

DETECTION OF HOOKWORM USING DEEP LEARNING

Maheswaran S¹, Indhumathi N², Sathesh S³, Srinithi C⁴, Sanjit A S⁵, Sriram R⁶

1 Associate Professor¹, Assistant Professor^{2,3}, Scholar^{4,5,6},

Department of ECE, Kongu Engineering College, Perundurai, TamilNadu, India

Email: mmaheswaraneie@gmail.com¹,indhunatarajan18@gmail.com²,sathesh808@gmail.com³,

srinithic2000@gmail.com⁴,sanjitsenthilkumar@gmail.com⁵

srirm11079@gmail.com⁶

ABSTRACT:

Hookworms are intestinal, blood-feeding, parasitic roundworms that cause types of infection known as helminthiasis. It has tubular structure with grayish white or pinkish semi-transparent body. The most serious effects of hookworm infection are blood loss leading to anemia, in addition to protein loss. Wireless Capsule Endoscopy (WCE) has become a widely used diagnostic technique to examine inflammatory diseases and disorders. To detect the hookworms two CNN networks, namely edge extraction network and hookworm classification network are used in the existing system. The proposed system has Median filter to remove noise from the dataset. Canny Edge detection is used to detect the edges of hookworm. This technique has low SNR (Signal to Noise Ratio) compared to other edge detection techniques. The architecture of CNN (Convolutional Neural Network) is modified to use it as a classifier. It classifies the hookworm and normal Wireless Capsule Endoscopy (WCE) dataset.

Key Words: Hookworms, Helminthiasis, Convolutional Neural Network, Signal to Noise Ratio, Wireless Capsule Endoscopy

I. INTRODUCTION:

Hookworms are parasitic roundworms that cause types of infection known as helminthiasis. They are blood-feeding parasites and occur in small intestine. Hookworm is common in areas with poor sanitation, less access to adequate water, and hygiene. Since Hookworms are found in the small intestine of humans, it is very difficult for the doctors to diagnose and find hookworms. Thus an endoscopy image plays a major role in detection of hookworms. These endoscopy images are taken with the help of pill camera. The camera is swallowed and it gives the pictures of the intestine. There are two main species of roundworm, belonging to the genera *Ancylostoma* and *Necator* which affects the humans. In other animals the most parasites are species of *Ancylostoma*. A duodenal worm is pale grey or slightly pink. The head is bent a touch in reference to the remainder of the body, forming a hook shape thus the name. The hook is at the front of the body. They have well-developed mouths with two pairs of teeth. The symptoms generally start with itchiness and a little rash caused by an allergy within the area that the larvae entered our skin. This is generally followed by diarrhea as the hookworms grow in our intestine. Other symptoms include abdominal pain, colic, or cramping and excessive crying in infants, intestinal cramps, nausea, fever and a loss of appetite.

II. LITERATURE REVIEW:

JY He et al., (2018) proposed deep hookworm detection framework (DHDF) for WCE (Wireless Capsule Endoscopy) images, which simultaneously models visual appearances and tubular patterns of hookworms.

They have used CNN technique to detect the hookworm. Two CNN networks- edge extraction network and hookworm classification network are seamlessly integrated in the proposed framework, which avoid the edge feature caching and speed up the classification. Two edge pooling layers are introduced to integrate the tubular regions induced from edge extraction network and the feature maps from hookworm classification network, leading to enhanced feature maps emphasizing the tubular regions.

VP Anubala et al., (2018) have done performance analysis of hookworm detection using deep convolutional neural network. Their proposed system consists of two convolutional neural networks, specifically edge withdrawal system along with hookworm categorization systems, which maintain a strategic distance from the edge quality reserving and accelerate the order. To combine the tubular areas got from the edge withdrawal system and the attribute map got from the hookworm taxonomy system, two edges pooling layers were proposed. Weiner filtering is used for noise removal and structured random forest techniques were used for edge extraction. To extract the features from the image K-means clustering was used. This technique has achieved an accuracy of about 89%.

K Pogorelov et al., (2019) proposed an approach for automatic detection of gastrointestinal tract infection of various lesions and abnormalities. The color features and texture features based approach is used for bleeding detection in the study. Color information is used to extract the lesion captured in the image. Color information is an important feature for the initial detection of bleeding images while texture can be used later to extract lesions in the detected images. The proposed approach can distinguish finely between borderline cases. Support vector machine (SVM) with a radial basis function for the classification of bleedy and normal images and achieves higher accuracy.

Aoki et al., (2020) used ResNet50 for automatic detection of blood in WCE images. The aforementioned deep CNN was trained using back-propagation and all layers of the network were fine-tuned using Stochastic Gradient Descent with a global learning rate of 0.0001. To ensure that all images were compatible with ResNet50, the latter were resized to 224×224 pixels. The authors used their own dataset to evaluate the performance obtaining an overall accuracy of 99.89%. The dataset was constituted by a total of 27,847 images (6503 images in which a bleeding region is present from 29 patients and 21,344 images without bleeding regions and acquired from 12 patients).

