

Received 20 July 2022, accepted 8 August 2022, date of publication 22 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3199661

RESEARCH ARTICLE

Do-It-Yourself Recommender System: Reusing and Recycling With Blockchain and Deep Learning

SACHI PANDEY¹, VIKAS CHOUHAN², DEVANSHI VERMA¹, SHUBHAM RAJRAH¹,
FAYADH ALENEZI³, RAJKUMAR SAINI^{2,4}, AND KC SANTOSH⁵

¹Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Ghaziabad 201204, India

²Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee 247667, India

³Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka 72388, Saudi Arabia

⁴EISLAB Machine Learning, Department of Computer Science, Electrical, and Space Engineering, Luleå Tekniska Universitet, 971 87 Luleå, Sweden

⁵Applied Artificial Intelligence (2AI) Research Laboratory, Department of Computer Science, University of South Dakota, Vermillion, SD 57069, USA

Corresponding authors: Sachi Pandey (sachipandey.231@gmail.com) and Rajkumar Saini (rajkumar.saini@itu.se)

This work was supported by the Applied AI Research Laboratory, University of South Dakota.

ABSTRACT Due to aggressive urbanization (with population size), waste increases exponentially, resulting in environmental damage. Even though it looks challenging, such an issue can be controlled if we can reuse them. To handle this, in our work, we design a machine learning and blockchain-oriented system that identifies the waste objects/products and recommends to the user multiple ‘Do-It-Yourself’ (DIY) ideas to reuse or recycle. Blockchain records every transaction in the shared ledger to enable transaction verifiability and supports better decision-making. In this study, a Deep Neural Network (DNN) trained on about 11700 images is developed using ResNet50 architecture for object recognition (training accuracy of 94%). We deploy several smart contracts in the Hyperledger Fabric (HF) blockchain platform to validate recommended DIY ideas by blockchain network members. HF is a decentralized ledger technology platform that executes the deployed smart contracts in a secured Docker container to initialize and manage the ledger state. The complete model is delivered on a web platform using Flask, where our recommendation system works on a web scraping script written using Python. Fetching DIY ideas using web-scraping takes nearly 1 second on a desktop machine with an Intel Core-i7 processor with 8 cores, 16 GB RAM, installed with Ubuntu 18.04 64-bit operating system, and Python 3.6 package. Further, we evaluate blockchain-based smart contracts’ latencies and throughput performances using the hyperledger caliper benchmark. To the best of our knowledge, this is the first work that integrates blockchain technology and deep learning for the DIY recommender system.

INDEX TERMS Deep learning, image recognition, municipal solid waste, blockchain, smart contract, hyperledger fabric, hyperledger caliper, recycle.

I. INTRODUCTION

According to a World Bank report, [1], approximately 2.01 billion metric tons of Municipal Solid Waste (MSW) is produced annually worldwide. The World Bank estimates that waste generation will increase to 3.40 billion metric tons by 2050. Out of total waste produced, it is estimated that 13.5% of it is recycled, while 5.5% is composted. The report

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang⁶.

also shows that 33% to 40% of waste generated worldwide is not appropriately managed and instead dumped or openly burned. This calls for responsible measures to manage waste properly. It is a challenging problem because, with urbanization, the easy availability and low cost of products have increased every individual’s carbon footprint. There is a tremendous increase expected in global waste production in the coming years. This will be a considerable challenge to various environmental bodies, scientific communities, governments, and humans in general, as it will affect all of us

drastically. The less we manage our waste, the more risk we put our planet in. Climate change is real, and it is affected by all factors that pose a disturbance to the ecological balance. In this regard, the proposed work attempts to curb the problem early by suggesting various DIY ideas to reuse the waste to the users.

The conventional waste management systems are doing well up to a certain level; however, we have wastes that can be efficiently reused or recycled. Therefore, there is a need for a more practical rather out-of-the-box approach. The rate at which we discard used products has increased directly in proportion to new products' availability. When a new design of shoes comes up, we usually discard the old ones even when they have not been used to their fullest. Suffice to say that the rate of discarding products directly generates more waste. More waste generation, combined with ineffective waste management methods (i.e., improper disposal of the waste causing pollution), has become one of the biggest threats to our planet today. A few solutions recognize the waste objects into various classes; however, they do not focus on what to do with the objects or how to reuse them [2]. The issue is the lack of focus on how to handle the problem of waste generation. Another issue is the number of classes that the objects are classified into. The waste generated can be classified into a wide variety of classes; however, it was observed that the object's output classes are less in number [3], [4], [5].

We developed a convincing notion to decrease waste generation as the current recommendation systems lack machine learning-based object recognition and correctness verification. The idea of recycling an item before discarding it has been around for decades, but the execution has been limited. To boost that implementation, we devised a simple recommendation system for Reusing Ideas specific to any particular object.

The mantra to combat resource scarcity is REUSE rather than labeling the waste as True Waste and finding effective ways to recycle it. The reason is very straightforward; recycling requires more resources than reusing. Recycling should be done when the reuse policy is hard to follow. In cases where reusing can not be done, the economic impact of waste is best known when the waste products are segregated. There are currently few effective systems used to segregate waste into various types at a household level. Nowadays, researchers utilize advantages (i.e., transparency, trust, non-repudiation, fairness) of blockchain technology in diverse applications such as payment system [6], cloud storage [7], file sharing [8], supply chain management (product traceability) [9], and healthcare [10]. Thus we utilized this technology in our proposed system. Our work aims to make a user-friendly, trusted recommendation and segregation system for households. This system recommends DIY reusing Ideas and suggests to the users what segregation class the waste belongs to segregate it themselves before discarding it to the municipal waste collection system. This encourages resource reuse and a helping hand in saving the environment by reducing the amount of waste going to be dumped. It gives suggestions

about the processes that can be performed on the waste object. It also provides a proper analysis of the waste product. For better understanding, we have divided our approach into two main modules.

- 1) Image Recognition Module (IRM): This machine learning (deep learning) module focuses on recognizing the object into a wide variety of classes using ResNet50 [11], a Deep Neural Network (DNN) trained on a huge image dataset [12], [13], [14].
- 2) Web Scrapping Module (WSM): WSM scraps the reuse recommendation ideas from the internet based on the IRM output, and these ideas are then presented to the user.

IRM recognizes the object in a user's submitted image, and WSM advises on how to reuse it. To enable verification of suggested recommendations in a trusted manner, we utilized blockchain technology in our work.

Blockchain is record-keeping technology that collects every transaction in the distributed and immutable ledger. Immutability limits record modification and establish trust in a blockchain network. The distributed network allows every network member to audit the existing transactions. In this work, blockchain transactions are verified collaboratively by the expert members/ auditors of Blockchain Network (BCN) to improve recommendation quality and maximize user trust. The Hyperledger Fabric (HF) blockchain platform is designed to serve as a foundation for developing modular applications or solutions. This HF platform provides exceptional levels of confidentiality, scalability, and resiliency. We use the HF platform to implement and test the proposed business logic written in the smart contracts. The Smart Contract (SC) is automatically executed when a particular event is initiated. We designed several smart contracts in the proposed work and deployed them on the Hyperledger Fabric blockchain platform. Expert members interact with BCN using smart contracts such as register, view, delete, and disable members. The DIY instance record deals with BCN using SCs to create, verify, and invalidate the DIY instance record.

