Detection of Diabetic Retinopathy Using Hybrid InceptionResNetV2-KELM Method

Musfiroh Musfiroh[1]*, Dian Candra Rini Novitasari[2]*, Lutfi Hakim[3]*, Adelia Damayanti[4]*, Dina Zatusiva Haq[5]*, Siti Nur Aisah[6]*
* Department of Mathematics, UIN Sunan Ampel Surabaya, Surabaya, Indonesia
musfiroh.rohmati22@gmail.com [1], diancrini@uinsa.ac.id [2], lutfihakimbungah@gmail.com [3], adeliadamayanti869@gmail.com [4] , zatusivad@gmail.com [5], aisahsitinur866@gmail.com [6]

Article Info

Article history:

Received 2025-12-07
Revised 2025-12-29
Accepted 2026-01-13

Keyword:

Convolutional Neural Network (CNN),
Diabetes Mellitus,
Diabetic Retinopathy,
InceptionResNetV2,
Kernel Extrem Learning Machine (KELM).

ABSTRACT

Diabetic Retinopathy (DR) is a complication of Diabetes Mellitus (DM), both type 1 and type 2 DM. Based on its severity, DR is divided into mild DR, moderate DR, severe DR, and proliferative DR stages. Manual detection is difficult because there is a fairly small difference between normal and DR. The Computer-Aided Diagnosis (CAD) system is a solution for detecting the severity of DR quickly and accurately so that DR sufferers do not get worse, which can cause blindness. This study uses fundus images from the Mesindor dataset consisting of four classes, namely normal, mild DR, moderate DR, and severe DR, with the InceptionResNetV2-KELM hybrid method. InceptionResNetV2 is used as a feature extraction and Kernel Extreme Learning Machine (KELM) as its classification. Several types of kernels are applied as model trials. The results show the highest sensitivity lies in the polynomial kernel experiment with a sensitivity value of 99.88%, an accuracy of 99.88%, and a specificity of 99.96%. The method used is able to detect very well and is quite time-effective compared to conventional CNN.

I. INTRODUCTION

Diabetic Melillus (DM) complications can cause kidney nephron damage called Diabetic Nephropathy, neurons in the brain are damaged called Diabetic Neuropathy, and Diabetic Retinopathy, which causes infection of the retina [1]. The World Health Organization (WHO) predicts diabetes as the seventh deadly disease. Recently, around 382 million people worldwide have suffered from DM, and 34.60% of them are reported to have Diabetic Retinopathy (DR) [2]. Cases of blindness worldwide are more than 2.60% caused by Diabetic Retinopathy. Almost all people with type 1 DM experience Diabetic Retinopathy, and more than 60% of people with type 2 DM also experience it [3]. According to the International Diabetes Federation, in 2019, 463 million people with DM aged 20-79 years. It is estimated that by 2045, it will reach 700 million people. In addition, DM sufferers mostly live in low and middle-income countries such as Indonesia and India [4].

Reducing the prevalence of DR is essential in preventing vision loss due to DR [5]. Early DR screening helps control and prevent blindness. Fundus imaging is one of the modalities that can capture the back (fundus) of the eye. The image can visualize the anatomical structures of the eye, such as the retina, optic disc, and macula. Manual screening of fundus images by a specialist requires high precision, laboriousness, is time-consuming, and is prone to human error [6]. DR image analysis is a very challenging task because the expressions caused by muscle movements are subtle and transient [7]. In addition, manual detection is very difficult because there is a very small difference between normal and DR images [8]. Computer-Aided Diagnosis (CAD) systems can diagnose the severity of DR quickly and accurately, allowing for timely patient care. CAD systems can also reduce computing costs and reduce skilled professional resources. The proposed algorithm includes stages such as preprocessing, feature extraction, and classification [9].

Previous studies have widely used CAD systems to detect DR. Eman et al.'s research conducted image segmentation to extract four pathological variations. Then, six significant features from the segmented pathological features will be extracted using the Gray-Level Co-occurrence Matrix (GLCM), and the classification method will be used using a Support Vector Machine (SVM). The experiment showed accuracy and sensitivity values of 81.42% and 74.62%, respectively, on Messidor data [10]. Gayathri et al. used CNN feature extraction with various classifiers (SVM, AdaBoost, Naive Bayes, Random Forest, and J48). The J48 method outperformed all other classifiers for the Messidor, IDRiD, and Kaggle datasets with an average accuracy of 99.89% for two-class classification and 99.59% for multi-class classification [5]. Therefore, using feature extraction methods to learn DR image features is crucial to improve classification performance.

CNN algorithm can learn image features very well. Habib Raj et al., using the CNN model VGGNet, obtained 95.41% accuracy, outperforming AlexNet, ResNet, and GoogleNet on DR images [11]. Yadav, et al. compared InceptionResNetV2, InceptionV3, Xception, MobileNetV2, VGG19, and DenseNet201 in classifying DR. The results obtained from the study showed that InceptionResNetV2 proved to be the best model for DR detection and classification with 82% accuracy [12]. Although CNN performed well, it experienced overfitting problems when dealing with small datasets [13]. In addition, adjusting many parameters and a complex hierarchical structure caused CNN to experience a time-consuming training process [14]. Separation of feature learning and classification is a solution to this problem.

Several studies use convolutional features in CNN architecture as feature learning, while the classification method uses a different method [9]. Qomariyah et al. conducted the identification of normal and NPDR classes with various CNN feature extraction trials, namely VGGNet16, VGGNet19, AlexNet, InceptionV3, InceptionResNetV2, GoogleNet, DenseNet201, and ResNet50. SVM classification and feature extraction using ResNet50 produced the best accuracy of 95.83% on 77 images from the Messidor base 12 and, while on 70 images from the Messidor database of base 13, the best accuracy was obtained at 95.24% with the InceptionV3-SVM method [15]. Shidqie Taufiqurrahman et al. Diagnosing multiclass DR using MobileNetV2-SVM resulted in kappa and accuracy of 0.925 and 85% [16]. Swapna et al. diagnosed diabetes using Heart Rate Variability (HRV) data from ECG signals. The study applied the CNN-LSTM method to obtain an accuracy of 95.10% [17]. Pratap et al. detected normal, mild, moderate, and severe DR cataracts from fundus images using CNN-SVM with the AlexNet model. The method resulted in an accuracy of 92.91% [18]. Based on this study, CNN, as a feature learner, is able to learn features from images.

The ELM method has attracted attention in many research fields due to its advantages in training speed, strong generalization performance, and higher accuracy than ANN and SVM [19], [20]. ELM parameters such as weights and biases are randomly assigned and do not need to be updated, making it very fast. In addition, the structure of ELM is simpler than all other gradient-based learning algorithms [21]. Nahiduzzaman et al. performed feature extraction using CNN and Singular Value Decomposition (SVD) to reduce input features from the results of CNN feature extraction. This study utilized the Extreme Learning Machine (ELM) method as a classifier to obtain accuracy and recall of 99.73% and 100% for the binary class. In comparison, the accuracy of the five DR classes was 98.09% for the APTOS-2019 dataset and 96.26% for the Messidor-2 data [22].