Kaushik Sekaran et al. (2019) suggested that detection of pancreatic cancer can be done with the combination of GMM technique and CNN. In this they have used Lump Feature Extraction (LFE) and EM algorithm[14-16]. The recognition rate was high compared to GLCM (Gray Level Co-occurrence Matrix), K-Means, K-Medoids and LBP (Local Binary Patterns). Lump Feature Extraction (LFE) and Lump Recognition Algorithm are used.

Maheswaran S et al. (2020) proposed acquisition of grain image, segmentation, extraction of features. Sathesh S et al. (2020) suggested open source software in single board computer (SBC) for compact processing.

III. PROPOSED SYSTEM

The proposed system will have more stages-preprocessing, edge detection, pickle file generation and model file generation. The main reason to get higher accuracy in this work is the changes made in the softmax

layer.

3.1 BLOCK DIAGRAM OF PROPOSED WORK



Fig.3.1 Block Representation of Proposed Method

3.1.1 PREPROCESSING:

The first step in image processing is preprocessing of the dataset. This plays an major role in the work because the noise in the dataset should be cleared effectively. Here Median filter is used to remove noise in the images. Since it is a non-linear digital filter it preserves the edges of the images and allows negative weights which are the biggest advantages of median filter.

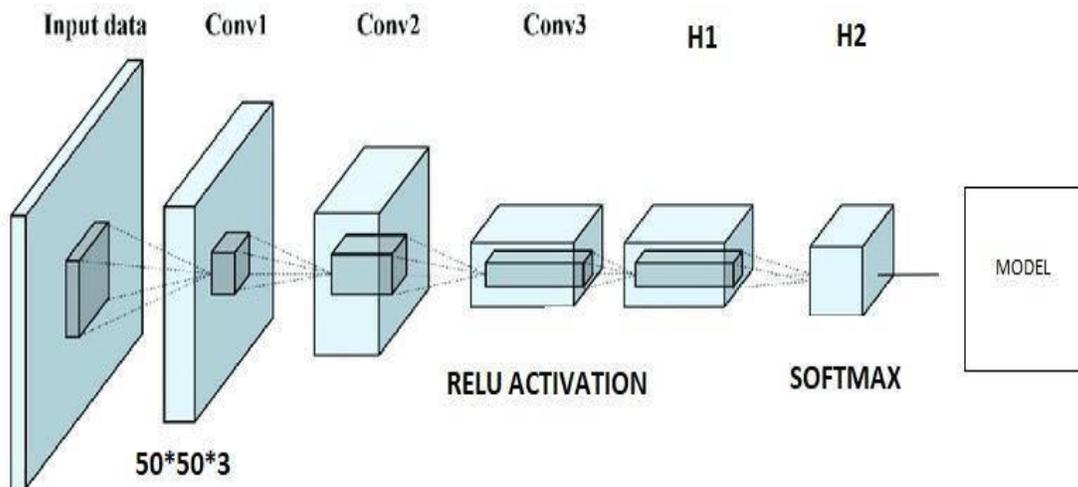
3.1.2 EDGE DETECTION:

After preprocessing, edge detection is done to find the boundaries of objects within images. It works by detecting discontinuities in brightness and is used for image segmentation and data extraction. Canny edge detection method is used in this paper because it has low SNR (Signal to Noise Ratio) value compared to other edge detection techniques like Sobel edge detector.

IV. PROPOSED ARCHITECTURE:

The architecture of proposed CNN technique is given below in the figure. A total of 5 layers are used-3 convolutional layers and 2 hidden layers.

Fig .4.1 Proposed Architecture



4.1 RELU ACTIVATION

The responsibility of activation function is to transform the summed weighted input from the node into the activation of the node or output for that input. The rectified linear activation function or ReLU is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. In other words it is used for rounding off the matrix values. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time and it has less complexity whereas sigmoid activation function uses exponential terms which make it complex.

4.2 SOFTMAX LAYER

The softmax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the softmax function is interpreted as the probability of membership for each class. It is used for problems in multi-class classification where class membership is required on more than two class labels.

The equation of softmax layer is modified in 3 stages to attain the higher accuracy value. The equation of softmax layer is given below.

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{-----} \quad 4.$$

σ = Softmax z = Input Vector

e^{z_i} = Standard Exponential Function for Input Vector K = Number of Classes in The Multi-Class Classifier

e^{z_j} = Standard Exponential Function for Output Vector

4.2.1 MODIFICATION 1

$$\sigma(z) = \frac{e^{z_i}}{z_i + \sum_{j=1}^K e^{z_j}} \quad \text{-----} \quad 4.2$$

Here e^{z_i} is added in the denominator. The standard exponential function for input vector term is added in the denominator.

