Medical Waste Classification using Deep Learning and Convolutional Neural Networks

Mark Verma, Arun Kumar, and Somesh Kumar, IEEE Senior Member
Atal Bihari Vajpayee - Indian Institute of Information Technology and Management
Gwalior, India
markvrma@gmail.com

***Abstract*—With the rise of attention to healthcare since the start of the century, which the recent pandemic has emphasized, the number of hospitals and clinics has increased exponentially. The growth in hospitals and patients has also resulted in increased medical waste. The different kinds of medical waste must be segregated and disposed of properly to prevent the spread of bacteria and viruses and cross-contamination. However, it is not economically feasible to hire a workforce that can segregate said waste. With the perceived popularity of deep learning and image classification systems, creating a Deep Learning model to categorize the different kinds of medical waste is possible. Hence using a deep learning-based classification method in which an appropriate pre-trained model is selected for practical implementation, followed by transfer learning methods to improve classification results, is appropriate. Different types of medical waste are grouped into umbrella categories(general, hazardous, infectious). An ideal situation would be where images are uploaded, and the machine can classify the presented waste appropriately with little to no waiting times. Three out of four of the modified pre-trained models with different architectures were able to achieve an accuracy above 95 percent.**

***Index Terms*—Deep Learning, Medical Waste Classification, Transfer Learning, Convolutional Neural Network(CNN).**

I. Introduction

As per the guidelines set by the government of India, medical waste is required to be segregated at the point of origin. Moreover hiring workforce to specifically segregate waste will only increase expenses for hospitals which would trickle down to patients. Some hospitals have outsourced this task to companies which results in a decline in the quality of medical waste segregation [1]. Negligence to not properly segregate medical waste can have far reaching consequences. From cross contamination of medical waste adversely affecting medical staff to improper disposal of medical waste which woulersely affect the environment. Examples of medical waste includes syringes, needles, infusion apparatus, bandages, gloves, gauze, blood bags among other items [2].Machine Learning algorithms can solve numerous practical problems, however, Deep Convolutional Neural Networks are better at effectively performing feature extraction and classification on images which is how we classify the medical waste [3]. A CNN is essentially made up of convolution layers, subsampling layers, and an output layer which are arranged in a feed-forward structure where convolution layers are followed by a sub-sampling layers, and the last convolution layer is followed by the output layer [4]. A through detailed explanation can be found in [4].

The goal is to apply Transfer Learning to an existing model using a custom-built dataset to classify medical waste into three categories: **General, Hazardous,** or **Infectious**. A commonly used transfer learning approach is to pre-train a neural network on the source domain (e.g., ImageNet, which is an image database containing more than fourteen million annotated images with more than twenty thousand categories) and then fine-tune it based on the instances from the target domain, which in our case is the medical waste [5]. The intuition behind Transfer Learning for image classification is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model for most general objects [6]. Fine-tuning the higher-order feature representations in the base model helps to make the model more relevant for the specific task. The different types of medical wastes to be classified are:

- Hazardous Waste, which consists of syringes, needles, infusion bags, medical bottles, blood bags
- Infectious Waste, which consists of used/unused gauze, bandages, gloves, and masks.
- General Waste which consists of paper cups, wrappers, and bottles.

In this work, we aim to create a Convolution Neural Network to detect and classify different kinds of medical waste. Due to a lack of adequately structured datasets that are publicly available for medical waste classification, this also requires creating a well-curated dataset [7]. Based on the accuracy of other Deep Learning Models that are used in image classification, the model will also be required to have an impressive accuracy. Hence different pre-trained models are tested and fine-tuned to achieve the model with the highest level of accuracy [5].

The rest of the paper is organized as, in Section 2, the proposed system and methodology are described, including workflow, creation of the dataset, and model Building. Section 3 will discuss model training and optimization. Section 4 delves into analyzing the results and discussion. The whole paper is concluded and future scope of the proposed system is discussed in Section. 5.

978-1-6654-7719-2/22/$31.00 © 2022 IEEE

II. Proposed System and Methodology

The proposed system separates general, hazardous, and infectious elements of medical waste. The proposed solution aims to create a web application that can detect medical waste and classify it. The image of the waste taken from a device will be uploaded to the website, where it will be processed and generate results based on the analysis of the approximate amounts of pre-determined classes in the image. The results generated will display the type of medical waste to the user.

