Chapter 7

Enhanced Images Grading

The grading of smart colposcopy images refers to evaluating and categorizing the visual appearance of cervical abnormalities observed in smart colposcopy images. The grading system, which categorizes abnormal cell changes into different grades based on severity. The CIN system typically includes grades CIN1, CIN2, and CIN3, with CIN3 being the most severe and potentially indicating the presence of precancerous or cancerous cells. During the grading process, specular reflection impacts the accuracy of grading process, leading to false predictions. Enhanced images overcome this problem by detecting and removing the glare region. This chapter compares the enhanced and non-enhanced images graded using the classification model to justify the need to address glare regions in smart colposcopy images during the grading process.

7.1 Overview of this Chapter

Specular reflection is a prevalent issue in medical imaging, notably in colposcopy images. It arises when light reflects off a shiny surface, resulting in bright spots or glare within the captured images. These reflections often obscure crucial details, potentially leading to inaccuracies in medical condition diagnoses and grading. To address this challenge, techniques have been developed to detect, segment, and remove specular reflections in colposcopy images using deep learning methods. The resulting images, termed enhanced images, are free from specular reflections. Deep learning classification models like DenseNet121, EfficientNet, and VGG19 are used to grade the enhanced images. In the context of smart colposcopy, the enhanced images obtained after detecting and removing specular reflection have shown promising results in improving grading accuracy. Smart colposcopy involves using advanced imaging technologies to assist healthcare professionals in diagnosing and assessing cervical abnormalities. Enhancing the quality of images improves the visibility of critical features, enabling more accurate interpretation and grading of the images. Deep learning models such as DenseNet121, EfficientNet, and VGG19 are well-suited for this task because they can learn complex patterns and features from images. These models are trained using enhanced and non-enhanced colposcopy images to check the performance of the grading process. The overview of this chapter is shown in the Figure 7.1

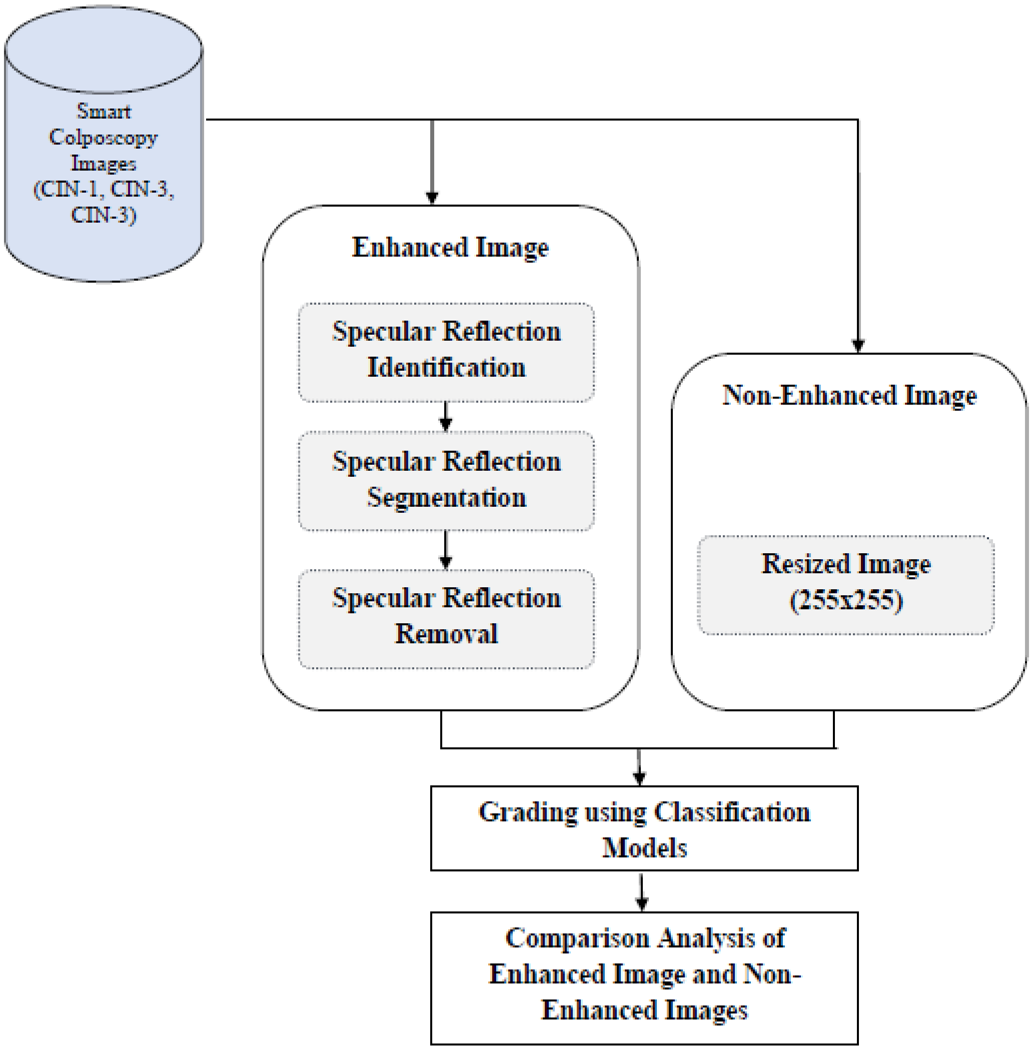


Figure 7.1. Overview of the Chapter for Improvising the Grading of Smart Colposcopy Images

During training, the models learn to differentiate between the three types of images, recognizing the distinctive characteristics of enhanced images that have undergone specular reflection removal. The training process involves presenting the models with various enhanced and non-enhanced images and corresponding labels indicating the grading or diagnosis. The models learn to extract relevant features and patterns from the images and map them to the appropriate grading or diagnosis category. Once trained, the models are tested with the unseen colposcopy images. In case of enhanced images, detecting and removing specular reflection allow the models to access more precise and more informative details, leading to higher accuracy in the grading process. The use of enhanced images in smart colposcopy grading offers several benefits. Firstly, it reduces the influence of specular reflection artifacts, which can interfere with interpreting essential features. By eliminating these artifacts, the models can focus on the actual structures and abnormalities in the images, improving the overall accuracy of the grading process. Furthermore, enhanced images provide better visualization of subtle details, enabling healthcare professionals to make more informed decisions. This can significantly impact patient care, as accurate grading plays a crucial role in determining the appropriate treatment and follow-up procedures.

7.2. Training CNN Classification Model for Grading of Smart Colposcopy Image

The CNN models such as DenseNet121, VGG19, and EfficientNet can effectively grade smart colposcopy images. These models are higher choices for image classification tasks and can takeout essential features from images to make accurate predictions.

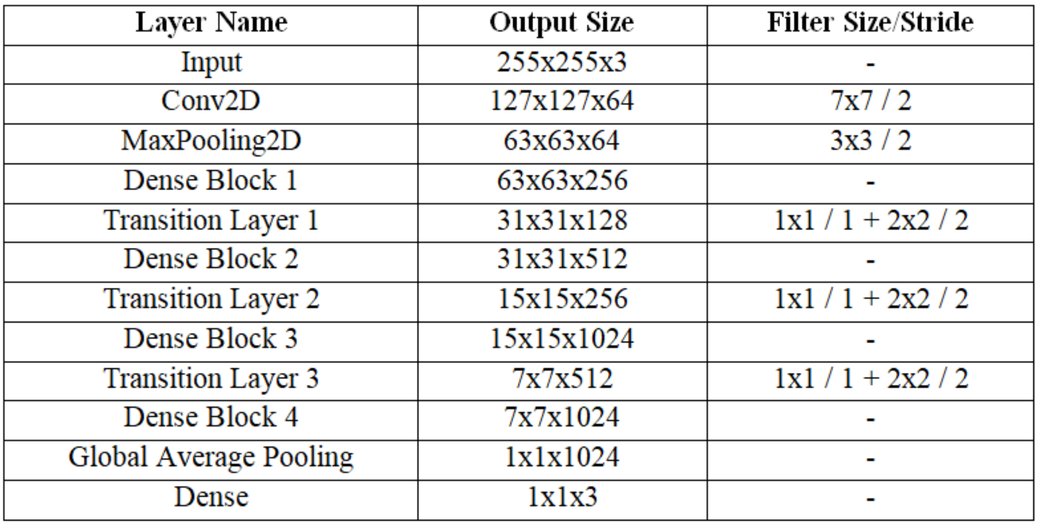
7.2.1. DenseNet121

The key idea behind DenseNet is to address the vanishing gradient problem and encourage feature reuse within the network by introducing dense connections between layers [133]. DenseNet-121's architecture consists of multiple dense blocks, transition layers, and a classification head. Here's a description of each component:

- **Dense Blocks**: It is the fundamental components of the DenseNet model. Convolutional layers with the same output shape make up each dense block. The output feature maps of each layer within a dense block are aggregated and fed as input to the following layers within the same block.
- **Transition Layers**: Transition layers are used to downsample the spatial dimensions of feature maps between dense blocks. It comprises of a 1x1 convolutional layer and a 2x2 average pooling layer. The 1x1 convolution reduces the feature map numbers, reducing computational complexity, by reducing the spatial dimensions by 2.
- **Classification Head**: The classification head is responsible for producing the final predictions. A global average pooling layer is utilized to decrease feature maps spatial dimensions to the size of 1x1. To generate class probabilities, an FC layer and a SoftMax are then used.

