Chapter 7

Deep Learning-based Approaches for Segmentation, Detection and Classification of Parasite Eggs from Microscopic Images

Deep learning (DL)-based methods represent the cutting edge in solving the majority of computer vision problems. These techniques utilize complex neural network architectures to automatically learn hierarchical representations of data, making them highly effective for tasks such as image segmentation, classification, and object detection. Convolutional Neural Networks (CNNs), a type of deep learning architecture specifically designed for processing visual data, become particularly prominent in this domain. Researchers across various fields have extensively utilized different CNN-based techniques and models to tackle a wide range of challenges related to image analysis. From medical imaging to autonomous driving, CNNs have demonstrated remarkable performance in accurately segmenting images, classifying objects, and detecting specific features within them. In the following sections, the applications of CNN-based algorithms are explored for automatic detection and identification of parasite eggs, highlighting the adaptability and effectiveness of these techniques in solving problems in biomedical imaging and diagnostics.

7.1 Segmentation of Parasite Egg Images using CNN-based Models

The literature indicates that among the several CNN-based models designed for semantic segmentation, U-Net [133] has been recognized as the preferred choice for researchers. U-Net is well-known for its ability to outperform traditional segmentation algorithms in a variety of computer vision applications, especially in medical image analysis. During this research work, multiple experiments are performed using UNet with various hyper-parameter configurations. The number of layers, epochs, learning rate, loss function, and optimizer are carefully adjusted to enhance performance. Further, various backbone networks, including VGG16, VGG19, and ResNet50, are explored to determine the most suitable architecture for our objective. The initial training approach consisted of treating segmentation as a binary classification task, with foreground objects labelled as white and background as black. The dataset, as mentioned in chapter 3 is used to train and validate the models. The dataset is divided into training and validation sets, as shown in Table 7.1.

Table 7.1: Number of images used in UNet-based segmentation process

Class	**Training Image**	**Validation/Test Image**
Roundworm or Ascaris	714	181
Hookworm or Necator	707	174
Whipworm or Trichuris	712	178
Total	2133	533

7.1.1 Training U-Net using Transfer Learning Approach

Initially, a U-Net architecture is trained using transfer learning, chosen due to the limited availability of annotated data specific to parasite egg segmentation tasks. By leveraging pre-trained weights from the models trained on large-scale datasets, learnt features are incorporated into our segmentation task. Specifically, pre-trained weights from VGG16, VGG19, and ResNet50 are used in this study. From the image dataset prepared for this task, 80% is used for training and the remaining 20% for testing and validation of the models. The hyper-parameter configurations utilized in our study are detailed in Table 7.2, which are based on both literature and our experimental findings.

The training is continued until the training loss falls below 0.1 and shows

Table 7.2: Hyper-parameters used during the training of UNet using the transfer learning technique

Parameter	**Value**
Image size	512
Batch size	32
Learning rate	Initial 0.005 and reduce by a factor of 0.1 every few epochs based on the validation loss stops improving
Optimizer	Adam
Loss Function	Cross-entropy loss
Maximum Epochs	200
Classification mode	Binary

no significant changes. Figure 7-1 illustrates how the training and validation losses change over epochs for the UNet model with ResNet50. Test results obtained from the models are recorded in terms of IoU, precision, recall, and F1 score, which are represented in Table 7.3, while Figure 7-2 shows a few examples of output images obtained from testing the models.

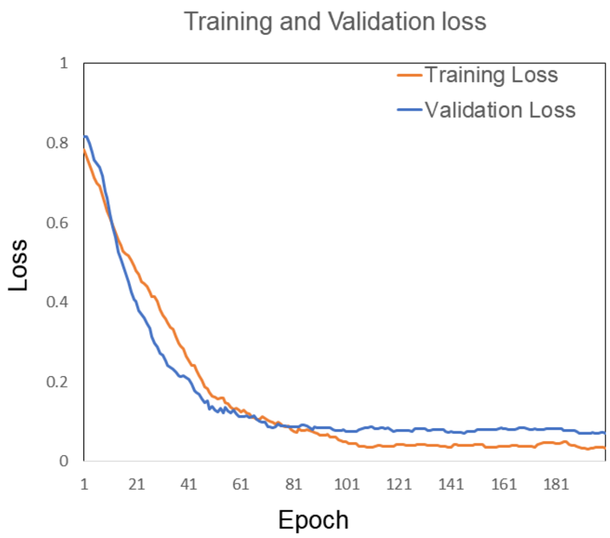


Figure 7-1: Training and validation loss of UNet using transfer learning with ResNet50 for binary segmentation.

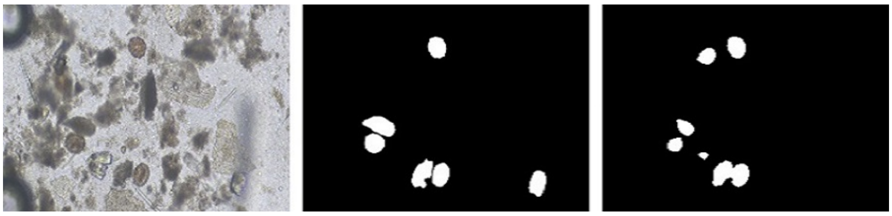
7.1.2 Training UNet with Random Weights

Further expansion of this research involves training a U-Net model from scratch with random weights. The model is trained for a maximum of 350 epochs with an initial learning rate of 0.002. Moreover, the experimentation also involves

Table 7.3: UNet segmentation results as binary classification with different pre-trained weights

Pre-Trained Weights	IoU	Precision	Recall	F1 Score
Vgg16	6.5	90.37	88.25	89.3
Vgg19	7.1	91.45	88.52	89.96
ResNet50	7.2	92.31	89.27	90.77