Once model training is completed, the models are converted to either a layer model or a graph model, depending on the pre-trained model and different layers used, from their original HD5 formats. These new converted models contain a model.json file with information about the weights and some sharded binary files. This model is then loaded using Tensorflowjs, where a user can upload an image and get a response on what type of medical waste is used. The image is converted to the appropriate shape that the model requires. The model returns a probability array, with each value being a probability of how much the object in the image belongs to a particular type of waste. A minimum threshold probability value of 0.4 has to be achieved for an image to be considered as a part of a medical waste type, else it outputs "Unknown" to the user. Figure 1 shows the proposed workflow, which shows the creation of the deep learning model, which will then take an image as an input and outputs an array of probabilities for each kind of medical waste, which will be reflected on the website.

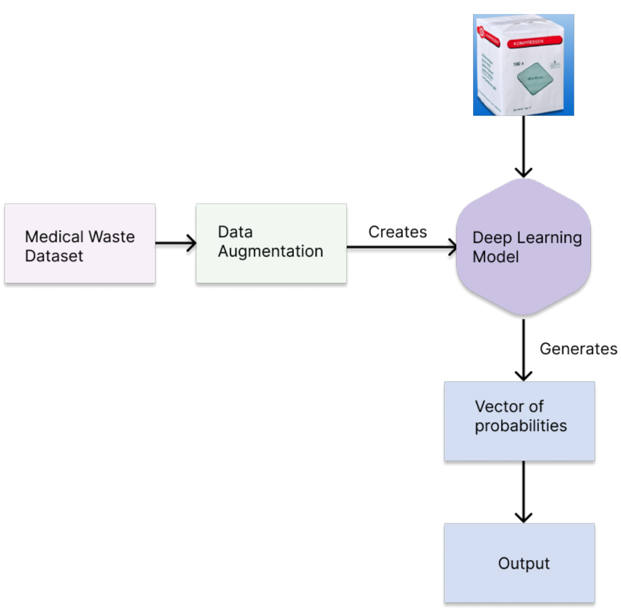

Fig. 1. Workflow of the system.

III. Methodology

The following steps are followed for the development of the proposed system.

Fig. 2. Examples of Images from the Dataset

A. Building the Dataset

Using JavaScript, URLs for the different medical waste types are collected from Google Images into a text file. This is followed by setting up a virtual environment and using a Python script that can use the text file to download the images from the URLs, check that the files are not corrupted, and save them with the appropriate name. The images are then manually sorted to check if the images downloaded are reliable to maintain quality. Figure 2 shows examples of the different kinds of medical waste images collected. The dataset created contains 11 different kinds of medical waste, with an even distribution through all the types to ensure that the model would not overfit the data.

Table 1 shows the number of images of the different kinds of medical waste collected.

TABLE I
Types of Medical Waste

Medical Waste	Number of Images
Plastic Cups	267
Plastic Bags	303
Wrappers	184
Infusion Apparatus	299
Infusion Bag	372
Infusion Bottle	315
Syringes and Needles	301
Tweezers	333
Gauze	345
Gloves	267
Masks	302

B. Data Augmentation

The goal of data augmentation is to increase the size of the dataset by applying various transformations to the existing dataset and prevents overfitting [8]. There are many different sub-categories of transformations. For the current dataset, traditional transformations are chosen. These include a combination of the affine image transformations and color modifications as well as geometric transformations [9]. It is essential to show a degree of restraint while applying different data transformations, as applying too many transformations could be detrimental to the accuracy of the model [10]. Figure 3 shows an example of data augmentation performed on one of the dataset's images.

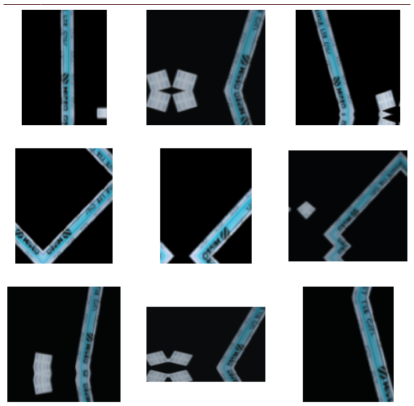

Fig. 3. Example of Data Augmentation on an Image

C. Model Building

Since the pre-trained model is used to get the general features and are transferable, these layers are frozen, and the pre-trained model is fine-tuned by the last few convolutional layers and fully connected layer [11]. These extra layers help adjust the model to work better for our particular use case. These layers include GlobalAveragePooling2D, BatchNormalization, Dropout Regularization, and Softmax Activation Function.

1) GlobalAveragePooling2D: This pooling block takes a tensor of the size that is most appropriate for the pre-trained model, since in this case the tensor is a form of an image, and computes the average of the width and height across all 3 channels. The output is a 1D tensor of size 3 [12].