The input layer represents the input image with dimensions of 255x255 pixels and 3 channels (RGB). The convolutional layer (Conv2D) applies 64 filters with a filter size of 7x7 and 2 as stride. It decreases the spatial dimensions of the input, resulting in an output size of 127x127x64. The pooling layer is set with the pool size of 3x3 and 2 as stride. It further reduces the spatial dimensions, resulting in an output size of 63x63x64. The block 1 consists of multiple convolutional layers that preserve the spatial dimensions, resulting in an output size of 63x63x256. The specific number of layers and their configurations depend on the DenseNet variant. The transition layer 1 combines a 1x1 convolutional layer to decrease the feature maps number to 128 and a 2x2 average pooling layer with the stride of value 2. It decreases the spatial dimensions to 31x31x128. Dense Block 2 follows the same principle as Dense Block 1, producing an output size of 31x31x512. Similar to transition layer 1, the layer 2 reduces the spatial dimensions to 15x15x256. The dense block 3 continues to increase the number of feature maps and maintains the spatial dimensions, resulting in an output size of 15x15x1024. The transition layer 3 reduces the spatial dimensions to 7x7x512. The dense block four further expands the number of feature maps while keeping the spatial dimensions intact, resulting in an output size of 7x7x1024. The global average pooling is used to reduce the spatial dimensions to 1x1x1024. It aggregates spatial information into a single value per feature map. The final layer is dense, with three units representing the number of classes in the classification task. The output, which shows the anticipated probability for each class, is 1x1x3. The network architecture for image grading is outlined in the Table.7.1.

Table 7.1 Design of the DenseNet 121 Network



Layer Name	Output Size	Filter Size/Stride
Input	255x255x3	-
Conv2D	127x127x64	7x7 / 2
MaxPooling2D	63x63x64	3x3 / 2
Dense Block 1	63x63x256	-
Transition Layer 1	31x31x128	1x1 / 1 + 2x2 / 2
Dense Block 2	31x31x512	-
Transition Layer 2	15x15x256	1x1 / 1 + 2x2 / 2
Dense Block 3	15x15x1024	-
Transition Layer 3	7x7x512	1x1 / 1 + 2x2 / 2
Dense Block 4	7x7x1024	-
Global Average Pooling	1x1x1024	-
Dense	1x1x3	-

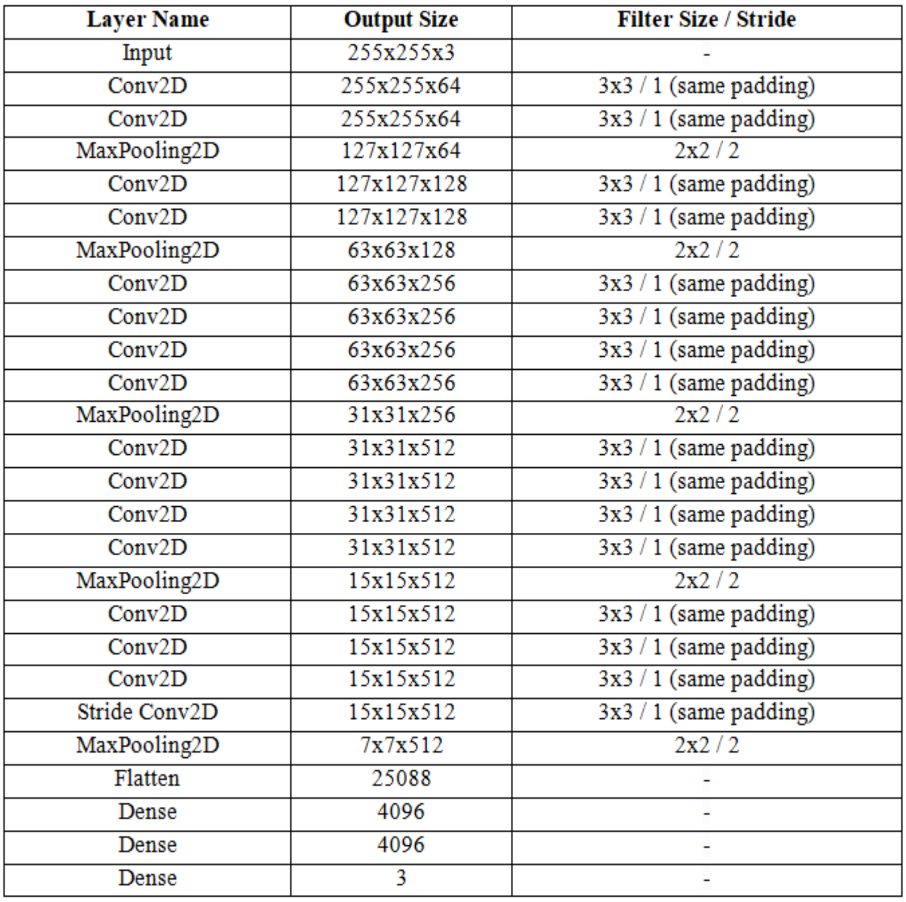
7.2.2 VGG19

It is part of the VGG family of models and has been widely used for image classification tasks [134]. The VGG19 architecture consists of 19 layers, including 16 convolutional layers and three fully connected layers. Here's a brief explanation of the components of VGG19:

- **Convolutional Layers**: It consists of 3x3 filters with 1 stride and a 1 padding. There are a total of 16 convolutional layers stacked one after another. These layers are responsible for learning local features and capturing spatial patterns in the input images.
- **Max Pooling Layers**: A max pooling layer with a 2x2 pool size and a stride of 2 is applied after all pair of convolutional layers. It reduces the feature maps' spatial dimensions and helps extract higher-level features while preserving important information.
- **Fully Connected Layers**: Following the convolutional layers, it has three FC layers. 4,096 neurons make up each completely linked layer, which proceeds by an activation function for ReLU. The final FC layer has three neurons, corresponding to the class numbers in the smart colposcopy images on which VGG models were trained.
- **SoftMax Activation**: In last layer, SoftMax activation is utilized to transform the outputs of the last FC layer into class probabilities. It allows the model to predict the input image by assigning probabilities to different classes.

The input layer represents the input image with dimensions of 255x255 pixels and 3 channels (RGB). The first Conv2D layer applies 64 filters with a filter size of 3x3 and 1 as stride. It employs the same padding size, which means the output size remains the ideal as the input size (255x255x64). The second Conv2D layer also applies 64 filters with a filter size 3x3 and a stride of 1. It maintains the same output size (255x255x64). The first MaxPooling2D layer is set with the size of 2x2 and a stride of 2. It reduces the spatial dimensions by a factor of 2 as 127x127x64. The third Conv2D layer applies 128 filters with a filter size 3x3 and a stride of 1. It maintains the same output size of 127x127x128. The fourth Conv2D layer also uses 128 filters with a filter size 3x3 and a stride of 1. The second MaxPooling2D layer reduces the spatial dimensions by a factor of 2 as 63x63x128. The fifth Conv2D layer applies 256 filters with a filter size 3x3 and a stride of 1. The sixth Conv2D layer also uses 256 filters with a filter size 3x3 and a stride of 1. The seventh Conv2D layer applies 256 filters with a filter size 3x3 and a stride of 1. The eighth Conv2D layer also uses 256 filters with a filter size 3x3 and a stride of 1. It keeps the same output size as 63x63x256. The third MaxPooling2D layer further reduces the spatial dimensions by a factor of 2 as 31x31x256. The ninth Conv2D layer applies 512 filters with a filter size 3x3 and a stride of 1. The tenth Conv2D layer also uses 512 filters with a filter size 3x3 and a stride of 1. The eleventh Conv2D layer applies 512 filters with a filter size 3x3 and a stride of 1. The twelfth Conv2D layer uses 512 filters with a filter size 3x3 and a stride of 1. The fourth MaxPooling2D layer reduces the spatial dimensions by a factor of 2 as 15x15x512. The thirteenth Conv2D layer applies 512 filters with a filter size 3x3 and a stride of 1. The fourteenth Conv2D layer also uses 512 filters with a filter size of 3x3 and a stride of 1. The fifteenth Conv2D layer applies 512 filters with a filter size 3x3 and a stride of 1. The sixteenth Conv2D layer uses 512 filters with a filter size of 3x3 and a stride of 1. It maintains the same output size as 15x15x512.