(a) Results obtained using Vgg19 as backbone network



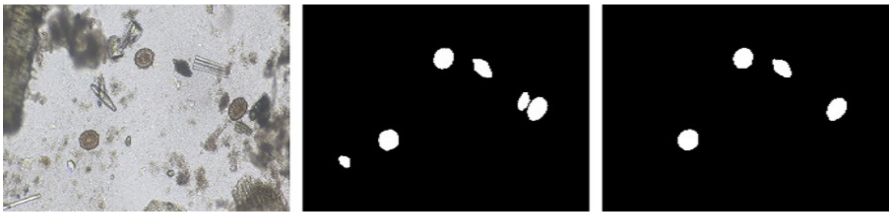
(b) Results obtained using ResNet50 as backbone network

Figure 7-2: Validation result of Unet using transfer learning: input image (left), ground truth (middle), and output segmented image (right)

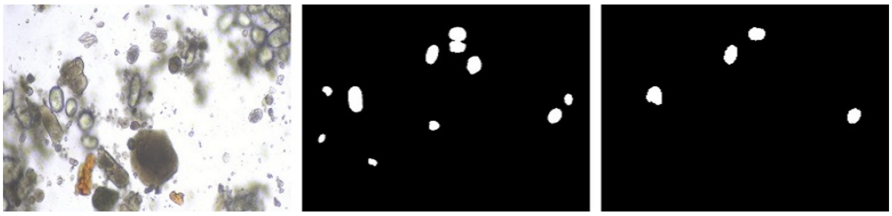
varying the number of layers in the original U-Net model. This is done to identify the optimal configuration that enables the U-Net model to achieve the highest accuracy in segmenting parasite egg images. It is observed that reducing the number of layers decreases performance, while increasing the number of layers does not yield significant changes in the results. Table 7.4 presents the results obtained from testing the original U-Net model trained from scratch, while a few test images are shown in Figure 7-3.

Table 7.4: UNet segmentation results as binary classification with random weights

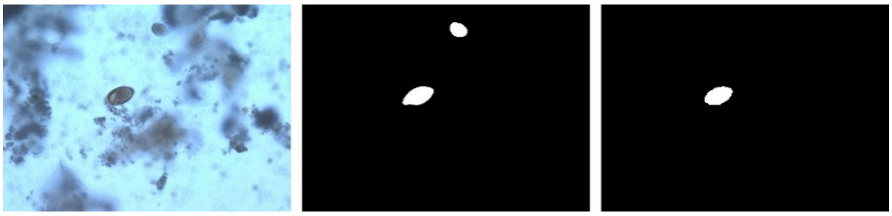
Architecture	IoU	Precision	Recall	F1 Score
Original UNet	6.8	90.65	88.72	89.67
UNet with two new layers	6.9	91.41	88.78	90.07
UNet by reducing one layer	6.3	87.53	86.27	86.89



(a) Image containing roundworm eggs



(c) Image containing hookworm eggs



(e) Image containing whipworm egg

Figure 7-3: Validation results of UNet training from scratch using random weights: image (left), ground truth (middle), and output segmented image (right)

7.1.3 Training UNet using Multi-Class Object

The study is extended by training U-Net with multi-class masks, distinguishing three classes of parasite eggs and single non-egg objects or debris. For this experimentation, a lightweight backbone network, VGG19, is considered and trained using transfer learning approach with the configuration as mentioned in Table 7.5. The validation results obtained from this model are presented in Table 7.6. A few examples of output segmented images of the model are also shown in Figure 7-4.

7.2 Classification of Parasite Eggs Using CNN models

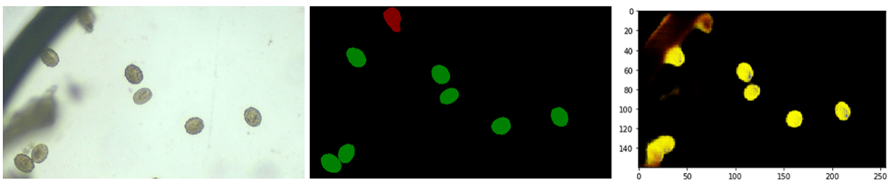
After the segmentation, the issue of accurately identifying different types of parasite eggs and non-egg objects is addressed using state-of-the-art Convolutional Neural Network (CNN)-based classification techniques. At first, the transfer learning method is used, where a pre-trained CNN model is fine-tuned to fit our parasite egg classification tasks. Secondly, a custom CNN architecture is designed, con-

Table 7.5: Hyper-parameters used during the training of UNet using transfer learning technique with multi-class objects

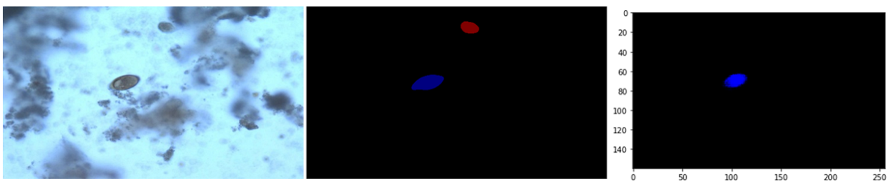
Parameter	**Value**
Image size	256
Batch size	32
Learning rate	Initial 0.001 and reduce by a factor of 0.1 every few epochs based on the validation loss stops improving
Optimizer	Adam
Loss Function	Dice loss
Maximum Epochs	350
Classification mode	Multi (three parasite eggs, one non-egg, and background)

Table 7.6: UNet segmentation results as multi-classification with random weights

Object Class	**IoU**	**Precision**	**Recall**	**F1 Score**
Roundworm	7.1	90.42	88.17	89.28
Hookdworm	6.6	87.54	84.81	86.15
Whipdworm	6.8	89.16	87.53	88.31
Non-Egg	6.3	86.75	84.22	85.47
Average	6.7	88.47	86.18	87.21



(a) Image containing roundworm eggs



(b) Image containing whipworm egg

Figure 7-4: UNet multi-class segmentation output: input image (left), ground truth (middle), and output segmented image (right)

figured, and trained to fit the specific requirements of our dataset. To enhance the generalization capabilities of the models and strengthen their capacity for diverse data scenarios, data augmentation strategies are employed, as discussed in Chapter 3. The training of the models are carried out by randomly dividing the dataset into two parts: 80% for training and 20% for validation. During the training process, the 10-fold cross-validation method is used to evaluate and improve the model's performance, aiming to produce a robust and reliable classification model.