4.2.2 MODIFICATION 2

$$\sigma(z) = \frac{e^{z_i}}{1 + \sum_{j=1}^K e^{z_j}} \quad \text{-----} \quad 4.3$$

Since the maximum probability is 1, it is added to the denominator.

4.2.3 MODIFICATION 3

$$\sigma(z) = \frac{e^{zi}}{\text{avg}(e^{zi}) + \sum_{j=1} e^{zj}} \text{-----4.4}$$

Here the average of exponential term was added in the denominator. The average of standard exponential function for input vector term is added in the denominator.

V.RESULTS AND DISCUSSION:

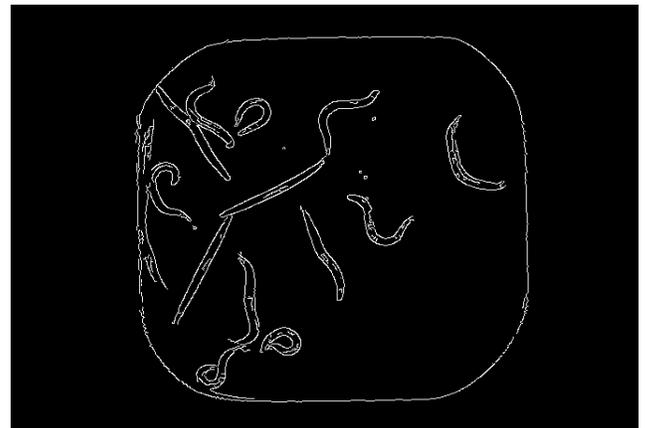
5.1 PREPROCESSING

The input image has noise in it and the edges were not clear as shown in the figure. After the combination of median filter and canny edge detection, the output image was clear and the edges were also visible.

5.1 Input image



5.2 Output image



5.2 PICKLE FILE GENERATION

The pickle file contains the features and labels of the dataset. Two pickle file x.pickle and y.pickle are generated. x.pickle will store the feature values. y.pickle will store the labels. The data stored in pickle file will be byte codes The data folder in the image contains two folders in it. They are Hookworm and Normal. Both the hookworm and normal images are uploaded in their respective folders to store the dataset.

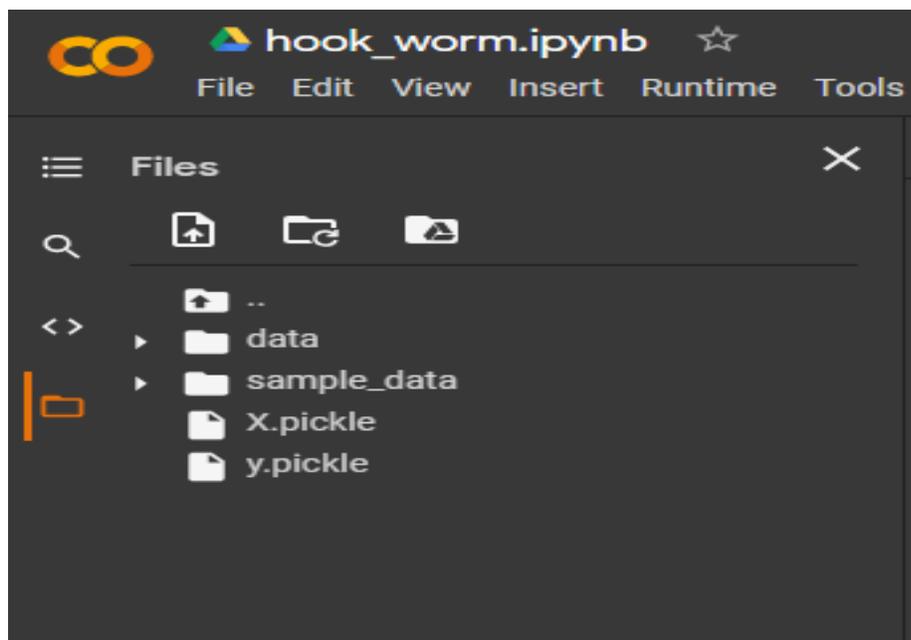


Fig.5.3 Pickle File Generation

5.3 MODEL FILE GENERATION

The model file contains the selected features of the dataset. It stores the class of the dataset available. The two classes are Hookworm and Normal. The generation of model file is given in the figure below. CNN.model is the model file generated. model.h5 and model.json are other formats of storing the model file generated.