Table 7.2 Design of the VGG19 Network



Layer Name	Output Size	Filter Size / Stride
Input	255x255x3	-
Conv2D	255x255x64	3x3 / 1 (same padding)
Conv2D	255x255x64	3x3 / 1 (same padding)
MaxPooling2D	127x127x64	2x2 / 2
Conv2D	127x127x128	3x3 / 1 (same padding)
Conv2D	127x127x128	3x3 / 1 (same padding)
MaxPooling2D	63x63x128	2x2 / 2
Conv2D	63x63x256	3x3 / 1 (same padding)
Conv2D	63x63x256	3x3 / 1 (same padding)
Conv2D	63x63x256	3x3 / 1 (same padding)
Conv2D	63x63x256	3x3 / 1 (same padding)
MaxPooling2D	31x31x256	2x2 / 2
Conv2D	31x31x512	3x3 / 1 (same padding)
Conv2D	31x31x512	3x3 / 1 (same padding)
Conv2D	31x31x512	3x3 / 1 (same padding)
Conv2D	31x31x512	3x3 / 1 (same padding)
MaxPooling2D	15x15x512	2x2 / 2
Conv2D	15x15x512	3x3 / 1 (same padding)
Conv2D	15x15x512	3x3 / 1 (same padding)
Conv2D	15x15x512	3x3 / 1 (same padding)
Stride Conv2D	15x15x512	3x3 / 1 (same padding)
MaxPooling2D	7x7x512	2x2 / 2
Flatten	25088	-
Dense	4096	-
Dense	4096	-
Dense	3	-

The fifth and final MaxPooling2D layer reduces the spatial dimensions by 2 as 7x7x512. The flatten layer reshapes the output from the previous layer into a 1D vector of size 25088. The first fully connected dense layer consists of 4096 neurons. The second fully connected dense layer also consists of 4096 neurons. The last layer has three neurons, representing class numbers in the classification task. VGG19 is a deep CNN architecture with 16 Conv layers, max pooling layers, 3 FC layers. It focuses on learning local features using convolutional layers and reduces spatial dimensions using max pooling. The FC layers and SoftMax activation function enable the model to make class predictions based on the learned features. The network architecture of the VGG19 model is represented in the Table.7.2.

7.2.3 Efficient Net

It is known for its efficiency and robust performance across various computer vision tasks, including image classification [135]. It combines several techniques to balance model size, computational efficiency, and accuracy. It introduces a compound scaling method that uniformly scales the network's depth, width, and resolution to optimize its performance. The architecture consists of several repeated blocks, including convolutional layers, squeeze-and-excitation blocks, and depth wise separable convolutions layer. Here's a brief explanation of the components of Efficient Net:

- **Convolutional Layers**: EfficientNet begins with convolutional layers to capture low-level features and spatial patterns in the input images. The specific number of convolutional layers varies depending on the EfficientNet variant (e.g., EfficientNet-B0, B1, B2, etc.).
- **Depthwise Separable Convolutions**: It employs "*Depthwise separable convolutions*", which divide the standard convolutional operation into a depthwise convolution for channel mixing. This technique significantly reduces the computational cost while maintaining representational capacity.
- **Squeeze-and-Excitation Blocks (SE):** It incorporates SE blocks to enhance the expressive power of the network. The blocks concentrate on using adaptively recalibrating feature maps to model channel-wise dependency. This mechanism helps the network allocate more attention to informative features and suppress less relevant ones.
- **Compound Scaling**: It helps to uniformly scales the network's depth, width, and resolution. The scaling coefficients are identified through a grid search to optimize both accuracy and efficiency. This approach allows EfficientNet models to achieve better performance while being computationally efficient.

The input layer represents an image with 255x255 pixels and 3 channels (RGB) dimensions. The first Conv2D layer applies 64 filters with a filter size of 3x3 and a stride of 2. It uses "same" padding, meaning the output size remains the same as the input size 255x255x64. This layer normalizes the activations of the previous layer to improve network stability and training speed. It results in an output size of 128x128x32. The Swish activation function is applied element-wise to the last layer's output. Swish is a nonlinear activation function that introduces a smooth, nonlinear behavior to the network. The output size remains the same as 128x128x32. The MBConvBlock1 represents a MobileNetV2-style inverted residual block. It consists of depthwise separable convolutions and expansion layers. The output size is reduced to 64x64x32.

The MBConvBlock2 applies a 1x1 convolution with a stride of 1, reducing the number of filters to 16. It preserves the spatial dimensions as 64x64x16. The MBConvBlock3 uses the 3x3 convolution with a stride of 2, increasing the number of filters to 24. It reduces the spatial dimensions by a factor of 2 to 32x32x24. The MBConvBlock4 represents another inverted residual block without any downsampling. The spatial dimensions remain the same as 32x32x24. The MBConvBlock5 applies a 3x3 convolution with a stride of 2, increasing the filters to 40. It reduces the spatial dimensions by a factor of 2 to 16x16x40. The MBConvBlock6 represents another inverted residual block without downsampling. The spatial dimensions remain the same as 16x16x40. The MBConvBlock7 applies a 3x3 convolution with a stride of 2, increasing the number of filters to 80. It reduces the spatial dimensions by a factor of 2 to 8x8x80. The MBConvBlock8 represents another inverted residual block without downsampling. The spatial dimensions remain the same as 8x8x80. The MBConvBlock9 applies a 3x3 convolution with a stride of 1, increasing the number of filters to 112. The MBConvBlock10 uses a 5x5 convolution with a stride of 2, increasing the number of filters to 192. It reduces the spatial dimensions by a factor of 2 as 4x4x192. The MBConvBlock11 represents another inverted residual block without downsampling. The MBConvBlock12 applies a 3x3 convolution with a stride of 1, increasing the number of filters to 320. The Conv2D layer applies a 1x1 convolution with a stride of 1, resulting in an output size of 4x4x1280. The batch normalization layer is used in this layer to normalize the activations of the previous layer, resulting in an output size of 4x4x1280. The Swish activation function is applied element-wise to the last layer's output, maintaining the output size as 4x4x1280. This layer conducts global average pooling, decreasing the spatial dimensions to 1x1x1280. It aggregates spatial information into a single value per feature map. The final Conv2D layer doesn't specify the filter size/stride, but it's likely used for the classification task. Its output size depends on the number of classes and the desired prediction outcome.

In summary, the given architecture represents a mobile-oriented CNN. It starts with a convolutional layer and applies a series of MobileNetV2-style inverted residual blocks with varying filter sizes, strides, and spatial dimension reductions. Batch normalization and Swish activation are used for normalization and nonlinear behavior, respectively. The architecture concludes with global average pooling and a final Conv2D layer, potentially for classification purposes. The architecture of the EfficieneNet model for grading of images is represented in the Table 7.3.

Table 7.3 Network Architecture of the EfficienNet

Layer Name	Output Size	Filter Size / Stride
Input	255x255x3	-
Conv2D	255x255x64	3x3 / 2 (same padding)
Batch Normalization	128x128x32	-
Swish	128x128x32	-
MBConvBlock1	64x64x32	-
MBConvBlock2	64x64x16	1x1 /1
MBConvBlock3	64x64x24	3x3 / 2
MBConvBlock4	32x32x24	-
MBConvBlock5	32x32x40	3x3 / 2
MBConvBlock6	16x16x40	-
MBConvBlock7	16x16x80	3x3 / 2
MBConvBlock8	8x8x80	-
MBConvBlock9	8x8x112	3x3 /1
MBConvBlock10	8x8x192	5x5 /2
MBConvBlock11	4x4x192	-
MBConvBlock12	4x4x320	3x3 /1
Conv2D	4x4x1280	1x1 / 1
Batch Normalization	4x4x1280	-
Swish	4x4x1280	-
Global Average Pooling	1x1x1280	1x1 / 3
Conv2D		-

7.3 Experimental Results and Discussion

For the implementation of the classification model, the following requirements are data collection, which includes the original images, image enhancement, and grading of smart colposcopy images. The dataset is sourced from the Kaggle dataset repository. The type 1 represents CIN 1, which is the initial stage of cancer; type2 represents CIN 2, which is the cancer stage; and type 3 represents CIN3, which means the final stage of cervical cancer. The dataset comprises 4870 smart colposcopy images across three cancer stages. The model is partitioned into 80% training and 20% testing ratio. Training is conducted using a dataset of dimensions (3870, 255, 255, 3), while testing involves images sized (1000, 255, 255, 3). The input size for training images is 255x255x3, with an output class of 3 corresponding to CIN1, CIN2, and CIN3.

