VLR Net-An Ensemble Model to Enhance the Accuracy of Classification of Colposcopy Images

Priyadarshini Chatterjee * , Shadab Siddiqui , Razia Sulthana Abdul Kareem

Posted Date: 31 October 2024

doi: 10.20944/preprints202410.2511.v1

Keywords: cervical cancer; colposcopy; deep learning; image; clustering

Article

VLR Net-An Ensemble Model to Enhance the Accuracy of Classification of Colposcopy Images

Priyadarshini Chatterjee [a,b]*, Shadab Siddiqui [c] and Razia Sulthana Abdul Kareem [d]

[a] Koneru Lakshmaiah Education Foundation, Department of Computer Science and Engineering, Aziznagar, Hyderabad, 500075, Telengana, India
[b] B V Raju Institute of Technology, Department of Computer Science and Engineering, Vishnupur, Narsapur, 502313, Medak, India
[c] Koneru Lakshmaiah Education Foundation,Department of Computer Science and Engineering, Aziznagar, Hyderabad, 500075, Telengana, India
[d] University of Greenwich, Old Royal Naval College, Department of Computer Science, United Kingdom
* Correspondence: bulan_babba@rediffmail.com (P.C.); priyadarshini.ch@bvrit.ac.in (P.C.)

Abstract: Early detection of cervical cancer is the need of the hour to stop the fatality from this disease. There have been various CAD approaches in the past that promise to detect cervical cancer in an early stage, each of them having some constraints.This study proposes a novel ensemble deep learning model, VLR (variable learning rate), aimed at enhancing the accuracy of cervical cancer classification. The model architecture integrates VGG16, Logistic Regression, and ResNet50, combining their strengths in an offbeat ensemble design. VLR learns in two ways: firstly, by dynamic weights from three base models, each of them trained separately; secondly, by attention mechanisms used in the dense layer of the base models. Hyperparameter tuning is applied to further reduce loss, fine tune the model's performance, and maximize classification accuracy. We performed K-fold cross-validation on VLR to evaluate any improvements in metric values resulting from hyperparameter fine tuning. We have also validated our model on images captured in three different solutions and on a secondary dataset. Our proposed VLR model outperformed existing methods in cervical cancer classification, achieving a remarkable training accuracy of 99.95% and a testing accuracy of 99.89%.

Keywords: cervical cancer; colposcopy; deep learning; image; clustering

1. Introduction

Despite being preventable if caught early, women are disproportionately affected by cervical cancer globally [23]. It happens when abnormal cells around the womb develop into cancer, mainly because of HPV. Within the cervix, glandular cells give way to squamous cells through a normal process [24]. These cells may develop abnormalities and eventually turn into cancer if they are exposed to HPV [25].

A colposcopy is a diagnostic procedure that looks attentively for disease-related indications in the cervix, vagina, and vulva. When abnormal findings are obtained from an HPV test or Pap smear, it is frequently advised [26]. A colposcopy's main goal is to find and treat precancerous lesions as soon as possible to avoid cervical cancer [20]. A subset of machine learning and artificial intelligence called deep learning replicates how a human brain processes information and forms patterns to be used in decision-making [27]. It incorporates multilayered neural networks that are capable of autonomous learning and intelligent decision-making [18]. It is an effective tool for creating complex models that can carry out operations that were previously believed to be exclusive to human intelligence, as it is able to learn from vast volumes of data [19]. A convolutional neural network was created by the Oxford Visual Graphics Group called VGG16 model, which is renowned for its tasteful yet straightforward design [31]. There are 16 layers in the network: 3 fully linked levels and 13 convolutional layers [32]. To minimize spatial dimensions, it takes advantage of uniform architecture with max-pooling layers and compact 3x3 filters. Microsoft Research unveiled ResNet50, a 50-layer deep convolutional neural network, as a member of the Residual Networks (ResNet) family [33]. By using skip connections to introduce residual learning, it solves the vanishing gradient issue in deep networks [34]. Gradients

 doi:10.20944/preprints202410.2511.v1

may travel straight across the network thanks to these connections, which makes it possible to train significantly deeper architectures [22]. Logistic regression is a universally used statistical model for multiclass and binary classification tasks, such as cervical cancer classification [35]. It calculates the feasibility that a given input belongs to a particular class, making it effective in distinguishing between different stages of cervical cancer based on features extracted from colposcopy images [36]. A decision tree is a supervised machine learning algorithm used for classification and regression tasks [37]. The decision tree has a root node, internal nodes, branches, and leaf nodes [38]. The root node is decided based on the characteristics of the data. Branches of the decision tree are the decision and direction the decision tree grows. The nodes in the last level are the final predicted values. The attention mechanism in neural networks is a method to strengthen the model's ability to focus on the most distinguishing parts of the input data [39]. In architectures combining dense layers, softmax, and GAP, the attention mechanism dynamically focuses on the most relevant features, which are then aggregated [40].

Experiments are constructed using 11,000 positive colposcopy images captured in three different solutions. This dataset is acquired from International Agency for Research on Cancer [41]-WHO. This ensemble model achieves best classification accuracy than the existing approaches and the contributions of this work are summarized as:

1. VLR is an ensemble of VGG16, ResNet50 and Logistic Regression models, making it possible to harness the strength of each of these individual models by extracting high level features and also increased interpretability by reducing the chance of over fitting.
2. The VLR learning process is optimized by utilizing dynamic weights and by incorporating attention mechanisms in the dense layers of the base models. Additionally, hyperparameter tuning is applied to minimize training loss, thereby increasing validation accuracy.
3. The study aims to examine the impact of different datasets on VLR model's accuracy. By applying VLR on different secondary datasets, the research evaluates how dataset variability affects the model's performance.

The paper has the following sections: section 2: proposed methodology, section 3: algorithm, section 4: experimental analysis , section 5: results and discussions, section 6: related work, and section 7: conclusion.

2. Proposed Methodology

The proposed methodology is designed for predicting accurately the stages of cervical cancer using colposcopy positive images as CIN1 (represents low-grade, mild changes in cervical cells), CIN2 (indicates moderate abnormal cell growth around the cervix), and CIN3 (Considered a pre-cancerous condition, with a higher risk of developing into invasive cervical cancer if untreated). There are some glitches in the existing approaches; incorrect use of dataset in predicting the type of cervical cancer [42], an imbalance between recall and precision leading to a higher count of false positives [43], and lower classification accuracy.

To deal with these glitches, the VLR is designed to work with two levels of feature extraction and two ways of learning with hyperparameter fine tuning and K-fold cross validation. We employed three base models; VGG16, ResNet50, and Logistic Regression, training each independently on a dataset of 11,000 positive cervigrams, which were divided in an 80:20 ratio for training and testing. We selected validation accuracy, training loss, and training time as key performance metrics, assigning them weights and then normalize them. We have used an attention mechanism that will compute static normalised weights for features based on the importance of each feature map for the current input. These weights are combined with the dynamic normalized weights and forward propagated to VLR. The VLR model optimizes its loss by learning from the weighted contributions of the three base models, with the weights assigned manually during forward propagation. Furthermore, the VLR also learns from the attention weight computed using predefined scores assigned on the basis of the features of highest importance. Moreover, the VLR model benefits from knowledge acquired from the pre-trained base models, combining their strengths to improve overall performance. Fig. 1 is the

 doi:10.20944/preprints202410.2511.v1

framework of VLR model. In the first stage we train the three base models noting their metrics, which will be forward propagated as dynamic weight to the second stage, where we calculate the normalized attention weight based on predefined scores on the features of highest importance. From the second stage, the combined forward propagated score is transferred to the third stage, wherein the ensemble model minimizes the validation loss using backward propagation. In the fourth stage to further minimize the loss of the ensemble model, a hyperparameter fine tuning is applied on learning rate and batch size. In the last stage, the model's performance is generalized using K-fold cross validation technique and validated on the images being separated as captured in three different solutions and also on a secondary dataset.

In the following section, we explain the six stages of the proposed architecture and their mathematical basis as shown in Figure 1. We have already described the motivation that led us to this finding in the beginning of this section.

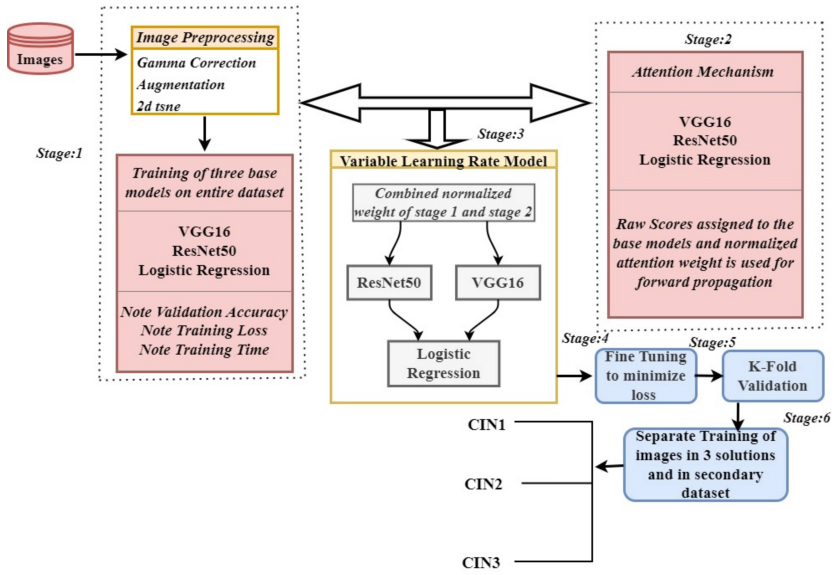

Figure 1. Pictorial Representation of the architecture.

2.1. Stage 1: Pre-Trained Base Models as Feature Extractor and Meta Learner

In this stage, the base models: VGG16, ResNet50 and LR are trained separately on 8800 training data and 2200 test data. The reason behind selecting these base models are; VGG16 acts as a feature extractor, ResNet50 extracts the features more deeply with residual connection and Logistic regression acts as a meta learner and prevents over fitting. The validation accuracy, training loss and training time of these three models are noted after 150 epochs and weights are assigned to them. These weighted metrics are normalized and added with the normalized weights of attention mechanism and forwarded to VLR as scores so that VLR can learn from these scores. The mathematical basis of this stage is as described below in equations 1, 2 and 3.

