
Jin Shi et al.	2	Indoor, Outdoor	85.06
Jianjun et al.	5	CRW, Gas Station, HR, VPL and Street	75.99
Proposed Work	4	CRW, HR, VPL, and RB/T	86.01

Although the performance using our framework outperformed the other two competing models by approximately 1%, a lot of research is required in this area to incorporate a scene classification module in an automated vehicle. Better scene representation and advanced classifiers would obtain higher results and assist the unmanned vehicle over densely populated streets under rigorous road and climate conditions.

5. ABLATION STUDY

Maintaining the ML hyper-parameters, the researchers conducted experiments using any two sets of features at a time from object-oriented, scene-level, and conventional features. Table 4 shows the classification accuracies on 20% of test samples which were randomly chosen from the available samples about each of the categories. The average accuracies computed reveal that the object-oriented features and conventional features are crucial in classifying the scene classes but the scene-level features are essential to enhance the performance as seen in Table 1. Also, object-oriented features play an important role in representing the scene as seen from the last column of Table 2. Merely using the scene-level and conventional features would not differentiate complex scene images. Thus dropping any of the features has a greater influence on the performance.

Table 4. Classification Accuracies for Various Feature Combinations

Class 1	Class 2	Accuracy - Test Object-oriented & Scene-level features	Accuracy – Test Object-oriented & Conventional features	Accuracy - Test Scene-level & Conventional features
CRW	HR	91.10	93.94	89.68
CRW	RB/T	89.25	90.45	87.25
CRW	VPL	91.43	91.98	88.39
HR	RB/T	90.22	92.62	89.88
HR	VPL	92.15	93.94	90.10
RB/T	VPL	89.98	90.00	86.97
Average	**90.69**	**92.16**	**88.71**	

6. CONCLUSIONS

This article introduces a road scene classification framework based on heterogeneous features that are extracted at image-level, object-level, and local–level. The features that are extracted, are column normalized, and the missing entries are filled using the column mean. The feature samples are separated for training and testing in an 80:20 ratio and further classified using the support vector machine. The classification results that are obtained, reveal the proposed scene classification framework in classifying two classes that showed higher performance despite partial occlusions, ill-illumination due to diverse weather conditions, low inter-class disparities, multi-class ambiguities, and data imbalance.

There can be an improvement in the classification accuracy by adding quality preprocessing to mitigate the uneven illumination effects from the scene, YOLOv5m transfer learning for common road scene objects, other than the objects found in the ImageNet dataset, and using custom CNN networks. Due to heterogeneous features and three networks (YOLv5m and VGG19), the time required to extract features from the scene images is large. The future work will be based on the concentration of considering more than two classes (multiclass model) that will consider three or four classes as the classifier.

7. REFERENCES:

[1] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, "The Pascal visual object classes (VOC) challenge", International Journal of Computer Vision, Vol. 88, No. 2, 2010, pp. 303-338.

[2] O. Russakovsky, J. Deng, Su H., J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, "Imagenet large-scale visual recognition challenge", International Journal of Computer Vision, Vol. 115, No. 3, 2015, pp. 211-252.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20-25 June 2009, pp. 248-255.

[4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, "Learning deep features for scene recognition using places database", Proceeding of the 27th International Conference on Neural Information Processing Systems, Vol. 1, 8 December 2014, pp. 487-495.

[5] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, "Places: A 10 million image database for scene recognition", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 40, No. 6, 2017, pp. 1452-1464.

[6] J. C. Rangel, M. Cazorla, I. García-Varea, J. Martínez-Gómez, E. Fromont, M. Sebban, "Scene classification based on semantic labeling", Advanced Robotics, Vol. 30, No. 11-12, 2016, pp. 758-769.

[7] I. Kostavelis, A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," Robot Autonomous Systems", Vol. 66, 2015, pp. 86-103.

[8] S. Wang, Y. Li, W. Liu, "Multi-class weather classification fusing weather dataset and image features", Proceedings of the CCF Conference on Big Data, Springer, Xi'an, China, 11-13 October 2018, pp. 149-159.

[9] I. Sikirić, K. Brkić, P. Bevandić, I. Krešo, J. Krapac, S. Šegvić, "Traffic scene classification on a representation budget", IEEE Transaction on Intelligent Transportation System, Vol. 21, No. 1, 2019, pp. 336-345.

[10] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets robotics: The KITTI dataset", International Journal of Robotics Research, Vol. 32, No. 11, 2013, pp. 1231-1237.

[11] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 770-778.

[12] V. O. O. Castelló, I. S. S. Igual, O. Del Tejo Catalá, J. C. Perez-Cortes, "High-Profile VRU Detection on

Resource-Constrained Hardware Using YOLOv3/v4 on BDD100K", Journal of Imaging, Vol. 6, No. 12, 2020, p. 142.

[13] A. Boukerche, Z. Hou, "Object Detection Using Deep Learning Methods in Traffic Scenarios", ACM Computing Surveys, Vol. 54, No. 2, 2021, pp. 1-35.

[14] M. Mobahi, S. H. Sadati, "An Improved Deep Learning Solution for Object Detection in Self-Driving Cars", In Proceedings of the 28th Iranian Conference on Electrical Engineering, Tabriz, Iran, 4-6 August 2020, pp. 5-9.

[15] H. F. Yoshi, T. Hirakawa, T. Yamashita, "Deep Learning-Based Image Recognition for Autonomous Driving", IATSS Research, Vol. 43, No. 4, 2019, pp. 244-252.

[16] J. Chen, T. Bai, "SAANet: Spatial Adaptive Alignment Network for Object Detection in Automatic Driving", Image and Vision Computing, Vol. 94, 2020, p. 103873.

[17] C. R. Kumar, "A Comparative Study on Machine Learning Algorithms Using Hog Features For Vehicle Tracking Furthermore Detection", Turkish Journal of Computer and Mathematics Education, Vol. 12, No. 7, 2021, pp. 1676-1679.

[18] N. A. M. Mai, P. Duthon, L. Khoudour, A. Crouzil, S. A. Velastin, "3D Object Detection with SLS-Fusion Network in Foggy Weather Conditions", Sensors, Vol. 21, No. 20, 2021, p. 6711.

[19] M. Mirza, C. Buerkle, J. Jarquin, M. Opitz, F. Oboril, K. U. Scholl, H. Bischof, "Robustness of Object Detectors in Degrading Weather Conditions", Proceedings of the IEEE International Intelligent Transportation Systems Conference, Indianapolis, IN, USA, 19-22 September 2021.

[20] G. Al-refai, M. Al-refai, "Road Object Detection Using Yolov3 and KITTI Dataset", International Journal of Advance Computing Science and Applications, Vol. 11, 2020, pp. 1-7.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems, Vol. 25, No. 2, 2012, pp. 1097-1105.

[22] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, A. Mouzakitis, "A Survey on 3D Object Detection Methods for Autonomous Driving Applications", IEEE Transaction on Intelligent Transportation Systems, Vol. 20, No. 10, 2019, pp. 3782-3795.

[23] B. Lidestam, B. Thorslund, H. Selander, D. Näsman, J. Dahlman, "In-Car Warnings of Emergency Vehicles Approaching: Effects on Car Drivers' Propensity to GiveWay", Frontiers in Sustainable Cities, Vol. 2, 2020, p. 19.

[24] A. Agafonov, A. Yumaganov, "3D Objects Detection in an Autonomous Car Driving Problem", Proceedings of the International Conference on Information Technology and Nanotechnology, Samara, Russia, 26-29 May 2020, pp. 1-5.

[25] M. Alqarqaz, M. B. Younes, R. Qaddoura, "An object classification approach for autonomous vehicles using machine learning techniques", World Electric Vehicle Journal, Vol. 14, No. 2, 2023, p. 41.

[26] S. A. Khan, H. J. Lee, Huhnkuk, "Enhancing object detection in self-driving cars using a hybrid approach", Electronics, Vol. 12, No. 13, 2023, p. 2768.

[27] J. Ni, K. Shen, Y. Chen, W. Cao, S. X. Yang, "An improved deep network-based scene classification method for self-driving cars", IEEE Transaction on Instrumentation and Measurement, Vol. 71, 2022, p. 5001614.

[28] R. Prykhodchenko, P. Skruch, "Road scene classification based on street-level images and spatial data", Array, Vol. 15, No. 2, 2022, p. 100195.

[29] X. Jia, Y. Tong, H. Qiao, M. Li, J. Tong, B. Liang, "Fast and accurate object detector for autonomous driving based on improved YOLOv5", Scientific Reports, Vol. 13, 2023, p. 9711.

[30] Y. Li, J. Wu, H. Liu, J. Ren, Z. Xu, J. Zhang, Z. Wang, "Classification of Typical Static Objects in Road Scenes Based on LO-Net", Remote Sensing, Vol. 16, No. 4, 2024, p. 663.

[31] G. Dogan, B. Ergen, "A new CNN-based semantic object segmentation for an autonomous vehicle in urban traffic scenes", International Journal of Multimedia Information Retrieval, Vol. 13, No. 11, 2024.

[32] A. Chaudhari, "Smart traffic management of vehicles using faster RCNN based deep learning method", Scientific Reports, Vol. 14, 2024, p. 10357.

[33] J. Guo, J. Wang, H. Wang, B. Xiao, Z. He, L. Li, "Research on Road Scene Understanding of Autonomous Vehicles Based on Multi-Task Learning", Sensors, Vol. 23, No. 13, 2023, p. 6238.

[34] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, "BDD100k: A diverse driving dataset for heterogeneous multitask learning", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13-19 June 2020, pp. 2636-2645.

[35] S. Goferman, L. Zelnik-Manor, A. Tal, "Context-aware saliency detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 10, 2012, pp. 1915-1926.

[36] R. McCall et al. "A taxonomy of autonomous vehicle handover situations", Transportation Research Part A, Policy and Practice, Vol. 124, 2019, pp. 507-522.

[37] C. Szegedy et al. "Going deeper with convolutions", Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015, pp. 1-9.

[38] S. K. Saroj, V. Ratna, R. Kumar, N. P. Singh, "Efficient Kernel-based Matched Filter Approach for Segmentation of Retinal Blood Vessels", Solid State Technology, Vol. 63, No. 5, 2020, pp. 7318-7334.

[39] M. Wulandari, R. Chai, B. Basari, D. Gunawan, "Hybrid Feature Extractor Using Discrete Wavelet Transform and Histogram of Oriented Gradient on Convolutional-Neural-Network-Based Palm Vein Recognition", Sensors, Vol. 24, No. 2, 2024, p. 341.

[40] P. Lakshmi, M. Sivagami, "LT-LBP-Based Spatial Texture Feature Extraction with Deep Learning for X-Ray Images", Journal of Computer Science, Vol. 20, No. 1, 2024, pp. 106-120.

[41] C. I. Ossai, N. Wickramasingha, "GLCM and statistical features extraction technique with Extra-Tree Classifier in Macular Oedema risk diagnosis", Biomedical Signal Processing and Control, Vol. 73, 2022, p. 103471.

[42] D. R. Sulistyaningrum, T. Ummah, B. Setiyono, D. B. Utomo, Soetrisno, B. A. Sanjoyo "Vehicle detection using histogram of oriented gradients and real Adaboost", Journal of Physics: Conference Series, Vol. 1490, 2020.

[43] M. D. Narlawar, D. J. Pete, "Occluded Face Recognition: Contrast correlation & edge preserving enhancement based optimum features on CelebA dataset", Journal of Harbin Engineering University, Vol. 44, No. 8, 2023, pp. 1192-1204.