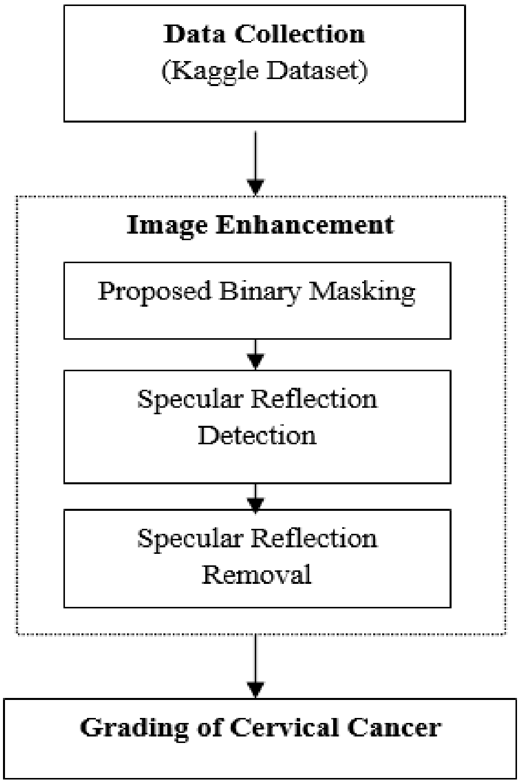


Figure 7.2 Workflow for Grading Cervical Cancer

7.3.1. Image Enhancement

The steps involved are specular reflection detection and specular reflection removal to enhance the smart colposcopy.

- **Specular Reflection Detection:** The 4870 images are majorly affected by the reflection artifacts. These artifacts are identified by the proposed threshold method, as discussed in Chapter 4. The fine-tuned UNet++ is trained for the prediction of SR on the images, as discussed in Chapter 5.
- **Specular Reflection Removal:** The image identified with the glare region is removed using the proposed bilateral-based convolution neural to fill the eliminated region on the colposcopy images, as discussed in Chapter 6. The image reflection

region identified and removed is called enhanced images, which are trained using deep learning for the grading process.

7.3.2 Model Training and Evaluation

The following steps in grading the smart colposcopy are model selection, training, and evaluation.

- **Model Selection**: The CNN based deep learning model like VGG19, DenseNet121 and EfficientNet model is considered for the grading of smart colposcopy images based on their severity as shown in the Figure.7.2.

- **Model Training**: Train the selected deep learning model using the training set. During training, feed the enhanced and non-enhanced images as input to the model and their corresponding labels as shown in the Figure 7.3. It is trained employing the Adam optimizer. The extension of stochastic gradient descent combines the benefits of adaptive learning rates and momentums. Adam automatically adapts the learning rate for each parameter during training. It calculates individual learning rates based on the historical gradients of each parameter, allowing it to converge faster and more effectively. It also incorporates the concept of momentum, which helps the optimizer accelerate the learning process by accumulating the influence of past gradients. It helps the optimizer navigate flatter regions of the loss landscape and avoid getting stuck in local minima. It uses the average, which makes the algorithm move towards minima at a faster pace. Categorical cross-entropy is a commonly used loss function for classification tasks, mainly when dealing with multiple classes (i.e., CIN1, CIN2 and CIN3). The formula for categorical cross-entropy loss is represented in the equation (7.1)

$$L = -\sum \left(y_{true} * \log\left(y_{pred} \right) \right) \tag{7.1}$$

where L represents the categorical cross-entropy loss, y_{true} true labels (ground truth) in one-hot encoded form. It is a vector of length equal to the number of classes, where 1 is the element that matches the true class, and all other elements are 0, y_{pred} predicted probabilities for each class generated by the model. It is also a vector of length equal to the number of classes, where each element represents the predicted probability of the corresponding class. The formula computes the element-wise multiplication of the true labels (y_{true}) and the logarithm of the predicted probabilities (y_{pred}). The individual terms are then summed across all classes to calculate the overall loss.

During training, the goal is to minimize the categorical cross-entropy loss by adjusting the model's parameters (weights and biases) through techniques like gradient descent or back propagation. This loss function penalizes the model more severely when it predicts significantly incorrect probabilities for the true class and encourages the model to learn accurate class probabilities over time. The batch size refers to the number of training examples or samples processed by the model in each iteration or update of the optimizer. It determines the number of samples propagated through the network and is used to compute the gradients and update the model's parameters. The model utilizes a batch size of 32, coupled with a learning rate of 0.0001. The learning rate is a hyper parameter that determines the step size at which the model's parameters (weights and biases) are updated during training. It controls the magnitude of the changes made to the parameters based on the computed gradients. The model undergoes training for 60 epochs, utilizing the Adam optimizer. Dropout is a regularization technique commonly used in deep learning models to prevent over fitting. It involves randomly setting a fraction of the input units or activations to zero during training, effectively "*dropping out*" those units. This helps reduce the model's reliance on specific features and encourages learning more robust and general sable representations. The dropout layer is configured at a rate of 0.5 to mitigate over fitting, particularly given the limited dataset size.

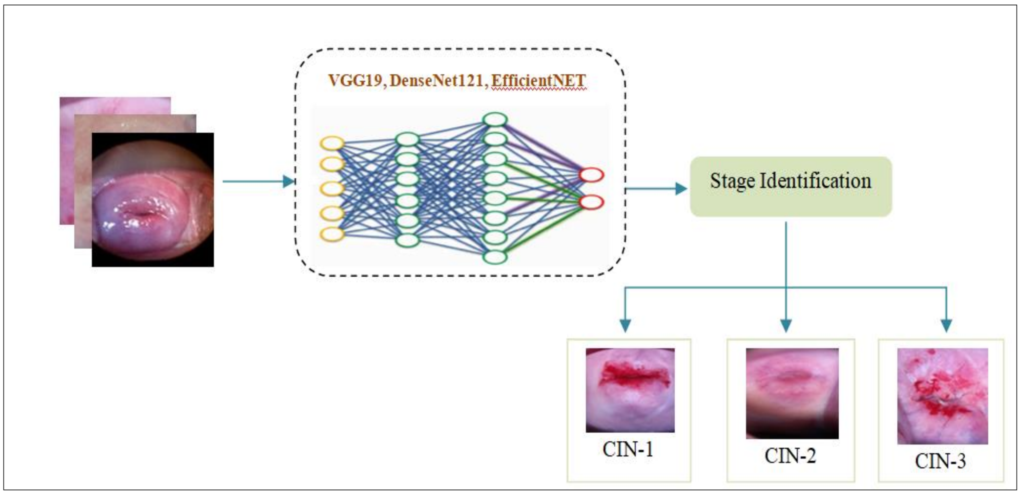


Figure 7.3 Implementation of Deep Learning Model on Enhanced Smart Colposcopy Images

The loss function employed is categorical cross-entropy, accompanied by the SoftMax activation function. The SoftMax function is a commonly used activation function in deep learning models, particularly for multiclass classification tasks. It takes a vector of real numbers as input and transforms them into a probability distribution over multiple classes, where the probabilities sum up to 1. The SoftMax function exponentials each input vector element and then normalizes it by dividing it by the sum of the exponential values. This normalization ensures that the output values lie between 0 and 1 and sum up to 1, effectively representing probabilities. The SoftMax function is commonly used as the final activation function in the output layer of a neural network for multiclass classification tasks. It transforms the raw outputs of the network into probabilities, allowing for easy interpretation and comparison of class probabilities. The weight initialization for the model classification is conventional, aiming to enhance the model's performance.

- **Model Evaluation:** The trained model is evaluated with metrics such as accuracy, precision, specificity and sensitivity for the grading of colposcopy images.