The main contributions to the paper are as follows:

- 1) The proposed approach works on a variety of daily use objects (12 classes). It recognizes the object in a given image and recommends ways to reuse the objects using DIY reusing ideas.
- 2) We designed and deployed the novel smart contracts that support verifying the DIY recommendations. Further, we captured and demonstrated the performance of smart contracts.
- 3) We performed extensive experiments in real-time. This was achieved through the Python library OpenCV, which uses the system's webcam to recognize the objects. We used 11,700 images collected from different sources to train the IRM model. Our model performs satisfactorily in the recognition and suggestions. Additionally, we present a comparative analysis with existing approaches.

The rest of the paper is organized as follows. Section II reviewed the related work. The proposed framework and smart contracts are presented in Section III. We present the description of the dataset, the experimental setting, and the results in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORKS

Deep learning with Convolutional Neural Network (CNN) has been used in variety of applications including image classification [2], [3], detection [16], and identification [17]. This section reviews the existing works that aim to recognize waste objects. In a study by Srinilta *et al.* [2], a system for waste segregation is developed using various transfer learning models like DenseNet-121, ResNet-50, MobileNet, and VGG-16. The dataset consists of about 9200 Municipal Solid Waste images collected from multiple sources. All the transfer learning models yield an accuracy of more than 80%. Waste items with different physical appearances can still fall into the same class because of the way these items are processed. For example, paper and glass are recyclable, so they can end up in the same class. One of the previous works lays the groundwork for how resnet50 performed better in classifying waste objects as compared to VGG-16, MobileNet V2, and DenseNet-121 classifiers [2]. Similarly, Aral *et al.* [3] classify the TrashNet dataset using various Deep Learning Models. TrashNet [13] is a very popular dataset released by Yang and Thung in 2016. It consists of 6 classes: paper, glass, cardboard, plastic, metal, and trash. This study uses various transfer learning models like Xception [18], MobileNet [16], DenseNet [19], and Inception V4 [20], where DenseNet-121 and DenseNet-169 [19] yield a recognition rate of nearly 95%. Rabano *et al.* [4] develop a Common Garbage Classification system using the TrashNet dataset. MobileNets [16] are used due to their capability for faster and more efficient computation. The models do many tasks, including classification and detection. The baseline model has an accuracy of 87.2%. In another study, Sreelakshmi *et al.* [5] segregate Plastic and Non-Plastic Wastes using Capsule Neural Networks. A system to segregate waste like plastic and non-plastic is of utmost importance, with huge waste being produced daily. Capsule-Net analyzes two datasets derived from public and private areas. It recognizes the objects placed on the conveyor belt. The accuracy achieved is 96.3% in Dataset 1 and 95.7% in Dataset 2.

The pollution caused by road transportation of the MSW collection system is a crucial source of environmental pollution which can result in lung cancer, asthma, allergies, and various breathing problems along with severe and irreparable damage. Therefore, Amal *et al.* [21] investigate an efficient computerized method for optimizing MSW collection that minimizes the environmental and other factors. Chen *et al.* [22] introduce an integrated transportation and storage system for radioactive waste processing in nuclear power plant decommissioning. This research used the technology of Radio Frequency Identification for detection and

information system. This further leads to adequate monitoring of the storage status. Later, Moslehi *et al.* [23] applies a Scenario-based Stochastic Programming (SSP) approach to deal with the uncertainties such as parameters of demand and Waste Electrical and Electronic Equipment (WEEE) return rate, which is obtained from the consumer. It solves the model using a nominal approach and an SSP approach via the epsilon-constraint and augmented epsilon-constraint methods to obtain optimal Pareto solutions and compare them.

The recent studies [24], [25], [26], [27], [28] provide blockchain-based solutions of waste management. França *et al.* [24] develop an information system to tackle integrity and quality assurance of information in a business model. The information system uses blockchain for solid waste management in a municipality using social currency. A solid waste management system was designed for people to earn Green coins in exchange for the solid household wastes that people take to the city hall. These Green Coins have a monetary value using which people can buy products from registered bakeries, supermarkets, drugstores, and grocery stores. The Municipality sells the waste to recycling companies completing the cycle. Falazi *et al.* [25] aim to provide a uniform way to deal with heterogeneous smart contracts by defining and creating the Smart Contract Invocation Protocol (SCIP). It helps invoke smart contract functions and to query past and future smart contract-related events uniformly, irrespective of the underlying blockchain technology. This paper aims to develop an abstract layer as a wrapper around blockchains, exposing a consistent unified set of operations that allow external applications to interact with smart contracts. This would eliminate the need to adapt to different protocols and heterogeneous Application Programming Interfaces (APIs). Recently, Mahapatra *et al.* [27] introduced an e-waste optimization solution based on the Hyperledger Fabric platform to recover precious metals from waste circuit materials. They use microwave heat treatment on electronic wastes to extract the most precious metals. Transfer learning models like ResNet [11], DenseNet [19], VGG [29], Inception [20] and other similar models provide deep architectures with pre-trained weights on Imagenet dataset, which make the task of object recognition relatively easier. Smart systems can be built upon them.

In contrast, the primary purpose of the proposed work is the waste classification and recommendation that provides ways to deal with the waste objects. In this work, we incorporated 12 classes with images collected and classified from various sources. The number of output classes is more than the existing approaches [2], [3], [4], [5]. Additionally, we have demonstrated in Table 1 that compares the proposed method with the existing works.

III. PROPOSED APPROACH

In this section, an elaborated plan is designed to come up with the desired blockchain-oriented framework. This framework consists of a DIY application divided into two modules- IRM and WSM. A representative system model of the proposed

TABLE 1. Comparison with existing approaches on waste recognition models.

Approach	# of Images	Architecture	Classes	Accuracy
1. Proposed work	11696	ResNet50	12	94.17%
2. Municipal Solid Waste Segregation with CNN [2]	9200	DenseNet/ResNet	4	91.30%
3. Classification of TrashNet Dataset Based on Deep Learning Models [3]	2527	DenseNet	5	95%
4. Material Classification of Recyclable Waste using the Weight and Size of Waste [15]	Hardware based project, Database created for size of objects	Arduino Mega Microcontroller used for detecting dimensions	10	93%
5. Common Garbage Classification using MobileNet [4]	2527	MobileNet	5	87.2%

blockchain-oriented framework is illustrated in Figure 1. In this framework, the DIY Blockchain Network (DBN) consists of several peer nodes, distributed applications, and smart contracts. DIY App is a distributed application that runs on a peer-to-peer DBN, and a smart contract consists set of business logic that are auto-triggered when a particular event is invoked. Our application provides an interface where a user can either upload an image or display the object through the webcam. The object image is then fed as an input to the Image Recognition model (IRM) to make an identification. The predicted class is then further fed to the Web Scraper, which fetches recommendations on reusing the object using DIY ideas. This is user-friendly as it can be used as a web application. The object recognition with the idea recommendation is demonstrated in Figure 2, where WSM uses the results from IRM to retrieve probable recommendations.