The KELM method improves ELM's shortcomings by applying a kernel function. The kernel function is applied because the optimal number of neurons in ELM can only be obtained by trial and error, resulting in a dimensionality catastrophe [23]. For brain tumor classification, Pashaei et al. used CNN feature extraction with several classification methods, such as KELM, MPL, Stacking, XGBoost, SVM, and RBF. KELM obtained the highest accuracy of 93.68% compared to using the MPL, Stacking, XGBoost, SVM, and RBF methods, which obtained accuracies of 88.80%, 86.91%, 87.33%, 87.51%, and 86.84%, respectively [24]. Ripon et al. applied CNN-KELM in the case of eye image recognition from facial images obtained remotely. The study compared feature extraction methods and classification methods. Haar wavelet, LBP, GLAC, and CNN were used for feature extraction experiments, while the classification method trials used SVM and KELM. The CNN-KELM method obtained the highest accuracy, precision, recall, and F1-measure of 99.54%, 99.61%, 99.60%, and 99.57% [25]. Novitasari et al. compared various feature extraction methods, namely GoogleNet, ResNet18, ResNet50, ResNet101, and DenseNet, in diagnosing four DR classes. In addition, the study also compared the ELM modification methods, namely KELM, MLELM, and DELM. Hybrid DenseNet-KELM requires a faster computing time than other ELM modifications, 42.75 seconds [26].

Based on the previous explanation, this study will identify multiclass Diabetic Retinopathy derived from Messidor data: normal, mild, moderate, and severe. This study aims to obtain an optimal model in an efficient time. The dataset has an imbalanced data problem and applies a rotation type augmentation. The hybrid CNN method with KELM is used to detect DR. The CNN method is used for feature extraction, while KELM is used for classification. The CNN architecture uses InseptionResNetV2. The extraction results are classified using KELM. Experiments with several kernels are applied to find the optimal model. Model evaluation is evaluated by a confusion matrix based on accuracy, sensitivity, and specificity.

II. Method

A. *Research Stages*

This study conducted Diabetic Retinopathy (DR) detection using a hybrid Convolutional Neural Network (CNN) type InceptionResNetV2 with Kernel Extreme Learning Machine (KELM). The flowchart of this study is shown in Figure 1.

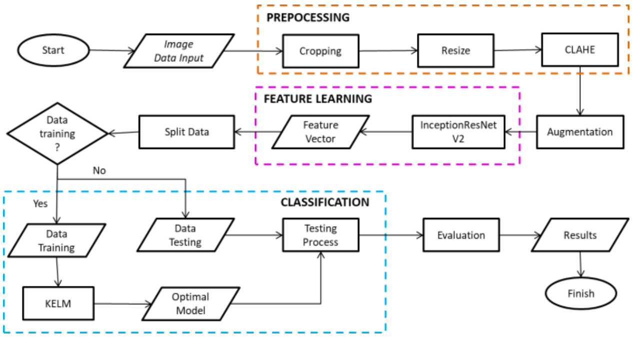


Figure 1. Research Flow Diagram

The input image data was first processed using cropping, resizing, CLAHE, and image augmentation. Cropping is used to remove unnecessary image portions, resizing to adjust the input size before entering the CNN model, CLAHE is used to enhance image contrast, and augmentation uses a rotation method to balance the classes. The next stage is feature extraction using feature learning from the InceptionResnetV2 CNN. The classification stage uses the KELM method, applying data partitioning using K-Fold Cross Validation (K-Fold). Classification results are evaluated using a confusion matrix.

B. *Data Collection*

This study used input from DR fundus image data. Data obtained from Messidor were as many as 1200 [27]. The data set contains four classes, namely normal, mild DR, moderate DR, and severe DR, each of which has a total of 516, 153, 247, and 254 data. Sample DR image data is shown in Figure 2. Fundus image data in the Messidor dataset has different sizes, namely 1440 × 960, 2240 × 1488, and 2304 × 1536.

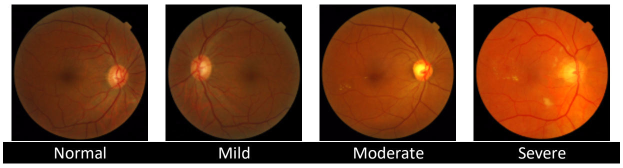


Figure 2. DR Image Data Sample

C. *Convolutional Neural Network (CNN)*

CNN is a development of the Multi-Layer Perceptron (MLP) neural network, built by imitating the pattern of neuron connections in the human visual cortex [28]. CNN has a high-depth network and has been widely applied to image data. CNN is more efficient in various image processing applications mainly due to its success in feature learning. The biggest advantage of CNN is that it can learn features ranging from basic features (such as shape, color, and texture) to more complex features. The main idea is to get local features from high input layers and transfer them to lower layers for more complex features [29]. Feature learning in CNN may be more efficient than manually designed features such as wavelet features, morphological features, shape features, etc.[30].

The architecture consists of an input layer of neurons, several hidden layers, and an output layer. Neurons in CNN have weight, bias, and activation functions, where each neuron in one layer is connected to each neuron in the next layer [31]. The general CNN architecture is shown in Figure 3. Based on the figure, CNN consists of two stages: the feature extraction stage used for feature extraction and the classification stage. The feature extraction stage in CNN consists of a convolution layer with ReLU activation and a pooling layer, while the classification stage contains a fully connected layer and softmax activation [30].

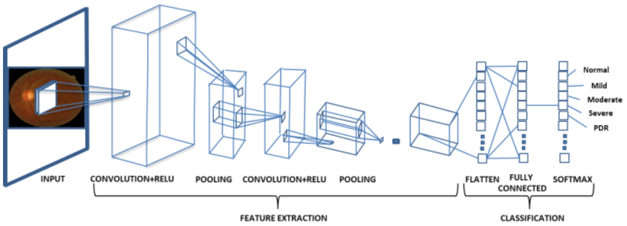


Figure 3. CNN Architecture

D. *InceptionResNetV2 (IRV2)*

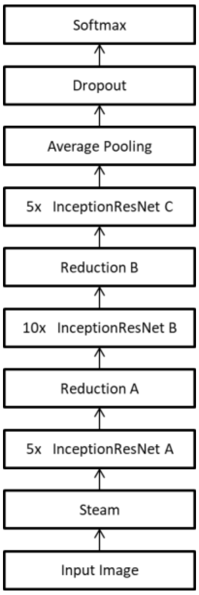


Figure 4. InceptionResNetV2 Architecture

IRV2 is a CNN architecture that combines the advantages of the Inception module and the Residual module to improve accuracy and reduce the number of calculations. IRV2 uses the Inception module because it requires less computational complexity than the basic inception module and uses residual connections to achieve good efficiency [32]. Basic inception has a very complicated algorithm, so the number of calculations and parameters used becomes a burden on improving network performance. On the other hand, the ResNet module in ResNet can result in poor feature extraction diversity compared to the inception module due to the imbalance between depth and width [33].