7.3.3. Comparison Analysis of the Performance of Enhanced and Non-Enhanced Images

The quality of cervical images plays a crucial role in the cancer grading process. The "*CIN1, CIN2, and CIN3*" data, obtained from Kaggle, are learned on pre-existing models such as DenseNet121, Vgg19, and efficient models. Similarly, images assessed for quality are trained using pre-trained models. Accuracy in image classification refers to measuring how well a machine learning model correctly predicts the class or label of an image. It is usually represented as a percentage and calculated by dividing the number of correctly classified images by the total number of images in the dataset. The accuracy for image classification can be calculated using equation (7.2).

$$Accuracy = \frac{NumberofCorrectlyClassifiedImages}{TotalNumberofImages} * 100 \qquad (7.2)$$

In this formula, the number of images correctly classified by the model is divided by the total number of images in the dataset. The result is multiplied by 100 to obtain the accuracy as a percentage. Precision is a performance metric that measures the proportion of correctly predicted positive instances out of the total cases predicted as positive. In other words, it quantifies the accuracy of positive predictions made by a model. The formula to calculate precision is as represented in the equation (7.3)

$$Precision = \frac{TruePostives}{(TruePostivies + FalsePositives)} \qquad (7.3)$$

In Equation 7.3, a true positive (TP) denotes the count of accurately predicted positive instances, while false positives (FP) represents the count of inaccurately predicted positive instances. Specificity is the performance metric measuring the proportion of correctly predicted negative instances out of the total ones predicted as negative. It quantifies the accuracy of negative predictions made by a model. The formula to calculate specificity is represented in the equation (7.4).

$$Specificity = \frac{TrueNegatives}{(TrueNegatives + FalsePositives)} \tag{7.4}$$

In equation 7.4, true negatives (TN) represents the count of correctly predicted negative instances, and false positives (FP) represents the count of incorrectly predicted negative instances. Sensitivity, also known as recall or true positive rate, is a performance metric measuring the ratio of positive instances correctly predicted among all positive instances. It quantifies the model's ability to determine the positive cases accurately. The formula to calculate sensitivity is represented in the equation (7.5).

$$Sensitivity = \frac{TruePostivies}{(TruePostivies + FalseNegatives)} \tag{7.5}$$

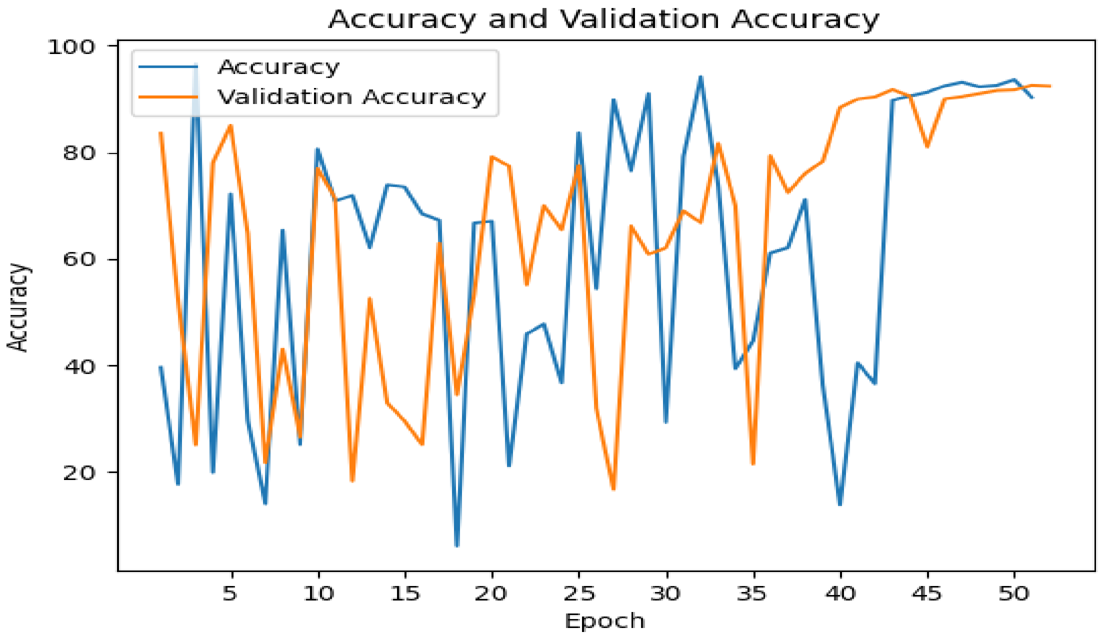
In equation 7.5, true positives (TP) represent the count of correctly predicted positive instances, and false negatives (FN) represents the count of incorrectly predicted negative instances. As demonstrated in Table 7.4, the study reveals that quality-assessed photos produce enhanced accuracy in grading cancer.

Table 7.4. Quantitative Analysis of the Classification of Smart Colposcopy Images

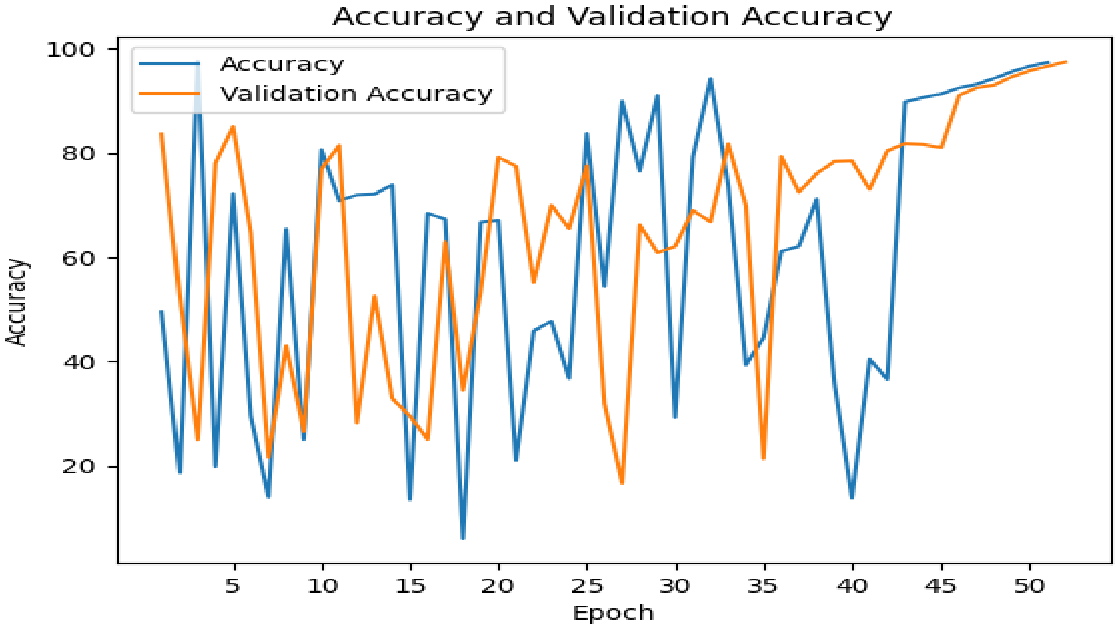
Metrics	**Without Enhancement**			**With enhancement**		
	DenseNet121	**Vgg19**	**EfficientNet**	**DenseNet121**	**Vgg19**	**EfficientNet**
Accuracy (%)	92.72	95.02	91.79	**97.32**	**96.25.**	**93.27**
Precision (%)	90.72	92.14	93.28	**96.89**	**95.74**	**95.74**
Specificity (%)	91.41	90.39	90.39	**94.73**	**91.39**	**91.39**
Sensitivity (%)	91.78	92.49	89.49	**92.12**	**92.79**	**91.79**

Table 7.4 compares performance metrics for three different neural network models, namely DenseNet121, Vgg19, and EfficientNet. The models were evaluated both without enhancement and with enhancement images. The evaluation was done on a specific task, likely a classification task, where the models were trained to predict certain classes. Table

7.4 shows that applying enhancement to the models generally improved performance across various metrics. All three models achieved higher accuracy, precision, specificity, and sensitivity with enhancement. DenseNet121 showed the most significant improvement in accuracy and precision after enhancement. The accuracy calculation of the DenseNet121, model for each epoch is represented in Figure 7.4.