[44] J. Shi, H. Zhu, Y. Li, Y. Li, S. Du, "Scene classification using deep networks combined with visual attention", Journal of Sensors, Vol. 2023, No. 1, 2023. (retracted)

Application of Artificial Vision Based on Convolutional Neural Networks for Predictive Detection of Faults in Electrical Distribution Line Insulators

Original Scientific Paper

Vicente Paul Astudillo
Instituto Tecnológico Superior Rumiñahui,
Department of Electrical
Rumiñahui, Ecuador
paul.astudillo@ister.edu.ec

Pablo Catota-Ocapana*
Instituto Tecnológico Superior Rumiñahui,
Department of Electrical
Rumiñahui, Ecuador
pablo.catota@ister.edu.ec

*Corresponding author

Abstract *– Insulators play a crucial role in transporting and distributing electrical energy. They separate the energized conductor from the metal structure and support the conductors against adverse weather conditions such as winds and rains. However, these devices lose their insulating and mechanical properties when exposed to climatic factors such as sun exposure, rain, dust, and environmental pollution. This is due to the forming of a cover of organic matter and breaks and fissures, which can trigger adverse effects such as generating electric arcs. For this reason, it is essential to identify these failures effectively. In this research, an innovative solution is proposed that involves the use of artificial vision integrated into uncrewed vehicles, using the YOLOv5 object detection technology based on convolutional neural networks, to analyze 3000 images of the insulators in search of signs of deterioration, such as the presence of organic matter, breaks or cracks. The results showed an accuracy of over 90% in detecting failures. Deploying YOLOv5 alongside an uncrewed vehicle allows for faster and more accurate inspection of insulators along power distribution lines in real-time. Furthermore, by using this artificial vision technology, detailed data on the condition of the insulators can be collected in an automated manner, which facilitates the planning of preventive and corrective maintenance actions. This not only reduces the costs associated with the maintenance of distribution lines but also contributes to improving the reliability and efficiency of the electrical system.*

Keywords*: electrical energy, artificial vision, efficiency, mechanical properties*

Received: May 27, 2024; Received in revised form: November 5, 2024; Accepted: November 8, 2024

1. INTRODUCTION

The conventional electrical system comprises three main stages: generation, transmission, and distribution. The distribution stage is crucial since it brings electrical energy to commercial and residential users [1], [2]. This system comprises structures (poles and towers) that support the conductors, which have metallic elements directly connected to the ground. Consequently, insulating elements are required to prevent leakage currents caused by surges or atmospheric discharges [3], [4], [5]. Insulators are essential components in distribution and transmission lines. Therefore, it is vital to carry out maintenance work because exposure to various weather factors can reduce their useful life, such as industrial pollution, dust, and acid rain. [1], [6], [7]. These factors cause the loss of its insulating characteristics, and due to this, there is a risk of leakage currents and unwanted electric arcs, which are one of the causes that cause the flow of electrical energy to be cut off. This would affect residential and large customers and the operation of the electrical power system. Therefore, it is essential to detect faults in the insulator.

Currently, different techniques are used for the inspection of insulators; for example, in remote locations, the inspection of insulators is carried out through visual observation and the operator's judgment [5], [8], [9]. On distribution lines, basket cars are used whenever there is a nearby access road, while on transmission lines, the inspection is carried out when the operator climbs the tower [10], [11], [12]. These activities require considerable time, especially considering the distribution lines in rural areas that are inaccessible to the use of the basket car. This approach is inefficient due to the travel time of the operators and the necessary equipment, as well as its economic implications. It is essential to highlight that the high number of insulators in the distribution lines makes continuous inspection of these elements complex [10], [11], [13]. Therefore, maintenance takes a corrective approach in most cases, with inspections performed following a catastrophic insulator failure [7], [14].

The electrical insulator must have highly resistive properties so as not to allow the circulation of electric current. Among the materials with these characteristics are porcelain, glass, and teflon. Currently, porcelain is one of the most used materials in the manufacture of insulators for distribution lines due to its outstanding dielectric properties [15], [16], [17] and its low production cost. There are various types of insulators, such as rigid or pin type and suspension type, the latter being the most commonly used in distribution lines [18].

Advanced techniques based on artificial intelligence (AI) are some of the tools developed to monitor insulator status, improve fault detection, and reduce operator dependence. The use of vision artificial intelligence for analyzing images provided by uncrewed vehicles is presented as an innovative solution. However, creating a database is a complex task that requires many images representing the insulators' possible states that could cause faults in the distribution lines. In addition, it is crucial to consider the limitations, such as the safety distances from the energized conductors. According to the Arconel regulation 001/018 [19], it is established that for a voltage level of 13.8 kV, a minimum distance of 6 meters must be maintained to avoid electromagnetic interference with the operation of the drone. The advancement of AI has improved the accuracy and efficiency of image analysis, especially with the use of deep learning (DL). Training data is collected and analyzed to diagnose the insulators' condition, significantly reducing analysis time.

1.1. RELATED WORK

The YOLOV5 algorithm (You Only Look Once) has fast detection characteristics and high precision; this is a technique that is used in artificial vision [20], [18], [21]. This technique uses various sampling methods such as residual blocks, bounding boxes, loss function, and non-maximum suppression. This algorithm allows for extracting the essential characteristics and giving a prediction with a high percentage of precision. This data analysis technique has been used in several investigations. In [15], they proposed a model based on YOLOv5, which increases speed compared to previous versions by using complex backgrounds and the variation of the loss function. Analyzing insulators allowed a significant increase in fps (Frames Per Second). Compared to the original Yolo algorithm. In [22], they developed a model with RCNN, an RPN (Regional Proposal Network) model; this algorithm allows analyzing images more efficiently to detect objects' characteristics by adding a convolutional layer, and the precision improves considerably.

In [23], a model based on YoloV3 is developed, which uses a convolutional neural network composed of 53 layers to detect each of the images in the database in order; this allows increasing precision and detecting the desired characteristics. In [8], a study of the effects of insulators was developed using a Fast R-CNN model, which is based on three stages of feature extraction, analysis, and search in the regression layer, which allowed the detection speed to be improved at approximately 40 fps. In [24], an OTSU segmentation method is used to detect insulator failures, separating the pixels of the objects within the image and the background pixels, allowing the images to be decomposed and their characteristics better analyzed. In [10], an InsuDet model is proposed based on a feature pyramid (FPN), which contains multi-scale feature maps from an input image, which is especially useful for detecting objects of different sizes within the same image. In [25], an analysis of insulators in a transmission line is conducted using the convolutional neural network ResNeSt (Residual Network with Split-Attention) to predict the damage that insulators sustain due to ultraviolet (UV) radiation. Table 1 shows the precision and sensitivity of each classifier mentioned above.

Table 1. Evaluation of classifier types

Model	Accuracy (%)	Sensitivity (%)
ResNeSt [26]	94.2	93.4
Fast R-CNN [13]	92.1	81.1
YoloV3 [27]	92.67	86.10
YoloV5 [9]	96.47	89.2
InsuDet [10]	71.5	64.05

When analyzing the artificial vision methods suitable for detecting faults in insulators, it is observed that there are several options, but each of them has advantages and disadvantages; therefore, in this work, it was decided to use YOLOV5, taking into consideration the multiple benefits it has, the main benefit is its pro-

cessing speed that can reach up to 45 fps. The deep learning object detection algorithm should be able to eliminate complex background interference when imaging. Considering the complex background of image data, we integrate the HorBlock module into the original base network to improve the network's ability to extract each image's features and increase the network's detection accuracy for minor insulator defects. Additionally, the algorithm's precision is a great advantage when analyzing the insulators in real-time, as it helps determine what type of maintenance the operator should perform (corrective, preventive, or none of the above).

1.2. WORK ORGANIZATION

To address the problems raised, a series of stages will be developed, allowing the failure detection process in the insulators to be carried out more simply. The stages contemplated in this investigation are described below.

- A database covering the central states of the insulators of the distribution lines in the cities of Salcedo and Latacunga was created. Images were captured using a drone in diverse environments, with different backgrounds and angles, to obtain a varied sample that faithfully represents actual conditions. This database comprises 3,000 images collected over three months to validate the possible changes that may occur in each of the elements every quarter.
- To optimize feature detection, a convolutional layer was incorporated into the deep learning process. This addition allows for exhaustive analysis of the images, facilitating the identification of potential states and improving the model's accuracy. The convolutional layer significantly improves the efficiency of feature extraction from input images, resulting in more efficient hierarchical learning and improved object detection at various contexts and scales. Furthermore, by learning relevant rather than specific features from the training images, the model better generalizes to new images in situations other than those in the training.
- The neural network is designed to analyze videos in real-time as static images of insulators. This allows analysis to be carried out remotely on distribution lines, contributing to a faster and more efficient detection rate.
- To validate the results, a confusion matrix will be used, a tool that allows the validation of the precision and sensitivity of the algorithm in various environments, backgrounds, and environments to which the distribution lines are subjected.

2. MATERIALS AND METHODS

The approach proposed in this study uses a drone to capture and transmit video and images of the porcelain insulators used in the 13.8 kV distribution lines. Using this method takes advantage of the advantages of drones for aerial inspection in an agile and precise way, allowing information on the status of the insulators to be obtained. When the data is collected, it is sent to a computer through a Wi-Fi communication network [12], where the fault detection model (YOLOV5) is executed, and the results are displayed. In addition, the model includes an offline mode that allows you to analyze videos and images stored on physical media. It is important to note that remote inspection provides immediate information on the status of the insulator. It is worth mentioning that using the offline model can improve the effectiveness of stopping the fault (Fig. 1).

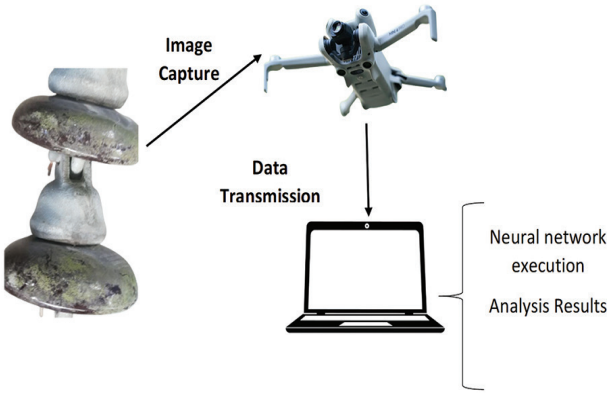


Fig. 1. Insulator Failure Detection System

The insulators analyzed using the proposed algorithm will be classified into three states: good, dirty, and broken. An insulator is considered good when it does not present any damage and, therefore, does not require any maintenance. On the other hand, an insulator will be considered dirty when it has a layer of plant and industrial matter, suggesting the need to carry out preventive and corrective maintenance. As for a broken and cracked insulator, it is an element that must be replaced immediately due to the decrease in its resistance, which compromises its integrity and considerably increases the risk of catastrophic failure.

2.1. DATASET COMPILATION

The experimental data for creating the dataset was collected in the province of Cotopaxi. The electrical distribution infrastructure covers an area of 6172.32 km [28] and is managed by Empresa Eléctrica Cotopaxi SA. The study focused specifically on the cities of Salcedo and Latacunga. The data set contains various types of porcelain insulators and chains of two or a maximum of three links at different angles. The images also present different backgrounds due to factors external to the insulators, such as conductors, poles, structures, vegetation, etc. The sample of the insulators analyzed can be seen in Fig. 2. The data comprises a total of 3000 images described in Table I. The resolution of the images is 1280x720 pixels to improve the fps rate.

Fig. 2. Insulator Samples

Table 2. Condition of the Insulators

State	Characteristics	Label	Number of samples
Good Insulator	Insulator in optimal conditions	Well	1000
Dirty insulator	Insulators with the presence of organic and inorganic dirt	Dirty	1000
Broken Insulator	Insulator with breaks and cracks	Broken	1000

2.2. ARCHITECTURE OF THE PROPOSED MODEL

YOLOv5 comprises 24 convolutional layers organized into three sections: extraction layer, fusion layer, and prediction layer. Added to these are two fully connected two layers, which allows for a latency of less than 25 ms [29]. The structure of the model is illustrated in Fig. 3; the input image enters the extraction layer through the "focus" module, which is responsible for dividing the original image and reconstructing it with a resolution of 418x418. Within the fusion layer, information from different levels of the network is combined through the "upsampling" process, increasing the resolution of the characteristics of the higher levels of the network so that it is coupled to the lower levels. In contrast, concatenation combines features to obtain a richer representation of information. The prediction layer uses the data from the fusion layer to generate object detections. Finally, the two fully connected layers are responsible for classifying the detected objects and determining their coordinates, along with the class of the detected object.