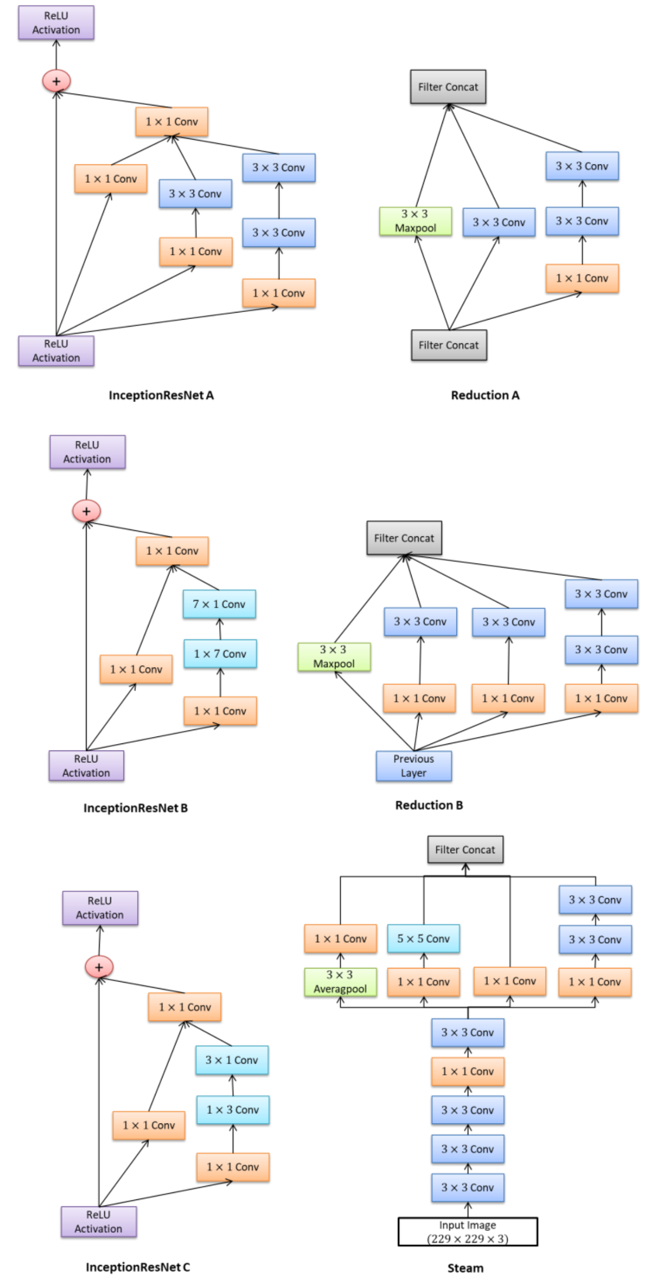
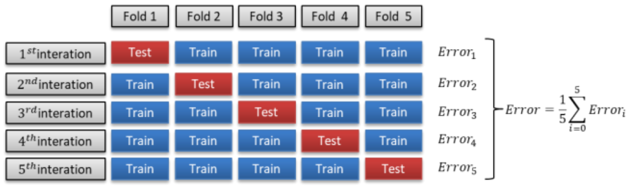


Figure 5. Structure of Each Layer of InceptionResNetV2

The feature extractor in IRV2 is based on the inception block, which uses split, transform, and merge functions. The inception module has several parallel convolutions and filter circuits (1 × 1, 3 × 3, 5 × 5, etc.) combined with the circuits in each branch [31]. The output obtained from each convolution branch is combined and given as input to the next convolution module [34]. The convolution layer uses different filters in the inception block such as 1 × 1 is used for dimension reduction, (2 × (3 × 3)) to factorize the filter into a smaller size, and (1 × 3, 3 × 1 and 1 × 7, 7 × 1) for asymmetric [35], [36]. Each Inception block is connected to a 1 × 1 convolution filter layer without activation function for dimension transformation to achieve input matching. This system compensates for the dimensionality reduction in the Inception block [32]. The residual network learning unit is applied to avoid the problem of complete gradient vanishing when training the Inception network model [33]. The interior module of the network includes Inception-Resnet-A, Inception-Resnet-B, and Inception-Resnet-C blocks, where the total number of frameworks in each module is 10, 20, and 10. The architecture of each IRV2 layer is shown in Figure 5 [32]. The architecture of IRV2 is demonstrated in Figure 4.

E. K-Fold Cross Validation (K-Fold)

K-Fold CV is a validation method used to select a model in applying learning problems with n iterations [37]. This method can avoid bias in data selection and overfitting problems. K-Fold CV randomly divides the original sample D into mutually exclusive subsets of equal size, namely $=$ $D_1 \cup D_2 \cup \cdots \cup D_k$. $D_i \cap D_j = \emptyset (i \neq j)$. The consistency of the data distribution in each subset is maintained as much as possible, namely from hierarchical sampling by D [38]. The original sample D is randomly divided into a number of k subsets. $K-1$ is used for training, and the remaining subset is used for testing. The algorithm is set for the training subsample in each partition, then the average error on the test subsample objects is calculated [39]. The use of K-Fold CV with $k = 5$ is shown in Figure 6.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
1^{st}interation	Test	Train	Train	Train	Train	$Error_1$
2^{nd}interation	Train	Test	Train	Train	Train	$Error_2$
3^{rd}interation	Train	Train	Test	Train	Train	$Error_3$
4^{th}interation	Train	Train	Train	Test	Train	$Error_4$
5^{th}interation	Train	Train	Train	Train	Test	$Error_5$

$$Error = \frac{1}{5}\sum_{i=0}^{5} Error_i$$

Figure 6. K-Fold Cross Validation with K=5

F. Kernel Extreme Learning Machine (KELM)

KELM is an extension of the ELM method proposed by Huang et al. The random determination of weights and biases in ELM causes large variations in classification accuracy in different trials [40]. ELM has a multicollinearity problem when the number of hidden layer nodes is often less than the number of training samples. The presence of multicollinearity when solving the Moore-Penrose Generalized Inverse matrix can cause $\boldsymbol{HH^T}$ singularity. As a result, the ELM output is random and has poor stability and generalization ability [41]. KELM overcomes this by adopting a kernel function to replace the ELM hidden layer mapping [40]. The kernel takes the hidden layer as an unknown feature that maps data from the input space to the feature space. Using kernels causes KELM to obtain better generalization performance than ELM [42]. In addition, the network becomes more stable and does not need to be determined randomly by weights and biases [40]. The KELM architecture is shown in Figure 7.

Based on the orthogonal projection method and ridge regression theory, the Moore–Penrose Generalized Inverse Matrix ($\boldsymbol{H}^{\dagger}$) can be calculated by Equation 1, and the output weight $\boldsymbol{\beta}$ can be calculated by adding a positive constant $\frac{I}{C}$ to balance the empirical and structural risks [43], [44]. Based on the Karush-Kuhn-Tucker theory, the Lagrange factor is introduced so that the ELM calculation uses Equation 2.

$$\boldsymbol{H}^{\dagger} = (\boldsymbol{H'H})^{-\boldsymbol{1}}\boldsymbol{H'} \qquad (1)$$

$$\boldsymbol{\beta} = \begin{cases} \left(\frac{I}{C} + \boldsymbol{H'H}\right)^{-1} \boldsymbol{H'Y} \text{ ,} if\ N > L \\ \boldsymbol{H'}\left(\frac{I}{C} + \boldsymbol{H'H}\right)^{-1} \boldsymbol{Y} \text{ ,} if\ N \leq L \end{cases} \qquad (2)$$

Where C is the regulation coefficient, $\boldsymbol{H'}$ is the transposed $\boldsymbol{H}$ matrix, and $\boldsymbol{I}$ is the identity matrix. Then, the ELM output function is shown in Equation 4.