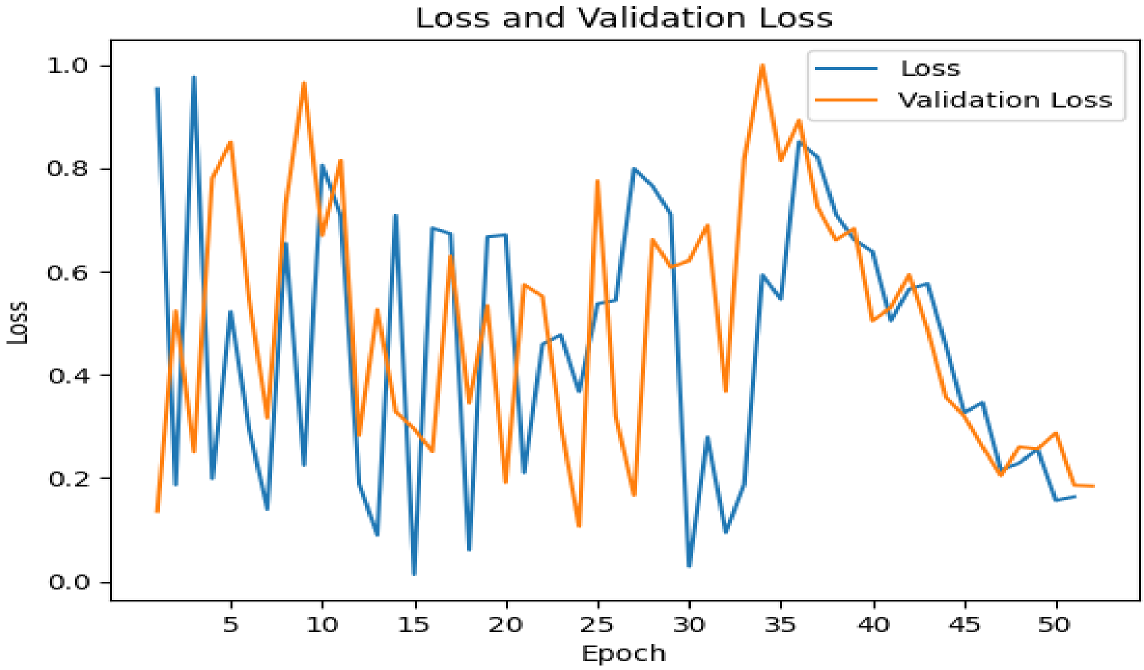
(a)



(b)

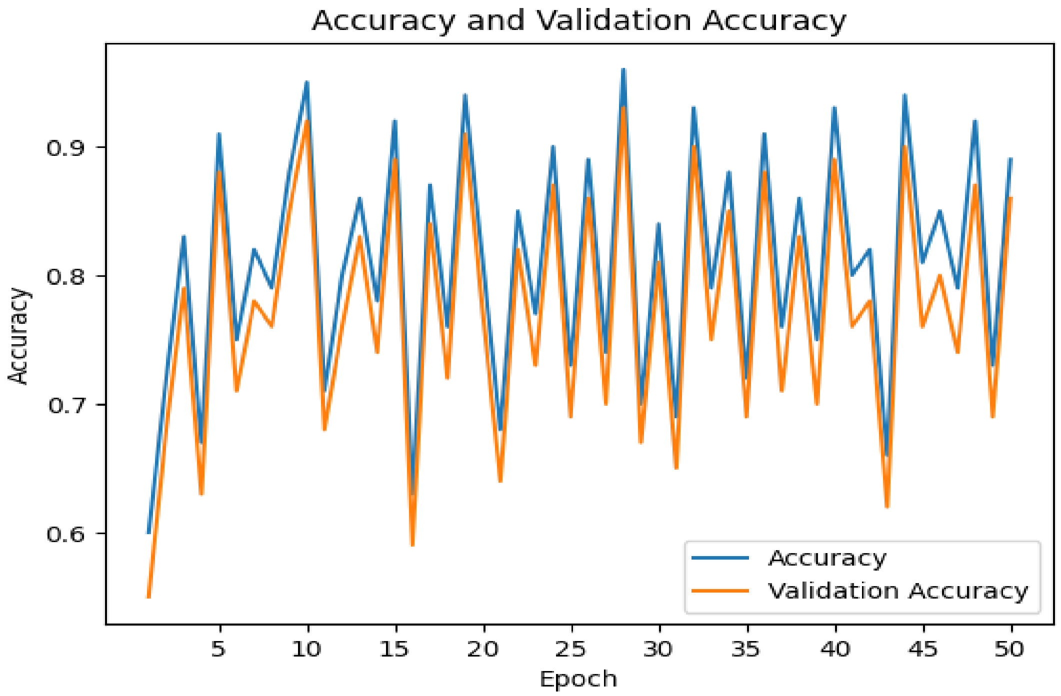
Figure 7.4 Grading Accuracy of the Smart Colposcopy Images using the DenseNet121 model. (a) Without Enhanced Images (b) With Enhanced Images

(a)

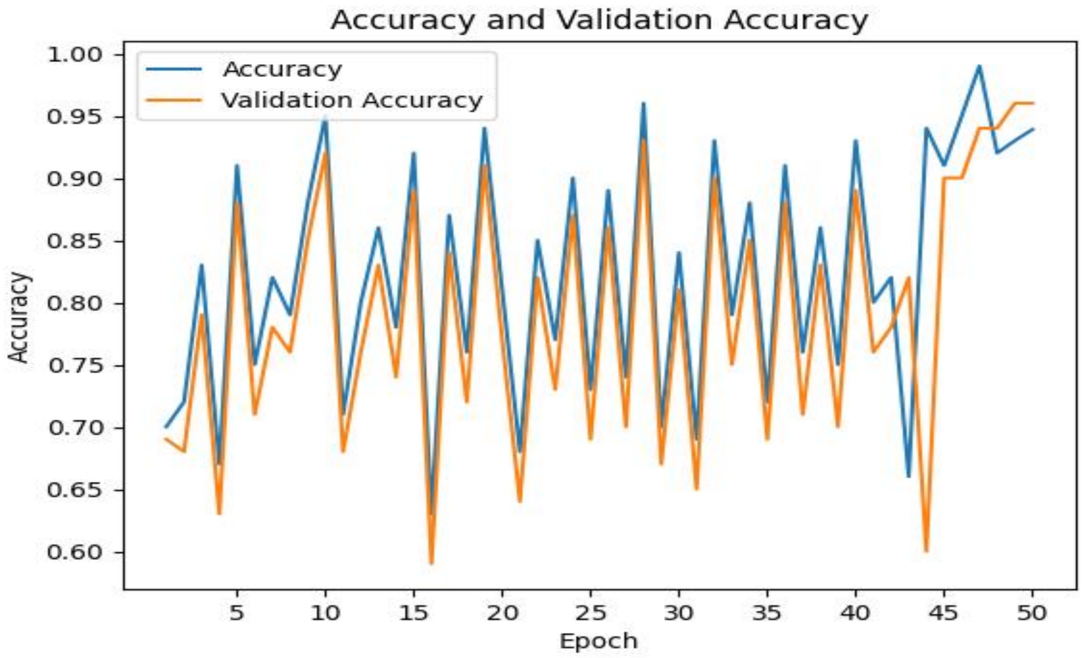


(b)

Figure.7.5 Grading Loss calculation of the Smart Colposcopy Images using the DenseNet121 model. (a) Without enhanced Images (b) With enhanced Images



(a)

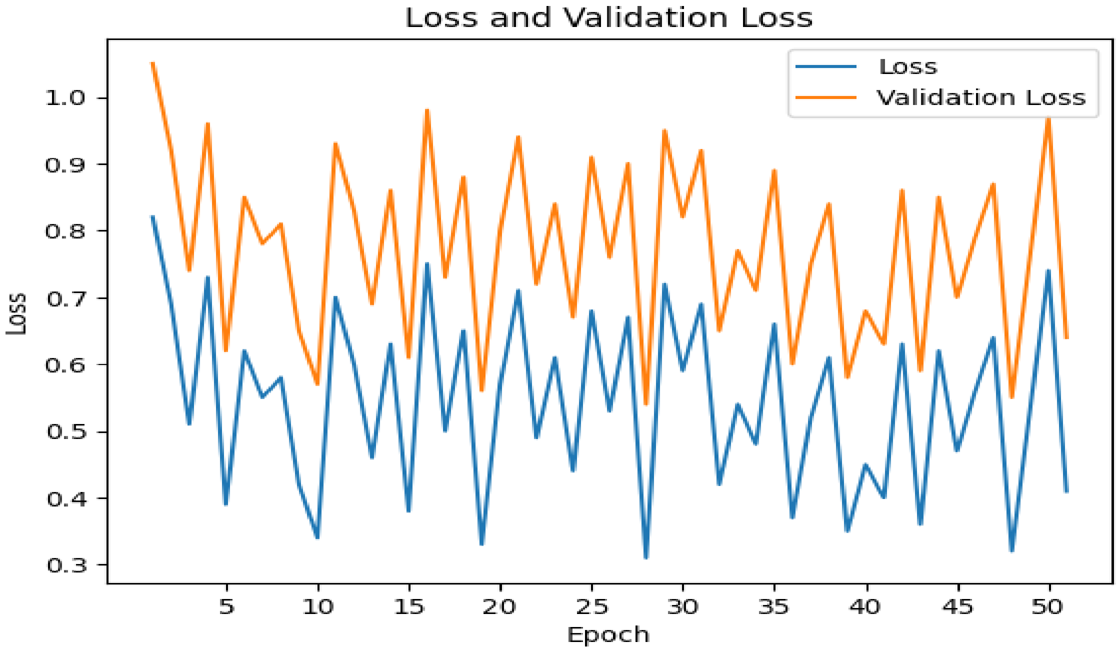


(b)