The deep architectures such as ResNet-50 trained over the ImageNet dataset can learn distinguishable representations of the input images without any handcrafted features. They have been proven efficient in handling images with background clutter and poor lighting conditions compared to traditional non-deep learning approaches. Such an architecture can be fine-tuned by training with task-specific images without much effort; it may be an obvious choice for further research. In the image recognition phase, ResNet-50 CNN architecture is used [11]. As a DNN consists of many layers, using a sigmoid activation function can lead to the vanishing gradient problem, thereby causing poor learning for deep networks. This problem can be solved using Residual Networks with a ReLU activation function. The residual blocks allow us to train many DNNs. Due to the effectiveness of the ResNet-50 (Residual Network) architecture, it is used as the base model, trained on the ImageNet dataset. The weights are retained; however, the output layer can be changed. Such a framework can be used with other networks. A dense layer is added with ReLU used as the activation function to overcome the vanishing gradient problem. This layer is followed by the drop-out layer, which helped overcome the problem of overfitting. After some hyperparameter tuning, the drop-out rate was set as 0.2. The Softmax function in the last layer produces the output between 0 and 1 so that it can be interpreted as probabilities. As there are 12 output classes, there will be a probability value corresponding to every class, and probabilities for all 12 classes add to 1. Adam is used for stochastic optimization [30]. Due to this algorithm's adaptive learning

rate, the model could achieve a stable model swiftly. With hyperparameter tuning, the learning rate of 0.0001 worked the best for our model.

In the initial phase of data pre-processing, data augmentation is performed to create more images in the training phase. Data augmentation helps deal with the problem of fewer data [31]. The images are first resized into 224×224 RGB channel and then normalized. The images' orientation is changed using various pre-processing techniques like horizontal and vertical flip, zoom in and zoom out, cropping, and rotating the images. During the training phase, these images are fed to the model.

The testing is done in real-time using OpenCV [32]. It is a library that aims at handling real-time computer vision problems. A mask is created enclosing the waste object, for which the prediction result is fetched and displayed on the screen itself, as shown in Figure 3. The result of the object recognition CNN model is used in the Web Scraping Module.

The next phase is the recommendation system. Web scraping is performed on the results obtained from IRM using a Python script. The recommendations suggested DIY ideas to reuse the waste object instead of disposing of them, as shown in Figure 4.

The two modules are combined together in a Flask web application [33], where it is possible for a user to either upload an image or directly input the image frame from the webcam and obtain prediction followed by reusing DIY recommendations.

A. SMART CONTRACTS

A transaction protocol that is particularly invoked based on predefined specific execution policies. It will perform specified actions according to the input arguments. However, we cannot update the installed smart contract (chaincode) for a dedicated channel. Instead, we can try the chaincode version to identify the difference between the currently upgraded chaincode and the previous chaincode. A smart contract is a domain-specific program linked to certain business operations, whereas a chaincode is a technical container that holds a collection of related smart contracts. With transactions contributed by applications, chaincode initializes and controls the ledger state. The world state of a chaincode can be kept isolated from the world state of other chaincodes by using a namespace. However, smart contracts in the same chaincode have shared direct access to the same world state, whereas

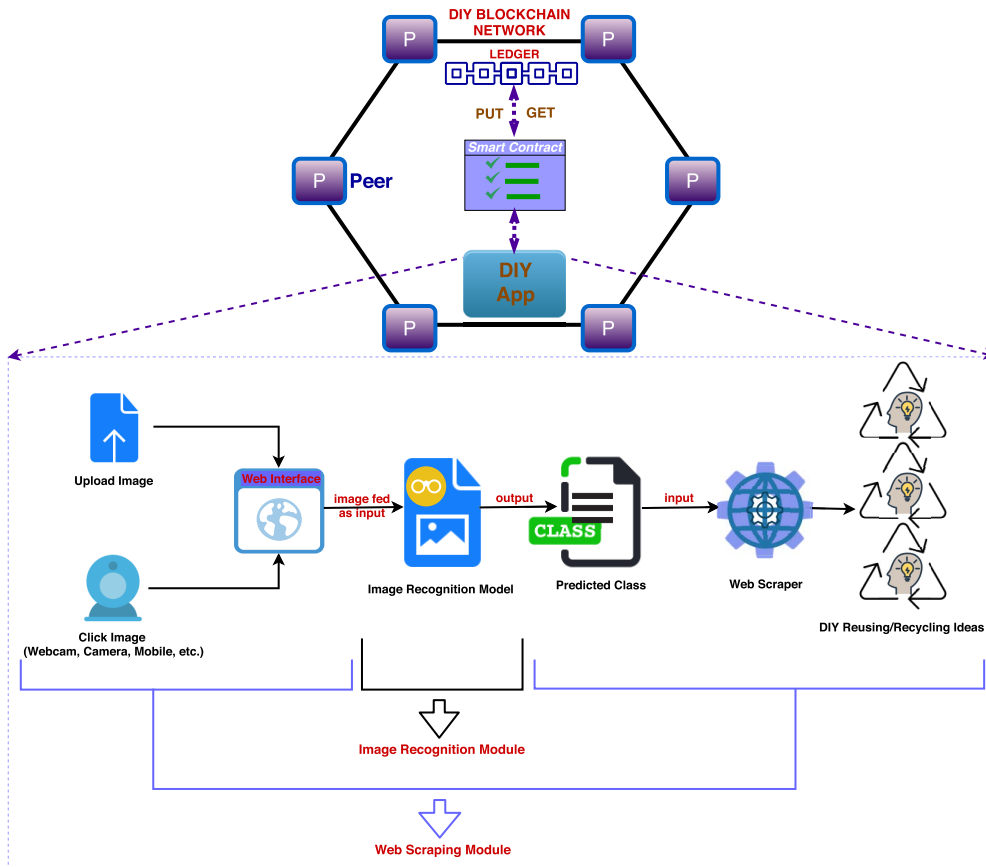


FIGURE 1. Blockchain-oriented Framework for the object recognition and idea recommendation.

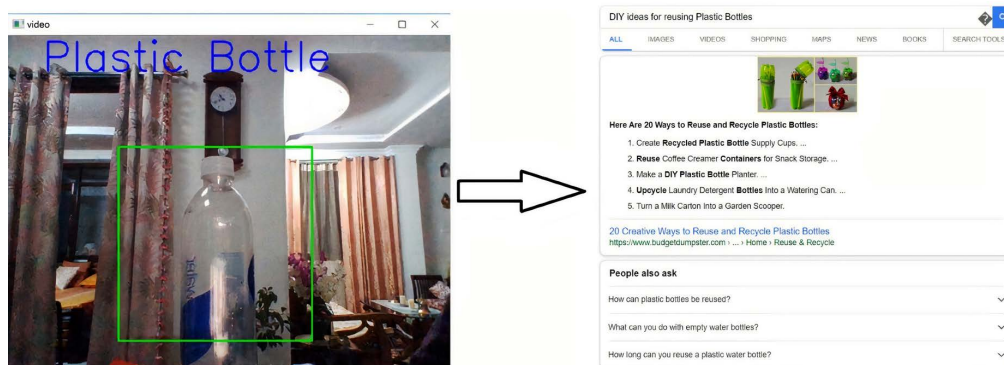


FIGURE 2. Prediction and recommendation of object.

smart contracts in other chaincodes do not. The fabric passes the stub parameter when a chaincode’s initialize, invoke, or query functions are called. This stub can be used to access ledger services, and transaction context and invoke another chaincode by calling APIs.

We designed and deployed several Smart Contracts (SC) in the blockchain platform, such as registering, viewing, deleting, and disabling expert members and creating, verifying, and invalidating instance records. We represent three essential contracts to deal with recommended ideas, i.e., insert DIY instance details, verify the DIY record, and invalidate the record.