$$\boldsymbol{Y} = \boldsymbol{H\beta}$$

$$\boldsymbol{Y_{out}} = \begin{cases} h(x)\left(\frac{I}{C} + \boldsymbol{H'H}\right)^{-1} \boldsymbol{H'Y} \text{ ,} if\ N > L \\ h(x)\ \boldsymbol{H'}\left(\frac{I}{C} + \boldsymbol{H'H}\right)^{-1} \boldsymbol{Y} \text{ ,} if\ N \leq L \end{cases} \qquad (4)$$

Since the mapping condition of the hidden layer feature $h(x)$ is unknown, the Mercer condition kernel matrix is introduced [41]. According to the Mercer condition, it can be applied to ELM to formulate KELM. The ELM kernel matrix is first defined by Equation 5.

$$\Omega_{ELM} = \boldsymbol{HH'} \in \mathbb{R}^{N\times N} \qquad (5)$$

Where $\Omega_{ij} = h(x_i)h(x_j) = k(x_i, x_j)$ and $k(.)$ is the kernel function. The kernel matrix Ω_{ELM} is used to replace the random matrix $\boldsymbol{HH'}$ in ELM, and the kernel function is used to map all samples from the N-dimensional input space to the high-dimensional hidden feature space, which can obtain more information from the original data, and improve the stability and generalization ability of the model [42]. Then $h(x)\boldsymbol{H'}$ can be defined as Equation 6, and $\boldsymbol{\beta}$ in KELM is shown in Equation 7.

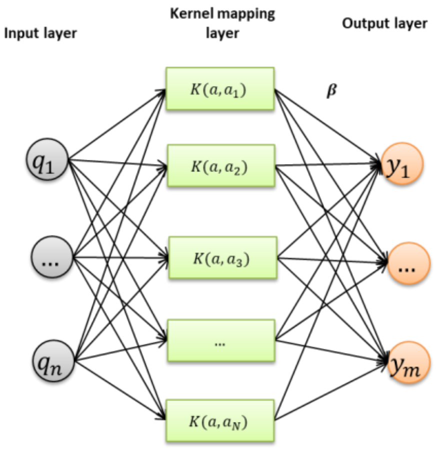


Figure 7. KELM Architecture

$$h(x)\boldsymbol{H'} = \begin{bmatrix} k(x, x_1 \\ \vdots \\ k(x, x_N \end{bmatrix} \qquad (6)$$

$$\boldsymbol{\beta} = \left(\frac{I}{C}\boldsymbol{I} + \Omega_{ELM}\right)^{-1} \boldsymbol{Y} \qquad (7)$$

Then, Equation 4 can be changed into Equation 8.

$$\boldsymbol{Y_{out}} = \begin{bmatrix} k(x, x_1 \\ \vdots \\ k(x, x_N \end{bmatrix}' \boldsymbol{\beta} \qquad (8)$$

Based on Equation 8, $h(x)$ is unrelated to the ELM output, while the ELM output is only related to the kernel function $k(x, y)$. Since the kernel function is an inner product, the number of hidden layer nodes does not affect the output, and there is no need to randomly set the initial weights and biases of the hidden layer. In addition, several kernel functions, such as Gaussian kernel, linear kernel, polynomial kernel, and wavelet kernel, are provided in the modelling [41]. The standard kernel functions are listed in Table 1 [42].

TABLE 1
TYPES OF KELM KERNELS

Category	Formula	Parameter
Linear	$k(x_i, x_j) = x_i\, x_j{}'$	–
Polynomial	$k(x_i, x_j) = (x_i\, x_j{}' + 1)^d$	$d \geq 1$
RBF	$k(x_i, x_j) = \exp\left(-\frac{\lVert x_i - x_j \rVert^2}{2\sigma^2}\right)$	$\sigma > 0$
Wavalet	$k(x_i, x_j) = \mathrm{Cos}\left(\frac{1.675\lVert x_i - x_j \rVert}{\sigma^2}\right)$ $\exp\left(\frac{-\lVert x_i - x_j \rVert^2}{d}\right)$	–

III. Results And Discussion

A. Data Preprocessing And Augmentation

Data is preprocessed to optimize the classification process. The preprocessing process uses manual cropping to remove unused retinal fundus image bumps. The InceptionRenNetV2 architecture requires an input image data size of 299 × 299 × 3, so the fundus image is resized to change the image size. The image data used has different brightness levels. The data requires handling to increase the contrast so that the image becomes sharper and clearer using CLAHE. The CLAHE process is applied to each image channel, namely the red, green, and blue channels. The CLAHE results on each channel are combined. A comparison of the histograms of the DR images before and after CLAHE can be seen in Figure 8. The figure shows that the histograms of the DR images after CLAHE are flatter than those of the DR images before CLAHE. The histogram values after CLAHE are more even and uniform.

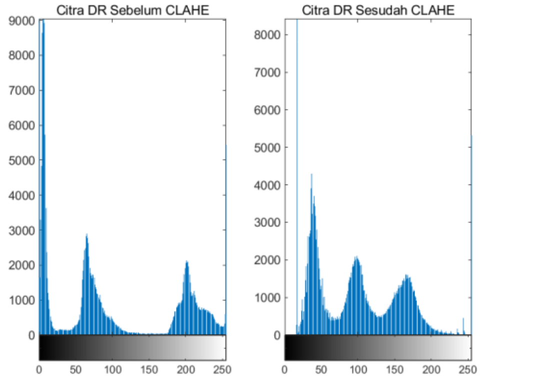


Figure 8. The histograms of the DR images before and after CLAHE

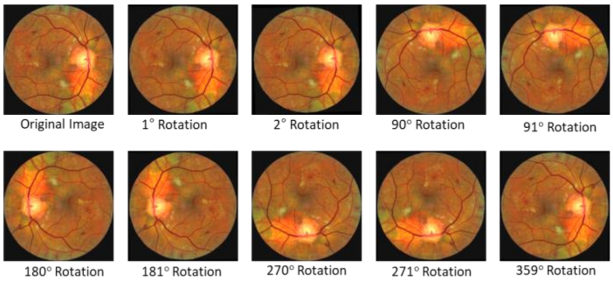


Figure 9. DR Sample Results of CLAHE and Rotation at Several Angles

The dataset used has imbalanced data, so it uses the Data Augmentation Process to increase the variance of image data and reduce overfitting in the system. The augmentation technique in this study uses one type of geometric augmentation: rotation. The rotation angle of this study uses angles ranging from 1° to 359° with a range of 1°. The results of CLAHE and augmentation are shown in Figure 9. Changes in the number of images before and after augmentation are shown in Table 2. The random sampling technique is applied to balance the data by taking 12960 data, consisting of 3240 in each class.

TABLE. 2
DATA CHANGES AFTER AUGMENTATION

DR Messidor Classes	Number of Classes		Random Sampling Results
	Before Augmentation	After Augmentation	
Normal	516	185.244	3240
Mild	153	54.927	3240
Moderate	247	88.675	3240
Severe	254	91.186	3240

B. Feature Extraction

The preprocessing results are processed to the feature extraction stage using InceptionResNetV2. The feature learning section in InceptionResNetV2 is used as feature extraction to obtain more complex features. Image data passes through the InceptionResNetV2 layer to produce feature maps as many as convolution kernels. Deep networks will extract more complex features in the next layer until the last layer in the InceptionResNetV2 feature learning. Changes in the feature map from an image to the last layer of feature learning are represented in Figure 10.