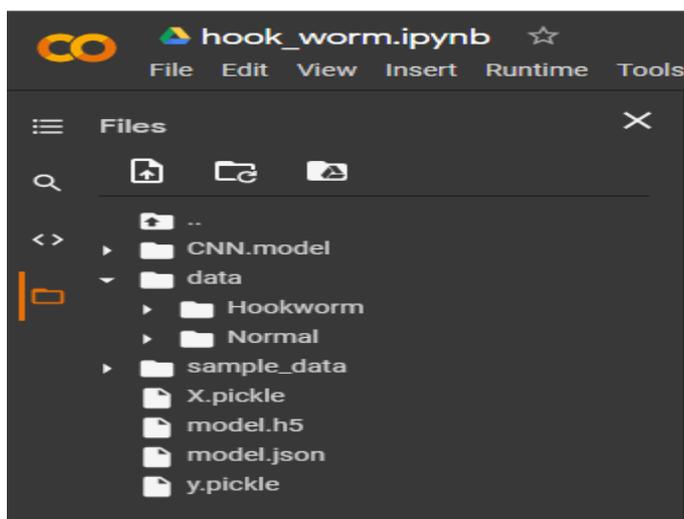


Fig 5.4 Model File Generation

VI.SOFTMAX LAYER MODIFICATIONS

6.1 MODIFICATION 1

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{-----4.1}$$

Here e^{z_i} is added in the denominator. In this modification accuracy rate of 66.67% was obtained. The loss rate of 53.67% is obtained. A total of 25 epochs were used in all types of modifications. The graph explaining the training and validation accuracy and also the training and validation loss of the first modification is given in the figure.

```

Epoch 11/25
1/1 [=====] - 0s 159ms/step - loss: 0.8377 - accuracy: 0.5000 - val_loss: 0.8020
Epoch 12/25
1/1 [=====] - 0s 158ms/step - loss: 0.6397 - accuracy: 0.5556 - val_loss: 1.0163
Epoch 13/25
1/1 [=====] - 0s 158ms/step - loss: 0.8924 - accuracy: 0.5000 - val_loss: 0.9615
Epoch 14/25
1/1 [=====] - 0s 158ms/step - loss: 0.8142 - accuracy: 0.5000 - val_loss: 0.7887
Epoch 15/25
1/1 [=====] - 0s 165ms/step - loss: 0.6181 - accuracy: 0.7222 - val_loss: 0.8245
Epoch 16/25
1/1 [=====] - 0s 168ms/step - loss: 0.6341 - accuracy: 0.5000 - val_loss: 0.9440
Epoch 17/25
1/1 [=====] - 0s 154ms/step - loss: 0.7381 - accuracy: 0.5000 - val_loss: 0.9520
Epoch 18/25
1/1 [=====] - 0s 157ms/step - loss: 0.7416 - accuracy: 0.5000 - val_loss: 0.8731
Epoch 19/25
1/1 [=====] - 0s 164ms/step - loss: 0.6116 - accuracy: 0.5278 - val_loss: 0.8390
Epoch 20/25
1/1 [=====] - 0s 152ms/step - loss: 0.5912 - accuracy: 0.5833 - val_loss: 0.9049
Epoch 21/25
1/1 [=====] - 0s 156ms/step - loss: 0.6231 - accuracy: 0.7778 - val_loss: 0.9507
Epoch 22/25
1/1 [=====] - 0s 159ms/step - loss: 0.6199 - accuracy: 0.7222 - val_loss: 0.9190
Epoch 23/25
1/1 [=====] - 0s 165ms/step - loss: 0.6022 - accuracy: 0.7222 - val_loss: 0.8720
Epoch 24/25
1/1 [=====] - 0s 159ms/step - loss: 0.5668 - accuracy: 0.7500 - val_loss: 0.8735
Epoch 25/25
1/1 [=====] - 0s 155ms/step - loss: 0.5367 - accuracy: 0.6667 - val_loss: 0.8983
Saved model to disk
    
```