Mathematical Basis

1. Each model $M_i \in \{M_{\text{VGG16}}, M_{\text{ResNet50}}, M_{\text{LR}}\}$ is trained on a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{h \times w \times c}$ and $y_i \in \{0, 1\}$. The training loss $\mathcal{L}_i$ for each model is minimized using the binary cross-entropy loss function:

$$\mathcal{L}_i = -\frac{1}{n} \sum_{j=1}^{n} \left[y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j) \right] \tag{1}$$

where $\hat{y}_j$ is the predicted probability for the positive class from model M_i.

 doi:10.20944/preprints202410.2511.v1

2. The validation accuracy (A_i) for each model is denoted as:

$$A_i = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}(\hat{y}_j = y_j) \tag{2}$$

Where $\mathbb{1}(\hat{y}_j = y_j)$ is an signal function that equals 1 if the predicted class $\hat{y}_j$ matches the true label y_j and 0 otherwise. The sum is then averaged over all n samples to compute accuracy.

3. We represent the training time (T_i) for each model simply as the time taken to train the model:

$$T_i = \text{Total time taken for training model } M_i \tag{3}$$

2.2. Stage 2: Attention Mechanism in Base Models

In this stage, we are implementing attention mechanism to the dense layer of the base models that provides weights based on predefined scores attributed to the base models. These raw scores are attributed to the models based on their capability to learn the feature of importance. The mathematical basis of this stage is as described below in equations 4, 5 and 6.

Mathematical Basis

1 Extract Features from Each Model

$$\mathbf{f}_{\text{VGG16}} = \text{VGG16}(X), \quad \mathbf{f}_{\text{ResNet50}} = \text{ResNet50}(X), \quad \mathbf{f}_{\text{LR}} = \text{LR}(X) \tag{4}$$

In the above equation X is the input data to the base models.

2 Calculate Normalized Attention Weights Using Predefined Scores

$$w_i = \frac{e^{s_i}}{\sum_{j \in \{\text{VGG16, ResNet50, LR}\}} e^{s_j}} \tag{5}$$

The **attention weight** for the base model i. It represents how much importance or contribution is given to a specific model's output. The tem e^{s_i} is the **exponential function** applied to the **predefined score** s_i of the model i. The term $\sum_{j \in \{\text{VGG16, ResNet50, LR}\}} e^{s_j}$ is the **sum of exponential values of scores** for all models: VGG16, ResNet50, and LR. The summation is done over all base models, allowing normalization so that the resulting attention weights w_i add up to 1.

3 Combine Features Using Weighted Attention

$$\mathbf{F}_{\text{combined}} = w_{\text{VGG16}} \cdot \mathbf{f}_{\text{VGG16}} + w_{\text{ResNet50}} \cdot \mathbf{f}_{\text{ResNet50}} + w_{\text{LR}} \cdot \mathbf{f}_{\text{LR}} \tag{6}$$

2.3. Stage 3: Forward Propagation and Backward Propagation of VLR

The process followed in stage 3 and it's mathematical basis is described below:

Step 1: Assign and Normalize dynamic Weights

We assign dynamic weights to each model based on three metrics: validation accuracy (A), training loss (L), and training time (T). Let the weights be w_A, w_L, and w_T, with respective values of 0.5, 0.2, and 0.3. The weights assigned to the metrics indicate their respective significance in evaluating the effectiveness of every base model.

For each model M, the combined dynamic performance weight W_M is calculated as:

$$W_M^{\text{dynamic}} = w_A \cdot A_M + w_L \cdot \frac{1}{L_M} + w_T \cdot \frac{1}{T_M} \tag{7}$$

Where A_M, L_M, and T_M are the validation accuracy, training loss, and training time of model M, respectively, for each of the three base models.

Step 2: Apply Static Attention Mechanism

 doi:10.20944/preprints202410.2511.v1

In addition to the dynamic weights, we apply a static attention mechanism to determine the importance of each model during forward propagation. Let W_M^{attn} denote the static attention weight obtained from an attention module that takes the feature maps of each base model as input.

$$W_M^{\text{attn}} = \text{AttentionModule}(M_{\text{features}}) \tag{8}$$

Where M_{features} represents the features extracted by the respective base models (e.g., VGG16, ResNet50, LR).

Step 3: Calculate Final Combined Weight

To determine the final contribution of each base model, we combine the dynamic weight and static attention weight using a hyperparameter α, which balances the influence of both components:

$$W_M^{\text{final}} = \alpha W_M^{\text{dynamic}} + (1 - \alpha) W_M^{\text{attn}} \tag{9}$$

Where $0 \leq \alpha \leq 1$ is a hyperparameter that controls the contribution of static versus attention-based weights. In this scenario, we set $\alpha = 0.7$ for dynamic component and $\alpha = 0.3$ to static components in order to balance the high pre-defined scores assigned to the models in dynamic normalized weight calculation of each models.

Step 4: Usage in Forward Propagation

During forward propagation, the output predictions of the base models ($M_{\text{VGG16}}, M_{\text{ResNet50}}, M_{\text{LR}}$) are combined using their respective W_M^{final} values to generate a weighted ensemble output:

$$\hat{y}_{\text{VLR}} = \sum_{M \in \{\text{VGG16,ResNet50,LR}\}} W_M^{\text{final}} \cdot \hat{y}_M \tag{10}$$

The final prediction by the VLR model is based on this ensemble, where models with higher combined weight W_M^{final} contribute more significantly to the prediction.

Step 5: Training Objective

Loss Function for Backpropagation: The VLR model optimizes its parameters using a differentiable loss function, such as cross-entropy loss, on the combined predictions. The combined weight W_M^{final} is used to determine the influence of each base model during prediction aggregation.

$$\mathcal{L}_{\text{VLR}} = -\frac{1}{n} \sum_j [y_j \log \hat{y}_{\text{VLR},j} + (1 - y_j) \log(1 - \hat{y}_{\text{VLR},j})] \tag{11}$$

Rationale for Weighting

The weights (w_A, w_L, w_T) are designed to prioritize models that exhibit strong validation performance while accounting for training efficiency and stability. The static attention mechanism further refines the weight contribution based on feature importance, making the VLR model adaptable to the specific input data. This balanced consideration ensures that the final VLR model learns effectively from the best features of each base model.

2.4. Stage 4: Hyperparameter Fine Tuning

The goal of hyperparameter optimization is to identify the best set of hyperparameters that minimize the validation loss $\mathcal{J}_{\text{val}}$, which subsequently reduces the training loss $\mathcal{J}_{\text{train}}$. The set of hyperparameters Θ includes parameters such as the learning rate α, batch size M, and the optimizer $\mathcal{A}$. The objective function for hyperparameter optimization is expressed as:

$$\Theta^* = \arg\min_{\Theta} \frac{1}{N} \sum_{n=1}^{N} \mathcal{J}_{\text{val}}^n(\Theta) \tag{12}$$

where $\Theta = \{\alpha, M, \mathcal{A}\}$, N is the number of cross-validation folds, and $\mathcal{J}_{\text{val}}^n(\Theta)$ represents the validation loss in the n-th fold.

 doi:10.20944/preprints202410.2511.v1

By optimizing $\mathcal{L}_{\text{val}}$, we indirectly reduce the training loss $\mathcal{L}_{\text{train}}$, leading to a model that generalizes better on unseen data.

2.5. Stage 5: K-Fold Cross Validation

K-Fold Cross Validation aims to assess the model's performance by dividing the dataset into K equal-sized folds. In this process, each fold is used as a validation set exactly once, while the remaining $K-1$ folds are utilized for training. The overall performance is estimated by averaging the results across all folds, providing a reliable measure of the model's generalizability. The objective function for calculating the average validation metric $\mathcal{M}_{\text{val}}$ over K folds is given by:

$$\mathcal{M}_{\text{val}} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{M}_{\text{val}}^{k} \tag{13}$$

where $\mathcal{M}_{\text{val}}^{k}$ represents the evaluation metric (such as accuracy, precision, recall, etc.) calculated on the k-th validation set. The goal is to minimize the variability across the folds to ensure consistency, often measured by:

$$\sigma_{\mathcal{M}} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left(\mathcal{M}_{\text{val}}^{k} - \mathcal{M}_{\text{val}}\right)^{2}} \tag{14}$$

where $\sigma_{\mathcal{M}}$ represents the standard deviation of the metric, providing insight into the model's stability and consistency across different validation sets.

2.6. Stage 6: Validating the Model on Different Datasets

In stage 1 to stage 5, the architecture uses 8,800 training data and 2,200 test data. In this stage we will evaluate the model on 3,582 training images and 896 test images captured in Lugol's iodine, 2,393 training images and 599 test images captured in acetic acid, 2,824 training images and 706 test images captured in normal saline. We are also validating the performance of the model on secondary dataset; Malhari containing 2232 train dataset and 558 test dataset.

2.7. Classification

The classification process is done by the ensemble prediction by the VLR model combining the normalized dynamic weight and normalized attention weight so that the training and validation loss can be adjusted and minimized accordingly.

Mathematical Basis

1. **Ensemble Prediction:** Use stored predictions and features from base models (VGG16, ResNet50, LR) on training and validation sets to perform ensemble prediction:

$$\hat{y}_{VLR} = \sum_{M \in \{\text{VGG16, ResNet50, LR}\}} W_M \, \hat{y}_M \tag{15}$$

2. **Calculate Final Combined Weights:**

$$W_M = 0.7 A_M + 0.2 \frac{1}{L_M} + 0.1 \frac{1}{T_M} \tag{16}$$

3. **Apply attention mechanism:**

$$W_M^{\text{attn}} = \text{AttentionModule}(\text{VGG16}_{\text{features}}, \text{ResNet50}_{\text{features}}, \text{LR}_{\text{output}}) \tag{17}$$

4. **Combine weights:**

 doi:10.20944/preprints202410.2511.v1

$$W_M^{\text{final}} = \alpha W_M + (1 - \alpha) W_M^{\text{attn}} \tag{18}$$

5. **Use final weights for ensemble prediction:**

$$\hat{y}_{VLR}^{\text{final}} = \sum_{M \in \{\text{VGG16, ResNet50, LR}\}} W_M^{\text{final}} \hat{y}_M \tag{19}$$

3. Algorithm

Algorithm 1 VLR Ensemble Model incorporating Attention Mechanism, Static Weights, Fine-Tuning, and K-Fold Cross Validation for VLR