The ‘create instance record’ contract creates an instance record with Instance Identity (IID), image data, number of ideas, and validity status. Then, it is inserted into the decentralized blockchain described in Algorithm 1. It takes an argument as an input and a particular chaincode (smart contract) stub interface, and it returns a corresponding response. This contract first defines the number of input DIY ideas (ideas_count). Then it appends all the ideas into the DIY list. Later, the ‘input’ instance structure is defined and initialized with the required information. The default validity status is set to true for the suggested ideas. The experts will verify it. Then it checks whether the instance record exists and returns

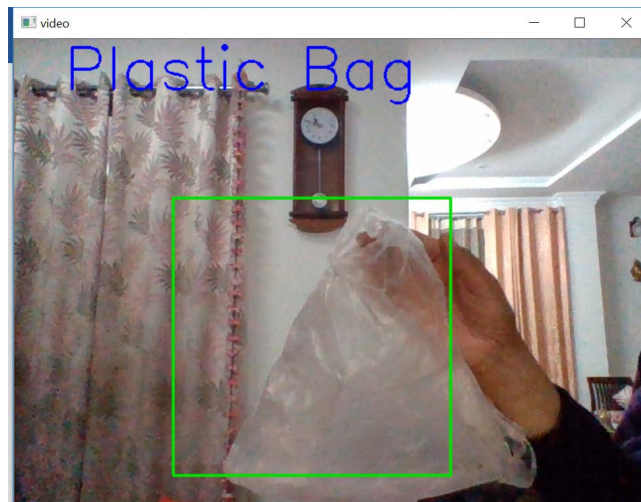


FIGURE 3. Prediction of a sample image – plastic bag.

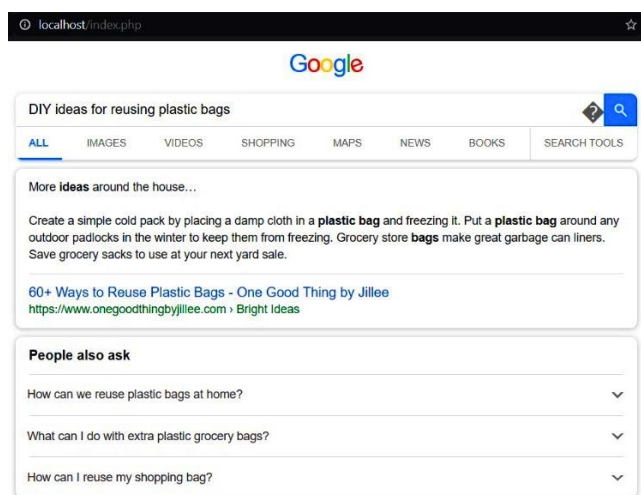


FIGURE 4. DIY recommendation on reusing recognized waste object – plastic bag.

an error if the instance record already exists. Later, it marshals the input instance and stores them into the blockchain corresponding to the key IID.

Algorithm 2 represents the ‘verify instance record’ SC that describes the procedure of viewing and verifying the DIY record from the blockchain platform. This contract takes an argument as an input and a particular chaincode stub interface and returns a corresponding response. This contract first initializes the instance id by the input argument. First, it invokes the ‘GetState’ method of the blockchain to retrieve stored information as a ‘diyRecordAsBytes’ if the record information exists; otherwise, it fails to get state and returns an error response. Next, the ‘diyrecord’ instance is created to store the unmarshal information of ‘diyRecordAsBytes.’ Then it calls the function ‘ViewRecord’ to view the recommended ideas to verify their correctness and returns their validity status. It invokes the ‘Invalidate Instance Record’ contract to reset the validity status to false if incorrect DIY recommendation. Finally, it returns a success response, including retrieved bytes information.

Algorithm 1 Smart Contract: Create and Insert Instance Record for Recommending Ideas

```

Input : stub ChaincodeStubInterface, args [ ]string
Output: Response

1 begin
2   ideas_count :← args[2]
3   var diylist DiyIdeas
4   for i :← 0; i < ideas_count; i ++ do
5     /* Diy ideas start from the third
6      index of the arguments list */
7     Append(diyList.Ideas, args[i + 3])
8   end
9   /* Initializing Instance structure
10  Values */
11  var input Instance
12  input.IID ← InitStructValue(args[0])
13  input.ImageData ← GetImageData(args[1])
14  input.DiyListsize ← InitStructValue(args[2])
15  input.DiyList ← diyList
16  input.Validity ← true
17  /* Check if instance record already
18  exists */
19  if IsDiyRecordExist(stub, input.IID) then
20    return Error(“This instance record”+ input.IID +
21    “is already exists.”)
22  end
23  /* Putting state in block */
24  PutState(input.IID, Marshal(input))
25  /* diy instance record has been entered
26  */
27  return Success(“diy instance record has been stored”)
28 end

```

Algorithm 2 Smart Contract: Verify Recommended Ideas

```

Input : stub ChaincodeStubInterface, args [ ]string
Output: Response

1 begin
2   iid :← args[0]
3   diyRecordAsBytes, err :← stub.GetState(iid)
4   if err != nil then
5     jsonResp :← “{\“Error\”:
6     \“Failed to get state for”+ iid + “\”}”
7     return Error(jsonResp)
8   end
9   var diyrecord Instance
10  Unmarshal(diyRecordAsBytes, &diyrecord)
11  validitystatus :← ViewRecord(diyrecord)
12  if validitystatus == false then
13    jsonResp :← “{\“Error\”:
14    \“This diy recommendation is wrong – for”+
15    iid + “\”}”
16    InvalidateInstanceRecord(stub, diyrecord.IID)
17    return Error(jsonResp)
18  end
19  return Success(diyRecordAsBytes)
20 end

```

The ‘invalidate instance record’ contract invalidates the stored record’s correctness for the recommended DIY recommendation, exhibited in Algorithm 3. This contract takes an argument as an input and a particular chaincode stub

Algorithm 3 Smart Contract: Invalidate Instance Record

```

Input : stub ChaincodeStubInterface, args [ ]string
Output: Response
1 begin
2   iid :← args[0]
3   diyRecordAsBytes, err :← stub.GetState(iid)
4   if err != nil then
5     jsonResp :← “{\“Error\”:
6       \“Failed to find record by” + iid + “\”}”
7     return Error(jsonResp)
8   end
9   var diyrecord Instance
10  Unmarshal(diyRecordAsBytes, &diyrecord)
11  diyrecord.Validity ← false
12  /* updating instance record          */
13  /* Putting state in block            */
14  PutState(diyrecord.IID, Marshal(diyrecord))
15  /* diy instance record has been updated */
16  return Success(“diy instance record has been updated”)
17 end

```

interface and returns a corresponding response. This contract initializes the variable instance id (*iid*) by the input argument. First, it invokes the ‘GetState’ method of the blockchain to retrieve stored information as a ‘diyRecordAsBytes’ if the record information exists; otherwise, it fails to get state and returns an error response. Next, the ‘diyrecord’ instance is built to store the unmarshal information of ‘diyRecordAsBytes.’ Then it resets the validity status of the stored record to false. After, it marshals the ‘diyrecord’ instance and stores them into the blockchain corresponding to the key IID. Finally, it returns a success response, including a message ‘diy instance record has been updated’.