Fig 6.1 Accuracy Rate 1

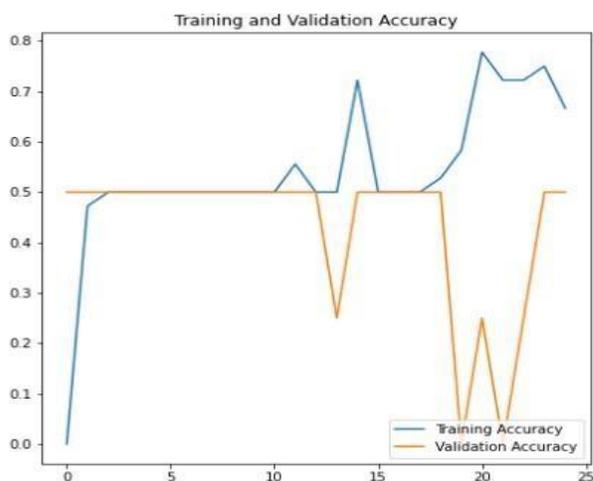


Fig 6.2 Accuracy Graph 1

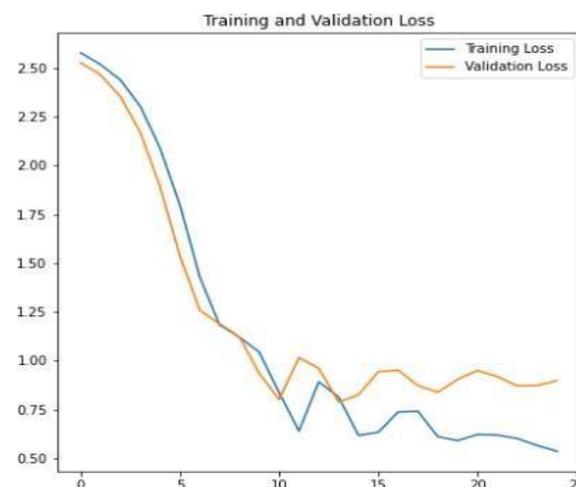


Fig 6.3 Loss Graph 1

6.2 MODIFICATION 2

$$\sigma(z) = \frac{e^{z_i}}{1 + \sum_{j=1}^K \dots} \quad \text{-----4.2}$$

Since the maximum probability is 1, it is added to the denominator. In this modification accuracy rate of 92.86% was obtained. The loss rate of 34.97% is obtained which is very less compared to other modification techniques. The graph explaining the training and validation accuracy and also the training and validation loss of the second modification is given in the figure. The epoch is set to 25 in order to avoid overfitting.

```

1/1 [=====] - 0s 110ms/step - loss: 1.8779 - accuracy: 0.5000 - val_loss: 1.5672
Epoch 10/25
1/1 [=====] - 0s 96ms/step - loss: 1.5647 - accuracy: 0.5000 - val_loss: 1.2259
Epoch 11/25
1/1 [=====] - 0s 108ms/step - loss: 1.2167 - accuracy: 0.5000 - val_loss: 0.9578
Epoch 12/25
1/1 [=====] - 0s 103ms/step - loss: 0.9618 - accuracy: 0.5000 - val_loss: 0.7834
Epoch 13/25
1/1 [=====] - 0s 97ms/step - loss: 0.8150 - accuracy: 0.5000 - val_loss: 0.6280
Epoch 14/25
1/1 [=====] - 0s 108ms/step - loss: 0.6461 - accuracy: 0.5000 - val_loss: 0.6592
Epoch 15/25
1/1 [=====] - 0s 111ms/step - loss: 0.6733 - accuracy: 0.5000 - val_loss: 0.7008
Epoch 16/25
1/1 [=====] - 0s 120ms/step - loss: 0.7236 - accuracy: 0.5000 - val_loss: 0.5498
Epoch 17/25
1/1 [=====] - 0s 100ms/step - loss: 0.5443 - accuracy: 0.9286 - val_loss: 0.6020
Epoch 18/25
1/1 [=====] - 0s 97ms/step - loss: 0.6603 - accuracy: 0.5000 - val_loss: 0.6089
Epoch 19/25
1/1 [=====] - 0s 98ms/step - loss: 0.6337 - accuracy: 0.5000 - val_loss: 0.4575
Epoch 20/25
1/1 [=====] - 0s 98ms/step - loss: 0.4729 - accuracy: 0.8571 - val_loss: 0.5081
Epoch 21/25
1/1 [=====] - 0s 102ms/step - loss: 0.4826 - accuracy: 0.7857 - val_loss: 0.5076
Epoch 22/25
1/1 [=====] - 0s 103ms/step - loss: 0.5786 - accuracy: 0.5714 - val_loss: 0.3333
Epoch 23/25
1/1 [=====] - 0s 99ms/step - loss: 0.3256 - accuracy: 1.0000 - val_loss: 0.3627
Epoch 24/25
1/1 [=====] - 0s 96ms/step - loss: 0.4197 - accuracy: 0.6429 - val_loss: 0.3313
Epoch 25/25
1/1 [=====] - 0s 106ms/step - loss: 0.3497 - accuracy: 0.9286 - val_loss: 0.2199
    
```

Fig 6.4 Accuracy Rate 2



Fig 6.5 Accuracy Graph 2

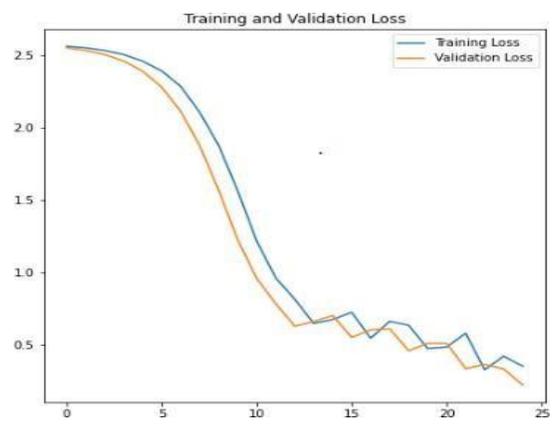


Fig 6.6 Loss Graph 2

6.3 MODIFICATION 3

$$\sigma(z) = \frac{e^{zi}}{\text{avg}(e^{zi}) + \sum_{j=1}^K e^{zj}} \tag{4.3}$$