Input: Training set size n; Number of epochs T_a; Number of cross-validation folds K; Pre-trained models: VGG16, ResNet50, LR; Initial weights W.
Output: Final VLR ensemble model after cross-validation.
Train Base Models: Train base models $M \in \{\text{VGG16}, \text{ResNet50}, \text{LR}\}$ on the entire training set of size n. Store model weights and predictions for later use.
for epoch $m = 90$ to T_a **do**
    Fine-tune VLR model using training set from epoch 90 to 200, adjusting hyperparameters (e.g., learning rate, batch size) to optimize metrics (accuracy, precision, recall, loss). Store fine-tuned weights and optimal hyperparameters Θ^*.
    Update Parameters: Update parameters Θ, η by gradient back-propagation.
    Initialize K-Fold Cross Validation:
    Set K (e.g., $K = 5$) and split the training dataset into K folds.
    for each fold $k = 1$ to K **do**
        Split Data: Use fold k as validation set, remaining $K - 1$ folds as training set.
        Load Fine-Tuned VLR Model: Initialize the VLR model with fine-tuned weights and optimal hyperparameters Θ^*.
        Ensemble Prediction:
        Use stored predictions and features from base models (VGG16, ResNet50, LR) on training and validation sets to perform ensemble prediction:

$$\hat{y}_{\text{VLR}} = \sum_{M \in \{\text{VGG16, ResNet50, LR}\}} W_M \hat{y}_M \tag{20}$$

        Calculate Final Combined Weights:

$$W_M = 0.7 A_M + 0.2 \frac{1}{L_M} + 0.1 \frac{1}{T_M} \tag{21}$$

        Apply attention mechanism:

$$W_M^{\text{attn}} = \text{AttentionModule}(VGG16_{\text{features}}, \text{ResNet50}_{\text{features}}, LR_{\text{output}}) \tag{29}$$

        Combine weights:

$$W_M^{\text{final}} = \alpha W_M + (1 - \alpha) W_M^{\text{attn}} \tag{22}$$

        Use final weights for ensemble prediction:

$$\hat{y}_{\text{VLR}}^{\text{final}} = \sum_{M \in \{\text{VGG16,ResNet50,LR}\}} W_M^{\text{final}} \hat{y}_M \tag{23}$$

        Evaluate Performance for Fold k**:** Compute and store validation metrics (accuracy, precision, recall, F1-Score, ROC-AUC).
    end for
    Calculate Average Performance Metrics for VLR: Compute average metrics (accuracy, precision, recall, F1-Score, ROC-AUC) across all K folds.
    VLR Validation on Different Solutions: Validate VLR on images captured in three different solutions separately.
    VLR Validation on Different Dataset: Validate VLR on images captured on a different dataset.
end for

4. Experimental Analysis

4.1. Experimental Setup

To validate the proposed architecture, we have conducted experiments on the same dataset (primary), separating the images captured in three different solutions [41]. We have also used secondary dataset, Malhari [51] to classify the three stages of Cervical Intraepithelial Neoplasia. The description of these two datasets are provided in section 4.2 and 4.3 in details. The datasets come from various hospitals and data collection centers, differing in aspects such as data size, image type, and acquisition protocols. This allows us to assess the proposed method's generalizability. We are also analyzing the model's adaptability by using K-Fold validation technique after fine tuning.

We have conducted the entire experiments in TPU-V28 high RAM environment. We have noted the following metrics in the experiment; training accuracy, validation accuracy, training loss, validation

 doi:10.20944/preprints202410.2511.v1

loss, precision, recall, F1 score and AUC. The base models were trained over 150 epochs with a batch size of 100 and 160 steps per epoch to obtain the results. VLR was trained for 200 epochs, altering the batch size and the learning rate for best validation accuracy.

4.2. Primary Dataset

We employed a dataset of 6,000 Colposcopy images provided by the International Agency for Research on Cancer[41]. The images were captured using three different solutions: Lugol's iodine, normal saline, and acetic acid. A key characteristic of these 6,000 cases is that the affected area or lesion is located within the T-Zone. We have done image augmentation to increase the image count to 11,000. There was a split of 80:20 for training and testing datasets, resulting in 8800 images considered for training and 2200 images for testing. Figure 2 and Figure 3 are the pictorial view of the dataset distribution of 11,000 images as train and test, and Figure 4 is a glimpse of dataset on which the research is performed.

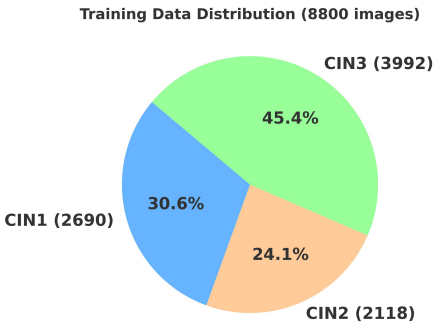

Figure 2. Distribution of training data into CIN1, CIN2, and CIN3.

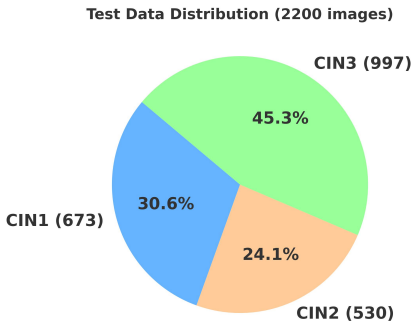

Figure 3. Distribution of test data into CIN1, CIN2, and CIN3.

 doi:10.20944/preprints202410.2511.v1

Figure 4. Visual depiction of samples in the Colposcopy dataset of CIN1, CIN2, and CIN3 classes.

4.2.1. Details of the Primary Dataset with Respect to Three Solutions

The graphical representation illustrates the distribution of image counts for three different solutions—Lugol's iodine, Acetic acid, and Normal saline—across original, augmented, and CIN classes (CIN1, CIN2, and CIN3). The original counts are significantly expanded through data augmentation, contributing to more diverse training data, which helps in improving the robustness of machine learning models. Specifically, Lugol's iodine has the highest augmented count (4478 images), followed by Normal saline (3530 images) and Acetic acid (2992 images). Figure 5 depicts the distribution of images captured in three different solutions. Subsection text.

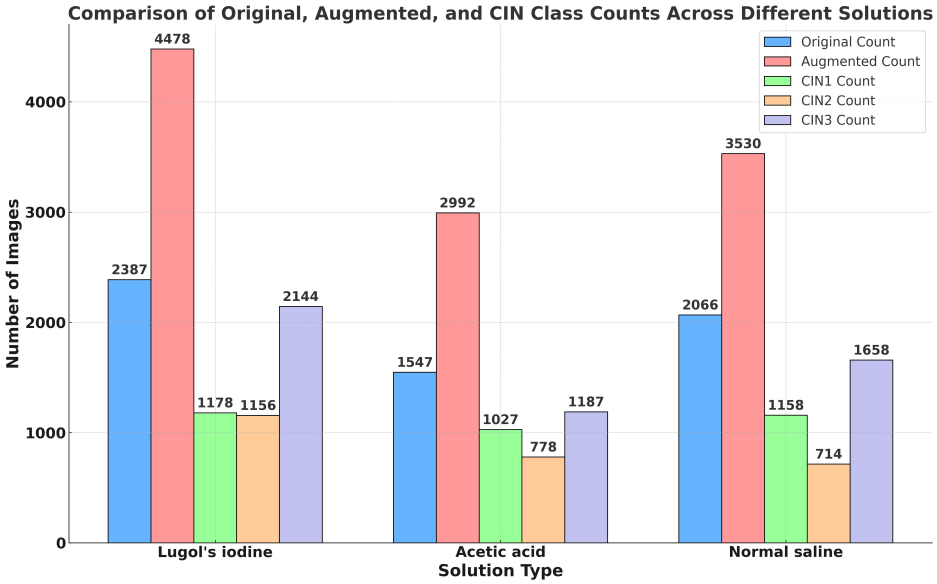

Figure 5. Distribution of dataset into CIN1, CIN2 and CIN3 in three different solutions.

 doi:10.20944/preprints202410.2511.v1

4.3. Secondary Dataset

The Malhari Colposcopy Dataset [51] can be particularly valuable when developing ensemble learning models, where multiple base models (such as VGG16, ResNet50, and Logistic Regression) are used, as it allows robust validation through both original and augmented data. This improves the reliability of predictions in real-world clinical settings. This dataset provides a rich foundation for advancing computer-aided diagnostic tools in the area of cervical cancer screening, with the potential to enhance the accuracy, robustness, and generalizability of automated models. There are a total 2,790 images divided with a train and test split of 80 : 20. There are a total of 901 total CIN1 images, a total of 931 images as CIN2 and a total of 960 images as CIN3. Figure 6 and Figure 7 depict the Malhari dataset distribution of train and test split.

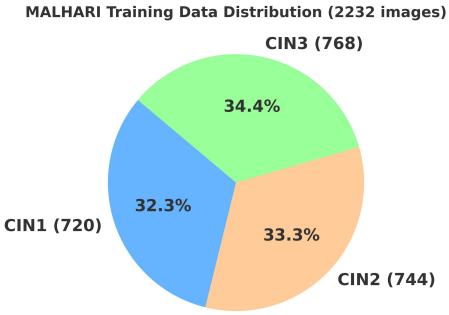

Figure 6. Malhari Training Data Distribution (2232 images).

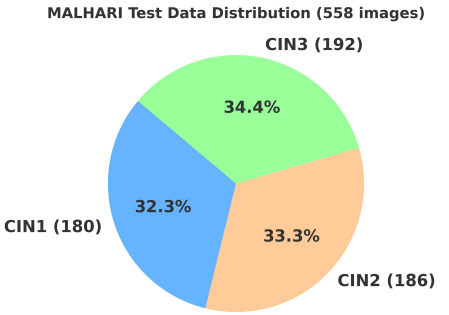

Figure 7. Malhari Test Data Distribution (558 images).

4.4. Data Preprocessing

We have implemented image preprocessing using these steps; a. gamma correction, b. data augmentation, and c. 2D t-sne. All the image preprocessing steps are performed on the primary dataset of 6,000 images.

4.4.1. Gamma Correction

In our experiment, we have considered the value of γ as 1.2 as some of the images in our dataset was very bright. Figure 9 shows gamma correction. This helped us to make the overexposed areas more discernible and improve diagnostic accuracy.

4.4.2. Data Augmentation

We have applied the data augmentation technique to increase the image count from 6,000 to 11,000. The augmentation techniques applied include a random rotation of up to 40° to introduce rotational variance, horizontal and vertical shifts of up to 20% of the image dimensions to improve

 doi:10.20944/preprints202410.2511.v1

robustness to positional changes, a shear transformation of 20% to simulate different perspectives, and random zooming in or out by up to 20% to introduce scale variations. Additionally, random horizontal flipping is used to help the model generalize better to mirrored versions of images. Any newly created pixels during these transformations are filled using the nearest pixel value, specified by the `fill_mode='nearest'` parameter. This data augmentation process helped to prevent over fitting by artificially expanding the training dataset, thereby making the model more generalized and capable of handling variations in new images. Figure 9 depicts the data augmentation result.