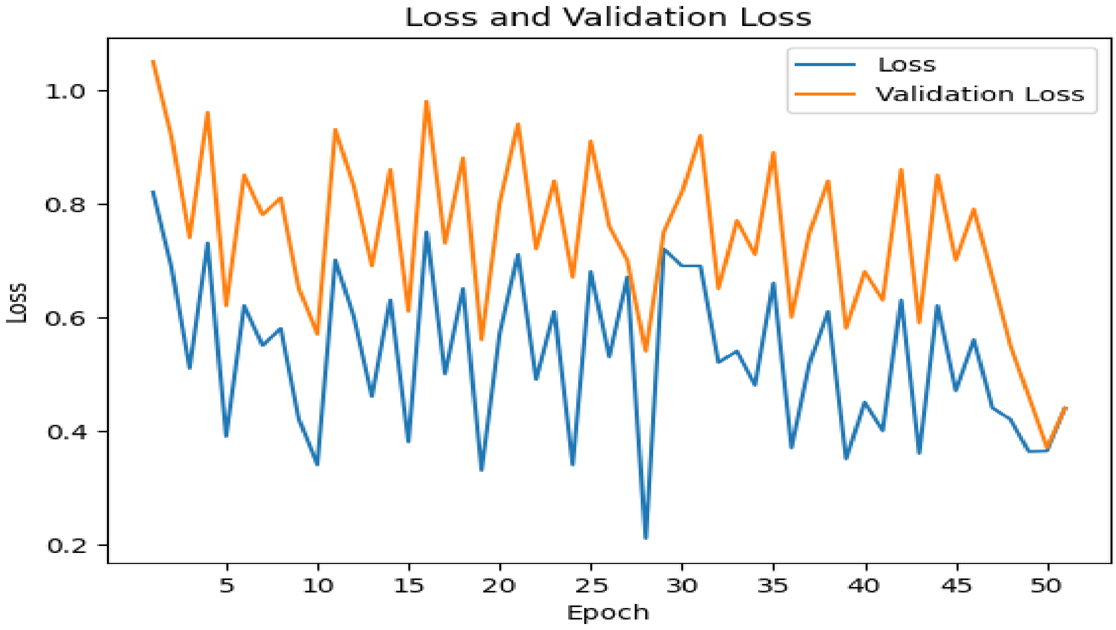
Figure 7.6 Grading Accuracy of the Smart Colposcopy Images using the VGG19 Model. (a) without enhanced Images (b) With enhanced Images

DenseNet121 model without enhancement grades the smart colposcopy images with an accuracy of 92.72% at the epoch value of 60. DenseNet121 model without enhancement grades the smart colposcopy images with an accuracy of 97.32% at the epoch value of 60.

Based on the analysis of the DenseNet121 model, the enhanced image has improvised the grading of smart colposcopy images with a loss value of 0.357, as shown in Figure 7.5(b). The accuracy calculation of the VGG 19 model for each epoch is represented in Figure 7.6.



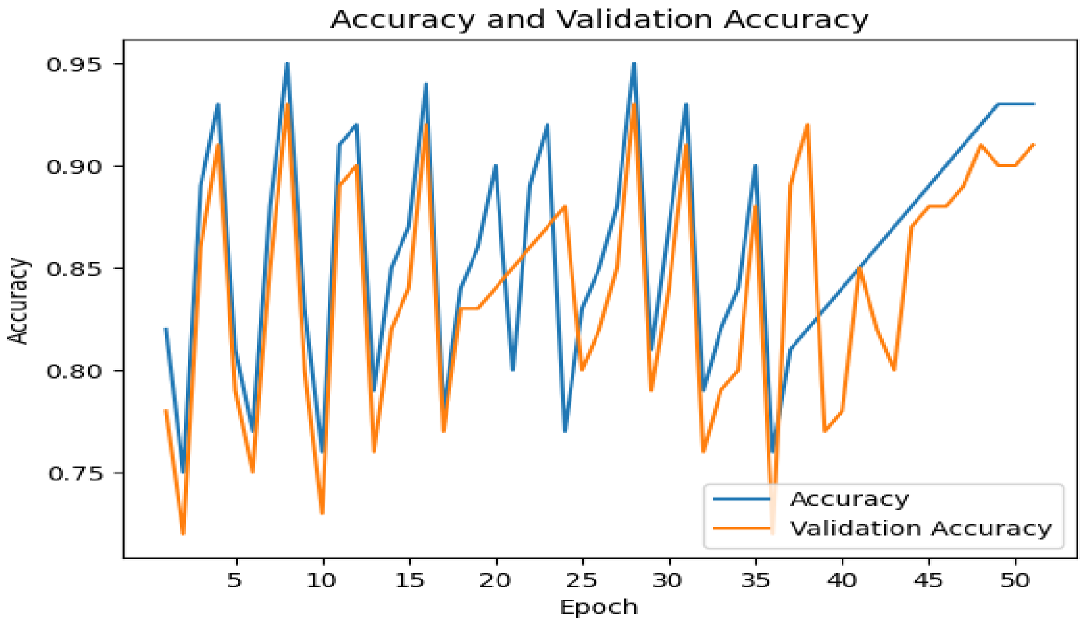
(a)



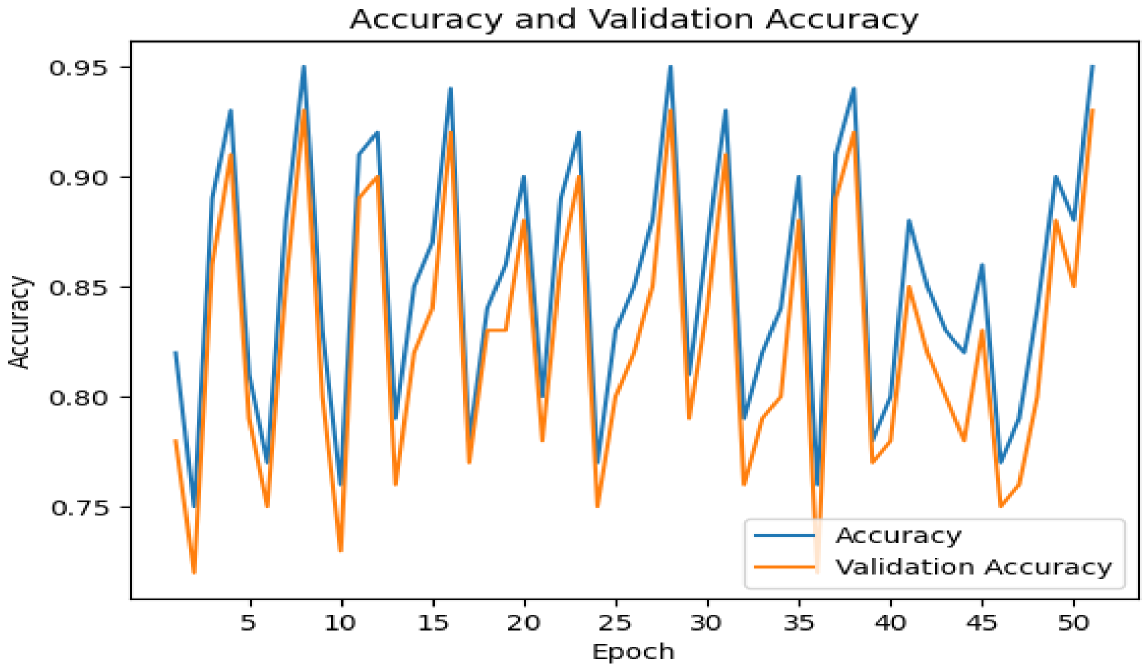
(b)

Figure 7.7 Grading Loss value of the Smart Colposcopy Images using the VGG19 model. (a) Without Enhanced Images (b) With Enhanced Images

The VGG19 model, without any enhancement, achieves a classification accuracy of 95.02% at the epoch value of 60. However, after applying the enhancement technique, the model's performance is further improved, and it grades the smart colposcopy images with an accuracy of 96.32% at the same epoch value of 60. Figure 7.6 represents the accuracy calculation of the VGG19 model for each epoch, showing the model's progress in achieving higher accuracy as the training proceeds. The accuracy steadily increases, reaching 95.02% at epoch 60 for the unenhanced images. However, after applying the enhancement, the accuracy increases even further to 96.32%, indicating the effectiveness of the enhancement in improving the model's grading performance. The enhancement process is visualized in Figure 7.6(b), showcasing how the enhanced images contribute to the improved grading results. The enhanced images aid the VGG19 model in better distinguishing and classifying different stages of cervical cancer, resulting in higher accuracy than the original unenhanced images. Furthermore, Figure 7.7 (b) illustrates the loss value associated with the enhanced images, indicating a value of 0.412. This loss value represents the model's ability to minimize the difference between the predicted and actual labels during training. A lower loss value suggests a better fit of the model to the data and improved performance in grading the smart colposcopy images.