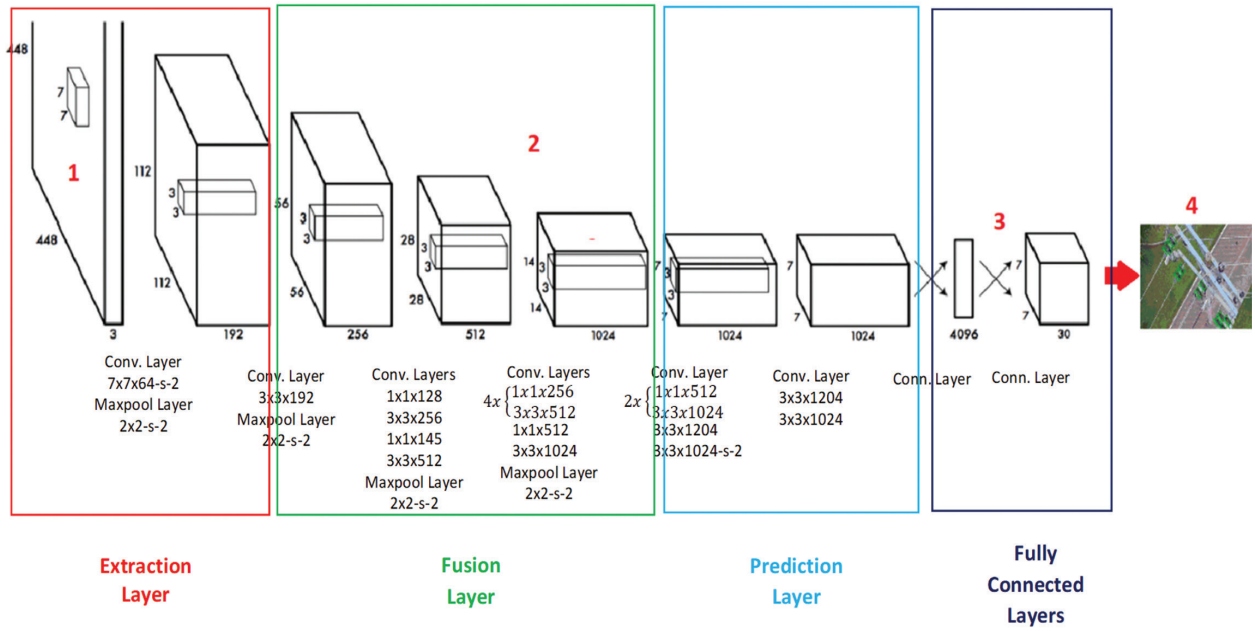


Fig. 3. Yolo architecture

Detection is based on probabilities, where the input image is divided into an SxS grid, generating bounding boxes with different sizes depending on the number of pixels in the input image. YOLO analyzes each bounding box to obtain the probability that they contain an object, predicting the possible classes. The object delimitation process is made up of five elements:

$$(x, y, w, h, k) \quad (1)$$

- (x, y) = Coordinates on the x, y axes.
- (w, h) = Size of the object.
- k = Value of perdition.

When generating the predictions by the neural network, the non-maximum suppression (NMS) algorithm is applied to eliminate overlapping and redundant boxes.

The bounding boxes are sorted according to their confidence score; the box with the highest score will be selected as the final box, along with the insertion over join (IoU) ratio [21], that exceeds the defined threshold. Finally, the final bounding boxes are obtained for each object detected in the image, along with its class. The probability of the class is given by the following:

$$Pc = Pr(Class) \cdot Pr(Obj) \cdot IoU \quad (2)$$

- Pr (Class) = Class Probability
- Pr (Obj) = Object Probability
- IoU = IoU value
- Pc = confidence scores

The class to which the object belongs can be predicted by obtaining a high Pc value. Subsequently, convolutional layers are applied to increase the resolution of the grids and improve the attributes through the use of the Local Binary Pattern (LBP). LBP is a texture descriptor used for attribute refinement, calculating the difference between the pixel and the values of neighboring pixels. The result is binarized to create a pixel texture with the best possible resolution. The refining is based on the following equation:

$$LBP(bx, by) = \left(\sum_{N=0}^{N-1} 2^p \; s(i_p - i_c) \right) \quad (3)$$

- (bx, by) = Location of the central pixel
- Ic = Gray Intensity Value
- Ip = Gray intensity value of the adjacent pixel
- N = Number of pixels in the image
- S = Activation function

With this approach, the aim is to improve detection accuracy, reduce possible false positives, and increase detection speed. To complement YOLO, neural networks present an architecture with intermediate layers that process information to extract specific features, from basic details such as edges and colors to the most complex textures between the input and output layers. These middle layers allow the neural network to learn increasingly complex representations of the input data as you go deeper into the network. Information flows through multiple layers, and each layer processes the information to extract specific features. The neural network becomes deeper as more intermediate layers are added, allowing it to learn increasingly abstract and complex features from the input data. By combining it with a CNN classifier, YOLO increases its prediction efficiency. Structures based on Residual Neural Network ([20]ResNet) are considered.

2.3. TRAINING OF THE PROPOSED MODEL

For the analysis and detection of failures, the total dataset images were taken and divided into a training set of 2100 images (70%) and a validation set of 900 images (30%) [23]. The selection of images was developed randomly to guarantee the diversity of both sets. Additionally, 1000 images of the training techniques were used for model validation. The evaluation algorithm was developed in Python. Google Colab was used for the training process. Google Colab is a free platform that allows you to run Python code and other programming languages directly from your web browser. Colab is based on Google Cloud and provides free access to hardware resources such as CPU, GPU, and TPU. These resources allow us to accelerate code execution and reduce training time, which had 2500 iterations, to minimize the error as much as possible. Four indices of the potential results obtained in the classification will be presented to evaluate the efficiency and precision of the algorithm. To validate the image classification methods using computer vision, a confusion matrix is used [22], [23], [25] as shown in Table 3.

Table 3. Confusion Matrix

Real value	Prediction	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

This matrix presents a synthesis of the results obtained through the algorithm, classifying the results into four categories that describe the agreement between the prediction and the actual state of the analyzed object. It is defined as true positive (TP) when the algorithm's prediction and the actual state of the object are positive, and false positive (FP) when the prediction is true. Still, the exact value of the object is negative, true negative (VN) when the prediction and the actual value of the object are also false, and false negative (FN) when the algorithm predicts negative. Still, the actual value of the object is positive.

By analyzing these confusion matrix values, the algorithm's performance in classifying different classes of images is evaluated, and possible classification errors are identified. These states are fundamental aspects of calculating model metrics, such as Precision (P), Sensitivity (R), and average sensitivity (mAP). These metrics provide a quantitative evaluation of the algorithm's classification performance, contributing to a deep understanding of the reliability and effectiveness in application environments.

$$P = \left(\frac{TP}{TP + FP} \right) \cdot 100\% \quad (4)$$

$$R = \left(\frac{TP}{TP + FN} \right) \cdot 100\% \quad (5)$$

$$AP = \int_0^1 P(r)\, dr \quad (6)$$

$$mAP = \sum_{n=1}^{N} \frac{AP(n)}{n} \quad (7)$$

For the calculation and subsequent results of the variables, a validation set of 900 images containing different states of insulators, utterly different from the training dataset, will be used. The set of images was analyzed image by image, filling the confusion matrix based on the algorithm's classification results. When studying each one, it is filled according to the classification described above within the confusion matrix,

which allows us to validate the precision and sensitivity of the algorithm. If the validation is unsatisfactory, retraining can be performed to improve performance.

Fig. 4 and Fig. 5 present the results of training the model using YOLOV5 to validate the insulator states. It was divided into two spaces for each evaluation according to the number of epochs within the training. Since the model converges in 1480 epochs, the results are presented up to that point. Regarding precision, it is verified that, for the validation data, the model achieves results above 92% from the one hundred and fortieth epoch of training, reaching a maximum of 93%, which validates the model's speed to obtain an adequate result. Regarding sensitivity, the total number of correct detections in percentage is validated, which reached a maximum value of 92% during training.

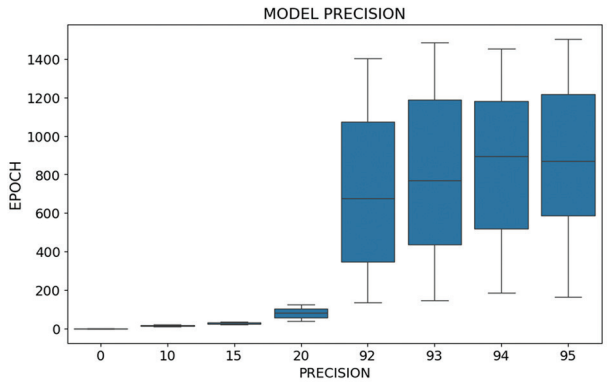


Fig. 4. Comparison of metrics during training (Model precision)

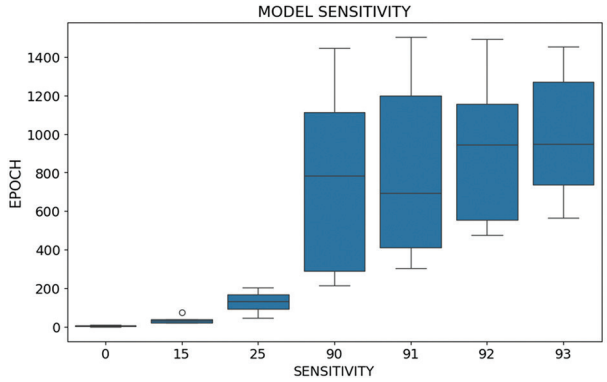


Fig. 5. Comparison of metrics during training (Model sensitivity)

3. EXPERIMENTAL RESULTS AND DISCUSSION

To validate the proposed algorithm, an evaluation was conducted on a 13.8 kV distribution network, different from the original sample, to verify its efficiency and sensitivity in detecting insulators in various environments and backgrounds. In Fig. 6, several results obtained in other posts of the test line are presented, where insulators in good condition can be observed, while, to a lesser extent, insulators that need maintenance.

The advantage of the proposed algorithm lies in its ability to automatically identify various states of the insulators, for which a color code has been assigned that facilitates its recognition, in addition to its identification label using VOC-Pascal.

Fig. 6. Detection Result

The proposed algorithm's efficiency was analyzed using the anchor box configuration and the loss function selection for the proposed states. The results obtained in the distribution line are presented in Table 4. Within the test, 1000 insulators were found.

Table 4. Confusion Matrix

True class	Well	Dirty	Broken
Well	**325**	fifteen	4
Dirty	eleven	**315**	4
Broken	4	4	**318**

Predicted class

When analyzing the confusion matrix, it is observed that % of the one thousand insulators identified in the distribution line sample, 32.5%, are insulators in good condition. In contrast, 31.5% were dirty insulators, for which it is necessary to perform preventive maintenance due to the high probability of failure. In comparison, 31.8% were classified as broken and cracked insulators, indicating the need for immediate corrective maintenance. Additionally, the algorithm presented several discrepancies in its predictions, with 4.2% of the test data classified incorrectly. This can be seen in Fig. 7, which shows the distribution of the insulators found in each kilometer of the test line.

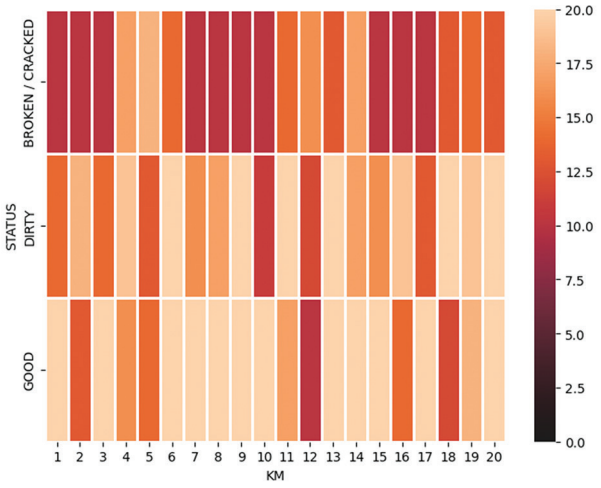


Fig. 7. Distribution of the insulators found

Finally, the precision and sensitivity of the detection of the distribution line insulators are calculated, as shown in Table 5. In it, an accuracy of 93% and an approximate sensitivity of 90% is observed. It is important to note that these parameters may vary depending on the drone's positioning angle and the isolator environment's lighting conditions.

Table 5. Algorithm Evaluation

State	Precision	Sensitivity	mAP
Well	0.94	0.91	0.90
Dirty	0.93	0.90	0.92
Broken	0.95	0.92	0.91

4. CONCLUSIONS

This article presents an improved method for detecting the state of insulators in 13.8 kV distribution lines using the YOLOv5 architecture. This approach can accurately and reliably measure insulators in various background contexts and viewing angles. An exhaustive experimental analysis shows that adding additional convolutional layers to YOLOv5 to generate the anchor boxes significantly improves the algorithm's accuracy and reliability in fault detection, reaching an accuracy rate of around 92%.

The average detection time per image for onboard processing is approximately two seconds, while for offline processing, it is one second. These processing times are considered in the context of an efficient flight route, considering both the drone's flight time and the number of distribution lines that must be analyzed.