Here the average of exponential term was added in the denominator. In this modification accuracy rate of 78.38% was obtained. The loss rate of 52.43% is obtained. The graph explaining the training and validation accuracy and also the training and validation loss of the third modification is given in the figure.

```

Epoch 11/25
1/1 [=====] - 0s 168ms/step - loss: 0.8788 - accuracy: 0.5135 - val_loss: 0.9539
Epoch 12/25
1/1 [=====] - 0s 180ms/step - loss: 0.7619 - accuracy: 0.5135 - val_loss: 0.6505
Epoch 13/25
1/1 [=====] - 0s 179ms/step - loss: 0.6120 - accuracy: 0.6486 - val_loss: 0.5484
Epoch 14/25
1/1 [=====] - 0s 177ms/step - loss: 0.6786 - accuracy: 0.4865 - val_loss: 0.5669
Epoch 15/25
1/1 [=====] - 0s 162ms/step - loss: 0.7627 - accuracy: 0.4865 - val_loss: 0.5600
Epoch 16/25
1/1 [=====] - 0s 166ms/step - loss: 0.6849 - accuracy: 0.4865 - val_loss: 0.5879
Epoch 17/25
1/1 [=====] - 0s 166ms/step - loss: 0.6023 - accuracy: 0.5676 - val_loss: 0.7385
Epoch 18/25
1/1 [=====] - 0s 167ms/step - loss: 0.6184 - accuracy: 0.7297 - val_loss: 0.8257
Epoch 19/25
1/1 [=====] - 0s 165ms/step - loss: 0.6664 - accuracy: 0.6757 - val_loss: 0.7209
Epoch 20/25
1/1 [=====] - 0s 164ms/step - loss: 0.6005 - accuracy: 0.7838 - val_loss: 0.5972
Epoch 21/25
1/1 [=====] - 0s 177ms/step - loss: 0.5621 - accuracy: 0.6216 - val_loss: 0.5607
Epoch 22/25
1/1 [=====] - 0s 166ms/step - loss: 0.6079 - accuracy: 0.5135 - val_loss: 0.5581
Epoch 23/25
1/1 [=====] - 0s 169ms/step - loss: 0.5935 - accuracy: 0.5135 - val_loss: 0.5756
Epoch 24/25
1/1 [=====] - 0s 176ms/step - loss: 0.5659 - accuracy: 0.5135 - val_loss: 0.6402
Epoch 25/25
1/1 [=====] - 0s 170ms/step - loss: 0.5243 - accuracy: 0.7838 - val_loss: 0.7352
    
```

Fig 6.7 Accuracy Rate 3

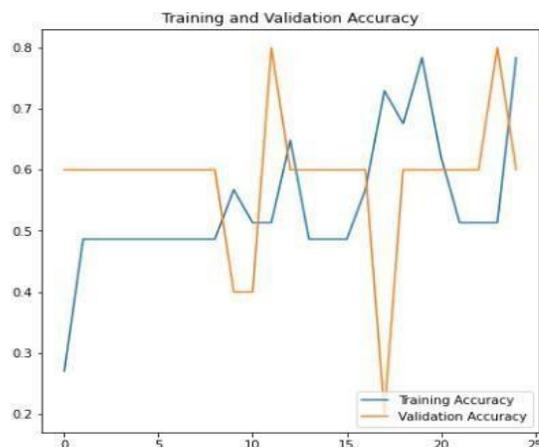


Fig 6.8 Accuracy Graph 3

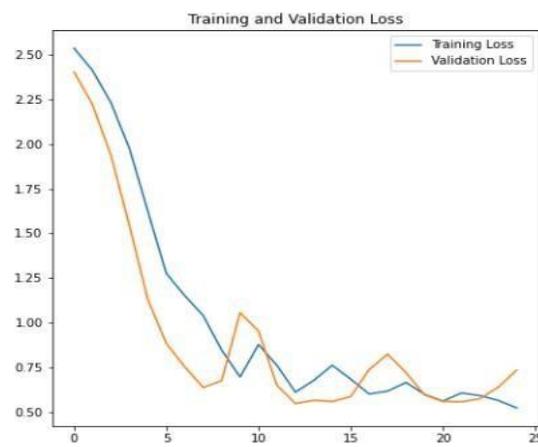


Fig 6.9 Loss Graph 3

6.4 TESTING STATE

After the completion of training phase, the testing occurs. In testing phase if the given image is a hookworm image then it displays as Hookworm as shown in the figure. The testing is based on the class of the image. Here the two classes of the image are Hookworm and Normal. The first class is assigned as Hookworm and the second class is assigned as Normal image. The output is based on the higher value of the class.