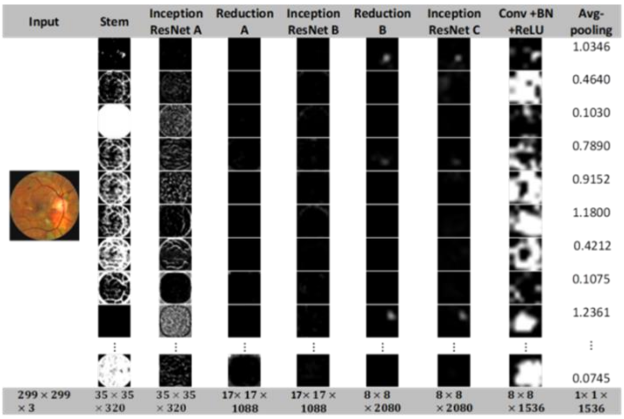


Figure10. Feature map results on CNN Architecture InceptionResNetV2

TABLE 3
FEATURE EXTRACTION RESULTS USING INCEPTIONRESNETV2

No	Feature 1	Feature 2	Feature 3	⋯	Feature 1536	Class
1	1.0346	0.4640	0.1030	⋯	0.1364	1
2	1.4937	0.4222	0.3865	⋯	0.2931	1
3	1.7704	0.1612	0.2999	⋯	0.1372	2
4	1.3112	0.7179	0.1966	⋯	0.5669	2
5	1.2735	0.7342	0.2586	⋯	0.3793	3
6	1.2773	0.4100	0.4393	⋯	0.1235	3
7	1.3604	0.5522	0.2307	⋯	0.0714	4
⋮	⋮	⋮	⋮	⋱	⋮	
12960	1.5560	0.2993	0.3494	⋯	0.1452	4

Based on changes in the feature map size, the InceptionResNet A, InceptionResNet B, and InceptionResNet C blocks maintain their input size. The input size differs from the Reduction A and B blocks, which tend to reduce the feature map to smaller with more channels than the previous input. The resulting feature map in each layer needs to be clarified as the InceptinResNetV2 layer deepens. It causes the feature map to learn the image very complexly in terms of texture, color, and edge. The feature learning process stops at the Global Average Pooling (GAP) layer. Each image subjected to feature extraction using this architecture gets 1536 features. Samples of feature extraction results using InceptionResNetV2 are shown in Table 3. The feature extraction results then become input at the classification stage using KELM.

C. Classification and K-Fold

The classification stage requires training and testing data, so dividing the data using K-Fold cross-validation (K-Fold) is necessary. This method can avoid bias in data selection and overfitting problems. The total data used is 12960. The data is divided using K-Fold with K = 5 so that the training and testing data are 10368 and 2592, respectively.

Several trials are required to obtain the optimum model in the DR classification process using the hybrid InceptionResNetV2-KELM method. InceptionResNetV2 is used as a feature extraction to obtain features from the image, while KELM is used as a classifier. The KELM method uses several trials based on the types of kernels. The kernels used include linear, polynomial, RBF, and wavelet kernels. Each trial produces a classification model, so the best model is selected based on system evaluation.

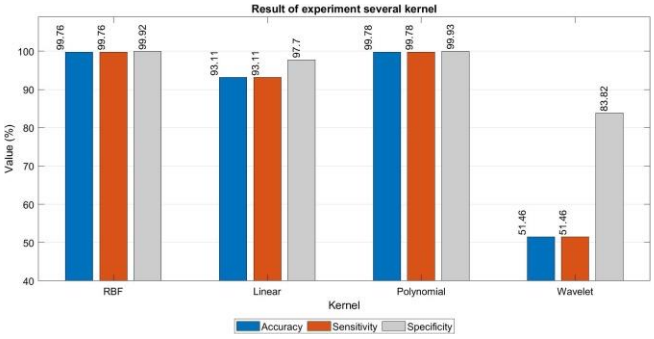


Figure 11. Kernel Evaluation Average

Based on Table 4, the experiment of several kernels resulted in insignificant differences in evaluation on each fold. The average computation time required was around 41 minutes, but fold 5 in the wavelet kernel took longer at around 49 minutes. Figure 11 compares the kernel experiments with a regularization coefficient (C) of 10, where the kernel parameter value uses $d = 2$ and σ is obtained from the standard deviation value. The figure shows that the polynomial kernel has the best accuracy compared to the linear, RBF, and wavelet kernels. The wavelet kernel has the lowest accuracy compared to other kernels. The polynomial kernel gets the highest evaluation value on the division of fold 2 data with sensitivity (Sen) / Recall, accuracy (Acc), specificity (Spec), precision (Prec), and F-Score (F1) of 99.88%, 99.88%, 99.96%, 99.88%, and 99.88% respectively.

TABLE 4
KERNEL TYPE EXPERIMENT EVALUATION RESULTS

Kernel	5-Fold	Sen (%)	Acc (%)	Spec (%)	Prec (%)	F1 (%)	Time
RBF	Fold 1	99.85	99.85	99.95	99.85	99.85	0h 41m 36s
	Fold 2	99.88	99.88	99.96	99.88	99.88	0h 41m 39s
	Fold 3	99.65	99.65	99.88	99.65	99.65	0h 41m 42s
	Fold 4	99.73	99.73	99.91	99.73	99.73	0h 41m 55s
	Fold 5	99.69	99.69	99.90	99.69	99.69	0h 41m 34s
Average		***99.76***	***99.76***	***99.92***	***99.76***	***99.76***	
Linear	Fold 1	93.29	93.29	97.76	93.28	93.27	0h 41m 34s
	Fold 2	93.79	93.79	97.93	93.77	93.77	0h 41m 36s
	Fold 3	92.59	92.59	97.53	92.60	92.57	0h 41m 33s
	Fold 4	93.60	93.60	97.87	93.63	93.59	0h 41m 31s
	Fold 5	92.59	92.59	97.53	92.57	92.57	0h 41m 32s
Average		***93.17***	***93.17***	***97.72***	**93.17**	**93.15**	
Polynomial	Fold 1	99.85	99.85	99.95	99.85	99.85	0h 41m 32s
	Fold 2	99.88	99.88	99.96	99.88	99.88	0h 41m 39s
	Fold 3	99.58	99.58	99.86	99.58	99.58	0h 41m 36s
	Fold 4	99.81	99.81	99.94	99.81	99.81	0h 41m 31s
	Fold 5	99.77	99.77	99.92	99.77	99.77	0h 41m 32s
Average		***99.78***	***99.78***	***99.93***	**99.78**	**99.78**	
Wavelet	Fold 1	52.08	52.08	84.03	52.11	52.08	0h 42m 5s
	Fold 2	51.00	51.00	83.67	51.00	50.99	0h 41m 54s

	Fold 3	50.93	50.93	83.64	50.97	50.94	0h 41m 54s
	Fold 4	51.00	51.00	83.67	50.99	50.98	0h 41m 52s
	Fold 5	52.35	52.35	84.12	52.40	52.35	0h 48m 59s
Average		***51.47***	***51.47***	***83.82***	**51.49**	**51.47**	

D. Confusion Matrix

Figure 12 shows the confusion matrix of the best model; the actual class is shown in the rows, while the predicted class is shown in the columns. There are mild, moderate, normal, and severe DR classes in the rows and columns of the confusion matrix above. The values in each box identify the amount of data from the actual class predicted by the model to be in the predicted class. The boxes on the diagonal of the matrix indicate the correctly predicted class, while the boxes outside the diagonal indicate the opposite. The data predicted according to the actual class in the normal, mild, moderate, and severe classes are 648, 648, 648, and 645, respectively. In the confusion matrix results, there are three incorrectly predicted data. The three data sets should be included in the severe class but are predicted to be in the moderate class.