This study highlights the importance of performing a detailed analysis of the different components of the electrical distribution system, especially porcelain insulators, which can be susceptible to various types of failures. The tests were carried out over three months, and the following results were obtained: 63.3% of the insulators had failures; the presence of tiny layers of contamination, plant matter, and small breaks and cracks in these insulators underlined the need for regular monitoring and maintenance.

Using the algorithm to detect the states of the distribution line insulators reduces the time necessary to identify possible problems and reduces the resources required for this task, which leads to significant savings for the company in charge of the management and maintenance of said distribution lines. With direction for future research, it is suggested that the algorithm be improved by integrating obstacle detection and avoidance capabilities and expanding the database to include polymer insulators to obtain a more robust and versatile model.

5. REFERENCES:

[1] Z. A. Siddiqui, U. Park, "A Drone Based Transmission Line Components Inspection System with Deep Learning Technique", Energies, Vol. 13, No. 13, 2020, p. 3348.

[2] S. F. Stefenon et al. "Fault detection in insulators based on ultrasonic signal processing using a hybrid deep learning technique", IET Science, Measurement & Technology, Vol. 14, No. 10, 2020, pp. 953-961.

[3] C. Liu, Y. Wu, J. Liu, Z. Sun, H. Xu, "Insulator Faults Detection in Aerial Images from High-Voltage Transmission Lines Based on Deep Learning Model", Applied Sciences, Vol. 11, No. 10, 2021, p. 4647.

[4] M. Liu, Z. Li, Y. Li, Y. Liu, "A Fast and Accurate Method of Power Line Intelligent Inspection Based on Edge Computing", IEEE Transactions on Instrumentation and Measurement, Vol. 71, 2022.

[5] H. Zhang et al. "Attention-Guided Multitask Convolutional Neural Network for Power Line Parts Detection", IEEE Transactions on Instrumentation and Measurement, Vol. 71, 2022.

[6] S. F. Stefenon et al. "Classification of insulators using neural network based on computer vision", IET Generation, Transmission & Distribution, Vol. 16, No. 6, 2022, pp. 1096-1107.

[7] J. Liu et al. "High precision detection algorithm based on improved RetinaNet for defect recognition of transmission lines", Energy Reports, Vol. 6, 2020, pp. 2430-2440.

[8] F. Li et al. "An automatic detection method of bird's nest on transmission line tower based on Faster_RCNN", IEEE Access, Vol. 8, 2020, pp. 164214-164221.

[9] Q. Li, F. Zhao, Z. Xu, J. Wang, K. Liu, L. Qin, "Insulator and Damage Detection and Location Based on YOLOv5", Proceedings of the International Conference on Power Energy Systems and Applications, Singapore, Singapore, 25-27 February 2022, pp. 17-24.

[10] X. Zhang et al. "InsuDet: A Fault Detection Method for Insulators of Overhead Transmission Lines Using Convolutional Neural Networks", IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021.

[11] Z. De Zhang et al. "FINet: An Insulator Dataset and Detection Benchmark Based on Synthetic Fog and Improved YOLOv5", IEEE Transactions on Instrumentation and Measurement, Vol. 71, 2022.

[12] Z. Liu, G. Wu, W. He, F. Fan, X. Ye, "Key target and defect detection of high-voltage power transmission lines with deep learning", International Journal of Electrical Power & Energy Systems, Vol. 142, 2022, p. 108277.

[13] W. Zhao, M. Xu, X. Cheng, Z. Zhao, "An Insulator in Transmission Lines Recognition and Fault Detection Model Based on Improved Faster RCNN", IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021.

[14] X. Huang, E. Shang, J. Xue, H. Ding, P. Li, "A Multi-feature Fusion-based Deep Learning for Insulator Image Identification and Fault Detection", Proceedings of IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, 12-14 June 2020, pp. 1957-1960.

[15] J. Zheng, H. Wu, H. Zhang, Z. Wang, W. Xu, "Insulator-Defect Detection Algorithm Based on Improved YOLOv7", Sensors, Vol. 22, No. 22, 2022, p. 8801.

[16] Q. Wen, Z. Luo, R. Chen, Y. Yang, G. Li, "Deep Learning Approaches on Defect Detection in High Resolution Aerial Images of Insulators", Sensors, Vol. 21, No. 4, 2021, p. 1033.

[17] W. Liu, Z. Liu, H. Wang, Z. Han, "An Automated Defect Detection Approach for Catenary Rod-Insulator Textured Surfaces Using Unsupervised Learning", IEEE Transactions on Instrumentation and Measurement, Vol. 69, No. 10, 2020, pp. 8411-8423.

[18] Z. Feng, L. Guo, D. Huang, R. Li, "Electrical Insulator Defects Detection Method Based on YOLOv5", Proceedings of IEEE 10th Data Driven Control and Learning Systems Conference, Suzhou, China, 14-16 May 2021, pp. 979-984.

[19] Agencia de Regulación y Control de Electricidad, "Regulación No. 001/18: Regulación de los precios de energía eléctrica para el servicio público de distribución", Quito, Ecuador, 2018.

[20] B. J. Souza, S. F. Stefenon, G. Singh, R. Z. Freire, "Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV", International Journal of Electrical Power & Energy Systems, Vol. 148, 2023, p. 108982.

[21] Z. Qiu, X. Zhu, C. Liao, D. Shi, W. Qu, "Detection of Transmission Line Insulator Defects Based on an Improved Lightweight YOLOv4 Model", Applied Sciences, Vol. 12, No. 3, 2022, p. 1207.

[22] S. Wang, Y. Liu, Y. Qing, C. Wang, T. Lan, R. Yao, "Detection of insulator defects with improved ResNeSt and region proposal network", IEEE Access, Vol. 8, 2020, pp. 184841-184850.

[23] M. W. Adou, H. Xu, G. Chen, "Insulator Faults Detection Based on Deep Learning", Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification, Xiamen, China, 25-27 October 2019, pp. 173-177.

[24] S. Fang, Z. Mingze, L. Sheng, W. Xiaoyu, C. Haiyang, "Fast detection method of insulator fault based on image processing technology", Proceedings of IEEE 5th Information Technology and Mechatronics Engineering Conference, Chongqing, China, 12-14 June 2020, pp. 400-406.

[25] J. Zhang, T. Xiao, M. Li, Y. Zhou, "Deep-Learning-Based Detection of Transmission Line Insulators", Energies, Vol. 16, No. 14, 2023, p. 5560.

[26] Z. Gao, G. Yang, E. Li, Z. Liang, "Novel Feature Fusion Module-Based Detector for Small Insulator

Defect Detection", IEEE Sensors Journal, Vol. 21, No. 15, 2021, pp. 16807-16814.

[27] H. Liang, C. Zuo, W. Wei, "Detection and Evaluation Method of Transmission Line Defects Based on Deep Learning", IEEE Access, Vol. 8, 2020, pp. 38448-38458.

[28] Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables, "Panorama Eléctrico 2024", https://controlrecursosyenergia.gob.ec/wp-content/uploads/downloads/2024/03/PanoramaElectricoXXI-Marzo-Baja.pdf (accessed: 2024)

[29] Y. Gu, P. Huang, J. Wang, L. Tang, J. Weng, X. Wang, "Insulator Defects Detection and Classification Method Based on YOLOV5", Proceedings of the 18th Annual Conference of China Electrotechnical Society, Nanchang, China, 15-17 September 2024, pp. 407-414.

A New Encryption Algorithm for Voice Messages on Social Media Using Magic Cube GF (2^8) Technology

Original Scientific Paper

Mohammed M. Al-Ezzi*
School of Computer Science and Engineering,
Central South University, China
mohd_soft2006@yahoo.com

Ministry of Higher Education and Scientific Research,
Baghdad, Iraq

Wang Weiping
School of Computer Science and Engineering,
Central South University, China
wpwang@csu.edu.cn

Abdul Monem S. Rahma
Computer Science Department
AL-Maarif University College, Iraq
Monem.rahma@uoa.edu.iq

Hasnain Ali Al mashhadani
School of Computer Science and Engineering,
Central South University, China
hasnainalmshhadani@gmail.com

Mazen R. Hassan
Department of Electrical Engineering Techniques,
Basrah Engineering Technical College,
Southern Technical University, Basrah, Iraq
mazen.hassan@stu.edu.iq

*Corresponding author

Abstract *– With the rise of multimedia technology, audio file encryption has become increasingly significant, especially for voice messages in popular social media applications like WhatsApp. Voice messages hold great social significance, and to ensure their security, they must be encrypted before being transmitted over the internet. This paper proposes an efficient algorithm to securely encrypt voice messages. The innovative algorithm is based on a magic cube to reduce the execution time of the advanced encryption standard (AES) cipher algorithm. This is achieved by replacing the MixColumn function with a 3 × 3 × 3 magic cube FG (2^8) irreducible polynomial. This work reduces the execution time of the AES cryptosystem and enhances complexity by utilizing additional keys generated by a 3 × 3 × 3 magic cube. to develop a block cipher algorithm that encodes audio files using two types of finite fields: GF (P) and GF (2^8). This algorithm places a key of three cells and a voice message of six cells on each face of a 3 × 3 × 3 magic cube. Time complexity and encryption quality are evaluated according to National Institute of Standards and Technology standards, and the differential attacks' peak signal-to-noise ratio is calculated. The total complexity achieved for both GF (P) =$256^9 \times 251^{18}$ and GF (2^8) = $256^9 \times 256^{18}$ is measured for comparison. Simulation results demonstrate a significant reduction in execution time and increased encryption complexity. Moreover, the magic cube with three faces (3 × 3 × 3) exhibits superior performance in terms of complexity and speed compared to the third-order magic square.*

Keywords: *GF (2^8) finite field, GF (P) finite field, Gaussian elimination, magic cube cryptography*

Received: March 23, 2024; Received in revised form: December 18, 2024; Accepted: December 9, 2024

1. INTRODUCTION

In today's world, when data is transferred between individuals, a high level of security is required. Cryptography focuses on the use of encryption and decryption algorithms to ensure private communication [1, 2, 3]. Due to the rapid development of the internet, wireless voice communication technology has become widely utilized. During the data transmission process, there is a risk of information leakage, making research on audio information encryption highly significant. Many chaos-based encryption algorithms, such as chaotic systems [4, 5] DNA coding, and classical logistic chaotic systems, are used to deal with speech data. In addition, traditional encryption algorithms, the advanced encryption standard (AES) [6-10], have been extensively employed in audio encryption and have yielded unsatisfactory results. AES encodes 128-bit plaintext blocks using master key blocks of 128, 192, or 256 bits. Accordingly, AES is referred to as AES-128, AES-192, and AES-256 based on the key sizes. Before generating the 128-bit ciphertext block, the plaintext block undergoes a predefined number

of rounds using the round function: 10 rounds for AES-128, 12 rounds for AES-192, and 14 rounds for AES-256. A 4 × 4-byte array can represent plaintext, cipher text, intermediate state blocks, and the primary key, which is implemented accordingly. The number of columns in the array is determined by dividing the key length by 32 [11]. The encryption method utilizes the following operations:

1. Sub-byte transformation or inverse sub-byte transformation: This technique involves a non-linear byte-to-byte transformation achieved through a multiplicative inverse operation followed by an affine transformation.
2. Shift rows: The shift row transformation is more significant than the initial arrangement because the state, cipher input, and output are treated as arrays of four 4-byte columns.
3. Mix columns: This operation analyses the state column-by-column, treating each column as a four-term polynomial.
4. Add round key: A round key is added to the state using an XOR operation.

The inverse cipher follows the same procedure as the encryption process but in the opposite direction. The inverse sub-byte transformation comes next after performing the Shift Rows operation in the opposite direction. next, the mixed columns operation is applied, and the add round key operation is performed. The resulting array (either plaintext or encrypted text) is obtained once these state operations are completed [12, 13]. Magic squares have a long history and have served various purposes. They have been the basis for many intelligence-testing games [14, 15]. A magic square of size, n × n is an arrangement of numbers from 1 to n^2 in a square such that the sum of every row, column, and diagonal is the same. Alpha magic squares consist of discrete words or numbers engraved or printed. They can be arranged vertically, horizontally, or diagonally to produce the same number or form the same words. When a dimension is added to a magic square [15-26], it becomes a "magic cube" towards computer ethics and information security, as computer security and computer ethics are important components of the management information system. The probability of constructing a magic cube is similar to that of constructing a magic square, especially for large values of n. Our proposed method addresses the weak delay [27] in the mix columns operation in the AES algorithm.