7.2.1 Training CNN model using Transfer Learning Technique

Training CNN Model with Four Classes of Image: In this approach, the VGG16 network [134] is selected for its simple yet efficient architectural design, which has proven effective across numerous classification tasks [135]. The original classification layers of the model are replaced with new fully connected layers, as shown in Figure 7-5. These newly connected classification layers provide the class probabilities for four classes of objects, including three parasite eggs and one non-egg object. The model is trained using pre-trained weights obtained from ImageNet [136] with the configuration detailed in Table 7.7.

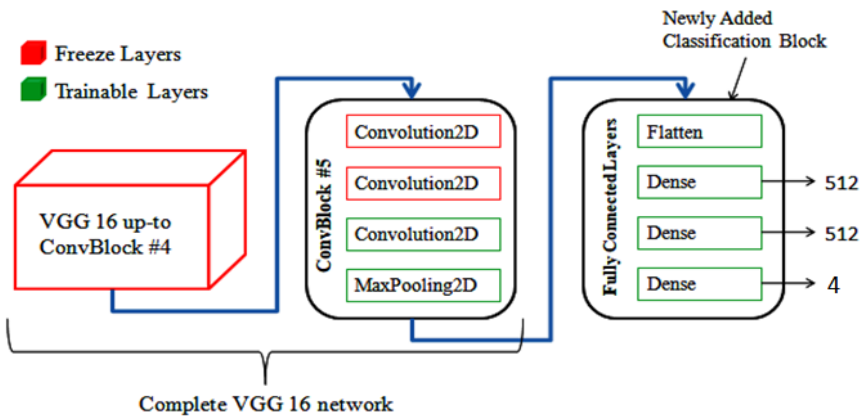


Figure 7-5: Modification of the VGG16 Model Used for Training Parasite Eggs Using the Transfer Learning Approach

The change in classification accuracy with the number of epochs during the training process is visualized in Figure 7-6. The model has achieved an overall training accuracy of 99.47% and a validation accuracy of 98.15%. Detailed validation results for each class are presented in Table 7.8 and the summary of the predicted and actual classification can be visualized through the confusion matrix provided in Figure 7-7.

7.2.2 Training a custom CNN model

Training CNN Model with Four Classes of Image: Inspired by the Vgg16 network, a smaller CNN model is designed in our work by incorporating convo-

Table 7.7: Values of Different Parameters Used in Training Using Transfer Learning Approach

Parameter	Value
Number of Class	4
Classes	Roundworm egg, Hookworm egg, Whipworm egg, and Non-Egg
Total Images/samples	14,650
Training Images	11720 (80%)
Validation Images	2930 (20%)
Input Image Size	120×120 (RGB)
Loss Function	Categorical Cross-Entropy
Optimizer	Adaptive Moment Estimation (Adam)
Activation function	ReLU (Rectified Linear Unit)
Batch size	32
Maximum Epoch	300
Initial learning Rate	0.001

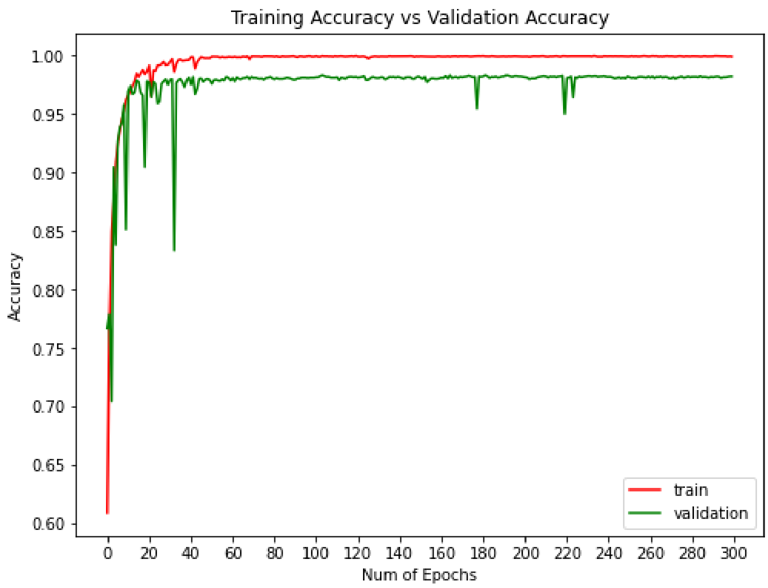


Figure 7-6: Change of Training and Validation Accuracy of Transfer Learning Approach - four classes

lutional, pooling, and fully connected layers. The generalized architecture of our model, showing the convolutional, pooling, and fully connected layers, is demonstrated in Figure 7-8. A convolutional kernel of size 3×3 is used with the same padding scheme in the architecture. Dropout or L2 regularization are also used to prevent overfitting and improve generalization. The model is trained with a maximum of 500 epochs and nearly similar hyper-parameters configurations as