The Hyperledger Fabric (HF) blockchain platform has been adapted to deploy smart contracts. The Linux Foundation launched hyperledger – an enterprise-grade – open-source distributed ledger system. HF is a modular blockchain architecture that serves as a platform for developing blockchain-based goods, solutions, and applications. It supports plug-and-play components such as consensus and membership services. Its modular and adaptable design caters to a wide range of business use cases. It also provides a novel approach to a consensus that enables performance at scale while maintaining privacy. Some of its benefits include channel privacy and confidentiality, modular design, efficient data transmission processing, chaincode functionality, and identity management. The HF component Membership Service Provider (MSP) provides an abstraction of membership operations. This membership identity service keeps track of user identities and authenticates all participants. The following are the essential characteristics that distinguish HF as a comprehensive and configurable Blockchain solution:

- **Assets:** It enables the network to exchange nearly anything with a monetary value.
- **Chaincode:** A set of programs implement the business logic governing how applications interact with the

ledger. The execution of chaincode is partitioned from transaction ordering and verification across node types, as well as network scalability and performance optimization.

- **Ledger features:** The shared ledger is immutable and stores the whole transaction history for each channel, as well as SQL-like querying capabilities for quick auditing and dispute resolution.
- **Channel privacy:** It competes with businesses and regulates industries that exchange assets on common network leverage channels and private data collectors to enable private and confidential multi-lateral transactions.
- **Members’ security and services:** Permission-based membership creates a trusted blockchain network in which participants are certain that all transactions can be detected and observed by authorized regulators and auditors.
- **Consensus:** A unique consensus procedure provides the organization with the flexibility and scalability it requires.

IV. EXPERIMENTS

This section presents the description of the dataset, implementation details, and the results obtained. Results are specified in terms of the accuracy obtained by the Image Recognition model, the time taken to recommend DIY ideas’ and smart contracts’ performance metrics.

A. DATASET DESCRIPTION

The dataset for the study contains around 11700 images and is collected from multiple sources. The dataset is a combination of images from ImageNet [12], TrashNet [13] and a Kaggle Dataset [14] also containing images with complex shapes and image background. These images are divided into 12 output classes, based on which recommendations will be made.

For the experiments, the dataset is split into training and validation data, which contains 80% and 20% data, respectively. The testing of the model is done in real-time, using OpenCV. The webcam captures the frame and converts it into a grid of 224 × 224 RGB format; the model predicts the result for this frame and displays it on the screen.

Data preparation is a delicate technique, more like an art that deals with incorrect labels, missing values, incorrectly formatted data, and sometimes even corrupted data. We have classified our dataset into 12 classes. The data distribution has been shown in Figure 5.

A screenshot of images used in the class “cardboard” is presented in Figure 6. The size of each image in our dataset is converted to a 224 × 224, including the three color (red, green, blue) channels RGB. This means that each pixel in a grid of 224 × 224 is represented by a 1D array containing three elements, each representing the three channels RGB. Consider the following representation for the input image as a grid of 224 × 224 as shown in Figure 7.

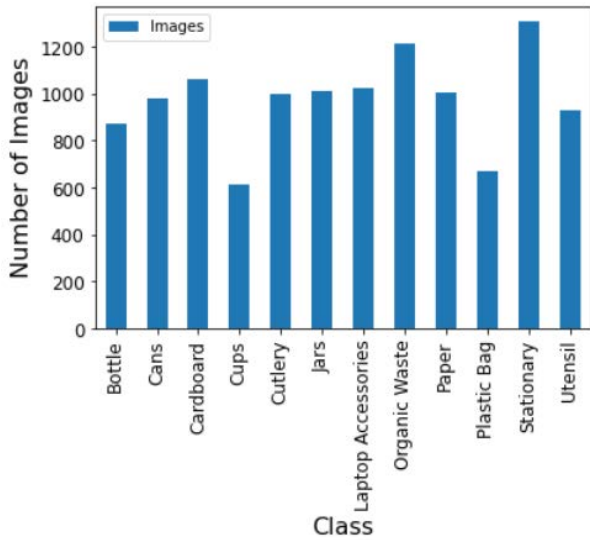


FIGURE 5. Data distribution over different waste classes.

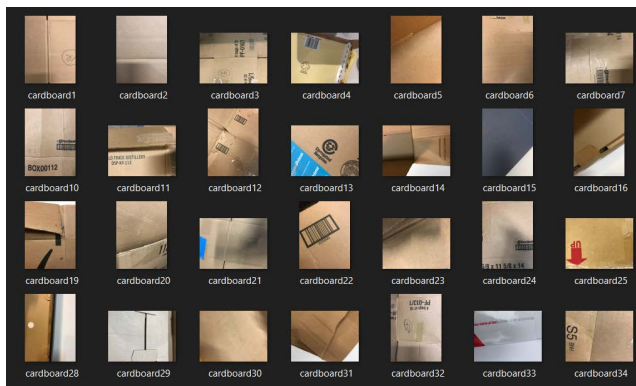


FIGURE 6. Sample images from waste class – cardboard.

B. IMPLEMENTATION DETAILS

The experimental setup comprises a desktop machine with Intel Core-i7 10510U with eight cores, 16 GB RAM, installed with ubuntu 18.04 64-bit operating system, and Python 3.6 package as it is compatible with Tensorflow. We have employed the SCs on the hyperledger fabric blockchain platform, and the performance is evaluated using the hyperledger caliper benchmark tool. Layers are added with Keras’ help to existing ResNet architecture, with pre-trained weights on ImageNet [12]. The following prerequisites are installed and configured before running the hyperledger fabric development:

- **Git client:** open-source version management system focused on speed and performance for small-scale and large-scale projects;
- **Docker engine:** docker components and services are used to build and run containers, which is a fundamental client-server technology;
- **Docker compose:** a tool for setting up and executing multi-container Docker applications. Python: a high-level programming language that is interpreted, interactive, and object-oriented;

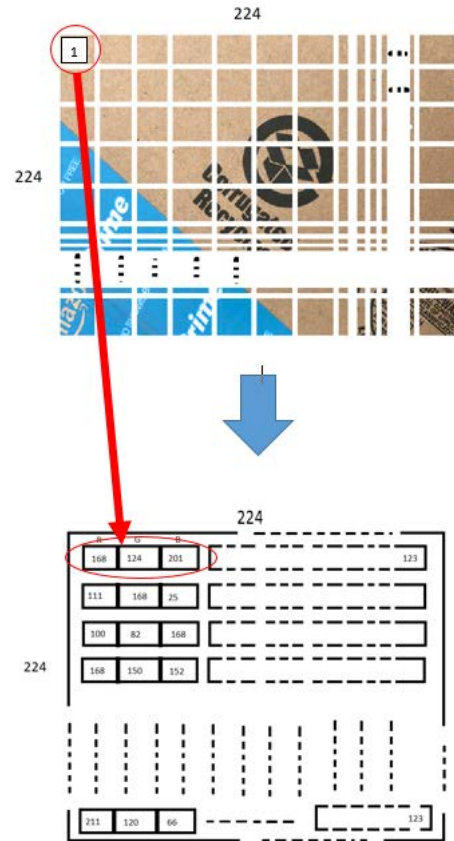


FIGURE 7. Conversion of a sample cardboard image into RGB input matrix.