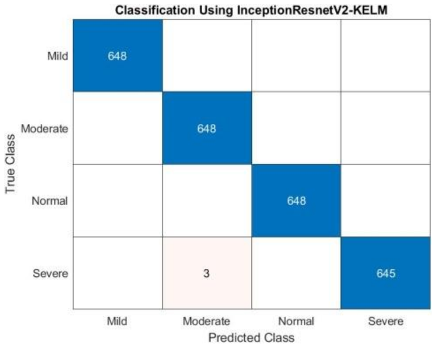
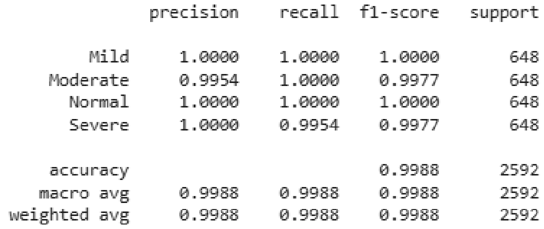


Figure 12. Fold-2 Confusion Matrix and Kernel Polynomial Results

Figure 13 shows that although excellent classification performance was achieved for classes Mild and Normal, a slight decrease was observed for classes moderate and Severe, where a small number of misclassifications caused the precision and recall values to decrease slightly by 99.54%. Nevertheless, the corresponding F1 score remained high at 99.77%, indicating strong classification performance.



```
              precision    recall  f1-score   support

        Mild     1.0000    1.0000    1.0000       648
    Moderate     0.9954    1.0000    0.9977       648
      Normal     1.0000    1.0000    1.0000       648
      Severe     1.0000    0.9954    0.9977       648

    accuracy                         0.9988      2592
   macro avg     0.9988    0.9988    0.9988      2592
weighted avg     0.9988    0.9988    0.9988      2592
```

Figure 13. Evaluate the confusion matrix of each class on fold2

E. Analysis

The experiment results in detecting diabetic retinopathy using the original InceptionResnetV2 are seen in Table 5. Based on these results, the average sensitivity/Recall, accuracy, specificity, precision, and F1-Score were 97.86%, 97.86%, 99.29%, 99.89, and 99.86 respectively. The experiment used k-fold with k = 5 and CNN parameters such as batch size (BS) 16, learning rate (LR) 0.01, and drop out (DO) 0.1. The InceptionResNetV2 method produces a model that has the highest evaluation with a faster time at fold 2.

TABLE 5
EVALUATION RESULTS USING INCEPTIONRESNETV2

5-Fold	**Sen (%)**	**Acc (%)**	**Spec (%)**	**Prec (%)**	**F1 (%)**	**Time**
Fold 1	96.45	96.45	98.82	96.53	96.44	47h 12m 48s
Fold 2	98.50	98.50	99.50	98.50	98.49	44h 35m 42s
Fold 3	98.15	98.15	99.38	98.16	98.15	57h 35m 9s
Fold 4	97.88	97.88	99.29	97.90	97.88	59h 20m 52s
Fold 5	98.34	98.34	99.45	98.35	98.34	40h 6m 59s
Average	**97.86**	**97.86**	**99.29**	**97.89**	**97.86**	

Figure 14 shows the difference in evaluation and training time when using InceptionResNetV2 and InceptionResNetV2-KELM. Based on the analysis of the results, the InceptionResNetV2 method produces a lower model evaluation than the hybrid InceptionResnetV2-KELM. Furthermore, its training time was significantly longer than the InceptionResNetV2-KELM hybrid method, which only took 42 minutes. The InceptionResNetV2-KELM hybrid method uses feature learning from the InceptionResNetV2 model and replaces the classification in InceptionResNetV2 with the KELM model, so that the training time is more effective and reduces computational complexity. KELM adopts the ELM principle by implementing kernel functions, so the training process is carried out without a backpropagation mechanism. This differs from component classification in InceptionResNetV2, which relies on backpropagation, potentially increasing computational complexity and model training time. Figure 15 shows the user interface of the detection system application used in this study.

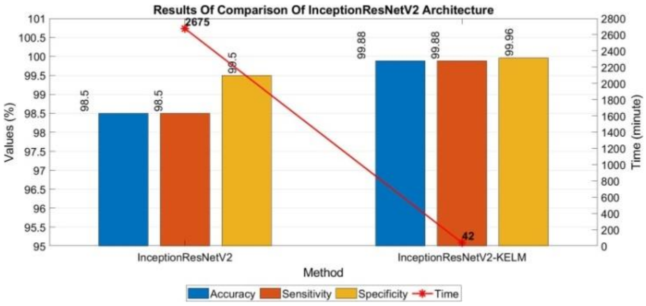
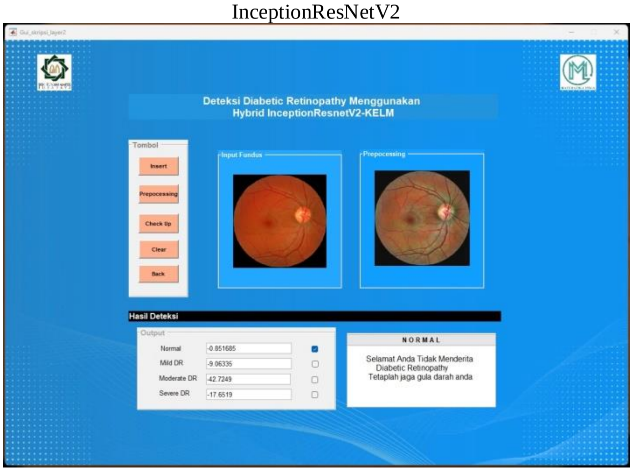


Figure 14. Comparison of InceptionResNetV2-KELM with InceptionResNetV2



Normal -0.851685 ☑
Mild DR -9.06335 □
Moderate DR -42.7249 □
Severe DR -17.6519 □

NORMAL

Selamat Anda Tidak Menderita
Diabetic Retinopathy
Tetaplah jaga gula darah anda

Figure 15. The detection system application interface

TABLE. 6
COMPARISON OF PREVIOUS RESEARCH USING MESSIDOR DATA

Source	Method	Sen (%)	Acc (%)	Spec (%)
(AbdelMaksoud et al., 2020)	Hybrid GLCM KELM	74.62	81.42	94.00
(Novitasari et al., 2023)	Hybrid DenseNet KELM	95.22	95.04	95.04
(Gayathri et al., 2020)	Hybrid CNN J48	-	99.75	0.99
Our research	Inception ResNetV2	98.50	98.50	98.50
	Hybrid Inception ResNetV2 KELM	**99.88**	**99.88**	**99.96**