Table 7.8: Validation result of CNN model using transfer learning - four classes

Class	Precision	Recall	F1-Score	Accuracy
Ascaris	0.98	0.99	0.99	98.78%
Necator	0.99	0.97	0.98	96.95%
Trichuris	0.98	0.99	0.98	98.63%
Non-Egg	0.98	0.98	0.98	98.24%
Average			0.98	98.16%

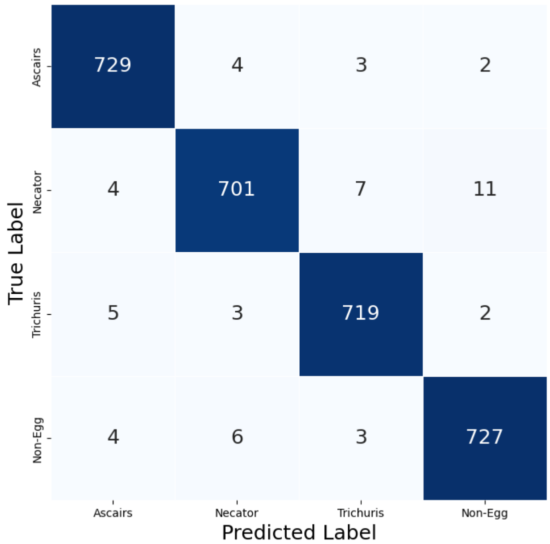


Figure 7-7: Confusion Matrix from validation of CNN classification model using transfer learning—four classes

those used in the previous section.

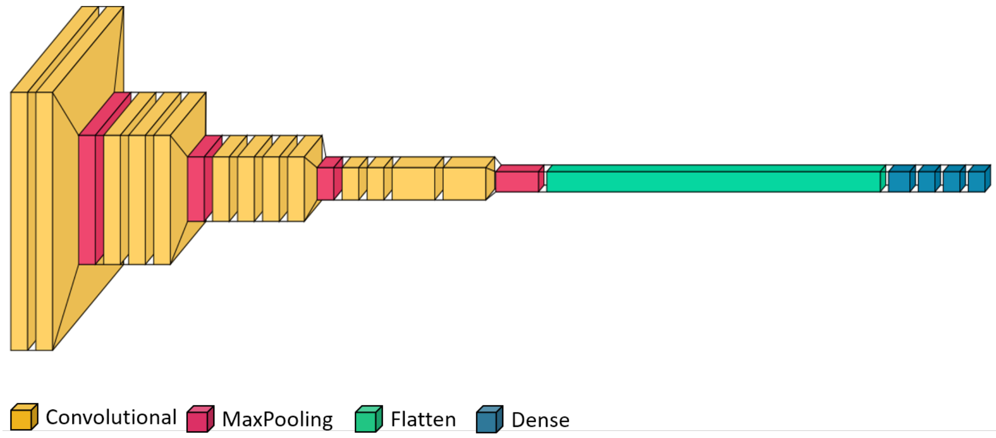


Figure 7-8: A summarised view of the custom CNN model designed for the classification of parasite eggs

Upon training, the model achieves an overall validation accuracy of 98.88% and a training accuracy of 99.16% using four classes of objects. The change in training versus validation accuracy during the training process is shown in Figure 7-9, while validation results are presented in Table 7.9 and can be visualized through Figure 7-10.

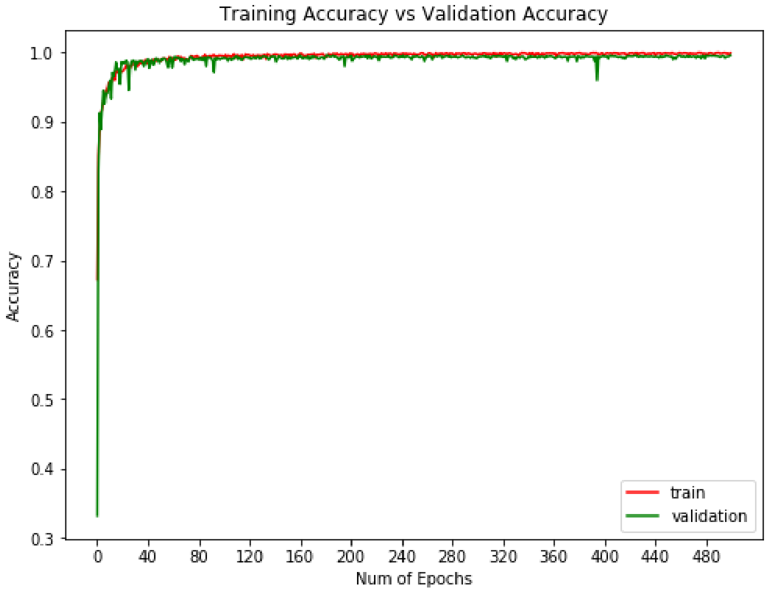


Figure 7-9: Change of Training and Validation Accuracy of Custom CNN Model - Four classes

Table 7.9: Validation result using a custom CNN model - Four classes

Class	Precision	Recall	F1-Score	Accuracy
Ascaris	0.98	1.00	0.99	99.58%
Necator	0.99	0.97	0.98	97.37%
Trichuris	0.99	0.99	0.99	99.44%
Non-Egg	0.99	0.99	0.99	99.08%
Average			0.99	98.88%

