Integrating AI and Computer Vision for Effective Foreign Object Detection on Airport Runways

Dr. CH. Sarada Devi, M. Tech, Ph. D[1], Ms. D. Rejitha, M. E, (Ph. D)[2], Abinaya. S[3]

[1]Head of the Department, Department of Computer Science and Engineering, Meenakshi College of Engineering K. K. Nagar Chennai, India

[2]Assistant Professor Department of Computer Science and Engineering, Meenakshi College of Engineering K. K. Nagar Chennai, India

[3]Department of Computer Science and Engineering, Meenakshi College of Engineering K. K. Nagar Chennai, India

---***---

Abstract-*The FOD-R Dataset consists of images representing common types of Foreign Object Debris (FOD) typically found on airport runways and taxiways. The dataset is primarily annotated with bounding boxes to support object detection tasks. An enhanced version, known as the FOD-xR Dataset, incorporates an additional feature in the form of color-coded labels that indicate the difficulty of debris removal. Green labels represent lightweight objects that are easier to remove, whereas red labels indicate heavier objects that require greater effort. This centralized labeling approach improves coordination among maintenance teams by enabling efficient planning and management of FOD removal operations. As a result, organizations can reduce operational delays, minimize aircraft damage, and improve overall airport safety. The FOD-R and FOD-xR datasets serve as valuable resources for developing intelligent FOD detection and removal systems.*

Index Terms-Foreign Object Debris (FOD), centralized management, dataset, bounding box annotation, object detection, color coding.

I. INTRODUCTION

Foreign Object Debris (FOD) poses a significant threat to aviation safety, as it can lead to aircraft damage, operational disruptions, injuries, and even fatalities. The financial losses associated with FOD incidents are also substantial due to aircraft repairs, flight delays, and maintenance costs. To address these challenges, researchers have increasingly explored the use of Machine Learning (ML) and Computer Vision (CV) techniques for the automatic detection and classification of FOD. The development of accurate and reliable ML-based detection systems requires high-quality datasets containing diverse FOD images captured under realistic operating conditions. Such datasets play a vital role in improving the accuracy, robustness, and efficiency of intelligent FOD detection models.

To facilitate research in this domain, the FOD-A dataset was introduced as a comprehensive benchmark for training and evaluating machine learning and computer vision algorithms. The object categories included in the dataset were selected based on recommendations from the Federal Aviation Administration (FAA) and findings from previous research, ensuring that the dataset represents a broad range of debris commonly found on airport runways and taxiways. Images were collected under varying environmental conditions, including different lighting levels and rainfall scenarios, to simulate real-world airport environments and improve the generalization capability of detection models.

The FOD-A dataset was also designed with scalability in mind, allowing new images and object categories to be incorporated while preserving annotation consistency. Multiple versions of the dataset are maintained to ensure that it remains comprehensive and up to date. Researchers can evaluate detection algorithms using a standardized version of the dataset and subsequently utilize newer releases to develop enhanced models capable of recognizing additional FOD categories. Figure 1 illustrates a sample image from the FOD-A dataset with annotated bounding boxes.

Although several publicly available image datasets contain common object categories such as vehicles, furniture, and household items, they do not adequately represent the specialized debris encountered in airport environments. Important FOD categories, including aircraft components, maintenance tools, luggage items, and metallic fragments, are largely absent from these general-purpose datasets. Therefore, only limited comparisons with conventional object detection datasets are presented, as they cannot effectively support airport-specific FOD detection tasks.

The remainder of this paper is organized as follows. Section II reviews the existing literature on FOD detection and related machine learning approaches. Section III describes the construction of the FOD-A dataset, including data collection procedures, annotation methodology, dataset statistics, and expansion strategies. Section IV presents the software packages, machine learning techniques, and evaluation methods used in the analysis. Finally, Section V summarizes the key findings, discusses the limitations of the current work, and outlines future research directions. The major contributions of this study include the development of a publicly accessible FOD-A dataset, the implementation of a scalable framework for image dataset generation, and the design and evaluation of advanced algorithms for automated FOD detection As a result, FOD-A is more suitable for FOD detection tasks as it provides a more accurate representation of airfield terrain and a greater variety of object categories with more descriptive object orders.

Fig. 1. Examples of FOD-A dataset images annotated with bounding boxes.