A comparison of the evaluation results of previous studies with this study is shown in Table 6. The extraction method using deep learning can learn features well. This study uses a hybrid InceptionResNetV2 with KELM to get an accuracy of 99.85%. The selection of CNN architecture and classification method affects the accuracy and training time. The study conducted by (AbdelMaksoud et al., 2020) used three extraction types: GLCM, area of ROIs, and bifurcation points. Messidor data was preprocessed and segmented first. Preprocessing uses a median filter, HEBPDS, for contrast enhancement, normalization, and resizing. Then, the segmentation process is carried out to extract four pathological variations, namely blood vessels (BV), exudates (EX), microaneurysms (MA), and hemorrhages (HM). Extraction with GLCM applied to BV segmentation produces 12 GLCM features; the area of ROIs is calculated from the four pathological variations, and the bifurcation points from BV segmentation are calculated. The feature extraction results are selected as the most relative and correlated using PCA. The PCA results were classified using MLSVM, obtaining accuracy, sensitivity, and specificity of 81.42%, 74.62%, and 94%, respectively. Research (Novitasari et al., 2023) detected Diabetic Retinopathy using hybrid CNN with ELM modifications, namely KELM, MLELM, and DELM. CNN extracts features used in the classification process using ELM modifications. This study conducted experiments using various CNN architectures: DenseNet, GoogleNet, ResNet18, ResNet50, and ResNet101. Before detection, Messidor data was processed by increasing the contrast in the image using CLAHE. The best architecture evaluation lies in the DenseNet architecture. The best results using the DELM method produced an average accuracy of sensitivity and specificity of 99.91% each. Research (Gayathri et al., 2020) detecting Diabetic Retinopathy using CNN for feature extraction and j48 as a classification method obtained an accuracy of 99.75%.

IV. Conclusion

The classification results using the InceptionResnetV2-KELM method obtained good accuracy because feature extraction using the feature learning section of the InceptionResNetV2 type CNN produces 1536 features. The resulting features can represent the image of Diabetic Retinopathy. The resulting feature map and the deep layer in InceptionResNetV2 must be clarified in each layer. KELM, as a classification method, can classify multiclass data well and in a relatively fast time. The best model results in detecting Diabetic Retinopathy using InceptionResNetV2-KELM obtained the highest sensitivity in the polynomial trial. The evaluation obtained from the model is a sensitivity of 99.88%, an accuracy of 99.88%, and a specificity of 99.96%. The training time on the model takes 41 minutes, 39 seconds faster than the original InceptionResNetV2.

References

[1] T. Shanthi and R. S. Sabeenian, “Modified Alexnet architecture for classification of diabetic retinopathy images,” *Comput. Electr. Eng.*, vol. 76, pp. 56–64, 2019, doi: 10.1016/j.compeleceng.2019.03.004.

[2] Z. Khan *et al.*, “Diabetic Retinopathy Detection Using VGG-NIN a Deep Learning Architecture,” *IEEE Access*, vol. 9, pp. 61408–61416, 2021, doi: 10.1109/ACCESS.2021.3074422.

[3] T. Agustin, E. Utami, and H. Al Fatta, “Implementation of Data Augmentation to Improve Performance CNN Method for Detectingg Diabetic Retinopathy,” *Int. Conf. Inf. Commun. Technol.*, pp. 83–88, 2020, doi: https://doi.org/10.1109/ICOIACT50329.2020.9332019.

[4] M. Murugappan, N. . Prakash, R. Jeya, A. Mohanarathinam, G. R.

Hemalakshmi, and M. Mahmud, “A novel few-shot classification framework for diabetic retinopathy detection and grading,” *Measurement*, vol. 200, p. 111485, Aug. 2022, doi: 10.1016/j.measurement.2022.111485.

[5] S. Gayathri, V. P. Gopi, and P. Palanisamy, “A lightweight CNN for Diabetic Retinopathy classification from fundus images,” *Biomed. Signal Process. Control*, vol. 62, no. November 2019, p. 102115, 2020, doi: 10.1016/j.bspc.2020.102115.

[6] N. Sambyal, P. Saini, R. Syal, and V. Gupta, “Modified U-Net architecture for semantic segmentation of diabetic retinopathy images,” *Biocybern. Biomed. Eng.*, vol. 40, no. 3, pp. 1094–1109, 2020, doi: 10.1016/j.bbe.2020.05.006.

[7] C. Mahiba and A. Jayachandran, “Severity analysis of diabetic retinopathy in retinal images using hybrid structure descriptor and modified CNNs,” *Meas. J. Int. Meas. Confed.*, vol. 135, pp. 762–767, 2019, doi: 10.1016/j.measurement.2018.12.032.

[8] M. Hayati *et al.*, “Impact of CLAHE-based image enhancement for diabetic retinopathy classification through deep learning,” *Procedia Comput. Sci.*, vol. 216, no. 2022, pp. 57–66, 2022, doi: 10.1016/j.procs.2022.12.111.

[9] D. C. R. Novitasari *et al.*, “Image Fundus Classification System for Diabetic Retinopathy Stage Detection Using Hybrid CNN-DELM,” *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040146.

[10] E. AbdelMaksoud, S. Barakat, and M. Elmogy, “A comprehensive diagnosis system for early signs and different diabetic retinopathy grades using fundus retinal images based on pathological changes detection,” *Comput. Biol. Med.*, vol. 126, p. 104039, 2020, doi: 10.1016/j.compbiomed.2020.104039.

[11] M. A. Habib Raj, M. Al Mamun, and M. F. Faruk, “CNN Based Diabetic Retinopathy Status Prediction Using Fundus Images,” *2020 IEEE Reg. 10 Symp. TENSYMP 2020*, no. June, pp. 190–193, 2020, doi: 10.1109/TENSYMP50017.2020.9230974.

[12] Y. Yadav, S. Chand, R. C. Sahoo, B. M. Sahoo, and S. Kumar, “Comparative analysis of detection and classification of diabetic retinopathy by using transfer learning of CNN based models,” *J. Intell. Fuzzy Syst.*, vol. 43, no. 1, pp. 985–999, 2022.

[13] M. Jena, D. Mishra, S. P. Mishra, and P. K. Mallick, “A Tailored Complex Medical Decision Analysis Model for Diabetic Retinopathy Classification Based on Optimized Un-Supervised Feature Learning Approach,” *Arab. J. Sci. Eng.*, vol. 48, no. 2, pp. 2087–2099, 2023, doi: 10.1007/s13369-022-07057-0.

[14] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, “Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines,” *J. Franklin Inst.*, vol. 357, no. 13, pp. 8925–8955, 2020, doi: 10.1016/j.jfranklin.2020.04.033.

[15] D. U. N. Qomariah, H. Tjandrasa, and C. Fatichah, “Classification of diabetic retinopathy and normal retinal images using CNN and SVM,” *Proc. 2019 Int. Conf. Inf. Commun. Technol. Syst. ICTS 2019*, pp. 152–157, 2019, doi: 10.1109/ICTS.2019.8850940.

[16] S. Taufiqurrahman, A. Handayani, B. R. Hermanto, and T. L. E. R. Mengko, “Diabetic Retinopathy Classification Using A Hybrid and Efficient MobileNetV2-SVM Model,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2020-Novem, pp. 235–240, 2020, doi: 10.1109/TENCON50793.2020.9293739.

[17] G. Swapna, K. P. Soman, and R. Vinayakumar, “Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1253–1262, 2018, doi: 10.1016/j.procs.2018.05.041.