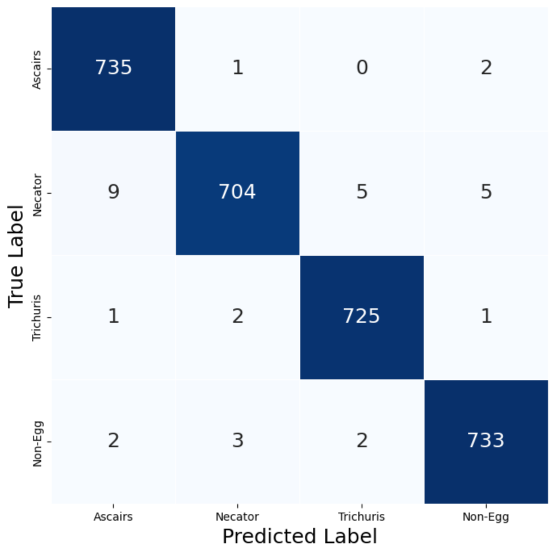


Figure 7-10: Confusion Matrix from validation of own CNN classification model - Four classes

Training CNN model with multiple classes of non-egg objects: Additionally, the model is trained to classify six classes of objects, including three types of parasite eggs and three types of non-egg objects. Here, the training epoch is set to 400. The model achieves 97.10% of training and 95.35% of validation accuracy. Change of training and validation accuracy is shown in Figure 7-11, while the validation results are presented in Table 7.10 and in the form of a confusion matrix as depicted in Figure 7-12.

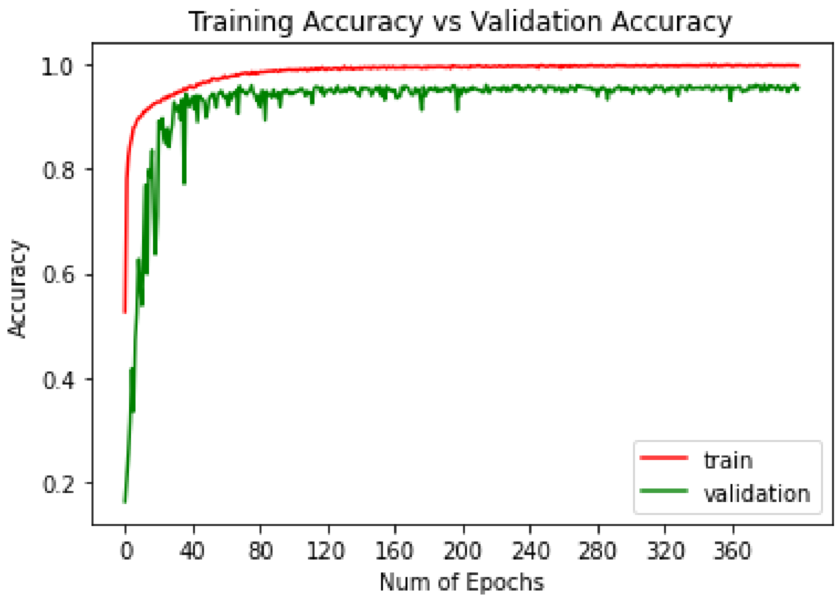


Figure 7-11: Change of Training and Validation Accuracy of Custom CNN model - Six classes

Table 7.10: Validation result using a custom CNN model - Six classes

Class	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Ascaris	0.97	1.00	0.98	99.73%
Necator	0.95	0.98	0.96	98.20%
Trichuris	0.94	0.99	0.97	99.44%
Non-Egg-1	0.95	0.91	0.93	90.72%
Non-Egg-2	0.96	0.93	0.94	92.91%
Non-Egg-3	0.96	0.91	0.94	90.97%
Average			0.95	95.35%

7.2.3 Analysis of Classification Results

Based on the above experiments, it is observed that training the CNN model with a single class of non-egg objects yields higher overall accuracy compared to training with multiple classes. However, a detailed analysis of the accuracy for

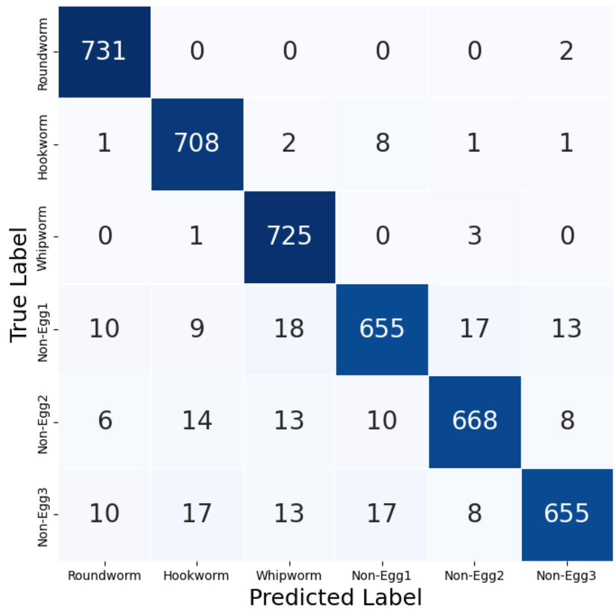


Figure 7-12: Confusion Matrix of the Validation Process of Custom CNN Model - Six classes

each type of parasite egg is performed, and it reveals an improvement in identification rates for hookworm eggs. The analysis is depicted in Figure 7-13. The model's overall accuracy is lower when trained with multiple non-egg classes, but it remains acceptable since our primary goal is to correctly identify the parasite eggs. Therefore, despite the reduced overall accuracy across the six classes, this classification process aligns with the primary objective of identifying parasite eggs.