4.4.3. 2d t-sne

The images in the dataset are acquired under different lighting conditions, magnifications, thereby increasing the the data's dimensionality. Therefore, we applied 2d-tsne to reduce the dimension of our dataset. Fig. 8 represents 2D t-sne embedded with images.

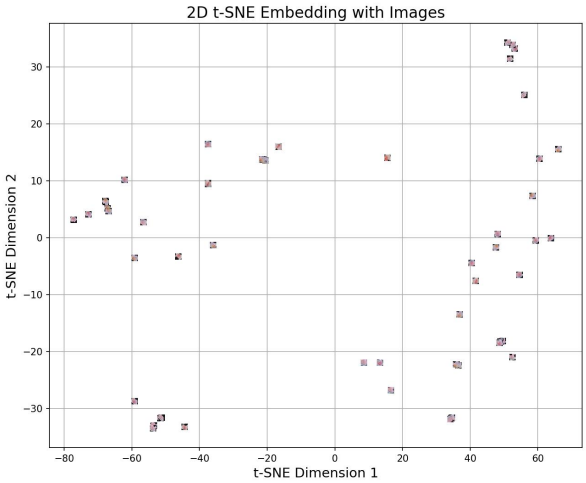

Figure 8. 2D t-SNE embedded with images.

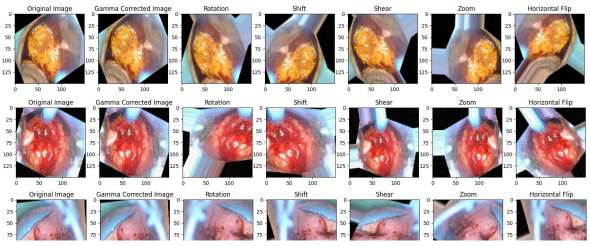

Figure 9. Original Image, Gamma correction, and data augmentation of CIN1, CIN2, and CIN3.

4.5. Experimental Setup of VLR

In this section and until section 4.7, all the experiments are carried out on 11,000 colposcopy images of the primary dataset.

We started with 8,800 training and 2,200 testing samples. The entire architecture of VLR model is explained by equation 24 and 25.

$$\text{VLR\_output} = \sum_{i=1}^{3} (w_i \cdot A_i \cdot \text{Model}_i(\text{input})) \tag{24}$$

Where: $\text{Model}_i(\text{input})$ represents the output of each model ($i = 1$ for VGG16, $i = 2$ for ResNet50, and $i = 3$ for Logistic Regression) when applied to the input data. w_i is the normalized static attention weight or predefined score assigned to each model ($w_1 = 2.5$ for VGG16, $w_2 = 3.0$ for ResNet50, and

 doi:10.20944/preprints202410.2511.v1

$w_3 = 1.0$ for Logistic Regression). A_i is the dynamic weight calculated based on validation accuracy, training loss, and training time.

The dynamic weights A_i can be computed as:

$$A_i = \alpha_1 \cdot \text{Accuracy}_i + \alpha_2 \cdot \frac{1}{\text{Loss}_i} + \alpha_3 \cdot \frac{1}{\text{Time}_i} \tag{25}$$

Where: $\alpha_1, \alpha_2, \alpha_3$ are the weighting factors for validation accuracy (0.5), training loss (0.2), and training time (0.3), respectively. Accuracy_i, Loss_i, and Time_i are the respective metrics for each model.

Dynamic Weights are assigned based on metrics such as validation accuracy, training loss, and training time, giving more importance to models with better performance and efficient training characteristics. We have tried to maintain a balance in assigning weights to these parameters.

Static Weights are computed through an attention mechanism, which dynamically adjusts the importance of each model's output based on the input features, allowing the VLR model to adaptively emphasize more relevant models for each specific instance.

The final weight is a blend of dynamic ($\alpha = 0.7$) and static components ($\alpha = 0.3$), ensuring both consistency and adaptability in the ensemble predictions. In this distribution of weight, we have prioritized dynamic components, as in the static components of assigning weight, all the models are already assigned very high raw scores. Table 1 shows the weight change over the epochs and how the loss contributed to VLR alters over the epochs. In order to represent the weight alteration trend, we are showing here results of every 20 epochs from 1 to 150th epoch. The last row of the table shows the value at the 150th epoch for all four models, with VLR plateauing after 90 epochs and reaching a training loss of **0.003317** and validation accuracy of **98.56%**. **It might seem that the accuracy is high, but in case of medical diagnostics, even small improvements in model performance can lead to better diagnostic outcomes. Reducing the loss might enhance the sensitivity and specificity of the model, potentially catching more true positives (actual abnormalities) and reducing false negatives (missed cases). This led us to carry out hyperparameter fine tuning in order to minimize the loss further.** Table 2 illustrates the working of the attention mechanism with predefined scores assigned to each model on the basis of their feature extraction capability.

Table 1. Epoch-wise Metrics for VGG16, ResNet50, LR, with VLR Loss and Final Combined Weight

Epoch	VGG16				ResNet50				LR				VLR Loss Contrib
	Val-Acc (%)	Train-Loss	Train-Time (sec)	Dyn-W_i^{norm}	Val-Acc (%)	Train-Loss	Train-Time (sec)	Dyn-W_i^{norm}	Val-Acc (%)	Train-Loss	Train-Time (sec)	Dyn-W_i^{norm}	
1	90.05	0.00912	3240	0.099	91.22	0.00880	2640	0.098	90.87	0.00850	220	0.088	0.004423
20	94.12	0.00750	3600	0.109	95.67	0.00710	3200	0.109	95.80	0.00690	180	0.098	0.003753
40	95.13	0.00730	3960	0.111	96.12	0.00690	3520	0.110	96.50	0.00660	140	0.100	0.003666
60	95.65	0.00720	4320	0.112	96.75	0.00680	3840	0.111	96.90	0.00640	100	0.101	0.003620
80	96.25	0.00711	4560	0.113	97.20	0.00660	3960	0.113	97.30	0.00620	60	0.103	0.003554
100	96.55	0.00700	4700	0.114	97.65	0.00650	4100	0.114	97.80	0.00600	40	0.104	0.003505
120	96.77	0.00711	4800	0.114	98.10	0.00631	3920	0.115	98.02	0.00349	20	0.135	0.003315
140	96.77	0.00711	4840	0.114	98.23	0.00631	3960	0.115	98.92	0.00349	17	0.136	0.003317
150	**96.77**	**0.00711**	**4860**	**0.114**	**98.23**	**0.00631**	**3960**	**0.115**	**98.92**	**0.00349**	**15**	**0.136**	**0.003317**

Formula	Final Combined Weight (W_i^{final}) = $\alpha W_i + (1-\alpha) W_i^{\text{attn}}$
Terms	α: Weightage parameter, $0 \leq \alpha \leq 1$ ($\alpha = 0.7$ for dynamic in this table) Dyn-W_i^{norm}: Dynamic Normalized Weight for model i W_i^{attn}: Static Attention Weight for model i ; VGG16 : 0.348, ResNet50: 0.574, LR: 0.078

 doi:10.20944/preprints202410.2511.v1

Table 2. Attention Weights Calculation for ResNet50, VGG16, and Logistic Regression based on scores of the attention module

Model	Raw-Score (S_i)	Exponential (e^{S_i})	Attention-Weight-Formula (W_i^{attn})	Attention-Weight-Value (W_i^{attn})
ResNet50	3.0	20.09	$\frac{e^{3.0}}{20.09+12.18+2.72}$	0.574
VGG16	2.5	12.18	$\frac{e^{2.5}}{20.09+12.18+2.72}$	0.348
Logistic-Regression (LR)	1.0	2.72	$\frac{e^{1.0}}{20.09+12.18+2.72}$	0.078

4.6. Experimental Setup of Hyperparameter Fine Tuning

If we observe closely Table 1, the loss from 120 to 150 epochs is fluctuating and not following the uniform decreasing trend. From 120th to 140th epoch, there is an increase in loss and from 140th epoch to 150th epoch there is a constant loss. To correct the loss trend and to make it more uniform, we fine-tuned the VLR model through hyperparameter optimization, particularly focusing on the learning rate, which minimized the training loss to **0.00175** with a validation accuracy of **99.89%**. There was a scope for decreasing the training loss of VLR so that the accuracy could be further increased. It was observed in the experimental setup 4.5 also that the learning rate, including all the other metrics of VLR did not change after 90th epoch. So, we have introduced hyperparameter fine tuning, changing the learning rate and the batch size, and noting the the corresponding validation loss and training loss for each checkpoint until 200th epoch. Table 3 shows how the learning rate and the batch size were altered to receive the desired training loss of **0.00175** and a validation accuracy of **99.89%**. We are presenting the checkpoints for every 10th epoch starting from 90th epoch until 200th epoch in Table 3. The values of TP, FP and FN is almost stable from 150th epochs in Table 3 .

Table 3. Hyperparameter Fine-Tuning Schedule for VLR Model Including Precision, Recall, and Confusion Matrix Calculation

Epoch	Learn Rate	Batch Size	Val Loss	Train Loss	Val Acc (%)	Precision (%)	Recall (%)	TP / FP / FN
90	0.001	100	0.0065	0.003317	98.56	98.10	97.80	2188 / 12 / 8
100	0.0005	100	0.0052	0.003304	99.10	98.25	97.85	2189 / 11 / 7
110	0.0001	80	0.0041	0.00325	99.45	98.40	97.88	2190 / 9 / 7
120	0.0001	60	0.0035	0.00275	99.60	98.50	97.90	2191 / 8 / 6
130	0.00005	60	0.0030	0.00250	99.68	98.55	97.92	2192 / 7 / 6
140	0.00005	50	0.0026	0.00230	99.72	98.60	97.95	2193 / 6 / 5
150	0.00001	50	0.0023	0.00214	99.77	98.65	97.97	2194 / 5 / 5
160	0.000005	40	0.0021	0.00205	99.80	98.68	97.99	2194 / 5 / 4
170	0.000005	40	0.0019	0.00200	99.82	98.70	98.00	2194 / 4 / 4
180	0.000001	35	0.0018	0.00192	99.85	98.72	98.03	2194 / 4 / 3
190	0.000001	35	0.0017	0.00185	99.87	98.74	98.05	2195 / 3 / 2
200	0.0000005	30	0.0015	**0.00175**	**99.89**	98.75	98.08	2195 / 3 / 2

4.7. Experimental Setup of K-Fold Cross Validation

K-fold cross-validation was performed with K set to 5, calculating metrics such as accuracy, precision, recall, F1-Score, and ROC-AUC, which were averaged across the folds to provide more stable and unbiased estimates. The training portion of each fold used 8800 samples, while the validation part used 2200 samples. This process was carried out after the 200th epoch and ran for an additional 100 epochs, using a batch size of 50. It was observed that the validation accuracy and validation loss stopped improving after 40 epochs, prompting the use of early stopping to halt the training process. During each iteration, one fold (20% of the training set, equating to 1760 samples) was used

 doi:10.20944/preprints202410.2511.v1

for validation, while the remaining four folds (7040 samples) were used for training. The separate test dataset of 2200 samples was kept aside for the final evaluation after completing all folds.