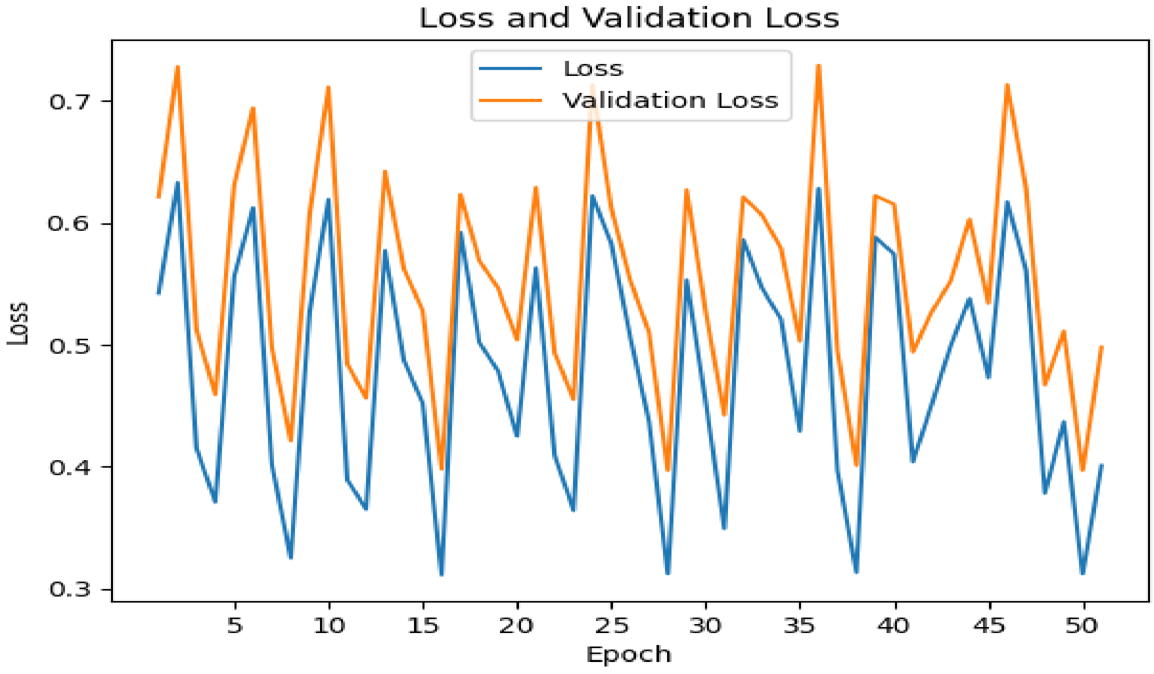


(a)

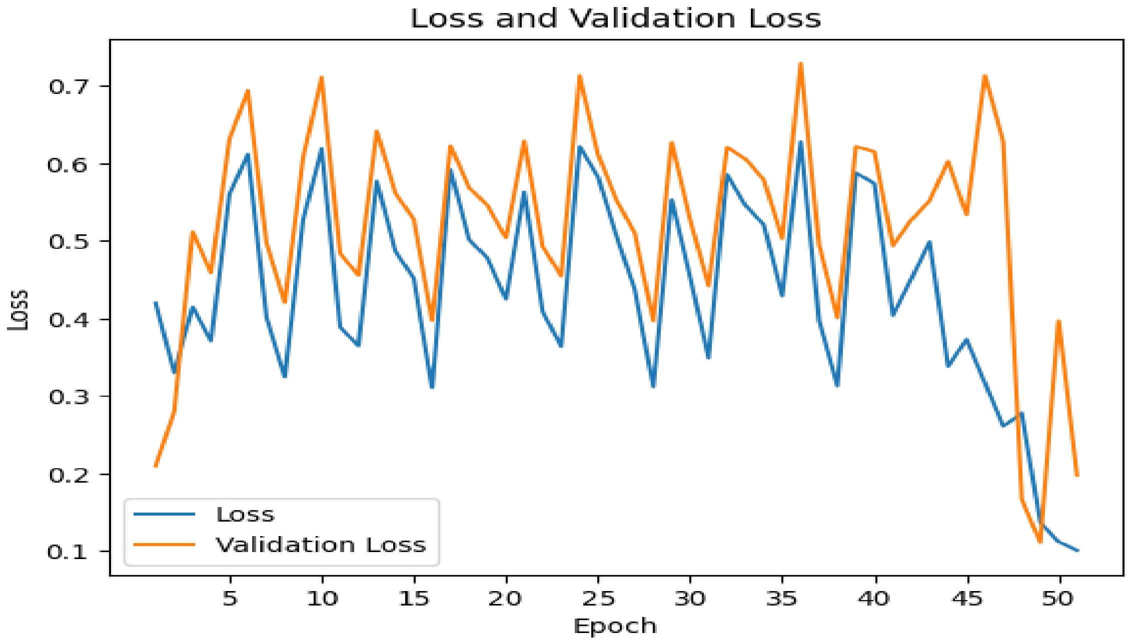


(b)

Figure 7.10 Grading accuracy of the Smart Colposcopy Images using Efficient model. (a) Without Enhanced Images (b) With Enhanced Images



(a)



(b)

Figure 7.9 Grading Loss of the Smart Colposcopy Images using Efficient Model. (a) Without enhanced Images (b) With Enhanced Images

The Efficient model, without any enhancement, achieves a classification accuracy of 93.27% at the epoch value of 60. However, after applying the enhancement technique, the model's performance is further improved, and it grades the smart colposcopy images with an accuracy of 91.79% at the same epoch value of 60. The enhancement process is visualized in Figure 7.8, showing how the enhanced images lead to improved grading results. The enhanced images help the model better identify and classify different stages of cervical cancer, resulting in a higher accuracy than the original unenhanced images. Furthermore, Figure 7.9 (b) illustrates the loss value associated with the enhanced images, indicating a value of 0.417. This loss value reflects the model's ability to minimize the difference between the predicted and actual labels during training. A lower loss value indicates a better fit of the model to the data and improved performance in grading the smart colposcopy images. Specular reflection, an artifact covering specific regions in medical images, resembles noise, leading to potential misdiagnosis. The removal of these artifacts enhances image clarity, reduces unwanted elements, and improves the visualization of cellular structures in smart colposcopy images. The accurate grading relies on precise identification of abnormal cell features. The enhanced images reduced the risk of misinterpretation, and ultimately enhancing diagnostic accuracy in medical assessments.

7.4. Summary

The application of the enhancement process led to remarkable improvements in accuracy for all three models: the enhanced efficient model achieved an accuracy of 91.79%, the enhanced VGG19 model achieved 96.32%, and the improved DenseNet121 model achieved an impressive accuracy of 97.32%, all at the epoch value of 60. These results underscored the proposed enhancement technique's significance in improving the models' grading performance. Moreover, the visual representations of the enhanced images and the associated loss values further validated the enhanced models' improved performance. The visualizations showcased the quality of the enhanced images and how they assisted the models in making more accurate and reliable predictions for cervical cancer grading. The associated loss values demonstrated the effectiveness of the models' training process with the enhanced images, resulting in minimized differences between the predicted and actual labels. The research comprehensively evaluated three CNN models for grading smart colposcopy images. The application of the enhancement technique significantly boosted the accuracy of the models, with the enhanced Efficient, VGG19, and DenseNet121 models achieving accuracies of 91.79%, 96.32%, and 97.32%, respectively, at the epoch value of 60. The visual representations and loss values further supported the enhanced models' improved performance. Overall, the findings highlighted the potential of the proposed enhancement technique for improving the accuracy and effectiveness of cervical cancer grading using smart colposcopy images, thereby contributing valuable insights to medical image analysis.