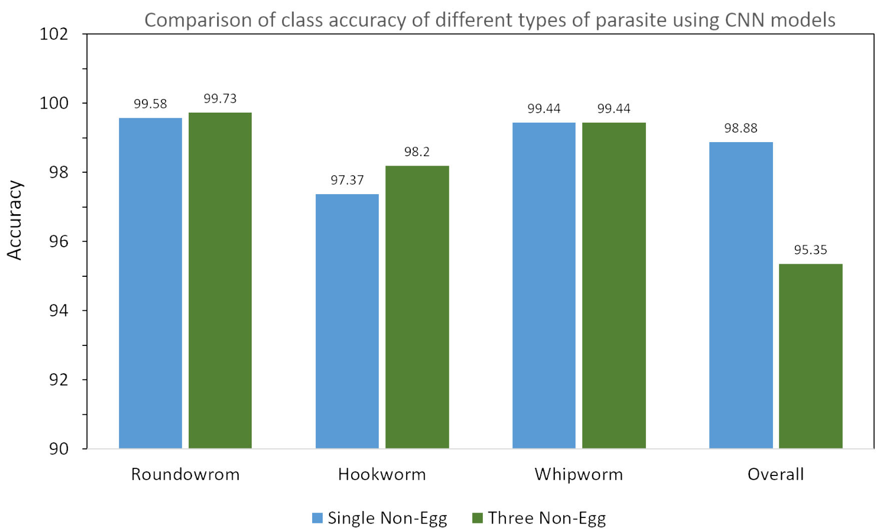


Figure 7-13: Comparison of classification accuracy of parasite eggs using CNN models with single and multiple Non-egg classes

A comparison of the performance between machine learning (ML) and CNN-based classification models is also conducted. The comparison of overall classification accuracy between ML-based models and the CNN-based models using four classes of data is illustrated in Figure 7-14. For this analysis, SVM and XGBoost classifiers, trained with texture and shape-based features, are considered, as they achieve overall better results. The comparative analysis of overall accuracy when using multiple non-egg classes is presented in Figure 7-15.

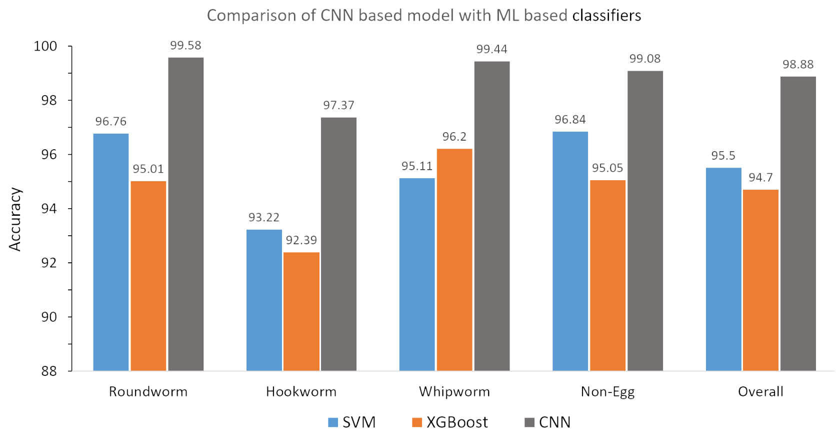


Figure 7-14: Comparison of classification accuracy of ML-based and CNN-based models - four classes

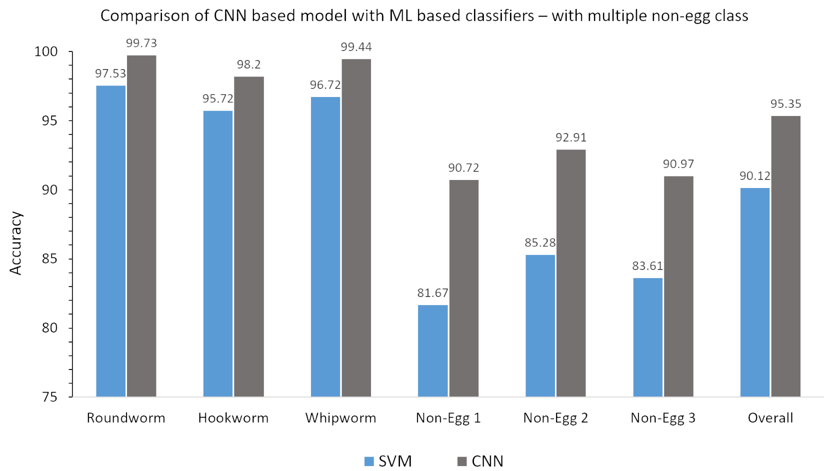


Figure 7-15: Comparison of classification accuracy of ML-based and CNN-based models - six classes

The analysis shows that CNN-based classification models outperform machine learning classifiers in accurately identifying different types of parasite eggs. One of the primary reasons for the higher performance of CNN models is their ability to learn hierarchical feature representations from raw input images. CNNs automatically learn features starting from low-level features like edges and textures and progressively capture more complex patterns from the images. In contrast, ML-based classifiers rely on handcrafted features, which may not be sufficient for effectively identifying all images in a dataset. Additionally, RGB images are used with CNNs, while the ML-based methods utilizes features extracted from grayscale images. Converting higher-dimensional RGB images to lower-dimensional grayscale images results in the loss of information, which may contribute to the lower performance of ML-based models.

7.2.4 Comparison of Classification Result with Previous Works

A comparative analysis is conducted to evaluate the results of the CNN-based classification approach in identifying different types of parasite eggs against some well-known works in this field. The comparison, shown in Table 7.11 reveals that our work successfully competes with previous studies and is capable of identifying various types of parasite eggs effectively, even with a limited amount of unique data.