II. RELATED WORK

Several studies and official publications have addressed the detection, classification, and management of Foreign Object Debris (FOD) in airport environments. Among these, the guidelines published by the Federal Aviation Administration (FAA) serve as the primary reference for identifying and categorizing FOD objects [1], [7], [10], [12], [17]. Since the FAA provides standardized definitions and recommendations for FOD management, the object classes included in the proposed dataset were selected based on these guidelines. The methodology adopted for choosing the object categories is discussed in detail in Section III-A.

A. Existing FOD Datasets

A number of publicly available datasets have been developed for FOD detection; however, many of them are limited in terms of object diversity. One widely used dataset primarily focuses on recognizing materials such as metal, plastic, and concrete [18]. Although these categories are useful for material classification, they do not represent the complete range of Foreign Object Debris identified by the FAA. Important FOD categories, including maintenance tools, brightly colored debris, natural objects, runway fragments, pavement chips, and other airport-specific materials, are not adequately represented in these datasets [1], [7].

To address these limitations, the FOD-A dataset was developed as a more comprehensive benchmark for airport FOD detection. Unlike existing datasets, FOD-A contains 31 distinct object categories, providing broader coverage of debris commonly encountered on airport runways and taxiways. The complete list of object categories is illustrated in Figure 3.

Another key difference between the existing material recognition dataset and FOD-A lies in the image characteristics and annotation strategy. The material recognition dataset mainly consists of close-up images and includes only three object categories, making it unsuitable for detecting the wide variety of debris encountered in real airport environments. In contrast, the FOD-A dataset contains wide-angle images annotated with bounding boxes and includes additional environmental information, such as lighting and weather conditions, to better simulate real-world operational scenarios.

Furthermore, the existing dataset contains approximately 500 annotated object samples distributed across three material categories. In comparison, FOD-A provides more than 1,376 annotated object instances spanning 31 different object classes. The increased number of samples, greater category diversity, and richer annotation make FOD-A a more effective resource for training and evaluating advanced machine learning and computer vision models for automated FOD detection.

II. FOD Detection Method Incorporated

In this study, data visualization techniques are employed to present complex information in a clear and intuitive manner, enabling better interpretation of the dataset. The visualization primarily illustrates the distribution of images across the two predefined classes: **high-risk** and **low-risk** FOD categories. Such graphical representations help researchers understand the balance of the dataset and identify class distributions before model training.

To generate these visualizations, the program first retrieves the list of category folders from the dataset directory using the command os.listdir (data_path). The retrieved folder names are stored in the variable categories, which contains the two classes: **high risk** and **low risk**. Numerical labels are then assigned to each category by generating a sequence of integers corresponding to the number of classes. Finally, a dictionary is created to map each category to its respective numerical label, resulting in the assignments **high risk → 0** and **low risk → 1**. This labeling scheme facilitates efficient preprocessing and training of machine learning models by converting textual class names into numerical representations.

III. DATASET

The Foreign Object Debris (FOD) dataset consists of a diverse collection of objects that commonly appear on airport runways and taxiways. These objects include aircraft fasteners such as nuts, bolts, washers, and cords used in aircraft assembly and engine maintenance. The dataset also contains aircraft components, including fuel caps, landing gear fragments, and other structural debris that may detach during aircraft operations.

In addition to aircraft-related objects, the dataset incorporates maintenance tools and equipment frequently used by ground personnel, along with feeding materials and other flight-line accessories. Furthermore, it includes fragments of runway infrastructure such as concrete pieces, asphalt particles, rubber expansion joints, flexible rubber fittings, joint seal materials, fibers, and filaments. Environmental debris generated from construction activities, including dust and ash, as well as weather-related contaminants such as snow, ice, and other natural pollutants, are also represented. This wide variety of object categories ensures that the dataset closely reflects real-world airport conditions and supports the development of robust machine learning models for accurate FOD detection and classification.

Fig. 2. The provided images depict FOD objects under different light-level categories in FOD-A. Specifically, the left image represents an example of a bright condition, the middle image depicts a dim condition, and the right image shows a dark condition.

III. DATASET DESCRIPTION

The FOD-A dataset categorizes environmental conditions into **dry** and **wet** weather classes, which collectively represent the majority of weather scenarios encountered in airport operations. In addition to the primary bounding-box annotations used for object detection, the dataset also includes supplementary annotations describing weather conditions and lighting levels. These additional labels provide contextual information that can support the development of more robust detection models capable of operating under varying environmental conditions.