4.8. Experimental Setup of Training of Images in Three Different Solutions and Training of Secondary Dataset

In section 4.2.1 and 4.3, the details of the data distribution of images captured in three different solutions and the secondary dataset are shown. Considering this distribution of the dataset, in this experiment, we are repeating stage 2.1, stage 2.2, and stage 2.3, separately on images captured in three different solutions: Lugol's iodine, acetic acid, and normal saline and on the Malhari dataset. On the basis of Table 1 and Table 2, Table 4 show the behaviour of VLR on images captured in three different solutions and Malhari dataset at the 150th epoch. **In ensemble model, higher training loss can still lead to better validation accuracy because combining multiple models enhances generalization by capturing diverse features and reducing overfitting. The weighted contributions from different models allow the ensemble to prioritize relevant predictions dynamically, improving performance on unseen data. Thus, even with slightly higher training loss, the ensemble leverages complementary strengths to achieve superior accuracy.**

Table 4. Behaviour of VLR on different datasets based on recalculated dynamic and static weights

Dataset	**VGG16**					**ResNet50**					**LR**					**VLR-Loss**	**VLR-validation-acc**
	Val-Acc (%)	Train-Loss	Static W_i^{attn}	Dynamic W_i	Final W_i^{final}	Val-Acc (%)	Train-Loss	Static W_i^{attn}	Dynamic W_i	Final W_i^{final}	Val-Acc (%)	Train-Loss	Static W_i^{attn}	Dynamic W_i	Final W_i^{final}	Train-Loss-Contrib	Final-Acc
Lugol's-iodine	91.67	0.00821	0.348	0.3352	0.3389	92.43	0.00731	0.574	0.3713	0.4670	94.42	0.00649	0.078	0.2935	0.1903	**0.00708**	**99.21**
Acetic-acid	90.17	0.00864	0.348	0.3360	0.3396	91.23	0.00791	0.574	0.3745	0.4705	92.12	0.00744	0.078	0.2895	0.1883	**0.00746**	**99.10**
Normal-saline	87.87	0.00978	0.348	0.3366	0.3400	88.73	0.00912	0.574	0.3764	0.4723	90.04	0.00842	0.078	0.2870	0.1865	**0.00802**	**99.02**
Malhari	93.45	0.00814	0.348	0.3332	0.3366	94.21	0.00701	0.574	0.3695	0.4642	95.66	0.00631	0.078	0.2973	0.1931	**0.00695**	**99.41**

From Table 4, the difference between the loss contributed towards VLR and the training loss of LR (showing highest accuracy) in the case of Lugol's iodine, acetic acid, and Malhari dataset is **0.0005**, **0.00002**, and **0.0006**, respectively, which is almost insignificant, so we can say that VLR is performing consistently as expected on the images in Lugol's iodine, acetic acid and Malhari dataset. In case of images captured in normal saline, VLR's training loss is less than LR's training loss and it's accuracy is very similar to LR's accuracy.

5. Results and Discussions

5.1. Results and Analysis of the Proposed Model

The experiment was implemented on training dataset of 8800 images and test dataset containing 2200 images. Table 5 presents the final values for the metrics used to evaluate each model. In case of VGG16, ResNet50, and Logistic Regression (LR), the values for validation accuracy, precision, recall, F1 score, ROC-AUC, training time, and training loss represent the results after completing the 150th epoch. However, for the Variable Learning Rate (VLR) model, since hyperparameter fine-tuning was introduced after the 90th epoch, the final values provided in Table 5 reflect the metrics obtained at the 200th epoch.

 doi:10.20944/preprints202410.2511.v1

Table 5. Model Performance and Training Metrics after 150th epochs for the base models and after 200 epochs for VLR

Model	Val-Acc (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)	Train Time (sec-(TPU-V28))	Train Loss
VGG16	96.77	95.47	96.12	95.84	93.255	4860	0.00711
ResNet50	98.23	96.65	97.24	97.02	95.854	3960	0.00631
LR	98.92	97.25	97.99	97.50	97.891	15	0.00349
VLR	99.89	98.75	98.08	98.41	99.991	5040	0.00175

5.1.1. Analysis of Trends of Precision and Recall

Figures 10 and 11, depicts the trends of recall, and precision of the base models for 150th epoch. Figure 12 are the trends of recall and precision of VLR from 90th epoch to 200th epoch. In medical diagnostics, the precision and recall plots are particularly important as they provide insight into how well the models perform. As the precision graphs for individual models are not entirely smooth, they exhibit a clear upward trend, indicating that the models are progressively reducing false positives. In the case of VGG16, ResNet50, and Logistic Regression models, the recall values are higher than their corresponding precision values. This suggests that these models are more focused on minimizing false negatives, which, in turn, leads to an increase in false positives. In contrast, the VLR model exhibits a precision higher than recall, which is ideal for medical diagnostics, as it reflects a model that effectively minimizes false positives. The plots, which range from 0 to 150 epochs with intervals of 10, clearly highlight these trends.

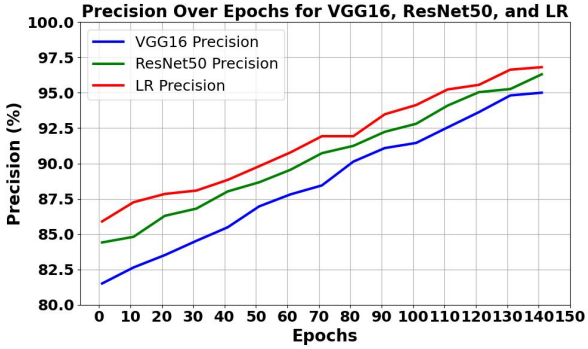

Figure 10. Precision plot of VGG16, ResNet50, and LR.

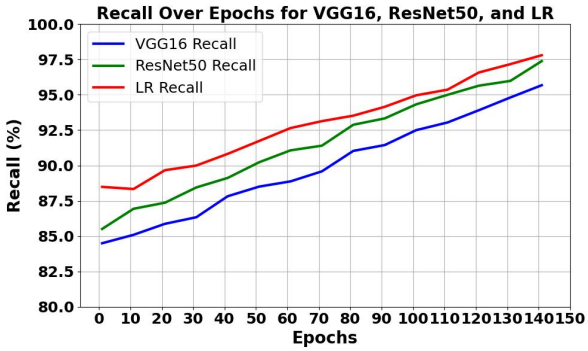

Figure 11. Recall plot of VGG16, ResNet50, and LR.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 October 2024 doi:10.20944/preprints202410.2511.v1

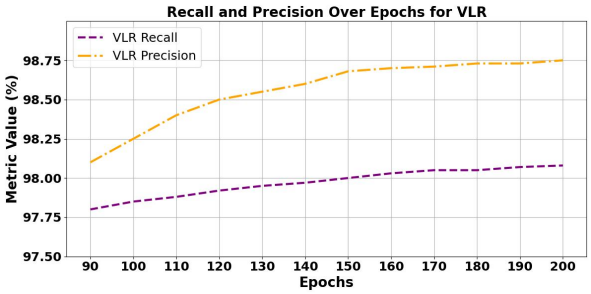

Figure 12. Recall and Precision curve of VLR from 90th to 200th epoch.

5.1.2. Analysis of trend of F1 score

Figure 13 is the F1 score of the four models. F1 score, which is an important metric saying how well the model balances between the recall and the precision. This histogram of VLR achieves the highest F1 score, indicates superior performance in balancing precision and recall compared to the other models.

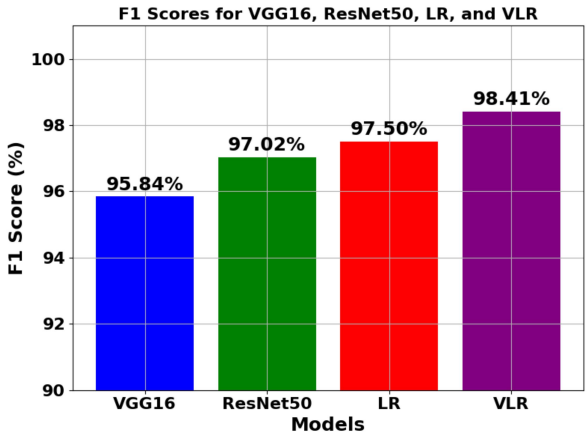

Figure 13. Analysis of F1 scores of three models.

5.1.3. Analysis of Using Normalized Dynamic and Static Attention Weight

ResNet50 has the highest contribution due to strong feature extraction and overall metrics, while VGG16 has a balanced role with moderate attention. Logistic Regression shows the lowest contribution, limited by its low attention weight despite high static metrics.

5.1.4. Analysis of ROC-AUC Graphs

Figure 14, 15, 16 and 17 are the ROC-AUC curve of the four models. As per the graphs, the VLR model shows the highest AUC (0.99) across all categories, indicating near-perfect classification. ResNet50 also performs well (AUC 0.96), followed by VGG16 (AUC 0.93). Logistic Regression achieves a strong AUC of 0.98, close to VLR, suggesting it is also a robust individual classifier despite lower feature extraction capabilities compared to VLR and ResNet50 but may not provide such high AUC for all datasets.

 doi:10.20944/preprints202410.2511.v1

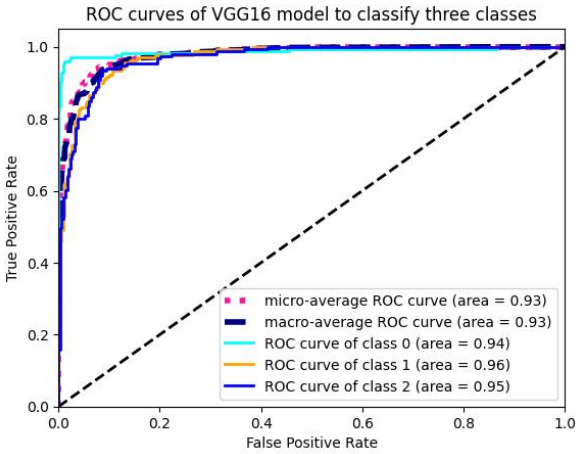

Figure 14. ROC-AUC curve of VGG16 model.