2) Batch Normalization: This layer applies transformations that maintain the mean output almost close to zero and standard output close to 1. This reduces the time required to train the model. Batch Normalization also works toward reducing internal covariate shift [13].

$$\mu = \frac{1}{n}\sum_{i=1} Z^i \tag{1}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}(Z^i - \mu)^2 \tag{2}$$

$$Z^i_{norm} = \frac{Z^i - \mu}{\sqrt{\sigma^2 - \epsilon}} \tag{3}$$

$$\mathsf{Z} = \gamma * Z^i_{norm} + \beta \tag{4}$$

The BatchNormalization layer first determines **mean** μ and **variance** $\boldsymbol{\sigma^2}$ of the activation values using (1) and (2), and then normalizes the activation vector Z^i with (3). This ensures that each neuron's output has a normal distribution across the batch. ϵ is used to stabilize the value in case there is noise. It then computes the layers output Z, where γ and β are parameters that are trained through gradient descent, using Exponential Moving Average to place more emphasis on the latest iterations.

3) Dropout Regularization: Deep neural networks can learn complicated relationships between inputs and outputs. However, if the training data set is limited, there is a high chance of overfitting the dataset. Dropout Regularization is a technique where neurons in a layer are randomly disconnected from the network, and training is then done on this diminished network. This results in multiple independent internal representations being learned by the network, which means the Deep Neural Network is not overly sensitive to certain weights of neurons [14]. Hence the Deep Neural Network will not overfit the data and is now better at generalization.

IV. Model Training and Optimization

GlobalAveragePooling2D, BatchNormalization, and **Dropout Regularization** are heavily prevalent in Deep Neural Networks that make use of Transfer Learning.

To compare the different models, accuracy and loss function graphs are plotted, and precision, recall, and F-score are calculated. The different pre-trained models selected are VGG19, InceptionV3, EfficientNetB7, and ResNet50. These models each have their advantages, for example, VGG19 has a relatively simple architecture but requires a lot of computation power, InceptionV3 is less computationally hungry but has trouble with being scaled up [15]. ResNet50 has a more complex architecture, while EfficientNetB7 is very sensitive to the slightest changes in hyperparameters [16].To calculate F-score, precision and recall, a separate test dataset of around 300 images is used with roughly the same distribution as the training dataset.

A. Precision

Precision is the percentage of how many results that were processed are correct [17]. Mathematically it is the ratio of True Positives to the sum of True Positives and False Positves as shown in (5) below.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

B. Recall

Recall is the percentage of how many correct results were found after processing [17]. Mathematically it is the ratio of True Positives(TP) to the sum of True Positives(TP) and True Negatives(TN) as shown in (6) below.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

C. F-Score

F-score is the harmonic mean of Precision and Recall as showin in (7) below [17].

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

All the code was tinkered with and executed on Kaggle since it provides 40 hours of uninterrupted GPU usage per week, with a more intuitive user interface compared to Google Colab. It should be noted that the image resolution for training

TABLE II
PRECISION, RECALL, AND F-SCORES OF THE 4 MODELS AGAINST THE DIFFERENT TYPES OF MEDICAL WASTE

	EfficientNetB7			InceptionV3			ResNet50			VGG19		
Types of Waste	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score	Precision	Recall	F-Score
General	0.934	0.934	0.934	0.934	0.934	0.934	1.000	0.9	0.947	0.875	0.934	0.903
Hazardous	0.980	1.000	0.981	0.980	1.000	0.990	0.962	1.000	0.983	0.980	1.000	0.990
Infectious	0.958	0.921	0.958	0.958	0.921	0.939	0.923	0.960	0.941	0.954	0.840	0.893

EfficientNetB7 had to be capped at 256x256 since Kaggle had limited memory space.

The variable parameters for the pre-trained models considered are, probability value for Dropout Regularization, number of neurons used for each dense layer and the number of extra layers used [6]. Each model is trained for 40 epochs.

V. RESULTS AND DISCUSSIONS

Table III shows that all the adjusted models were able to get their accuracy values above 93%, and three out of the four models were able to get their accuracy above 95% and loss value less than 0.15. Observing Fig. 4, Fig. 5, Fig. 6, and Fig. 7, we see a common trend between the four figures, with a sudden inflection point third epoch for all the models. However, EfficientNetB7 is able to achieve convergence for both accuracy and loss function. Analyzing table II we observe that among the four models, EfficientNetB7 has consistently performed well with regards to precision, recall, and F-score among all three types of waste, with InceptionV3 and ResNet50 close behind. Fig x shows an example of the website a potential user could be using, where an image is uploaded and the results displayed.