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 02030101 \\ 01020301 \\ 01010203 \\ 03010102 \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} S'1 \\ S'2 \\ S'3 \\ S'4 \end{bmatrix} = \begin{bmatrix} 0e0b0d09 \\ 090e0b0d \\ 0d090e0b \\ 0b0d090e \end{bmatrix} \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \end{bmatrix} \quad (2)$$

This multiplication operation can be computationally expensive, especially with large input matrices.

To overcome this, we introduce a 3 × 3 × 3 magic cube [22, 25, 28] in GF (2^8) irreducible polynomial [28-30], which allows for faster processing of the matrices [31]. This approach involves using a more complex key in GF $(2^8)^9$ and voice messages with two types, GF $(251)^{18}$ and $(2^8)^{18}$. The corresponding results demonstrate the new block cipher algorithm is proposed that utilizes three faces of a 3 × 3 × 3 magic cube irreducible polynomial instead of Mixcolumn. The technique proposal focuses on improving and enhancing the security of encrypted messages by increasing the complexity and decreasing encryption and decryption time. Hence, the protection of encrypted audio messages is effectively guaranteed. Thereby, it can be applied to secure and encrypt communications, messages, and voice messages on platforms that hold significant social importance. The contributions of this paper are as follows:

Development of a new and innovative symmetric block cipher algorithm.

- Utilize a magic cube with three faces and 27 cells, where the sum of each row, column, or diagonal is equal.
- The block cipher incorporates 9 keys and is represented by a system of 18 linear equations, solved using Gaussian elimination.
- The algorithm's key complexity enhances security and makes hacking more challenging.
- Implement the magic cube GF (2^8) algorithm to increase complexity and achieve high speed.
- The algorithm is computationally inexpensive.
- It provides a secure environment for transmitting and receiving audio files, specifically voice messages.

The rest of the paper is organized as follows: Section 1 provides an overview of related work. Section 2 describes the proposed methodology, while Section 3 describes the algorithm used to build the magic cube. Section 5 thoroughly evaluates the results, and finally, Section 6 concludes the paper.

2. RELATED WORKS

In [31], the cipher employs a substitution permutation network structure with six dimensions for parallel encryption of 128-bit data blocks in six directions, enhancing processing efficiency for large data volumes. The proposed multi-dimensional symmetric cipher algorithm focuses on encryption and does not address decryption procedures. However, the research paper does not provide information on the proposed algorithm's comparative analysis with other symmetric block ciphers to showcase its efficiency and effectiveness.

An advanced method named 3D-BERC for encrypting images in three dimensions, using Rubik's cube principles and bit-level encryption, was introduced in

[32]. This advanced encryption scheme ensures image security by implementing effective permutation and diffusion techniques to scramble and spread changes across cipher images. Experimental results and simulation analysis prove this. However, there is no mention of a comparative study with existing encryption schemes to demonstrate the superiority or effectiveness of the proposed 3D-BERC method.

The authors in [21] have proposed a data-hiding strategy that utilizes modification directions (EMD) to embed large payload information without causing distortion. The traditional EMD technique encodes a secret digit using the $(2n+1)$–ary system, with a maximum payload capacity of 1.161 bits per pixel (bpp). This study involves concealing a secret digit using the $(3n)$ 3–ary notational system. The secret digit is buried within a group of 3 pixels selected based on a random sequence. This process results in a payload of $log2(n)$. An increase in the dimension, n, of the neighborhood set leads to a higher payload. Nevertheless, the research primarily addresses the effectiveness of embedding in terms of efficiency and visual quality of the stego image without delving into a thorough analysis of the computational complexity or processing time involved in the embedding process.

In [20] introducing a cryptographic method known as the symmetric magic cube. This method focuses on constructing magic cubes of the form $m \times 2l$. It enables encryption and decryption of various types of images, including numeric digits and special characters, while also addressing the issue of repetition in the ciphertext. However, the proposed algorithm was applied only to medical images, and no other files were taken into consideration to demonstrate the validity of the application

The paper [33] proposes a lightweight image encryption algorithm based on Rubik's Revenge cube move patterns. The study primarily focuses on creating highly nonlinear S-boxes derived from the cube's permutations for improved security and effectiveness. However, the algorithm's efficiency in practice needs to be evaluated further to ensure it can handle real-world image encryption requirements.

The paper [34] presents a multiple remote sensing image (MRSI) encryption scheme that enhances salient image regions' security and transmission efficiency. The scheme uses a 4D-IDTLN chaotic system and knowledge-oriented and vision-oriented saliency techniques to create a mask contour positioning model (MCPM) and then fuses the MRSIs into a cube. The encryption is then further encrypted using closed-loop diffusion. The security of the proposed scheme is evaluated, showing higher security and better transmission efficiency. However, the paper does not address potential decryption methods or the process.

This paper [24] proposes a method for recognizing and restoring the color block of the magic cube using machine vision. The magic cube color block is recognized by machine vision, followed by image color space transformation and K-means clustering. The color block information is packaged and sent to the cube explore 5.14 through negotiation. This paper focused on a specific type of robot (a magic cube-solving robot), limiting the generalizability of the findings to other robotics applications involving color recognition and manipulation.

This paper [22] proposes a multiple remote sensing image (MRSI) encryption scheme based on saliency extraction and magic cube circular motion to improve salient regions' security and transmission efficiency in remote sensing images. The scheme provides two tiers of privacy protection for airport locations in the images. First, a 4D improved discrete tabu learning neuron (4D-IDTLN) chaotic system is proposed, which exhibits rich dynamic behaviors. Second, the salient regions of the images are classified and extracted using knowledge-oriented saliency (KOS) and vision-oriented saliency (VOS) techniques to create a mask contour positioning model (MCPM), which is then encrypted. The MRSIs are fused into a cube, encrypted using magic cube circular motion and chaotic sequences, and further encrypted using closed-loop diffusion. The security of the proposed encryption scheme is evaluated, indicating that it provides higher security and better transmission efficiency for MRSIs. However, the paper focuses on encryption techniques but does not delve into potential decryption methods or address the decryption process, which is crucial for understanding the complete security framework. Proposed Methodology

This paper addresses the security concerns related to voice messages on social media platforms. To tackle this problem, a new block cipher algorithm is proposed, which utilizes the three faces of a $3 \times 3 \times 3$ magic cube. The algorithm will be discussed in detail in the remaining sections.

3. PROPOSED ALGORITHM TO BUILD A MAGIC CUBE USING GF (2^8)

The audio file [35] media is divided into two main parts:

The first part is the structure, which contains information about the file, such as file type (single channel, dual channel, etc.), sampling rate, representation depth, file length (duration), and any other information that defines the file's properties [4].

The second part is the data containing the audio samples' numerical values. These samples are arranged chronologically according to the sampling rate, and each sample represents the sound level at a particular moment. The digital audio file is split into digital audio files [36]. The structure (header), which contains information about the file and the data, represents the audio's numeric values. These two parts go hand in hand and together form the digital audio file [4]. The proposed algorithm for encrypting the data of any audio file has preserved the file structure, and the algorithm's output is an encrypted audio file.

A finite field, commonly known as a Galois field, is named after the mathematician 'Evariste Galois', who discovered it. A restricted number of components characterizes finite fields and are increasingly used, particularly in translating computer data. The designated domains are extensively used in several scientific disciplines, including mathematics, programming, and number theory. The finite field was used for the decimal number, with the whole field constructed upon a prime number (P), represented succinctly as GF(P).

A polynomial is termed an "irreducible polynomial" if it cannot be expressed as the product of two or more lower-degree polynomials. Additionally, all non-zero elements possess a multiplicative inverse. Irreducible polynomials are utilized in various encryption methods, particularly in modern encryption techniques such as AES and elliptic curve cryptography.

A finite field has limited elements, and all outcomes of operations conducted inside it are within its range. We can execute mathematical operations for polynomial expressions like addition, subtraction, multiplication, division, and positive integer exponents.

The transition from decimal to polynomial numbers has occurred due to the need for modern electronics to operate on 8 bits. The coefficients of the polynomial may rely on 8 bits, represented as GF (2^8). To elucidate the need to transition to GF (2^8), using GF(P) will provide all resultant numbers that are less than the selected (P), given the present circumstances[11].

AES Where provides a convenient and efficient representation of operations on bytes. The arithmetic in GF (2^8) is typically implemented using a primitive polynomial of degree eight, which defines the structure of the field [11, 37-40].

A magic cube [21, 41, 42] is similar to a magic square[45], but it has multiple sizes, such as order 3, 4, 5, 6, 7, and so on. The size of a magic square determines the number of elements it contains, with larger sizes having more numbers.

In the AES algorithm, the mix columns transformation involves a computationally expensive multiplication operation, especially when dealing with large input matrices. To mitigate this, the multiplication operation can be replaced by utilizing a $3 \times 3 \times 3$ magic cube in GF (2^8), filled with the cells from 1 to 27.

3.1. DETAILS OF THE ALGORITHM

Based on the known characteristics of a $3 \times 3 \times 3$ magic cube, as shown in Fig. 1, the cube can be divided into three faces, each resembling a 3×3 square. This division allows us to obtain six equations for the rows, six for the columns, and six for the main and secondary diagonals. In total, we have 24 equations derived from the magic cube structure.

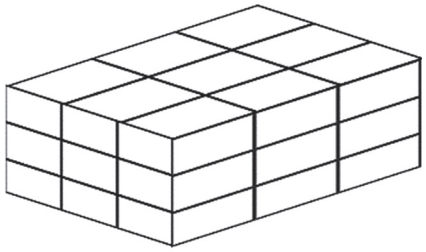


Fig. 1. Illustration of a $3 \times 3 \times 3$ magic Cube dissected into three faces

Face 1 of the magic cube consists of 1 to 9 cells. As a result, the message to be encrypted will contain only six cells corresponding to the number of equations derived from the rows, columns, and diagonals of face 1. The key will occupy the remaining three positions on face 1, as depicted in Fig. 2. Two equations are excluded from the resulting sums obtained from the equations. These excluded sums will be treated as encrypted voice messages, and their number will be comparable to the number of positions in the message (as shown in Figs. 2b and 2f).

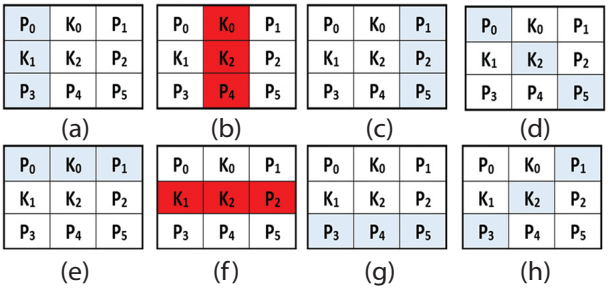


Fig. 2. Cube face 1. The highlighted sums are used in the magic cube algorithm

Face 2 of the magic cube consists of 1 to 9 cells. As a result, the message will contain six cells, equivalent to the number of equations. There will be three remaining positions designated for the key, as shown in Fig. 3. The resulting sums obtained from these equations will be treated as encrypted voice messages, and their number will correspond to the number of positions in the message. Two equations have been removed from the calculations (as depicted in Figs. 3j and 3n).

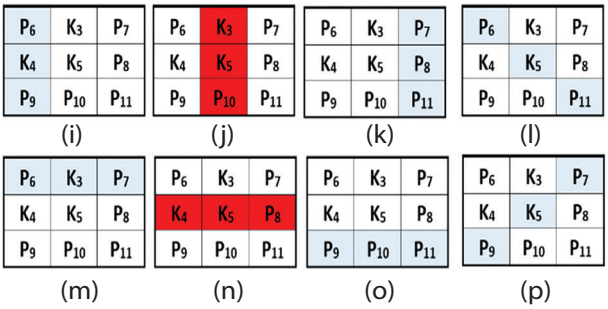


Fig. 3. Cube face 2. The highlighted sums are used in the magic cube algorithm.

Face 3 of the magic cube consists of 1 through 9 cells. Consequently, the message will consist of six cells, equal

to the number of equations, and the remaining key positions will only contain three cells, as illustrated in Fig. 4. The sums obtained from these equations will be treated as encrypted voice messages. Their number will match the number of message positions. The calculations have excluded two equations (as shown in Figs. 4r and 4w).