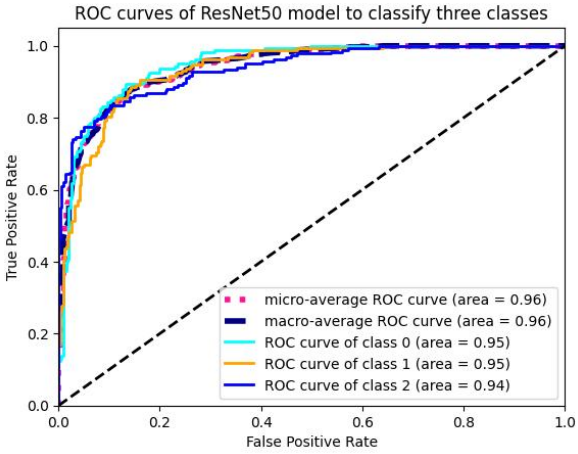

Figure 15. ROC-AUC curve of ResNet50 model.

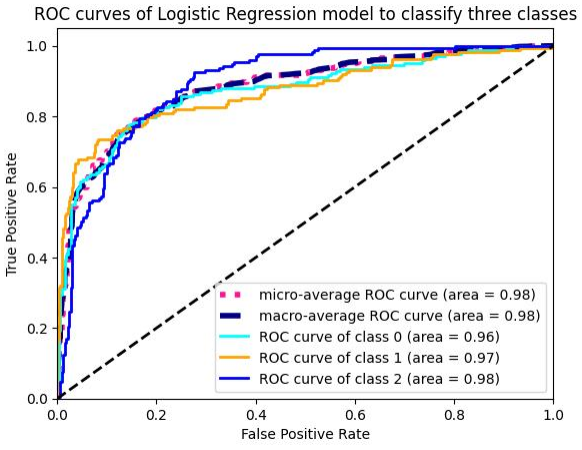

Figure 16. ROC-AUC curve of Logistic Regression model.

 doi:10.20944/preprints202410.2511.v1

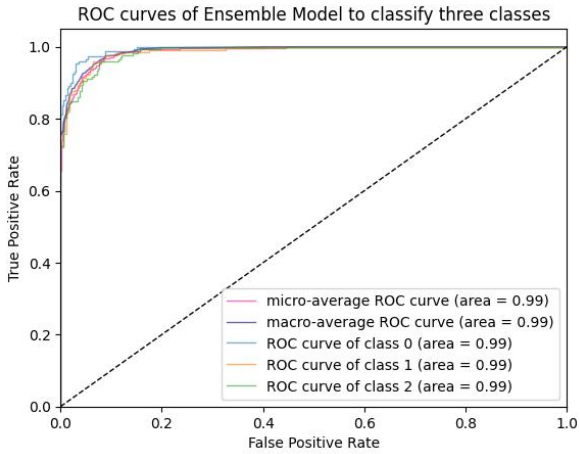

Figure 17. ROC-AUC curve of Variable Learning Rate model.

5.2. Confusion Matrix Calculation of the Four Models

Figure 18 is the confusion matrix plot of Table 6. The confusion matrices for the four models (VGG16, ResNet50, LR, and VLR) show a clear trend of improvement across the models. The VGG16 and ResNet50 models exhibit more mis-classifications between the classes, as indicated by higher false positives and false negatives. As we move to Logistic Regression (LR), the number of mis-classifications decreases significantly, indicating better performance. VLR shows the fewest misclassifications, with almost all predictions being correct, as reflected in the high diagonal values (True Positives). The plot of the confusion matrix suggests that VLR provides the highest classification accuracy among the models, followed closely by LR, with VGG16 and ResNet50 performing slightly worse.

Table 6. Formulas and Confusion Matrix Values for VGG16, ResNet50, LR, and VLR on Test Data (2200 Images)

Metric	**Formula**	**VGG16 (%)**	**ResNet50 (%)**	**LR (%)**	**VLR (%)**
Validation Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	96.77	98.23	98.92	99.89
Precision (P)	$\frac{TP}{TP+FP}$	95.47	96.25	97.25	98.75
Recall (R)	$\frac{TP}{TP+FN}$	96.12	97.24	97.99	98.08
True Positives (TP)	$TP = R \times (TP + FN)$	2115	2160	2189	2195
False Positives (FP)	$FP = \frac{TP}{P} - TP$	30	20	7	3
False Negatives (FN)	$FN = \frac{TP}{R} - TP$	55	20	4	2

 doi:10.20944/preprints202410.2511.v1

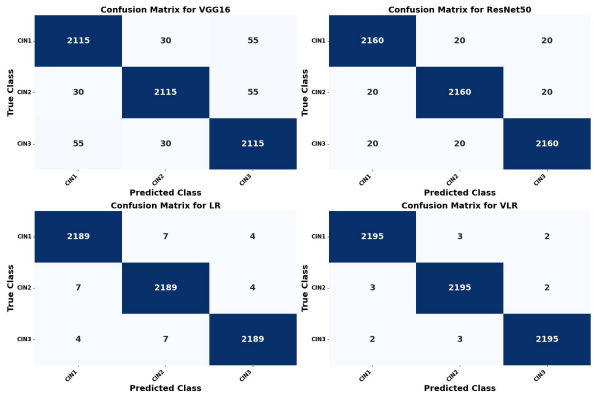

Figure 18. Heat map visualization highlighting the classification performance of each model.

5.3. K-Fold Cross Validation Results of VLR

Table 7 presents the final results of the K-fold cross-validation, with K set to 5, depicting very consistent performance metrics across K-folds until the 50th epoch, as early stopping was employed. The minimal standard deviation across all metrics, indicates stability in the model's ability to generalize well. The high consistency in validation accuracy (99.89%) and ROC-AUC (99.99%) suggests excellent discriminative power, while training loss shows negligible variation, confirming reliable learning without overfitting. Precision, recall, and F1-score also exhibit minimal variance, reflecting balanced and stable classification performance.

Table 7. K-Fold Cross Validation Results for VLR Model after Fine-Tuning

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean (%)	Standard Deviation (%)
Validation Accuracy (%)	99.90	99.88	99.91	99.87	99.89	99.89	0.02
Training Accuracy (%)	99.96	99.95	99.95	99.94	99.95	99.95	0.01
Precision (%)	98.76	98.77	98.74	98.75	98.76	98.75	0.01
Recall (%)	98.09	98.08	98.11	98.06	98.07	98.08	0.02
F1-Score (%)	98.47	98.45	98.48	98.44	98.46	98.46	0.02
ROC-AUC (%)	99.99	99.98	99.99	99.98	99.99	99.99	0.01
Training Loss	0.00176	0.00174	0.00175	0.00173	0.00175	0.00175	0.00002

5.4. Result Analysis of VLR on Images Captured in Three Solutions and Malhari Dataset

From Table 8, the Malhari dataset shows the best overall performance in terms of precision, recall, and F1 score, indicating better model robustness on this dataset. There is a slight drop in the model's performance on normal saline, suggesting that additional fine-tuning or adjustments might be needed for this solution.

Table 8. Performance of VLR on images captured using three different solutions and Malhari dataset

Solution	Train Accuracy (%)	Validation Accuracy (%)	Precision%	Recall-%	F1-Score%
Lugol's iodine	99.41	99.21	98.69	97.81	98.22
Acetic acid	99.23	99.10	98.14	97.23	97.64
Normal saline	99.14	99.02	97.12	96.04	96.56
Malhari	**99.74**	**99.41**	**98.62**	**98.11**	**98.36**

 doi:10.20944/preprints202410.2511.v1

5.5. *Final Classification by the VLR Model on the Primary Dataset*

Figure 19 is the test result of the classification done by the ensemble model (VLR). These test results are verified by the oncologist as CIN1, CIN2 and CIN3. First row is being classified as CIN1, second and third row are classified as CIN2 and the last two rows are classified as CIN3.

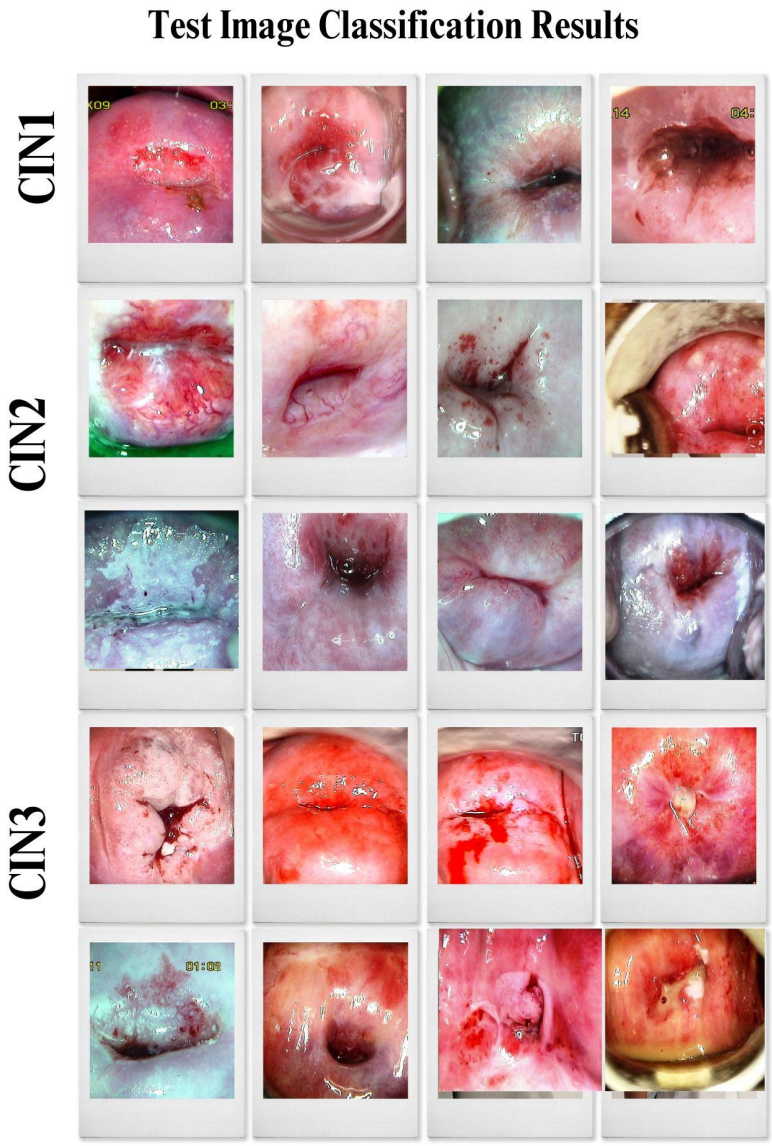

Figure 19. Test Dataset is classified by VLR as CIN1, CIN2 and CIN3.