TABLE III
ACCURACY AND LOSS OF DIFFERENT MODELS

	Accuracy	Loss
EfficientNetB7	0.9555	0.1280
VGG19	0.9338	0.1720
InceptionV3	0.9649	0.1010
ResNet50	0.9572	0.1133

VI. CONCLUSION

A competent deep convolutional neural network is built that can classify images into three categories, **General, Hazardous, and Infectious**. Around 3200 images are used to train the models, which can primarily achieve above 95% accuracy and have performed remarkably well on test data. Once the model is loaded, the overall processing time is less than 2 seconds, which is practical for real-world usage. The model is only trained to classify one type of waste at a time. If the model is confronted with an image of multiple types of medical waste, it will be unable to distinguish between the

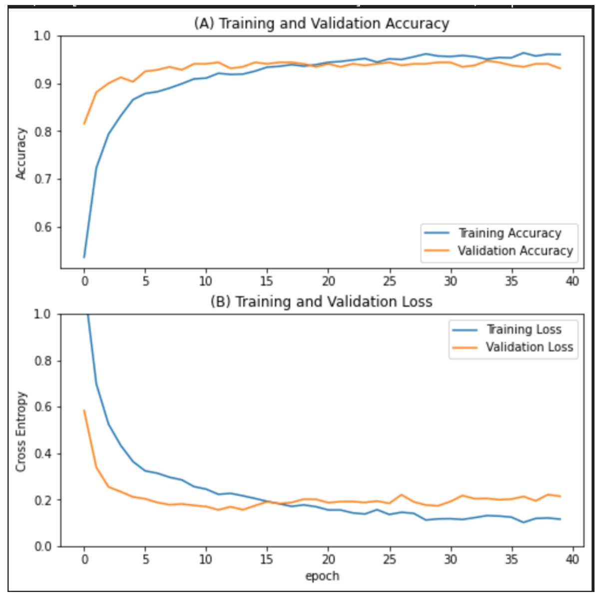

Fig. 4. ResNet50 - (A) Training and Validation Accuracy (B) Training and Validation Loss

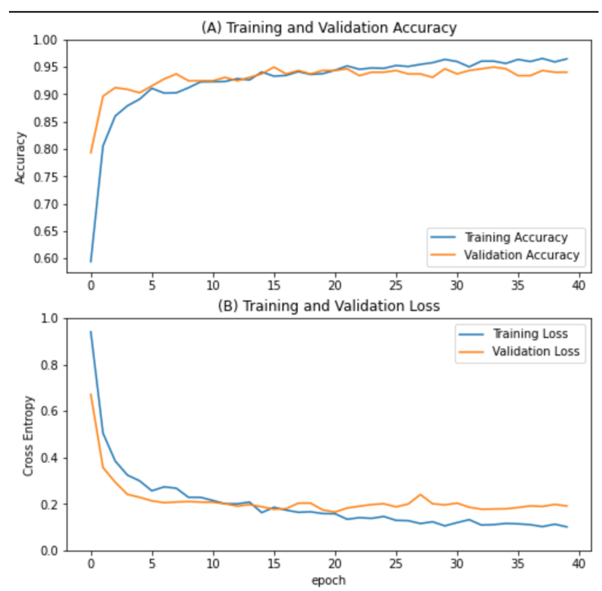

Fig. 5. InceptionV3 - (A) Training and Validation Accuracy (B) Training and Validation Loss

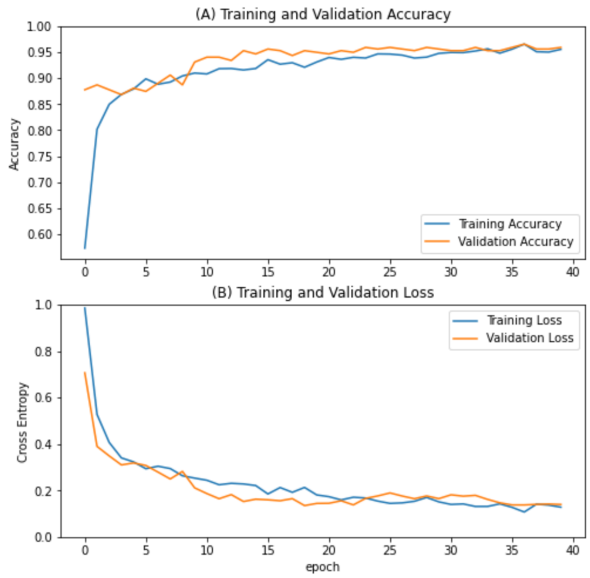

Fig. 6. EfficientNetB7 - (A) Training and Validation Accuracy (B) Training and Validation Loss