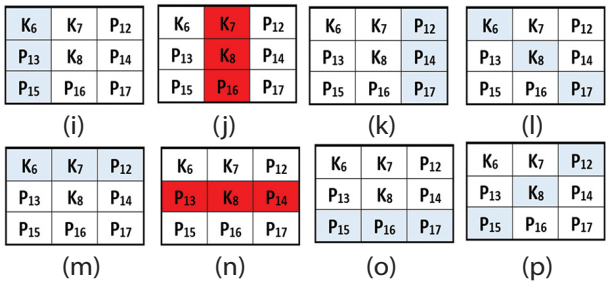


Fig. 4. Cube face 3. The highlighted sums are used in the magic cube algorithm

A total of 18 equations are to be employed in the proposed method. The positions and values of the keys within the magic cube are not restricted to fixed locations. Instead, a flexible selection process is employed, and the locations are expected to be chosen using GF (2^8), as shown in Fig. 5.

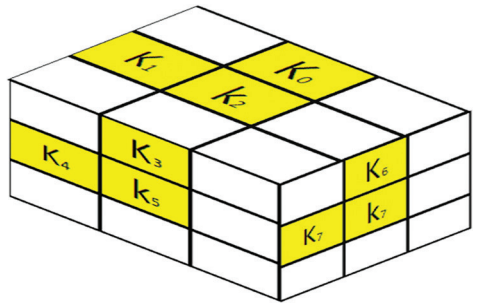


Fig. 5. The keys presumed selected for GF (2^8).

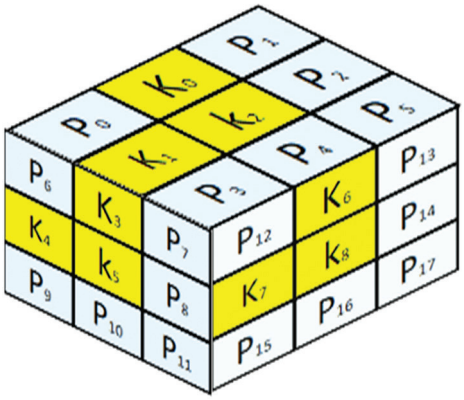


Fig. 6. The message on the created magic cube (colored sections) and keys (remaining areas)

Fig. 6 depicts the formation of 18 sites corresponding to 18 sums in 3 × 3 × 3 magic cubes after filling in the essential locations. Referring to Fig. 6, the sums of the 18 equations for each of the 3 faces are found, and the equations are formed as in (3).

$$
\begin{aligned}
&e1: P0 + K1 + P3 = Sum1\\
&e2: P1 + P2 + P5 = Sum2\\
&e3: P0 + K0 + P1 = Sum3\\
&e4: P0 + K0 + P1 = Sum4\\
&e5: P3 + P4 + P5 = Sum5\\
&e6: P1 + K2 + P3 = Sum6\\
&e7: P6 + K4 + P9 = Sum7\\
&e8: P7 + P8 + P11 = Sum8\\
&e9: P6 + K5 + P11 = Sum9\\
&e10: P6 + K3 + P7 = Sum10\\
&e11: P9 + P10 + P11 = Sum11\\
&e12: P7 + K5 + P9 = Sum12\\
&e13: K6 + P13 + P15 = Sum13\\
&e14: P13 + P14 + P17 = Sum14\\
&e15: K6 + K8 + P17 = Sum15\\
&e16: K6 + K7 + P12 = Sum16\\
&e17: P15 + P16 + P17 = Sum17\\
&e18: P12 + K8 + P5 = Sum18
\end{aligned}
\qquad (3)
$$

Consequently, we obtain 18 quanta representing the encoded voice message to be sent to the recipient. Each data packet will be encrypted using the magic cube algorithm. The keys will be placed in the agreed-upon positions on the recipient's end. In contrast, the remaining positions will remain unknown, with their number matching the number of encoded voice messages. The issue will be solved mathematically by arranging the elements so that the primary diameter does not equal zero at any of its coordinates. The computations and analyses in this work used the Gaussian elimination method to solve the equations and derive the communication.

After completing and implementing this work, a further development was made by replacing the phonetic letters GF (P) with the key complexity of GF (2^8). This modification was introduced because of the magic cube, as illustrated in Fig. 7.

Algorithm 1-a: The proposed algorithm for encryption symmetric cipher based on a magic cube.

Input: voice message, key values, and key positions.
Output: Cipher text.
Begin:
Step 1: Placing the key values in the agreed positions.
Step 2: The remaining positions are filled with the values of the message.
Step3: Find the final results for each equation of (1), of which there are 18 equations. The end result of the algorithm will be the encrypted voice message sent to the other party
End.

Algorithm 1-b: The proposed algorithm for decryption symmetric cipher based on a magic cube.

Input: Cipher text, key positions, and key value.
Output: voice message
Begin:
Step 1: The key value is placed at the agreed positions.

Step 2: The remaining positions will remain unknown. They will be found by solving 18 mathematical after arranging the equations by not making the main diagonal = 0. The final results will be the original data (voice message).

End.

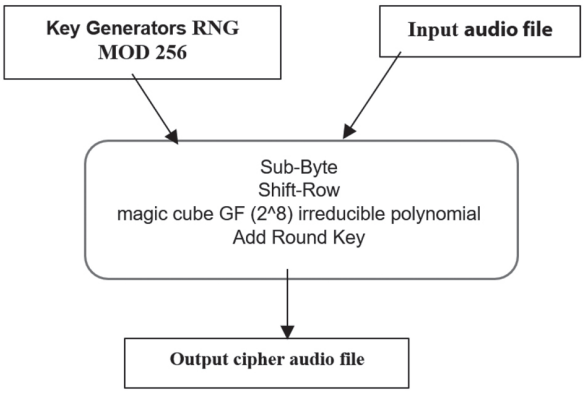


Fig. 7. Diagram replacement of a magic cube Instead of a mix column for the proposed AES technique for encrypting audio file

4. EVALUATION

Cryptology is concerned with encryption and decryption using specific algorithms. To ensure message integrity and security, the decryption process requires a private key that is kept confidential between the communicating parties. It is important to maintain the privacy of the key to protect the encrypted messages. This study will discuss the speed, complexity, and statistics measures defined by the (NIST), (PSNR) tests. These measures will be applied to analyze the performance of the proposed algorithm for voice messages (encryption and decryption). The statistics of the voice messages will be plotted and compared with earlier algorithms that operate on similar data types, considering both types of finite fields. The proposed algorithm was implemented using Python 3.10.5 and Jupyter Notebook Anaconda 3. The simulation was conducted with an Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz, 2.60 GHz, and 8.00 GB (7.88 GB usable) of RAM. The operating system used was a 64-bit version with x64-based processor.

4.1. COMPLEXITY OF THE KEY

Key complexity is a computed statistic that quantifies the difficulty of a brute-force attack on a cryptographic key. It indicates the number of attempts required to crack the key successfully through an exhaustive search.

Using GF (P), the solution to the $3 \times 3 \times 3$ magic cube is the value of the prime number employed raised to the power of the number of keys, represented by three keys per face:

For GF (2^8), the key's strength is 2^8 raised to the power of 3:

the complexity of face 1 key using GF (2^8) = 256^3 (7)

the complexity of face 2 key using GF (2^8) = 256^3 (8)

the complexity of face 3 key using GF (2^8) = 256^3 (9)

5.2. GENERALIZED COMPLEXITY OF THE PROPOSED SYSTEM

The data complexity of the proposed system is constant for both variants, GF (P) and GF (2^8). It is determined by the number of possible ASCII codes, which is 256, raised to the power of the number of message sites in the system, which is 18. This represents the total number of possible combinations of the message data. The total complexity of the system, when using GF (2^8) and keeping the keys secret, can be calculated as in (14):

complexity in face 1 using GF (P) = $256^3 \times 251^6$ (10)

complexity in face 2 using GF (P) = $256^3 \times 251^6$ (11)

complexity in face 3 using GF (P) = $256^3 \times 251^6$ (12)

total complexity using GF (P) = $256^9 \times 251^{18}$ (13)

total complexity using GF (2^8) = $256^9 \times 251^{18}$ (14)

The complexity of the $3 \times 3 \times 3$ magic cube system is compared with the system based on a magic square of order 3 (MS3) in Fig. 8, using both types of finite field.

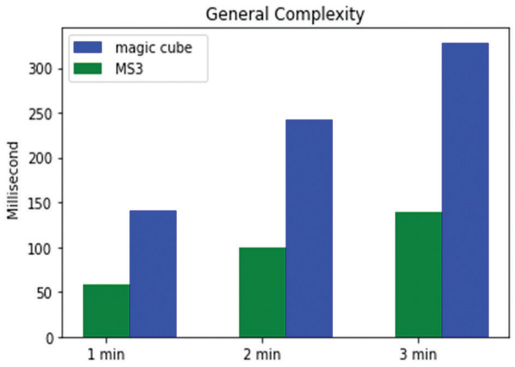


Fig. 7. The general complexity of the MS3 system and the $3 \times 3 \times 3$ magic cube system

4.2. EXECUTION TIME

The encryption and decryption times were measured for audio messages using GF (P) and GF (2^8), as shown in Table 1, and for voice messages using GF (p) and GF (2^8), as shown in Table 1.

Table 1 shows that the proposed algorithm is faster than the original. Using the finite field to type GF (P) and GF (2^8) irreducible polynomial gives a relatively good advantage as it gives faster execution. correspondingly, the original AES algorithm suffers from high calculation and computational overhead problems. However, the encryption and decryption process is equal in execution time. In the proposed algorithm, additional calculations are involved in performing Gaussian demodulation for decoding.

The corresponding plots illustrating the execution times can be found in Figs. (9-10). Furthermore, Figs.

(11-12) Compare the execution times of the proposed magic cube algorithm and the older MS3 method.

Table 1. Compare the execution time of the magic cube algorithm and the original AES algorithm for encrypting and decrypting voice

GF	Audio file size (MB)	Audio file duration (min.)	Basic AES encryption time (ms) Ref.[45]	Encryption time (ms)	Decryption time (ms)
GF (2^8)	1.61	1	61	140.670	142.560
GF (2^8)	2.76	2	82	241.392	243.490
GF (2^8)	3.90	3	14	328.218	330.115
GF (251)	1.61	1	61	184.394	186.415
GF (251)	2.76	2	82	316.028	326.567
GF (251)	3.90	3	14	362.494	365.631

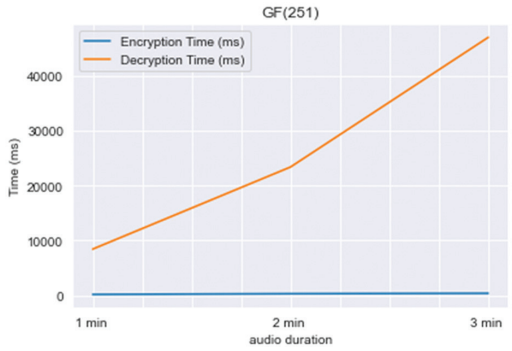


Fig. 9. Comparison of the execution time of the proposed algorithms using GF (P)

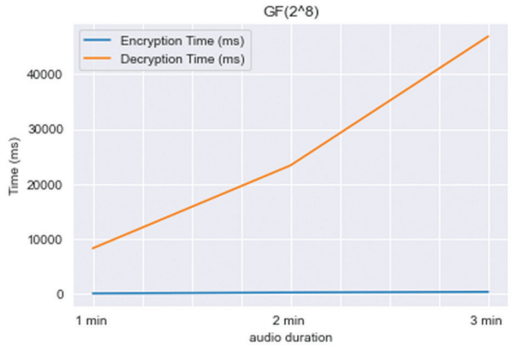


Fig. 10. Comparison of the execution time of the proposed algorithms using GF(2^8)

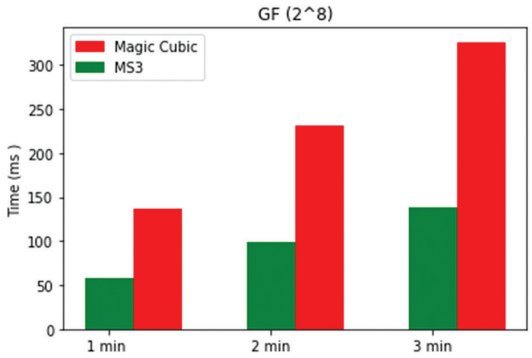


Fig. 11. Encryption time (m.s) and decryption time (m.s) when using the magic cube GF(P) algorithm for a voice

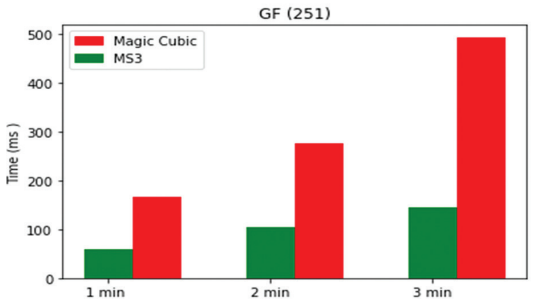


Fig. 12. A comparison in terms of execution time between MS3 and magic cube GF (2^8) for a voice

4.3. PSNR TEST

PSNR measures the ratio between the original signal and the encrypted signal. When it comes to encrypted audio, a lower PSNR [42] indicates a higher noise presence in the cryptogram. This means that the ciphertext is more resistant to attacks. The mathematical equation the PSNR [44], of the voice messages can be calculated using equation (15).