A hookworm image is given as input and the matrix has the single value of 1.0 as a first element. The first value is for hookworm and the second value is for normal image. Here only a single value has displayed in first element of matrix. Therefore the output will be displayed as Hookworm

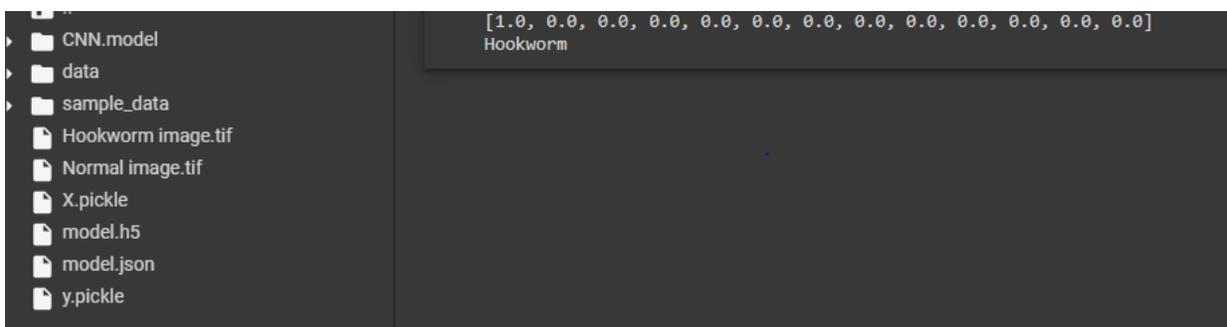


Fig 6.10 Hook Worm Image Testing

In second step a normal image is given as input and it prints as normal as shown in the figure. The first value is around 0.409 and the second value is around 0.590. On comparing both the values 0.590 is higher of the two. It is the second value of the matrix. Therefore the output displays as Normal.

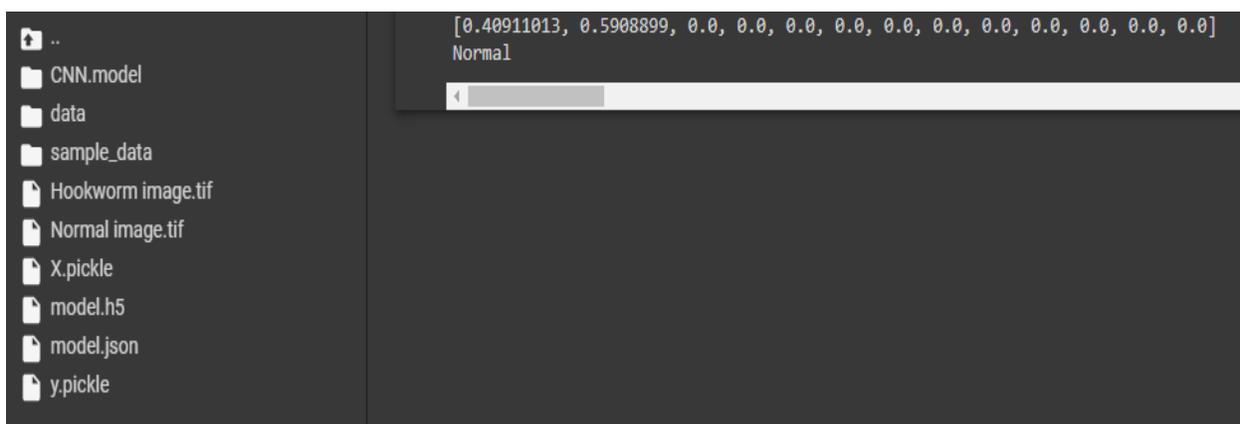


Fig 6.11 Normal Image Testing

VII.CONCLUSION

According to the Centers for Disease Control and Prevention, hookworm infections occur in an estimated 576 to 740 million people worldwide. It mainly affects people in developing nations in the tropics and subtropics due to poor sanitation. Hookworms infected about 428 million people in 2015. Heavy infections can occur in both children and adults, but are less common in adults. They are rarely fatal. Hookworm infection is a soil-transmitted helminthiasis and classified as a neglected tropical disease. Wireless Capsule Endoscopy (WCE) images were used to detect hookworms that are present in the small intestine of humans.

A total of 1407 images of both Hookworm and Normal images were used. The original image size is 696 x 520 pixels. Median filter is used to remove the noise in the image since it is the most effective method to remove noise from the datasets. Canny edge detection is used to detect the edges of the hookworm as it has low SNR value compared to other edge detection techniques. CNN algorithm is used as a classifier and the softmax layer equations were modified to obtain better accuracy. An accuracy rate of about 92.86% is achieved which is higher than the existing system. Therefore, the proposed method can improve the classification performance of hookworm images and then assist doctors in diagnosing diseases.