6. Related Work

6.1. *Comparative Analysis Based on Colposcopy Dataset*

Table 9 shows the comparative analysis among VLR and the existing cervical cancer classification techniques on the colposcopy dataset. It is found that each approach achieves remarkable results. DL [4] achieved remarkable recall value as 100%. Compared to all the other models, VLR achieves better accuracy, recall, precision and F1-score. Bold values indicate the higher performance.

 doi:10.20944/preprints202410.2511.v1

Table 9. Performance of other models on Colposcopy dataset

Model	Accuracy (%)	Precision	Recall	F1-Score
MFEM-CIN[7]	89.2	92.3	88.16	90.18
DL[4]	92	79.4	100	97.58
CNN[44]	87	75	88	80.55
CNN[45]	90	NA	NA	NA
CNN[46]	84	96	99	97.47
DenseNet-CNN[47]	73.08	44	87	58.44
CLD Net[48]	92.53	85.56	NA	NA
Kappa[49]	67	89	33	76.44
Caps Net[50]	99	NA	NA	NA
E-GCN[3]	81.85	81.97	81.78	81.87
VLR	**99.95**	**98.75**	**98.08**	**98.41**

6.2. Comparison with Existing Methods

We have used the dataset acquired from International Agency for Research on Cancer. We have done a systematic year-wise review of similar work on cervical cancer classification shown in Table 10. Our proposed model achieved the best result in terms of accuracy, precision, and recall, and we have also adhered to the use of the correct dataset.

Table 10. A Comprehensive Review of Deep Learning Techniques for Cervical Cancer Classification on Various Datasets

Reference	Dataset	Method	Accuracy	Remarks
[7]2024	Colposcopy	CNN, Transformer	89.2%	Employs a hybrid model of MFEM-CIN and Transformer.
[12]2024	Pap Smear	CNN, ML, DL	99.95%	Combines computational tools like CNN-LSTM hybrids, KNN, and SVM.
[16]2024	Pap Smear	Deep Learning	92.19%	Employs CerviSegNet-DistillPlus for classification.
[15]2024	Biopsy	Machine Learning	98.19%	Employs ensemble of machine learning algorithms for classification.
[17]2024	Colposcopy	Deep Learning	94.55%	Has used a hybrid deep neural network for segmentation.
[2]2023	Pap Smear	Deep Learning	97.18%	Uses deep learning techniques integrated with advanced augmentation techniques such as CutOut, MixUp, and CutMix.
[4]2023	Colposcopy	Deep Learning	92%	Uses predictive deep learning model.
[30]2022	Pap Smear	CNN, PSO, DHDE	99.7%	Applies CNN for a multi-objective problem, and PSO and DHDE are used for optimization.
[14]2022	Cytology	ANN	98.87%	Uses artificial Jellyfish Search optimizer combined with an ANN.

 doi:10.20944/preprints202410.2511.v1

Table 10. *Cont.*

Reference	Dataset	Method	Accuracy	Remarks
[9]2021	Colposcopy	Deep Learning	92%	Uses Deep neural techniques for cervical cancer classification.
[11]2021	Colposcopy	Deep Learning	90%	Using deep neural network generated attention maps for segmentation.
[10]2021	Colposcopy	Residual Learning	90%, 99%	Employed residual network using Leaky ReLU and PReLU for classification.
[29]2021	MR-CT Images	GAN	-	Uses a conditional generative adversarial network (GAN).
[21]2021	Pap Smear	Biosensors	-	Uses biosensors for higher accuracy.
[5]2021	Cervical Pathology Images	SVM, k-Nearest Neighbors, CNN, RF	90%-89.1%	Uses ResNet50 model as a feature extractor and selects k-NN, Random Forest, and SVM for classification.
[1]2021	Colposcopy	CNN	99%	Uses Faster Small-Object Detection Neural Networks.
[13]2021	Pap Smear	Deep Convolutional Neural Network	95.628%	Constructs a CNN called DeepCELL with multiple kernels of varying sizes.
[28]2020	MRI Data of Cervix	Statistical Model	-	A statistical model called LM is used for outlier detection in lognormal distributions.
[3]2020	Colposcopy	CNN	81.95%	Employs a graph convolutional network with edge features (E-GCN).
[8]2020	Colposcopy	Deep Learning	83.5%	Has used K-means for classification, CNN and XGBoost to combine the CNN decesion.
[6]2019	Colposcopy	CNN	96.13%	Uses a recurrent convolutional neural network for classification of cervigrams .
Our Method	Colposcopy	Deep Neural Network	99.95%	An ensemble model called Variable Learning Rate specially designed to increase the accuracy.

6.3. Gaps identified and corrective measures taken

Considering Table 9 and Table 10, we have identified the following gaps, and we have provided the corrective measures shown in Table 11.

Table 11. Gaps identified and corrective measures

Sl.No.	Gaps	Corrective Measures
1.[3],[9]	less classification accuracy	VLR projects a training accuracy of **99.95%**
2.[12],[29]	variations in dataset used for classification	correct dataset used; colposcopy
3.[4],[44]	recall values are higher than precision	VLR projects precision as **98.75%** and recall as **98.08%**

7. Conclusion

We have used a total of 11,000 positive Colposcopy images in our architecture. In the preprocessing step, we have used gamma correction, t-sne and then K-means clustering. The Silhouette score and

 doi:10.20944/preprints202410.2511.v1

the Purity scores were used to analyze the quality of the clusters. We have used base models, VGG16, ResNet50 and Logistic regression , trained them separately on the 11,000 Cervigram images. We have noted the validation accuracy of 96.77% for VGG16, 98.23% for Resnet50, 98.92% for logistic regression, and 99.89% for VLR. We have used forward and backward propagation for VLR model to make the model for robust. As the target in the research work was to enhance the accuracy of classification, we have achieved that by reducing the loss of the VLR model using hyperparameter fine tuning. K-fold cross validation technique was used to generalize the performance of VLR. The results of K-fold technique was almost close to the results after fine tuning. The performance of VLR was also validated on the images captured using three different solutions and a secondary dataset. It was found that images captured in secondary dataset reflected best performance. VLR, being an ensemble model has shown better validation accuracy on different dataset because of it's weighted learning.

Author Contributions: All the authors have equally contributed to this research.

Funding: This work is not funded from any external sources.

Data Availability Statement: The data which is used for this research was being gathered directly from IARC, WHO.

Conflicts of Interest: The authors have no conflicts of interests related to this research.

References

1. Elakkiya, R., Subramaniyaswamy, V., Vijayakumar, V., and Mahanti, A., *Cervical cancer diagnostics healthcare system using hybrid object detection adversarial networks*, IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 4, pp. 1464–1471, 2021.
2. Youneszade, N., Marjani, M., and Pei, C. P., *Deep learning in cervical cancer diagnosis: architecture, opportunities, and open research challenges*, IEEE Access, vol. 11, pp. 6133–6149, 2023.
3. Li, Y., Chen, J., Xue, P., Tang, C., Chang, J., Chu, C., Ma, K., Li, Q., Zheng, Y., and Qiao, Y., *Computer-aided cervical cancer diagnosis using time-lapsed colposcopic images*, IEEE Transactions on Medical Imaging, vol. 39, no. 11, pp. 3403–3415, 2020.
4. Youneszade, N., Marjani, M., and Ray, S. K., *A predictive model to detect cervical diseases using convolutional neural network algorithms and digital colposcopy images*, IEEE Access, vol. 11, pp. 59882–59898, 2023.
5. Zhang, S., Chen, C., Chen, F., Li, M., Yang, B., Yan, Z., and Lv, X., *Research on application of classification model based on stack generalization in staging of cervical tissue pathological images*, IEEE Access, vol. 9, pp. 48980–48991, 2021.
6. Yue, Z., Ding, S., Zhao, W., Wang, H., Ma, J., Zhang, Y., and Zhang, Y., *Automatic CIN grades prediction of sequential cervigram image using LSTM with multistate CNN features*, IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 3, pp. 844–854, 2019.
7. Chen, P., Liu, F., Zhang, J., and Wang, B., *MFEM-CIN: A lightweight architecture combining CNN and Transformer for the classification of pre-cancerous lesions of the cervix*, IEEE Open Journal of Engineering in Medicine and Biology, vol. 5, pp. 216–225, 2024.
8. Luo, Y.-M., Zhang, T., Li, P., Liu, P.-Z., Sun, P., Dong, B., and Ruan, G., *MDFI: multi-CNN decision feature integration for diagnosis of cervical precancerous lesions*, IEEE Access, vol. 8, pp. 29616–29626, 2020.
9. Pal, A., Xue, Z., Befano, B., Rodriguez, A. C., Long, L. R., Schiffman, M., and Antani, S., *Deep metric learning for cervical image classification*, IEEE Access, vol. 9, pp. 53266–53275, 2021.
10. Adweb, K. M. A., Cavus, N., and Sekeroglu, B., *Cervical cancer diagnosis using very deep networks over different activation functions*, IEEE Access, vol. 9, pp. 46612–46625, 2021.
11. Yue, Z., Ding, S., Li, X., Yang, S., and Zhang, Y., *Automatic acetowhite lesion segmentation via specular reflection removal and deep attention network*, IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3529–3540, 2021.
12. Bappi, J. O., Rony, M. A. T., Islam, M. S., Alshathri, S., and El-Shafai, W., *A novel deep learning approach for accurate cancer type and subtype identification*, IEEE Access, vol. 12, pp. 94116–94134, 2024, doi: 10.1109/ACCESS.2024.3422313.
13. Fang, M., Lei, X., Liao, B., and Wu, F.-X., *A deep neural network for cervical cell detection based on cytology images*, SSRN, vol. 13, pp. 4231806, 2022.