[18] T. Pratap and P. Kokil, “Computer-aided diagnosis of cataract using deep transfer learning,” *Biomed. Signal Process. Control*, vol. 53, p. 101533, 2019, doi: 10.1016/j.bspc.2019.04.010.

[19] Z. Wang, S. Lou, S. Liang, and X. Sheng, “Multi-Class Disturbance Events Recognition Based on EMD and XGBoost in φ-OTDR,” *IEEE Access*, vol. 8, pp. 63551–63558, 2020, doi: 10.1109/ACCESS.2020.2984022.

[20] Z. Chen, K. Gryllias, and W. Li, “Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine,” *Mech. Syst. Signal Process.*, vol. 133, p. 106272, 2019, doi: 10.1016/j.ymssp.2019.106272.

[21] V. Lahoura *et al.*, “Cloud computing-based framework for breast cancer diagnosis using extreme learning machine,” *Diagnostics*, vol. 11, no. 2, pp. 1–19, 2021, doi: 10.3390/diagnostics11020241.

[22] M. Nahiduzzaman, M. R. Islam, S. M. R. Islam, M. O. F. Goni, M. S. Anower, and K. S. Kwak, “Hybrid CNN-SVD Based Prominent Feature Extraction and Selection for Grading Diabetic Retinopathy Using Extreme Learning Machine Algorithm,” *IEEE Access*, vol. 9, pp. 152261–152274, 2021, doi: 10.1109/ACCESS.2021.3125791.

[23] C. Hou, Y. Li, X. Chen, and J. Zhang, “Automatic modulation classification using KELM with joint features of CNN and LBP,” *Phys. Commun.*, vol. 45, p. 101259, 2021, doi: 10.1016/j.phycom.2020.101259.

[24] A. Pashaei, H. Sajedi, and N. Jazayeri, “Brain tumor classification via convolutional neural network and extreme learning machines,” *2018 8th Int. Conf. Comput. Knowl. Eng. ICCKE 2018*, no. Iccke, pp. 314–319, 2018, doi: 10.1109/ICCKE.2018.8566571.

[25] K. S. Nawaz Ripon, L. Ershad Ali, N. Siddique, and J. Ma, “Convolutional Neural Network based Eye Recognition from Distantly Acquired Face Images for Human Identification,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2019-July, no. July, pp. 1–8, 2019, doi: 10.1109/IJCNN.2019.8852190.

[26] D. Candra, R. Novitasari, and R. E. Putra, “An Effective Hybrid Convolutional-Modified Extreme Learning Machine in Early Stage Diabetic Retinopathy,” *Int. J. Intell. Eng. Syst.*, vol. 16, no. 2, pp. 401–413, 2023, doi: 10.22266/ijies2023.0430.32.

[27] G. G. Guillaume PATRY, “Messidor-2.” .

[28] Y. Liu, H. Pu, and D. W. Sun, “Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices,” *Trends Food Sci. Technol.*, vol. 113, no. April, pp. 193–204, 2021, doi: 10.1016/j.tifs.2021.04.042.

[29] M. Z. Islam, M. M. Islam, and A. Asraf, “A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images,” *Informatics Med. Unlocked*, vol. 20, 2020, doi: 10.1016/j.imu.2020.100412.

[30] Y. D. Zhang, C. Pan, X. Chen, and F. Wang, “Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling,” *J. Comput. Sci.*, vol. 27, pp. 57–68, 2018, doi: 10.1016/j.jocs.2018.05.005.

[31] M. Desai and M. Shah, “An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN),” *Clin. eHealth*, vol. 4, no. 2021, pp. 1–11, 2021, doi: 10.1016/j.ceh.2020.11.002.

[32] R. M. Kamble *et al.*, “Automated Diabetic Macular Edema (DME) analysis using fine tuning with inception-resnet-v2 on oct images,” *2018 IEEE EMBS Conf. Biomed. Eng. Sci. IECBES 2018 - Proc.*, vol. cl, pp. 442–446, 2019, doi: 10.1109/IECBES.2018.8626616.

[33] J. Wang, X. He, S. Faming, G. Lu, H. Cong, and Q. Jiang, “A Real-Time Bridge Crack Detection Method Based on an Improved Inception-Resnet-v2 Structure,” *IEEE Access*, vol. 9, pp. 93209–93223, 2021, doi: 10.1109/ACCESS.2021.3093210.

[34] H. Munusamy, J. M. Karthikeyan, G. Shriram, S. Thanga Revathi, and S. Aravindkumar, “FractalCovNet architecture for COVID-19 Chest X-ray image Classification and CT-scan image Segmentation,” *Biocybern. Biomed. Eng.*, vol. 41, no. 3, pp. 1025–1038, 2021, doi: 10.1016/j.bbe.2021.06.011.

[35] K. N, L. V. Narasimha Prasad, C. S. Pavan Kumar, B. Subedi, H. B. Abraha, and V. E. Sathishkumar, “Rice leaf diseases prediction using deep neural networks with transfer learning,” *Environ. Res.*, vol. 198, no. April, p. 111275, 2021, doi: 10.1016/j.envres.2021.111275.

[36] M. Sadat Shahabi, A. Shalbaf, and A. Maghsoudi, “Prediction of drug response in major depressive disorder using ensemble of transfer learning with convolutional neural network based on EEG,” *Biocybern. Biomed. Eng.*, vol. 41, no. 3, pp. 946–959,

2021, doi: 10.1016/j.bbe.2021.06.006.

[37] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, “Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking,” *IEEE Access*, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.

[38] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, “Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization,” *Geosci. Front.*, vol. 12, no. 1, pp. 469–477, 2021, doi: 10.1016/j.gsf.2020.03.007.

[39] M. Artur, “Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features,” *Procedia Comput. Sci.*, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.

[40] S. Pang, X. Yang, X. Zhang, and X. Lin, “Fault diagnosis of rotating machinery with ensemble kernel extreme learning machine based on fused multi-domain features,” *ISA Trans.*, vol. 98, no. xxxx, pp. 320–337, 2020, doi: 10.1016/j.isatra.2019.08.053.

[41] H. Liu, Y. Zhang, and H. Zhang, “Prediction of effluent quality in papermaking wastewater treatment processes using dynamic kernel-based extreme learning machine,” *Process Biochem.*, vol. 97, no. July, pp. 72–79, 2020, doi: 10.1016/j.procbio.2020.06.020.

[42] Q. Chen, H. Wei, M. Rashid, and Z. Cai, “Kernel extreme learning machine based hierarchical machine learning for multi-type and concurrent fault diagnosis,” *Meas. J. Int. Meas. Confed.*, vol. 184, no. January, p. 109923, 2021, doi: 10.1016/j.measurement.2021.109923.

[43] S. Chen, C. Gu, C. Lin, Y. Wang, and M. A. Hariri-Ardebili, “Prediction, monitoring, and interpretation of dam leakage flow via adaptative kernel extreme learning machine,” *Meas. J. Int. Meas. Confed.*, vol. 166, p. 108161, 2020, doi: 10.1016/j.measurement.2020.108161.

[44] L. Gan, X. Zhao, H. Wu, and Z. Zhong, “Estimation of remaining fatigue life under two-step loading based on kernel-extreme learning machine,” *Int. J. Fatigue*, vol. 148, no. February, p. 106190, 2021, doi: 10.1016/j.ijfatigue.2021.106190.