REFERENCES

- [1] He, J. Y., Wu, X., Jiang, Y. G., Peng, Q., & Jain, R. (2019). Hookworm detection in wireless capsule endoscopy images with deep learning. *IEEE Transactions on Image Processing*, 27(5), 2379-2392.
- [2] Anubala, V.P., Muthukumar, N. and Nikitha, R., 2018, December. Performance Analysis of Hookworm Detection using Deep Convolutional Neural Network. In 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 348-354). IEEE.
- [3] Sekaran, K., Chandana, P., Krishna, N. M., & Kadry, S. (2020). Deep learning convolutional neural network (CNN) With Gaussian mixture model for predicting pancreatic cancer. *Multimedia Tools and Applications*, 79(15), 10233-10247.
- [4] Aoki, T., Yamada, A., Kato, Y., Saito, H., Tsuboi, A., Nakada, A., ... & Tada, T. (2020). Automatic detection of blood content in capsule endoscopy images based on a deep convolutional neural network. *Journal of gastroenterology and hepatology*, 35(7), 1196-1200.
- [5] Li, C., Konomis, D., Neubig, G., Xie, P., Cheng, C., & Xing, E. (2017). Convolutional neural networks for medical diagnosis from admission notes. *arXiv preprint arXiv:1712.02768*.
- [6] Saito, H., Aoki, T., Aoyama, K., Kato, Y., Tsuboi, A., Yamada, A., ... & Tada, T. (2020). Automatic detection and classification of protruding lesions in wireless capsule endoscopy images based on a deep convolutional neural network. *Gastrointestinal endoscopy*, 92(1), 144-151.
- [7] Diamantis, D. E., Koutsiou, D. C. C., & Iakovidis, D. K. (2019, May). Staircase detection using a lightweight look-behind fully convolutional neural network. In *International Conference on*

Engineering Applications of Neural Networks (pp. 522- 532). Springer, Cham.

- [8] Jia, X., & Meng, M. Q. H. (2017, July). Gastrointestinal bleeding detection in wireless capsule endoscopy images using handcrafted and CNN features. In 2017 39th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 3154-3157). IEEE.
- [9] Pogorelov, K., Suman, S., Azmadi Hussin, F., Saeed Malik, A., Ostroukhova, O., Riegler, M., ... & Goh, K. L. (2019). Bleeding detection in wireless capsule endoscopy videos—Color versus texture features. *Journal of applied clinical medical physics*, 20(8), 141-154.
- [10] Vani, V., & Prashanth, K. M. (2020). Ulcer detection in Wireless Capsule Endoscopy images using deep CNN. *Journal of King Saud University-Computer and Information Sciences*.
- [11] Nakai, H., Fujimoto, K., Yamashita, R., Sato, T., Someya, Y., Taura, K., ... & Nakamoto, Y. (2021). Convolutional neural network for classifying primary liver cancer based on triple-phase CT and tumor marker information: a pilot study. *Japanese journal of radiology*, 1-13.
- [12] Maheswaran, S., Vivek, B., Sivaranjani, P., Sathesh, S., and Vignesh K.P. (2020). Development of Machine Learning Based Grain Classification and Sorting with Machine Vision Approach for Eco-Friendly Environment. *Journal of Green Engineering*, vol. 10, no. 3, pp. 526-543.
- [13] S, Sathesh, Pradheep V A, Maheswaran S, Premkumar P, Gokul Nathan S, and Sriram P, "Computer Vision Based Real Time Tracking System to Identify Overtaking Vehicles for Safety Precaution Using Single Board Computer." *Journal of advanced research in dynamical and control systems*, vol. 12, no. SP7, July 2020, pp. 1551–61. *DOI.org (Crossref)*, doi:10.5373/JARDCS/V12SP7/20202258.
- [14] P. Prabu, Ahmed Najat Ahmed, K. Venkatachalam, S. Nalini, R. Manikandan, Energy efficient data collection in sparse sensor networks using multiple Mobile Data Patrons, *Computers & Electrical Engineering*, Volume 87, 2020,
- [15] V.R. Balaji, Maheswaran S, M. Rajesh Babu, M. Kowsigan, Prabhu E., Venkatachalam K, Combining statistical models using modified spectral subtraction method for embedded system, *Microprocessors and Microsystems*, Volume 73, 2020.
- [16] Malar, A.C.J., Kowsigan, M., Krishnamoorthy, N. S. Karthick, E. Prabhu & K. Venkatachalam (2020). Multi constraints applied energy efficient routing technique based on ant colony optimization used for disaster resilient location detection in mobile ad-hoc network. *Journal of Ambient Intelligence and Humanized Computing*, 01767-9.