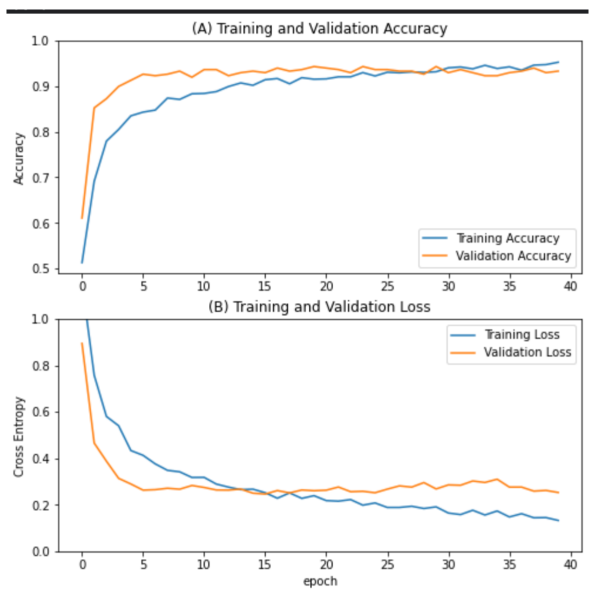

Fig. 7. VGG19 - (A) Training and Validation Accuracy (B) Training and Validation Loss

different kinds of waste successfully. This will require the help of computer vision, which will identify the different types of medical waste and label them with rectangular boxes.

Proceeding to increase the size of the dataset to further identify and categorize a greater variety of medical waste and individually labeling the images(using software like LabelImg), it would be possible to create a computer vision program using Tensorflowjs that could categorize the different kinds of medical waste in real time without needing to upload images first. This could be used in IoT devices which could be programmed to identify and segregate medical waste, which

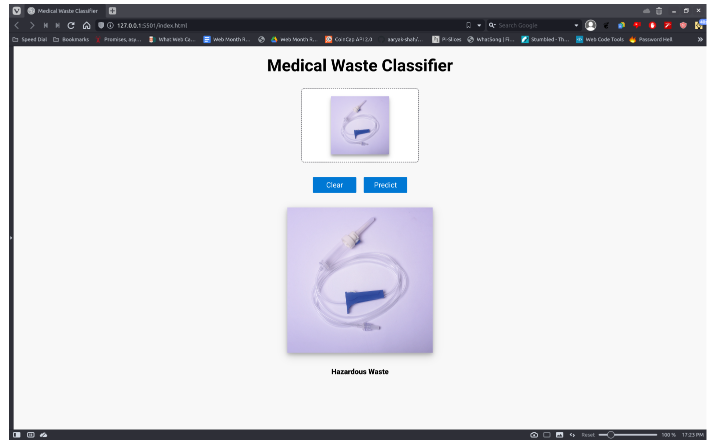

Fig. 8. A Proof of Concept Frontend for the Model where the user can upload images of different medical waste and have them immediately

can be of immense practical use to hospitals.

References

[1] P. Datta, G. Mohi, and J. Chander, "Biomedical waste management in india: Critical appraisal," *Journal of Laboratory Physicians*, vol. 10, p. 6, 01 2018.

[2] W.-T. Tsai, "Analysis of medical waste management and impact analysis of covid-19 on its generation in taiwan," *Waste Management & Research*, vol. 39, no. 1_suppl, pp. 27–33, 2021, pMID: 33666120.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[4] H. Ejaz and S. Mehdi, "Convolution neural network based image classifier," 01 2021.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. PP, pp. 1–34, 07 2020.

[7] H. Zhou, X. Yu, A. Alhaskawi, Y. Dong, Z. Wang, Q. Jin, X. Hu, Z. Liu, V. Kota, M. Abdulla, S. Ezzi, B. Qi, J. Li, B. Wang, J. Fang, and H. lu, "A deep learning approach for medical waste classification," *Scientific Reports*, vol. 12, 02 2022.

[8] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 12 2017.

[9] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117–122.

[10] J. Jin, A. Dundar, and E. Culurciello, "Robust convolutional neural networks under adversarial noise," 11 2015.

[11] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 11 2014.

[12] M. Lin, Q. Chen, and S. Yan, "Network in network," 12 2013.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 02 2015.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 06 2014.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.

[17] D. Powers, "Evaluation: From precision, recall and f-factor to roc, informedness, markedness and correlation," *Mach. Learn. Technol.*, vol. 2, 01 2008.