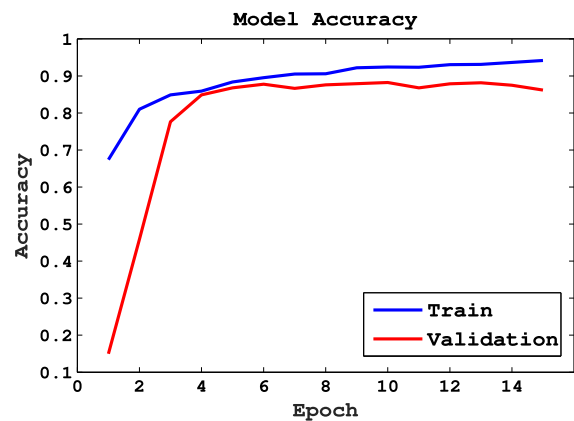


FIGURE 8. Performance: accuracy with respect to number of epochs.

- **npm:** an online repository of JavaScript modules, many of which are particularly drafted for node; and
- **Node:** a JavaScript-based framework for developing real-time applications in a short amount of time.

Data augmentation is performed to increase the dataset, which results in a better generalized model that prevents overfitting. Hyperparameter tuning is done for the learning rate, drop-out rate, optimizer, and the number of epochs. Adam is the finalized optimizer used with a learning rate equal to 0.0001 and a drop-out rate equal to 0.2. The model yields the best performance after 15 epochs. Beyond that, it starts

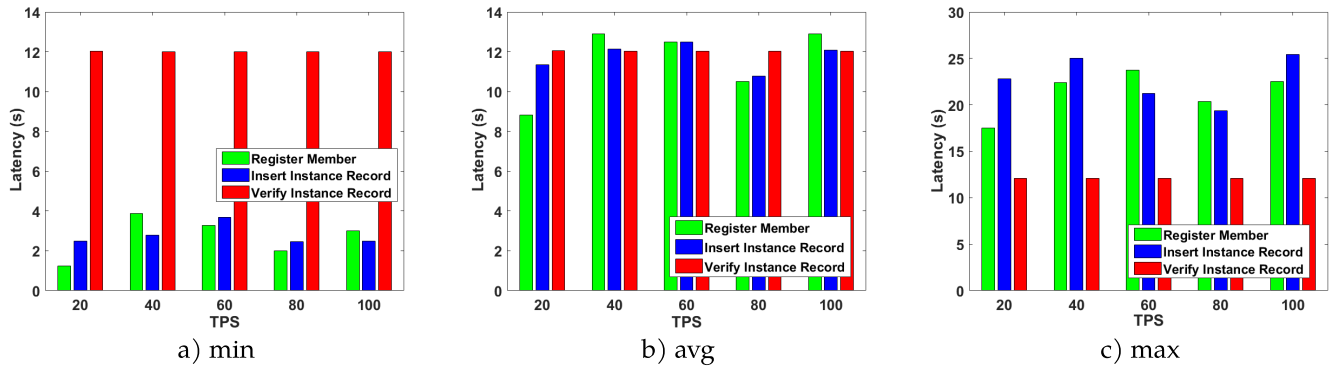


FIGURE 9. Initialization and verification operations latencies of smart contracts.

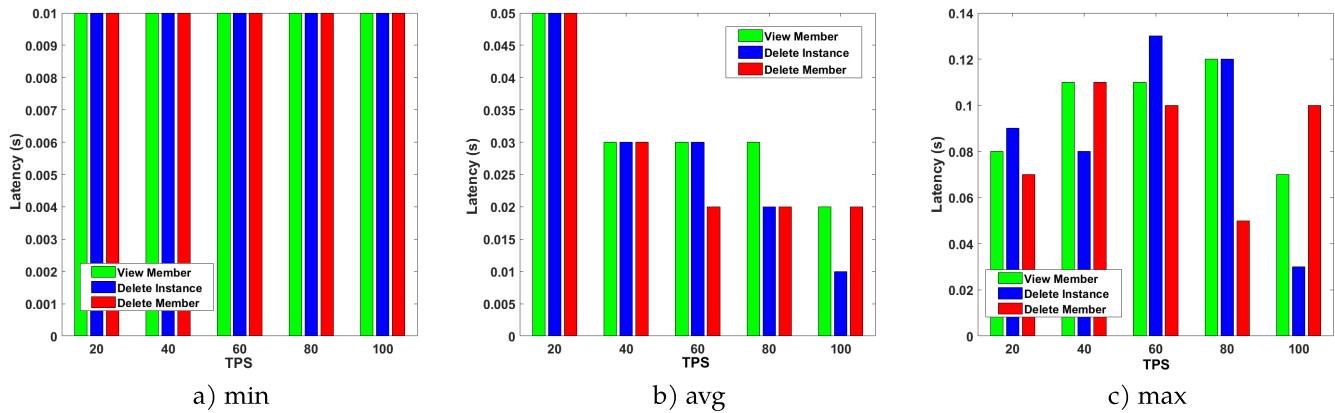


FIGURE 10. Reading and removing operations latencies of smart contracts.

TABLE 2. Data distribution of Test data.

SN	Class	# of images
1.	Bottle	201
2.	Cans	179
3.	Cardboard	177
4.	Cups	208
5.	Cutlery	196
6.	Jars	91*
7.	Laptop Accessories	195
8.	Organic Waste	195
9.	Paper	150
10.	Plastic Bag	206
11.	Stationery	204
12.	Utensils	203

* manually downloaded (source: Google)

over-fitting. The performance of the training model on the validation data is checked at the end of every epoch.

C. RESULTS

This section provides a quantitative evaluation of the proposed model’s accuracy and the average time taken to fetch recommendations. As mentioned above, the dataset has been collected from various sources like TrashNet, Kaggle, and ImageNet. This dataset was then classified into 12 classes according to the requirement of our project. There are about 11700 images in the dataset. The training is done on 15 epochs.

The model yields a training and validation accuracy of 94.17% and 86.18%, respectively, on the entire dataset. The accuracy graph is shown in Figure 8. The graph shown in Figure 8 demonstrates how well the predicted class performs against the actual class (ground truth). As the accuracy curve keeps increasing with every epoch, the model is learning well. The gap between the training and validation accuracy indicates the level of overfitting. As this gap is less in this graph, it implies that the model does not overfit.

The objective of the experiments is to check whether the object is being recognized correctly. This is done by using the system’s webcam and displaying the object in front of it, which then identifies this image. This helps us to check how well our model is performing in real-time. We tested our model in real-time by showing various objects such as plastic bottles, newspapers, magazines, plastic bags, cardboards, and papers. It was able to predict the classes correctly, and then it redirected us to various DIY reusing ideas.

In the next phase of experimentation, the Image Recognition system results are used further for obtaining reusing and recycling DIY idea recommendations. This process of fetching recommendations takes about 1 second on average. Also, there is a wide variety of output classes, due to which the recommendations are more precise and object-based.