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (15)$$

Where MSE is the mean square error used to measure the error in voice messages, equation (16) calculates the MSE for voice messages.

$$MSE = \frac{1}{M \times N} \sum_{i,j} (A[i,j] - B(i,j)^2 \quad (16)$$

In this context, M and N denote the width and height of the audio, respectively, whereas (i, j) indicates the position of the sample value point. A and B denote the original and encrypted voice messages, respectively, while representing the highest value inside the message. shown in Table 2 presents the results.

Table 2. illustrates the outcomes of the (PSNR and MSE) tests conducted on both the input and output voice messages

Audio file duration second	Audio file size	MSE	PSNR
96.6 s	256 bytes	0.00	Inf dB
165.6 s	256 bytes	0.00	Inf dB
234 s	256 bytes	0.00	Inf dB
96.6 s	251 bytes	0.00	Inf dB
165.6 s	251 bytes	0.00	Inf dB
234 s	251 bytes	0.00	Inf dB

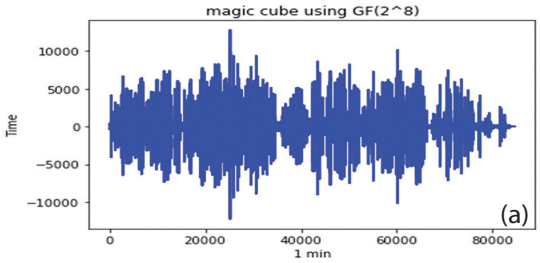
4.4. ANALYSES OF NIST TESTS

The proposed magic cube algorithm comprehensively evaluated its encryption capabilities using seven statistical tests recommended by the NIST. These tests assess the randomness of the binary sequences generated by the algorithm, ensuring its effectiveness in encryption. One of the devised tests is the randomization test, which examines the encrypted output audio file of the encryption methods. This test encompasses various

randomization techniques, including frequency tests, frequency testing within blocks, and entropy tests. The results of these tests are presented in Table 3. In particular, the entropy measure is utilized to assess the randomness of the binary sequences. A higher entropy value indicates a higher probability when the number of zeros and one's equals (entropy = 1.000000). A lower entropy value suggests a lower probability when the number of ones and zeros differs (entropy 0.000000). When employing various blocks, any variation in the distribution of ones and zeros leads to a corresponding variation in the entropy value.

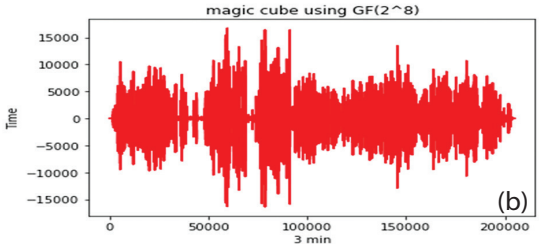
Table 3. The NIST test results for the two different forms of finite field

GF	Audio file size (MB.)	Audio duration (min.)	Block Frequency	Cumulative Sums	FFT	Frequency	Longest Run	Rank	Runs
GF (2^8)	1.61	1	1.000000	0.288242	0.767097	0.144127	1.000000	0.000000	0.674990
GF (2^8)	2.76	2	1.000000	0.727622	0.468160	0.654721	1.000000	0.000000	0.342806
GF (2^8)	3.90	3	1.000000	0.917917	0.468160	0.654721	1.000000	0.000000	0.342806
FG (251)	1.61	1	1.000000	0.115559	0.123812	0.057780	1.000000	0.000000	0.843325
FG (251)	2.76	2	1.000000	0.359368	0.468160	0.371093	1.000000	0.000000	0.852179
FG (251)	3.90	3	1.000000	0.727622	0.468160	0.371093	1.000000	0.000000	0.456057
Condition Final results			<=1.000000 **All success**	<=1.000000 **All success**	<=1.000000 **All success**	<=1.000000 **All success**	<=1.000000 **All success**	<=1.000000 **All success**	<=1.000000 **All success**



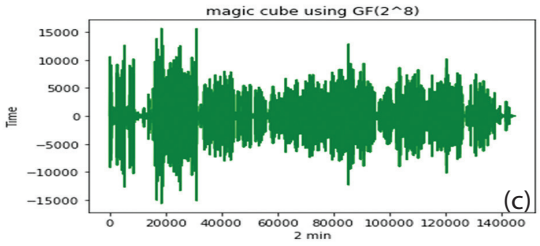
Sample Cipher voice messages 1min. GF (2^8)

b5000000c4000000ce0000003201000004
0100004600000130100008a0100000c00
0000b5000000df0100001c0000003e0100
006b010000d4000000a300000031010000
cc000000b5000000c4000000ce00000032



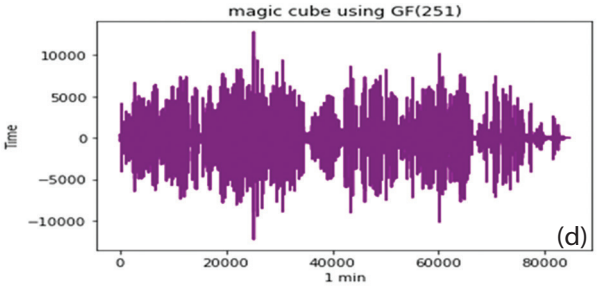
Sample Cipher voice messages 2 min. GF (2^8)

00308000001060000e0050000260500008
3020000e2080000a203000020050000600
300004d0500003007000052030000a70a0
0001a07000061050000b0070000c708000
0a1050000030800000 1060000e00500002



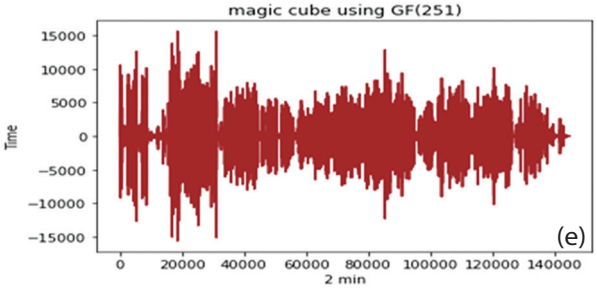
Sample Cipher voice messages 3 min. GF (2^8).

99020000b70100004c0100009501000061010
0009502000086010000530400 00ae030000be
0300006203000048040000b4020000ad03000
0a7020000f3020000680400 00c10200009902
0000b70100004c0100009501000061010000 95



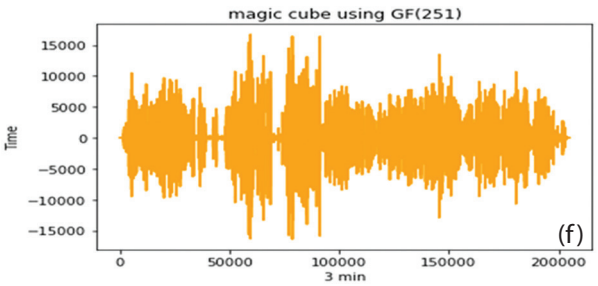
Sample Cipher voice messages 1min. GF (251)

df0100001c0000003e0100006b010000d4
000000a300000031010000cc000000b5
000000c4000000ce00000032010000040
100004100000c000000b50000000a3000
00031010000cc000000b5000000c40000



Sample Cipher voice messages 2 min. GF (251)

000061050000b0070000c7080000a1050
000030800000 1060000e0050000260500
0083020000e2080000a20300002005000
0600300004d0500003007000052030000
a70a00001a07000061050000b0070000c



Sample Cipher voice messages 3 min. GF (251)

ce0000003201000004010000460000001 30
100008a0100000c000000b5000000df01000
01c0000003e0100006b010000d4000000a3
00000031010000cc000000b5000000c40000
00ce0000003201000004010000460000001

Fig. 13. Samples of encoded voice messages with corresponding Cipher voice messages using : (a) 1 min. GF(2^8), (b) 2 min. GF(2^8), (c) 3 min. and (d) 1 min. GF(P), (e) 2 min. GF(P), (f) 3 min. GF(P)

5. CONCLUSION

Developing a new block cipher algorithm by replacing the MixColumn function with a $3 \times 3 \times 3$ magic cube GF (2^8) irreducible polynomial. Introduced several noteworthy characteristics. Firstly, it exhibited a higher key complexity and an increased number of equations compared to the MS3 algorithm GF (2^8), where the magic cube was larger but resulted in faster execution. Additionally, two peculiarities were observed in the case of voice messages: GF (P) demonstrated faster execution time, while the GF (2^8) variant showed lower execution times. The complexity of the keys in the algorithm depends on the desired level of security and the specific implementation. In the magic cube algorithm, the keys play a crucial role in the encryption and decryption processes, involving linear equations solved using Gaussian elimination. The complexity of the keys enhances the randomness and unpredictability between messages, as confirmed by statistical analysis using NIST and PSNR tests. The complexity of the algorithm is enhanced by utilizing the three faces of the magic cube for keys. Consequently, this would increase the system's overall security and complexity. The proposed algorithm can be applied to various platforms, including smartphones, computers, and other devices involved in the exchange of voice messages. Its effectiveness in securing voice message communications makes it a suitable choice for ensuring privacy and security in such scenarios.

6. REFERENCES:

[1] R. Dunn, J. Kim, Z. A. Poucher, C. Ellard, K. A. Tamminen, "A Qualitative Study of Social Media and Electronic Communication among Canadian Adolescent Female Soccer Players", Journal of Adolescent Research, Vol. 39, No. 2, 2024.

[2] J. Hofhuis, J. Gonçalves, P. Schafraad, B. Wu, "Examining strategic diversity communication on social media using supervised machine learning: Development, validation and future research directions", Public Relations Review, Vol. 50, No. 1, 2024, p. 102431.

[3] L. S. Macca, J. Ballerini, G. Santoro, M. Dabić, "Consumer engagement through corporate social responsibility communication on social media: Evidence from Facebook and Instagram Bank Accounts", Journal of Business Research, Vol. 172, 2024, p. 114433.

[4] X. Wang, Y. Su, "An Audio Encryption Algorithm Based on DNA Coding and Chaotic System", IEEE Access, Vol. 8, 2020, pp. 9260-9270.

[5] D. Herbadji, A. Herbadji, I. Hadad, A. Belmeguenai, N. Derouiche, "An Enhanced Logistic Chaotic Map based tweakable Speech encryption algorithm", Integration, Vol. 97, 2024, p. 102192.

[6] A. Vishwakarma, B. Singh, "Implementation Study of AES Standard for IoT Systems", Proceedings of the IEEE Global Conference on Computing, Power and Communication Technologies, New Delhi, India, 23-25 September 2022.

[7] H. J. Mohammed, A. H. Al-Adhami, Y. Yaseen, L. Abed, "A Developed Cryptographic Model Based on AES Cryptosystem", AIP Conference Proceedings, Vol. 2400, 2022.

[8] B. J. Al-Khafaji, A. M. S. Rahma, "Proposed new modification of AES algorithm for data security", Global Journal of Engineering and Technology Advances, Vol. 12, No. 3, 2022, pp. 117-122.

[9] C. Zhang, Y. Jia, L. Zhu, Z. Zhang, "Research on Simple Power Consumption Based on AES Algorithm", Proceedings of the IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms, Changchun, China, 24-26 February 2023, pp. 1883-1886.

[10] K. Li, H. Li, G. Mund, "A reconfigurable and compact subpipelined architecture for AES encryption and decryption", EURASIP Journal on Advances in Signal Processing, Vol. 2023, No. 1, 2023.

[11] W. Stallings, "Cryptography and Network Security: Principles and Practice", Pearson Education, 2020.

[12] J.-J. Wang, Y.-H. Chen, G.-H. Liaw, J. Chang, C.-C. Lee, "Efficient schemes with diverse of a pair of circulant matrices for AES MixColumns-InvMixcolumns transformation", Communications of the CCISA, Vol. 26, No. 2, 2020, pp. 1-20.