Table 7.11: Comparison of results with the works that used CNN for classification of parasite eggs

Work	Parasite Egg Class	Classification Result
N. Butploy et al., 2021 [94]	1	93.33%
F. Grijalva et al., 2022 [137]	6	98.66%
T. Suwannaphong et al. 2023 [95]	4	98.25%
V. Savitha et al. 2023 [138]	5	99.72%, 98.20%, 99.51%, 99.47%, 99.46%
Our work	**3**	**99.73%, 98.20% and 99.44%**

7.3 CNN-based Object Detection Technique for Parasite Egg Detection

Convolutional Neural Networks (CNNs) have emerged as a basis in object detection due to their remarkable ability to learn hierarchical representations of visual data. CNN-based object detection techniques typically involve multiple stages, beginning with a convolutional backbone network such as VGG, ResNet, or EfficientNet, which extracts high-level features from the input image. These features are then fed into a detection head comprising layers responsible for localization and classification. One of the pioneering architectures in this domain is the Region-based Convolutional Neural Network (R-CNN) [3] family, including Faster R-CNN [139] and Mask R-CNN. R-CNN introduced a two-stage approach to object detection. In its first stage, it generates a set of region proposals by selective search, effectively identifying potential object locations in the image. These proposals are then passed through a convolutional neural network, typically pre-trained on ImageNet, to extract fixed-length feature vectors. Finally, these features are fed into support vector machines (SVMs) for classification and regression to refine the bounding boxes. The process of detecting objects using R-CNN can be visualized in Figure 7-16.

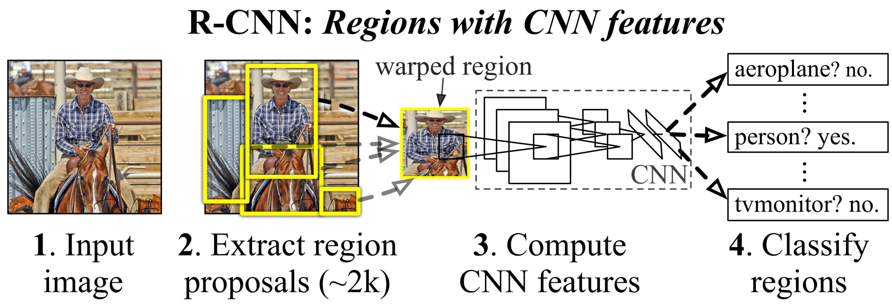
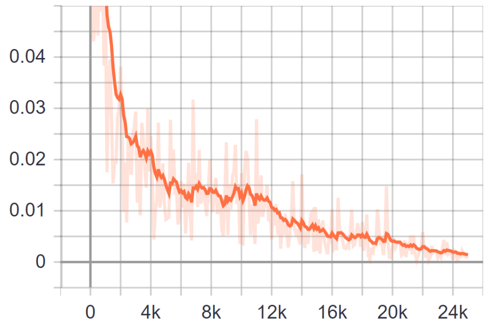


Figure 7-16: Working principle of R-CNN [3]

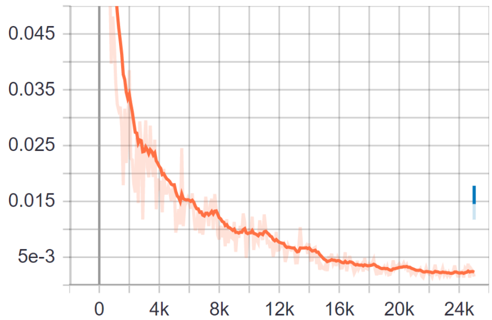
Faster R-CNN, introduced by Shaoqing Ren et al. [139], addresses the computational bottleneck of R-CNN by integrating the region proposal step into the network itself. It introduces the Region Proposal Network (RPN), which shares convolutional features with the subsequent object detection network. The RPN generates region proposals directly from the feature maps, allowing for efficient region proposal generation without the need for external algorithms. By integrating the region proposal network with the convolutional backbone, Faster R-CNN achieves remarkable performance improvements over its predecessor. Additionally, Faster R-CNN introduces anchor boxes, which serve as reference bounding boxes to predict object locations and sizes, further enhancing detection accuracy.

In this study, faster-RCNN technique with the Inception-V2 as a backbone network is utilized to detect the parasite eggs in the images of our database. The size, aspect ratio, number of anchor boxes, maximum number of bounding box proposals, NMS threshold, and other training parameters of the original model are fine-tuned accordingly to fit the shape and size of the parasite eggs in the images. To provide training ground truths, all three classes of parasite eggs as well as several non-egg objects that resemble parasite eggs are annotated, as mentioned in chapter 3. The number of training epochs is set to 30,000, but the training is carried out for 24,790 epochs until the overall loss dropped to less than 0.01.

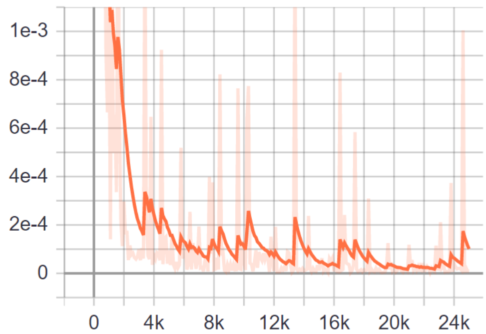
Changes of various loss functions, such as classification loss, bounding box regression loss, detection loss, and overall or total loss, are recorded during the training process, which can be visualized in Figure 7-17.