We tested our model on the test data (Table 2) taken from open images dataset v6.¹ Further, we demonstrated the

¹<https://storage.googleapis.com/openimages/web/index.html>

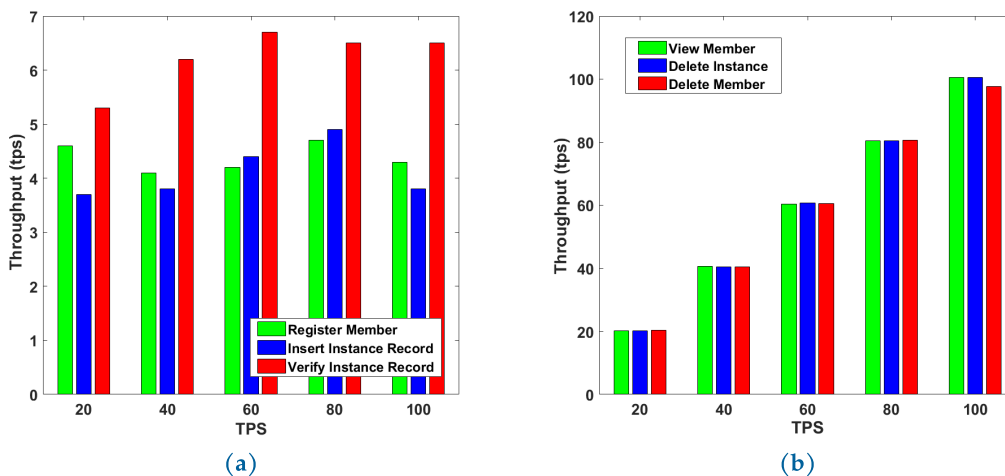


FIGURE 11. Smart contracts' throughput (a) initialization and verification operations, (b) reading and removing operations.

TABLE 3. Confusion matrix.

Bottle	134	18	3	9	0	11	4	3	0	1	12	6
Cans	13	113	3	15	0	9	16	0	0	2	5	3
Cardboard	0	1	104	1	0	0	20	0	19	3	29	0
Cups	1	7	2	138	3	3	9	8	0	1	6	30
Cutlery	1	7	15	5	81	2	9	1	1	2	60	12
Jars	2	4	2	1	0	73	0	1	0	0	4	4
Laptop Acc.	3	1	5	1	0	0	164	1	1	0	17	2
Organic Waste	5	0	0	1	1	0	6	168	0	4	10	0
Paper	0	0	9	3	0	0	3	0	105	0	30	0
Plastic Bag	4	1	3	2	1	5	3	2	5	175	4	1
Stationery	5	6	7	0	3	0	15	1	0	0	167	0
Utensils	2	16	2	23	4	1	22	40	1	2	14	76

confusion matrix and our results in Tables 3 and 4, respectively. The performance of the IRM can be improved by training the model with more images for object classes with less Precision, Recall, and F1-score.

D. PERFORMANCE OF SMART CONTRACTS

The caliper benchmark captured the performance of deployed smart contracts for 20 to 100 transactions per second. The performance is measured based on transaction latencies and throughputs. Latency performance metrics are grouped into three sets, i.e., minimum, average, and maximum. The latencies of initialization contacts (i.e., member registration and insert recommended idea) and instance verification contract are shown in Figure 9. As noted from the graph, the optimal latencies take the least 1.25s, an average of 8.81s, and the highest at 12.07s. Then the reading and removing operations latencies of smart contracts are disclosed in Figure 10. The optimal latencies of these contracts are consistent, ranging between 0.01-0.13. Reading and removing operations are more reliable than other operations because it does not involve other complex operations. Besides, Figure 11 illustrates the throughputs of the deployed contracts. The throughput performance metrics are grouped into two sets: (a) Initialization and Verification operations latencies of smart contracts, and (b) Reading and removing operations latencies of smart contracts. As seen from Figure 11(a), the throughput of initialization and verification

TABLE 4. Quantitative results of IRM.

	Precision	Recall	F1-score	Support
Bottle	0.79	0.67	0.72	201
Cans	0.65	0.63	0.64	179
Cardboard	0.67	0.59	0.63	177
Cups	0.69	0.66	0.68	208
Cutlery	0.87	0.41	0.56	196
Jars	0.7	0.8	0.75	91
Laptop Acc.	0.61	0.84	0.7	195
Organic Waste	0.75	0.86	0.8	195
Paper	0.8	0.7	0.74	150
Plastic Bag	0.92	0.85	0.88	206
Stationery	0.47	0.82	0.59	204
Utensils	0.57	0.37	0.45	203
Accuracy			0.68	2205
Macro Avg	0.71	0.68	0.68	2205
Weighted Avg	0.71	0.68	0.68	2205

contracts varies between 3.7-6.7tps. However, the maximum throughputs are achieved at 20.2-100.5tps, shown in Figure 11(b), because the throughputs of reading and removing operations are more efficient than other operations since they only execute lightweight query operations to the databases.

V. CONCLUSION

Resources are scarce. Reusing and recycling not only save our limited resources, but they also save the environment from pollution. The proposed work does exactly that by giving

recommendations to the users. In this paper, we proposed a deep neural network to identify the waste object and a blockchain-oriented recommendation system to suggest the user DIY reusing and recycling ideas. We have employed ResNet50 as a base architecture, and its performance has been enhanced by implementing several different techniques. On a dataset of size approximately 11700 images that were collected from various sources (12 classes), we have obtained results from the 'Image Recognition Model' that could potentially be used to obtain reusing ideas in the recommendation system. In our study, blockchain technology has enabled transaction verifiability and assisted in better and more realistic decision-making.

Increasing classes and dataset size is our immediate plan to avoid similarity issues among objects. To avoid possible misclassification, we also intend to do more analytical work on the waste objects the user wants to discard. It would include analyzing various markings, texts, or symbols available in objects. Additionally, trusted stored recommendation records will be utilized to minimize decision-making efforts and maximize trustworthy accuracy. This work can be extended to a global scale to reuse the items by other people. However, to make it on a global scale, there is a need for awareness among the public and connecting different technologies to make it a circular system. In such a system, there should be the opportunity to connect municipal corporations, recycling facilities, and second-hand stores to make reusable waste readily available to people who may need it. On another note, as future work, we aim at integrating our application with social media platforms where users can share their "Green Profiles". The more the users share their reused objects, the "Greener" their profile will be. Our reward-based system attracts more users to reuse more waste objects/products, which will positively impacting the universe.