[13] A. Vasselle, A. Wurcker, "Optimizations of Side-Channel Attack on AES MixColumns Using Chosen Input", Proceedings of the ACM SIGSAC Conference on Computer and Communications, 2017.

[14] J. Sesiano, "Sources and Studies in the History of Mathematics and Physical Sciences Magic Squares Their History and Construction from Ancient Times to AD 1600", Springer, 2019.

[15] N. Rani, V. Mishra, S. R. Sharma, "Image encryption model based on novel magic square with differential encoding and chaotic map", Nonlinear Dynamics, Vol. 111, No. 3, 2023, pp. 2869-2893.

[16] M. Tahbaz, H. Shirgahi, M. R. Yamaghani, "Evolutionary-based image encryption using Magic Square Chaotic algorithm and RNA codons truth table", Multimedia Tools and Applications, Vol. 83, No. 1, 2024, pp. 503-526.

[17] H. K. Wang, G. B. Xu, D. H. Jiang, "Quantum grayscale image encryption and secret sharing schemes based on Rubik's Cube", Physica A: Statistical Mechanics and its Applications, Vol. 612, 2023.

[18] A. De Schepper, J. Schillewaert, H. Van Maldeghem, M. Victoor, "Construction and characterisation of the varieties of the third row of the Freudenthal-Tits magic square", Geometriae Dedicata, Vol. 218, No. 1, 2024.

[19] X. Zhang, M. Liu, "Multiple-image encryption algorithm based on the stereo Zigzag transformation", Multimedia Tools and Applications, Vol. 83, No. 8, 2024, pp. 22701-22726.

[20] N. Rani, S. R. Sharma, V. Mishra, "Grayscale and colored image encryption model using a novel fused magic cube", Nonlinear Dynamics, Vol. 108, No. 2, 2022, pp. 1773-1796.

[21] J. J. Ranjani, F. Zaid, "Pseudo magic cubes: A multidimensional data hiding scheme exploiting modification directions for large payloads", Computers and Electrical Engineering, Vol. 89, 2021, p. 106928.

[22] C. Cai, Y. Wang, Y. Cao, B. Sun, J. Mou, "Multiple remote sensing image encryption scheme based on saliency extraction and magic cube circular motion", Applied Intelligence, Vol. 54, No. 8, 2024, pp. 5944-5960.

[23] S. I. Hernández, L. F. del Castillo, R. M. del Castillo, A. García-Bernabé, V. Compañ, "Memory kernel formalism with fractional exponents and its application to dielectric relaxation", Physica A: Statistical Mechanics and its Applications, Vol. 612, 2023.

[24] M. U. Hassan, A. Alzayed, A. A. Al-Awady, N. Iqbal, M. Akram, A. Ikram, "A Novel RGB Image Obfuscation Technique Using Dynamically Generated All Order-4 Magic Squares", IEEE Access, Vol. 11, 2023, pp. 46382-46398.

[25] N. Rani, V. Mishra, B. Singh, "Piecewise symmetric magic cube: application to text cryptography", Multimedia Tools and Applications, Vol. 82, No. 13, 2023, pp. 19369-19391.

[26] A. Santos, M. López de Haro, "A heuristic approach for the densest packing fraction of hard-sphere mixtures", Physica A: Statistical Mechanics and its Applications, Vol. 612, 2023.

[27] K. Gavaskar, "AES Algorithm using Dynamic Shift Rows, Sub Bytes and Mix Column Operations for Systems Security wih Optimal Delay", Research Square, 2022, https://www.researchsquare.com/article/rs-1973978/v1 (accessed: 2024)

[28] R. Gupta, A. Rai, "A class of permutation quadrinomials over finite fields", Communications in Algebra, Vol. 52, No. 4, 2024.

[29] M. Singh, D. Sehrawat, "Equal-degree factorization of binomials and trinomials over finite fields", Journal of Applied Mathematics and Computing, Vol. 70, No. 2, 2024, pp. 1647-1672.

[30] M. Yu, J. Xia, J. E. Feng, S. Fu, H. Shen, "Event-Triggered Synchronization of Multiagent Systems Over Finite Fields", IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 71, No. 1, 2024, pp. 370-374.

[31] O. A. Dawood, O. I. Hammadi, K. Shaker, M. Khalaf, "Multi-dimensional cubic symmetric block cipher algorithm for encrypting big data", Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 6, 2020, pp. 2569-2577.

[32] H. Zhu, L. Dai, Y. Liu, L. Wu, "A three-dimensional bit-level image encryption algorithm with Rubik's cube method", Mathematics and Computers in Simulation, Vol. 185, 2021, pp. 754-770.

[33] A. Yousaf, A. Razaq, H. Baig, "A lightweight image encryption algorithm based on patterns in Rubik's revenge cube", Multimedia Tools and Applications, Vol. 81, No. 20, 2022, pp. 28987-28998.

[34] H. K. Wang, G. B. Xu, D. H. Jiang, "Quantum gray-scale image encryption and secret sharing schemes based on Rubik's Cube", Physica A: Statistical Mechanics and its Applications, Vol. 612, 2023.

[35] Z. N. Al-kateeb, S. J. Mohammed, "A novel approach for audio file encryption using hand geometry", Multimedia Tools and Applications, Vol. 79, No. 27-28, 2020, pp. 19615-19628.

[36] X. Wang, Y. Su, "An Audio Encryption Algorithm Based on DNA Coding and Chaotic System", IEEE Access, Vol. 8, 2020, pp. 9260-9270.

[37] M. B. Lin, J. H. Chuang, "The Design of a High-Throughput Hardware Architecture for the AES-GCM Algorithm", IEEE Transactions on Consumer Electronics, Vol. 70, No. 1, 2024, pp. 425-432.

[38] A. Malal, C. Tezcan, "FPGA-friendly compact and efficient AES-like 8 × 8 S-box", Microprocessors and Microsystems, Vol. 105, 2024, p. 105007.

[39] S. Gupta, M. Singh, M. Harish, "On the Study of Families of Linearized Polynomials over Finite Fields", Contemporary Mathematics, Vol. 4, No. 3, 2023.

[40] A. Cintas-Canto, M. M. Kermani, R. Azarderakhsh, "Reliable Architectures for Finite Field Multipliers Using Cyclic Codes on FPGA Utilized in Classic and Post-Quantum Cryptography", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 31, No. 1, 2023, pp. 157-161.

[41] O. A. Dawood, A. M. S. Rahma, A. M. J. Abdul Hossen, "Generalized method for constructing magic cube by folded magic squares", International Journal of Intelligent Systems and Applications, Vol. 8, No. 1, 2016, pp. 1-8.

[42] S. Dhawan, "Secure and resilient improved image steganography using hybrid fuzzy neural network with fuzzy logic", Journal of Safety Science and Resilience, Vol. 5, No. 1, 2024.

[43] S. M. Kareem, A. M. S. Rahma, "An innovative method for enhancing advanced encryption standard algorithm based on magic square of order 6", Bulletin of Electrical Engineering and Informatics, Vol. 12, No. 3, 2023, pp. 1684-1692.

[44] S. Talasila, G. Vijaya Kumar, E. Vijaya Babu, K. Nainika, M. Veda Sahithi, P. Mohan, "The Hybrid Model of LSB—Technique in Image Steganography Using AES and RSA Algorithms", Proceedings of the International Conference on Soft Computing and Signal Processing, Vol. 2, Hyderabad, India, June 2023.

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia.

About this Journal

The International Journal of Electrical and Computer Engineering Systems publishes original research in the form of full papers, case studies, reviews and surveys. It covers theory and application of electrical and computer engineering, synergy of computer systems and computational methods with electrical and electronic systems, as well as interdisciplinary research.

Topics of interest include, but are not limited to:

- Power systems
- Renewable electricity production
- Power electronics
- Electrical drives
- Industrial electronics
- Communication systems
- Advanced modulation techniques
- RFID devices and systems
- Signal and data processing
- Image processing
- Multimedia systems
- Microelectronics
- Instrumentation and measurement
- Control systems
- Robotics
- Modeling and simulation
- Modern computer architectures
- Computer networks
- Embedded systems
- High-performance computing
- Parallel and distributed computer systems
- Human-computer systems
- Intelligent systems
- Multi-agent and holonic systems
- Real-time systems
- Software engineering
- Internet and web applications and systems
- Applications of computer systems in engineering and related disciplines
- Mathematical models of engineering systems
- Engineering management
- Engineering education

Paper Submission

Authors are invited to submit original, unpublished research papers that are not being considered by another journal or any other publisher. Manuscripts must be submitted in doc, docx, rtf or pdf format, and limited to 30 one-column double-spaced pages. All figures and tables must be cited and placed in the body of the paper. Provide contact information of all authors and designate the corresponding author who should submit the manuscript to https://ijeces.ferit.hr. The corresponding author is responsible for ensuring that the article's publication has been approved by all coauthors and by the institutions of the authors if required. All enquiries concerning the publication of accepted papers should be sent to ijeces@ferit.hr.

The following information should be included in the submission:

- paper title;
- full name of each author;
- full institutional mailing addresses;
- e-mail addresses of each author;
- abstract (should be self-contained and not exceed 150 words). Introduction should have no subheadings;
- manuscript should contain one to five alphabetically ordered keywords;
- all abbreviations used in the manuscript should be explained by first appearance;
- all acknowledgments should be included at the end of the paper:
- authors are responsible for ensuring that the information in each reference is complete and accurate. All references must be numbered consecutively and citations of references in text should be identified using numbers in square brackets. All references should be cited within the text;
- each figure should be integrated in the text and cited in a consecutive order. Upon acceptance of the paper, each figure should be of high quality in one of the following formats: EPS, WMF, BMP and TIFF;
- corrected proofs must be returned to the publisher within 7 days of receipt.

Peer Review

All manuscripts are subject to peer review and must meet academic standards. Submissions will be first considered by an editor-in-chief and if not rejected right away, then they will be reviewed by anonymous reviewers. The submitting author will be asked to provide the names of 5 proposed reviewers including their e-mail addresses. The proposed reviewers should be in the research field of the manuscript. They should not be affiliated to the same institution of the manuscript author(s) and should not have had any collaboration with any of the authors during the last 3 years.

Author Benefits

The corresponding author will be provided with a .pdf file of the article or alternatively one hardcopy of the journal free of charge.

Units of Measurement

Units of measurement should be presented simply and concisely using System International (SI) units.

Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003

Published: semiannually

Copyright

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.

Postal Address

Faculty of Electrical Engineering,
Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b
31000 Osijek, Croatia

IJECES Copyright Transfer Form

(Please, read this carefully)

This form is intended for all accepted material submitted to the IJECES journal and must accompany any such material before publication.

TITLE OF ARTICLE (hereinafter referred to as "the Work"):

COMPLETE LIST OF AUTHORS:

The undersigned hereby assigns to the IJECES all rights under copyright that may exist in and to the above Work, and any revised or expanded works submitted to the IJECES by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the complete Work and all incorporated parts of the Work. Otherwise he/she warrants that necessary permissions have been obtained for those parts of works originating from other authors or publishers.

Authors retain all proprietary rights in any process or procedure described in the Work. Authors may reproduce or authorize others to reproduce the Work or derivative works for the author's personal use or for company use, provided that the source and the IJECES copyright notice are indicated, the copies are not used in any way that implies IJECES endorsement of a product or service of any author, and the copies themselves are not offered for sale. In the case of a Work performed under a special government contract or grant, the IJECES recognizes that the government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official government purposes only, if the contract/grant so requires. For all uses not covered previously, authors must ask for permission from the IJECES to reproduce or authorize the reproduction of the Work or material extracted from the Work. Although authors are permitted to re-use all or portions of the Work in other works, this excludes granting third-party requests for reprinting, republishing, or other types of re-use. The IJECES must handle all such third-party requests. The IJECES distributes its publication by various means and media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various collections, databases and other publications. The IJECES publisher requires that the consent of the first-named author be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes. If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IJECES publisher assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Authors of IJECES journal articles and other material must ensure that their Work meets originality, authorship, author responsibilities and author misconduct requirements. It is the responsibility of the authors, not the IJECES publisher, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it.

- The undersigned represents that he/she has the authority to make and execute this assignment.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
- The undersigned agrees to indemnify and hold harmless the IJECES publisher from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.

_____ _____

Author/Authorized Agent **Date**

CONTACT
International Journal of Electrical and Computer Engineering Systems (IJECES)
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b
31000 Osijek, Croatia
Phone: +38531224600,
Fax: +38531224605,
e-mail: ijeces@ferit.hr