14. Devarajan, D., Alex, D. S., Mahesh, T. R., Kumar, V. V., Aluvalu, R., Maheswari, V. U., and Shitharth, S., *Cervical cancer diagnosis using intelligent living behavior of artificial jellyfish optimized with artificial neural network*, IEEE Access, vol. 10, pp. 126957–126968, 2022.
15. Sahoo, P., Saha, S., Mondal, S., Seera, M., Sharma, S. K., and Kumar, M., *Enhancing computer-aided cervical cancer detection using a novel fuzzy rank-based fusion*, IEEE Access, vol. 11, pp. 145281–145294, 2023, doi: 10.1109/ACCESS.2023.3346764.
16. Al Qathrady, M., Shaf, A., Ali, T., Farooq, U., Rehman, A., Alqhtani, S. M., and Alshehri, M. S., *A novel web framework for cervical cancer detection system: A machine learning breakthrough*, IEEE Access, vol. 12, pp. 41542–41556, 2024.
17. He, Y., Liu, L., Wang, J., Zhao, N., and He, H., *Colposcopic image segmentation based on feature refinement and attention*, IEEE Access, vol. 12, pp. 40856–40870, 2024.
18. Ramzan, Z., Hassan, M. A., Asif, H. M. S., and Farooq, A., *A machine learning-based self-risk assessment technique for cervical cancer*, Current Bioinformatics, vol. 16, no. 2, pp. 315–332, Feb. 2021.
19. Parra, S., Carranza, E., Coole, J., Hunt, B., Smith, C., Keahey, P., Maza, M., Schmeler, K., and Richards-Kortum, R., *Development of low-cost point-of-care technologies for cervical cancer prevention based on a single-board computer*, IEEE Journal of Translational Engineering in Health and Medicine, vol. 8, pp. 1–10, 2020.
20. Huang, P., Zhang, S., Li, M., Wang, J., Ma, C., Wang, B., and Lv, X., *Classification of cervical biopsy images based on LASSO and EL-SVM*, IEEE Access, vol. 8, pp. 24219–24228, 2020.
21. Ilyas, Q. M., and Ahmad, M., *An enhanced ensemble diagnosis of cervical cancer: A pursuit of machine intelligence towards sustainable health*, IEEE Access, vol. 9, pp. 12374–12388, 2021.
22. Nour, M. K., Issaoui, I., Edris, A., Mahmud, A., Assiri, M., and Ibrahim, S. S., *Computer-aided cervical cancer diagnosis using gazelle optimization algorithm with deep learning model*, IEEE Access, 2024.
23. Jacot-Guillarmod, M., Balaya, V., Mathis, J., Hübner, M., Grass, F., Cavassini, M., Sempoux, C., Mathevet, P., and Pache, B., *Women with cervical high-risk human papillomavirus: Be aware of your anus! The ANGY cross-sectional clinical study*, Cancers, vol. 14, no. 20, pp. 5096, 2022, doi: 10.3390/cancers14205096.
24. Bucchi, L., Costa, S., Mancini, S., Baldacchini, F., Giuliani, O., Ravaioli, A., Vattiato, R., et al., *Clinical epidemiology of microinvasive cervical carcinoma in an Italian population targeted by a screening programme*, Cancers, vol. 14, no. 9, pp. 2093, 2022, doi: 10.3390/cancers14092093.
25. Tantari, M., Bogliolo, S., Morotti, M., Balaya, V., Buenerd, A., Magaud, L., et al., *Lymph node involvement in early-stage cervical cancer: Is lymphangiogenesis a risk factor?*, Cancers, vol. 14, no. 1, pp. 212, 2022.
26. Cho, B.-J., Choi, Y. J., Lee, M.-J., Kim, J. H., Son, G.-H., Park, S.-H., and Kim, H.-B., *Classification of cervical neoplasms on colposcopic photography using deep learning*, Scientific Reports, vol. 10, no. 1, pp. 13652, 2020, doi: 10.1038/s41598-020-70490-4.
27. Yao, K., Huang, K., Sun, J., and Hussain, A., *PointNu-Net: Keypoint-assisted convolutional neural network for simultaneous multi-tissue histology nuclei segmentation and classification*, IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, no. 1, pp. 802–813, Feb. 2024, doi: 10.1109/TETCI.2023.3281864.
28. Jiang, X., Li, J., Kan, Y., Yu, T., Chang, S., Sha, X., Zheng, H., and Wang, S., *MRI-based radiomics approach with deep learning for prediction of vessel invasion in early-stage cervical cancer*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 3, pp. 995–1002, 2020.
29. Baydoun, A., Xu, K. E., Heo, J. U., Yang, H., Zhou, F., Bethell, L. A., Fredman, E. T., et al., *Synthetic CT generation of the pelvis in patients with cervical cancer: A single input approach using generative adversarial network*, IEEE Access, vol. 9, pp. 17208–17221, 2021.
30. Kaur, M., Singh, D., Kumar, V., and Lee, H.-N., *MLNet: Metaheuristics-based lightweight deep learning network for cervical cancer diagnosis*, IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 10, pp. 5004–5014, 2022.
31. Jiang, Z.-P., Liu, Y.-Y., Shao, Z.-E., and Huang, K.-W., *An improved VGG16 model for pneumonia image classification*, Applied Sciences, vol. 11, no. 23, pp. 11185, 2021, doi: 10.3390/app112311185.
32. Tammina, S., *Transfer learning using VGG-16 with deep convolutional neural network for classifying images*, International Journal of Scientific and Research Publications, vol. 9, no. 10, pp. 143–150, 2019.

33. Kareem, R. S. A., Tilford, T., and Stoyanov, S., *Fine-grained food image classification and recipe extraction using a customized deep neural network and NLP*, Computers in Biology and Medicine, vol. 175, pp. 108528, 2024, doi: 10.1016/j.compbiomed.2024.108528.
34. Ali, H., and Chen, D., *A survey on attacks and their countermeasures in deep learning: Applications in deep neural networks, federated, transfer, and deep reinforcement learning*, IEEE Access, 2023.
35. Prakash, A. S. J., and Sriramya, P., *Accuracy analysis for image classification and identification of nutritional values using convolutional neural networks in comparison with logistic regression model*, Journal of Pharmaceutical Negative Results, pp. 606–611, 2022.
36. Das, P., and Pandey, V., *Use of logistic regression in land-cover classification with moderate-resolution multispectral data*, Journal of the Indian Society of Remote Sensing, vol. 47, no. 8, pp. 1443–1454, 2019, doi: 10.1007/s12524-019-00974-2.
37. Alassar, Z., *Decision Tree as an Image Classification Technique*, Department of City and Regional Planning, Faculty of Architecture, Akdeniz University, vol. 18, no. 4, pp. 1–7, 2020.
38. Lee, J., Sim, M. K., and Hong, J.-S., *Assessing Decision Tree Stability: A Comprehensive Method for Generating a Stable Decision Tree*, IEEE Access, vol. 12, pp. 90061–90072, 2024, doi: 10.1109/ACCESS.2024.3419228.
39. Luo, Z., Li, J., and Zhu, Y., *A deep feature fusion network based on multiple attention mechanisms for joint iris-periocular biometric recognition*, IEEE Signal Processing Letters, vol. 28, pp. 1060–1064, 2021, doi: 10.1109/LSP.2021.3079850.
40. Chen, Z., Han, X., and Ma, X., *Combining contextual information by integrated attention mechanism in convolutional neural networks for digital elevation model super-resolution*, IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp. 1–16, 2024, doi: 10.1109/TGRS.2024.3423716.
41. International Agency for Research on Cancer (IARC), *IARC: International Agency for Research on Cancer*, 2023. Available: https://www.iarc.who.int/.
42. Jiang, X., Li, J., Kan, Y., Yu, T., Chang, S., Sha, X., Zheng, H., and Wang, S., *MRI-based radiomics approach with deep learning for prediction of vessel invasion in early-stage cervical cancer*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 3, pp. 995–1002, 2020.
43. Wang, C., Zhang, J., and Liu, S., *Medical ultrasound image segmentation with deep learning models*, IEEE Access, vol. 11, pp. 10158–10168, 2023, doi: 10.1109/ACCESS.2022.3225101.
44. Skerrett, E., Miao, Z., Asiedu, M. N., Richards, M., Crouch, B., Sapiro, G., Qiu, Q., and Ramanujam, N., *Multicontrast Pocket Colposcopy Cervical Cancer Diagnostic Algorithm for Referral Populations*, BME Frontiers, vol. 2022, pp. 9823184, 2022, doi: 10.34133/2022/9823184.
45. Gaona, Y. J., Malla, D. C., Crespo, B. V., Vicuña, M. J., Neira, V. A., Dávila, S., and Verhoeven, V., *Radiomics diagnostic tool based on deep learning for colposcopy image classification*, Diagnostics, vol. 12, no. 7, pp. 1694, 2022, doi: 10.3390/diagnostics12071694.
46. Allahqoli, L., Laganà, A. S., Mazidimoradi, A., Salehiniya, H., Günther, V., Chiantera, V., Goghari, S. K., Ghiasvand, M. M., Rahmani, A., Momenimovahed, Z., and Alkatout, I., *Diagnosis of cervical cancer and pre-cancerous lesions by artificial intelligence: A systematic review*, Diagnostics, vol. 12, no. 11, pp. 2771, 2022.
47. Zhang, T., Luo, Y.-M., Li, P., Liu, P.-Z., Du, Y.-Z., Sun, P., Dong, B., and Xue, H., *Cervical precancerous lesions classification using pre-trained densely connected convolutional networks with colposcopy images*, Biomedical Signal Processing and Control, vol. 55, article no. 101566, Jan. 2020, doi: https://doi.org/10.1016/j.bspc.2019.101566.
48. Bai, B., Du, Y., Liu, P., Sun, P., Li, P., and Lv, Y., *Detection of cervical lesion region from colposcopic images based on feature reselection*, Biomedical Signal Processing and Control, vol. 57, pp. 101785, 2020, doi: 10.1016/j.bspc.2019.101785.
49. Tanaka, Y., Ueda, Y., Kakubari, R., Kakuda, M., Kubota, S., Matsuzaki, S., Okazawa, A., Egawa-Takata, T., Matsuzaki, S., Kobayashi, E., and Kimura, T., *Histologic correlation between smartphone and colposcopic findings in patients with abnormal cervical cytology: Experiences in a tertiary referral hospital*, American Journal of Obstetrics and Gynecology, vol. 221, no. 3, pp. 241.e1–241.e6, Sept. 2019, doi: 10.1016/j.ajog.2019.04.039.

 doi:10.20944/preprints202410.2511.v1

50. Zhang, X., and Zhao, S.-G., *Cervical image classification based on image segmentation preprocessing and a CapsNet network model*, International Journal of Imaging Systems and Technology, vol. 29, no. 1, pp. 19–28, 2019, doi: 10.1002/ima.22291.
51. Malhari Colposcopy Dataset original, aug & combined, *Malhari Colposcopy Dataset original, aug & combined*, 2024. Available: https://www.kaggle.com/datasets/srijanshovit/malhari-colposcopy-dataset/suggestions?status=pending&yourSuggestions=true.