REFERENCES

- [1] *Global Waste to Grow by 70 Percent by 2050 Unless Urgent Action is Taken: World Bank Report*, World Bank, Washington, DC, USA, 2018.
- [2] C. Srinilta and S. Kanharattanachai, "Municipal solid waste segregation with CNN," in *Proc. 5th Int. Conf. Eng., Appl. Sci. Technol. (ICEAST)*, Jul. 2019, pp. 1–4.
- [3] R. A. Aral, S. R. Keskin, M. Kaya, and M. Haciomeroglu, "Classification of TrashNet dataset based on deep learning models," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2058–2062.
- [4] S. L. Rabano, M. K. Cabatuan, E. Sybingco, E. P. Dadios, and E. J. Calilung, "Common garbage classification using MobileNet," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–4.
- [5] K. Sreelakshmi, S. Akarsh, R. Vinayakumar, and K. P. Soman, "Capsule neural networks and visualization for segregation of plastic and non-plastic wastes," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 631–636.
- [6] C. Lin, D. He, X. Huang, and K.-K.-R. Choo, "OBFP: Optimized blockchain-based fair payment for outsourcing computations in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3241–3253, 2021.
- [7] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based cloud storage system with CP-ABE-based access control and revocation process," *J. Supercomput.*, vol. 78, no. 6, pp. 7700–7728, Apr. 2022.
- [8] L.-Y. Yeh, C.-Y. Shen, W.-C. Huang, W.-H. Hsu, and H.-C. Wu, "GDPR-aware revocable P2P file-sharing system over consortium blockchain," *IEEE Syst. J.*, early access, Jan. 18, 2022, doi: 10.1109/JSYST.2021.3139319.
- [9] J. Moosavi, L. M. Naeni, A. M. Fathollahi-Fard, and U. Fiore, "Blockchain in supply chain management: A review, bibliometric, and network analysis," *Environ. Sci. Pollut. Res.*, pp. 1–15, Feb. 2021.
- [10] U. Chelladurai and S. Pandian, "A novel blockchain based electronic health record automation system for healthcare," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 1, pp. 693–703, Jan. 2022.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [13] G. Thung and M. Yang. (2016). *Dataset of Images of Trash; Torch-Based CNN for Garbage Image Classification*. Accessed: May 23, 2020. [Online]. Available: <https://github.com/garythung/trashnet>
- [14] S. Sekar. (2019). *Waste Classification Data*. Accessed: Mar. 20, 2021. [Online]. Available: <https://www.kaggle.com/techsash/waste-classification-data>
- [15] M. A. Rahmad, N. S. Midi, S. A. Zaini, S. H. Yusoff, and S. Y. Mohamad, "Material classification of recyclable waste using the weight and size of waste," in *Proc. 7th Int. Conf. Comput. Commun. Eng. (ICCCCE)*, Sep. 2018, pp. 44–49.
- [16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [17] P. Kumar and A. Sharma, "Segmentation-free writer identification based on convolutional neural network," *Comput. Electr. Eng.*, vol. 85, Jul. 2020, Art. no. 106707.
- [18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
- [19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [20] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google Res. Blog*, Jun. 2015.
- [21] A. Louati, L. H. Son, and H. Chabchoub, "Smart routing for municipal solid waste collection: A heuristic approach," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 1865–1884, May 2019.
- [22] Y.-C. Chen, H.-M. Sun, R.-S. Chen, J.-H. Yeh, X. Fan, and I.-H. Chou, "Intelligent radioactive waste process cloud-based system for nuclear power plant decommissioning," *J. Ambient Intell. Humanized Comput.*, pp. 1–11, Jun. 2018.
- [23] M. S. Moslehi, H. Sahebi, and A. Teymouri, "A multi-objective stochastic model for a reverse logistics supply chain design with environmental considerations," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 7, pp. 1–24, Jul. 2020.
- [24] A. S. L. França, J. Amato Neto, R. F. Gonçalves, and C. M. V. B. Almeida, "Proposing the use of blockchain to improve the solid waste management in small municipalities," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118529.
- [25] G. Falazi, U. Breitenbücher, F. Daniel, A. Lamparelli, F. Leymann, and V. Yussupov, "Smart contract invocation protocol (SCIP): A protocol for the uniform integration of heterogeneous blockchain smart contracts," in *Proc. Int. Conf. Adv. Inf. Syst. Eng. Cham, Switzerland: Springer*, 2020, pp. 134–149.
- [26] A. Dua, A. Dutta, N. Zaman, and N. Kumar, "Blockchain-based e-waste management in 5G smart communities," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 195–200.
- [27] R. P. Mahapatra, S. S. Srikant, S. Pandey, V. Chouhan, and R. B. Rao, "Trusted and transparent blockchain-enabled e-waste optimization to recover precious metals with microwave heating," *J. Sci. Ind. Res.*, vol. 80, no. 7, pp. 623–629, 2021.
- [28] N. Gupta and P. Bedi, "E-waste management using blockchain based smart contracts," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 915–921.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.

- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.
- [31] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.
- [32] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*, Sebastopol, CA, USA: O'Reilly, 2008.
- [33] M. Grinberg, *Flask Web Development: Developing Web Applications With Python*. Sebastopol, CA, USA: O'Reilly, 2018.



SHUBHAM RAJRAH received the B.Tech. degree from the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, India. His research interests include machine learning and deep learning.



FAYADH ALENEZI received the B.Sc. degree (Hons.) in electrical engineering electronics and communications track from Jouf University, Sakaka, Saudi Arabia, in 2012, the M.S. degree in electrical engineering from Southern Illinois University Carbondale, IL, USA, in 2015, and the Ph.D. degree in electrical engineering from The University of Toledo, OH, USA, in 2019. He is currently an Assistant Professor with the Department of Electrical Engineering, Jouf University.

He has authored many journals and conference papers. His research interests include artificial intelligence, image processing, signal processing, image enhancement, machine learning, neural networks, and facial recognition.



RAJKUMAR SAINI received the Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee, India. He is currently a Post-doctoral Researcher with the EISLAB Machine Learning, Luleå University of Technology, Sweden. His research interests include computer vision, machine learning, pattern recognition, human-computer interface, brain signal analysis, and digital image processing.



KC SANTOSH received the Ph.D. degree in computer science-artificial intelligence from the INRIA Nancy Grand Est Research Center, France. Prior to that, he worked as a Research Fellow with the U.S. National Library of Medicine (NLM), National Institutes of Health (NIH). He worked as a Postdoctoral Research Scientist with the LORIA Research Center, Université de Lorraine in direct collaboration with industrial partner ITESOFT, France. He also served as a Research Scientist with

the INRIA Nancy Grand Est Research Center. He is currently the Chair of the Department of Computer Science (CS), University of South Dakota (USD). His research projects, primarily in applied AI, are funded (of more than \$2m) by multiple agencies, such as SDCRGP, Department of Education, National Science Foundation, and Asian Office of Aerospace Research and Development. He completed leadership and training programs for the Deans/the Chairs (organized by the Councils of Colleges of Arts & Sciences, USA, in 2021, and President's Executive Leadership Institute (PELI), USD, in 2021). He is highly motivated/interested in academic leadership. To name a few, he was a proud recipient of the Cutler Award for Teaching and Research Excellence (USD 2021), the President's Research Excellence Award (USD 2019), and the Ignite Award from the U.S. Department of Health & Human Services (HHS 2014).

...



SACHI PANDEY received the Ph.D. degree from the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, India. She is currently working as an Associate Professor with the SRM Institute of Science & Technology, and having ten and half years of experience. She has completed Postdoctoral Fellowship with the Department of Computer Science & Engineering, Indian Institute of Technology, Roorkee, India. So far, she is supervising a Ph.D. scholars and has guided many graduate and postgraduate students. She has published and granted patents in her domain. She has research publications in many reputed journals and conferences. She has served as the program committee member for many international conferences. Her research interests include wireless sensor networks, machine learning, and blockchain.

She has published and granted patents in her domain. She has research publications in many reputed journals and conferences. She has served as the program committee member for many international conferences. Her research interests include wireless sensor networks, machine learning, and blockchain.



VIKAS CHOUHAN received the M.Tech. and Ph.D. degrees in computer science and engineering from the Indian Institute of Technology, Roorkee, India. He has several research publications in reputed journals and conferences. He served as the program committee member and a reviewer for several international conferences. He has been selected as one of the recipient of the Visvesvaraya Research Fellowship for the Ph.D. program. His research interests include cloud computing, cloud

security & privacy, information security & reliability, and blockchain technology.



DEVANSHI VERMA received the B.Tech. degree from the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, India. Her research interests include machine learning and data science.