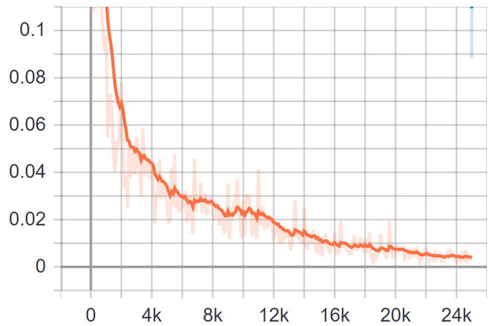
(a) Classification loss: Loss for the classification of detected objects into various classes



(b) Localization loss: Loss of the Bounding Box regressor for the Region Proposal Network (RPN)



(c) Objectness Loss: Loss of the Classifier that classifies if a bounding box is an object of interest or background



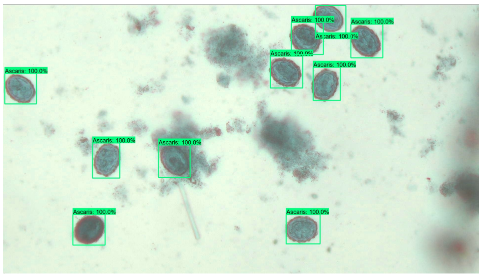
(d) Total Loss: Overall loss of the classification and bounding box regression

Figure 7-17: Visualization of different Losses from the training of Faster-RCNN model

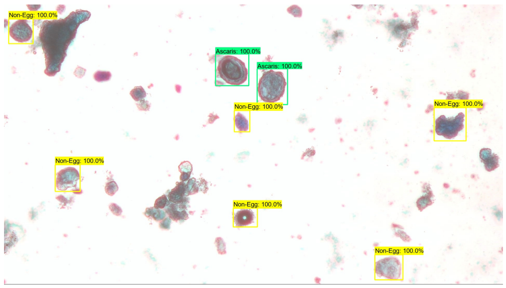
A few test results obtained from the model are shown in Figure 7-18. The results of the evaluation process are recorded in terms of mean average precision at different IoU, and average recall, which are provided in Table 7.12.

Table 7.12: Validation results of Faster-RCNN objects detection model

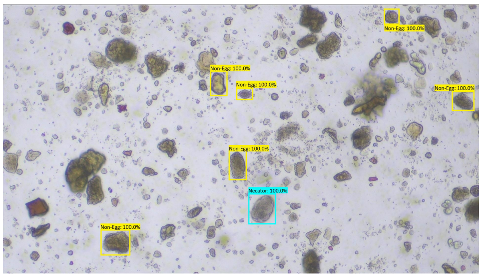
Class	Precision	Recall	mAP (IoU=0.5:0.95)	mAP (IoU=0.5)	mAP (IoU=0.75)	F1 Score
Roundworm	0.90	0.87	0.85	0.90	0.88	0.88
Hookworm	0.88	0.85	0.83	0.88	0.86	0.86
Whipworm	0.89	0.87	0.86	0.91	0.88	0.88
Non-Egg	0.88	0.84	0.82	0.86	0.83	0.86
Average	**0.89**	**0.86**	**0.84**	**0.88**	**0.86**	**0.87**



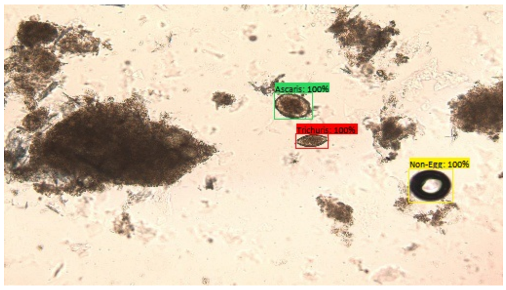
(a) Output image detecting Ascaris egg



(b) Output image detecting Ascaris egg and Non-Egg objects



(c) Output image detecting Necator egg and Non-Egg objects



(d) Output image detecting Ascaris and Trichuris egg

Figure 7-18: Examples of detecting various objects in images using the object detection model.

7.3.1 Comparison of Object Detection Result with Previous Works

A comparison is made between the results of the CNN-based object detection model and some well-known works in this field as depicted in Table 7.13.

Table 7.13: Comparison of our result with the previous works that used CNN-based object detection models for the detection of parasite eggs.

Work	Method	Parasite Egg Class	Result
Nago Q. Viet et al. [96], 2019	Faster-RCNN	8	mAP = 0.9767
A. Kitvimonrat et al. [97], 2020	Faster-RCNN, RetinaNet, and CenterNet	2	mAP = 0.793, 0.674, and 0.50 respectively
Satish Kumar et al. [99], 2023	YOLOv5	5	mAP = 97% (Approx)
Our work	Faster-RCNN	Three parasite eggs and One non-egg class	mAP = 0.88

From the comparative analysis, it is observed that our method achieves slightly lower mean average precision compared to several previous works, mainly due to the smaller and less diverse dataset used.

7.4 Conclusion

This chapter presents the application of deep learning for the segmentation, classification, and detection of parasite eggs. Key contributions and insights of the chapter include:

- **Effective Semantic Segmentation with U-Net**: The U-Net architecture shows promising results in distinguishing parasite eggs from image backgrounds. However, further improvement is needed through diverse image datasets and better annotations. The binary segmentation mode outperforms multi-class segmentation in speed and accuracy, likely due to class imbalances of different objects and parasite eggs.

- **Transfer Learning in Binary Segmentation**: Achieves the highest Intersection over Union (IoU) of 7.2 and F1 score of 90.65 using a transfer learning approach in binary segmentation mode. Addressing overlapping objects remains a challenge, which is potentially solvable with post-processing methods such as the circular Hough transform and watershed algorithm.

- **High Accuracy in Classification**: CNN-based classification models, including transfer learning with VGG16 and a custom CNN architecture, achieve nearly 99% accuracy. This highlights the potential of CNNs over traditional machine learning methods for classifying various types of parasite eggs.

- **Promising Object Detection with Faster-RCNN**: The Faster-RCNN architecture shows acceptable performance in object detection, particularly in terms of mean average precision (mAP) and recall. However, improved detection accuracy is obtainable using higher computational resources, a larger image dataset, and more precise annotations.

- **Enhancing Parasitic Egg Analysis and Diagnostics**: The application of deep learning techniques significantly enhances the accuracy of parasite detection in microscopic images. Despite the need for improvements in dataset diversity, sample quantity, and annotation quality, the research highlights the potential of deep learning in medical diagnostics.