The dataset consists of **1,376 images**, including **690 images containing high-risk FOD objects** and **686 images containing low-risk FOD objects**. Analysis of the dataset indicates that most debris objects occupy relatively small regions within the images, making accurate detection a challenging task. Figure 1 illustrates sample images and highlights the predominance of small-sized objects.

To create the dataset, images of commonly occurring FOD were collected using both handheld cameras and **Unmanned Aerial Vehicles (UAVs)**. UAV-based image acquisition enabled the collection of data from different heights and distances, while handheld cameras captured closer views with greater variations in viewing angles. Data collection was initially performed in video (MP4) format to facilitate large-scale acquisition. However, not every video frame contained relevant debris objects, resulting in the presence of unnecessary or empty frames that could negatively affect dataset quality.

To overcome this limitation, a dedicated command-line utility was developed to streamline the data preparation process. The tool automatically trims videos to relevant intervals, extracts individual frames as image files, and removes unnecessary data. Additionally, the utility standardizes image formats and resolutions, thereby improving dataset consistency and simplifying future dataset expansion. This automated workflow significantly enhances the efficiency of image collection and preprocessing.

IV. INCORPORATED PACKAGES

A. TensorFlow

TensorFlow is a widely adopted open-source framework used for implementing machine learning and deep learning applications across various domains, including computer vision, speech recognition, natural language processing, and sentiment analysis. In the proposed system, TensorFlow serves as the primary backend framework for developing and training the Convolutional Neural Network (CNN) model. It is also utilized for image preprocessing operations such as data reshaping and tensor manipulation. Beyond computer vision applications, TensorFlow has been extensively employed in research areas including information retrieval, geographic data analysis, text summarization, drug discovery, and defect detection.

B. Keras

The proposed model utilizes **Keras**, a high-level deep learning library built on top of TensorFlow, to simplify the design, implementation, and deployment of neural network architectures. Keras provides an intuitive interface for constructing CNN models and supports scalable execution across different computing platforms. In this work, Keras is used to define the CNN layers, configure model parameters, compile the network, and perform data preprocessing operations such as converting categorical labels into binary class matrices. Its modular design facilitates rapid experimentation and efficient model development.

C. OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source software library extensively used for image processing, computer vision, and machine learning applications. It offers a wide range of functionalities for image enhancement, feature extraction, object detection, and image analysis. In FOD detection applications, OpenCV can be employed for tasks such as edge detection, image enhancement using histogram analysis, and object feature extraction. Within the proposed framework, OpenCV is primarily used for image preprocessing operations, including resizing images, converting color formats, and preparing image data for input into the deep learning model. These preprocessing steps contribute to improved model performance and computational efficiency

V. THE PROPOSED METHOD

The proposed framework consists of two major components: a cascade classifier for preliminary object localization and a pre-trained Convolutional Neural Network (CNN) for object classification. The CNN architecture comprises two two-dimensional convolutional layers followed by fully connected dense layers that perform the final classification of Foreign Object Debris (FOD) into predefined risk categories.

Algorithm 1: Foreign Object Detection

Input: A labeled image dataset containing two classes: **High Risk** and **Low Risk**.
Output: A classified runway image indicating the detected category of foreign object debris.

Image Pre-processing

For every image in the dataset, the following operations are performed:

1. Display and categorize the images into **high-risk** and **low-risk** classes.
2. Convert the original RGB image into a grayscale image.
3. Resize the grayscale image to a fixed resolution of **100 × 100 pixels**.
4. Normalize the pixel values and reshape the image into a four-dimensional tensor suitable for CNN input.

CNN Model Construction

The classification model is built using the following sequence of layers:

1. A two-dimensional convolution layer containing **200 filters**.
2. A second convolution layer with **100 filters**.
3. A **Flatten** layer to transform extracted feature maps into a one-dimensional feature vector.
4. A fully connected (**Dense**) layer consisting of **64 neurons**.
5. A final output layer with **two neurons**, corresponding to the **High Risk** and **Low Risk** classes.

After constructing the network architecture, the dataset is divided into training and testing subsets, and the CNN model is trained to classify runway images based on the detected foreign objects.

A. Data Processing

Data preprocessing is a crucial stage in machine learning that transforms raw data into a structured format suitable for analysis and model training. The primary objective of preprocessing is to improve data quality, remove inconsistencies, and organize the information into a format that accurately represents the relationships among data elements. Proper preprocessing enhances the efficiency of feature extraction and improves the overall performance of machine learning models.

In the proposed system, the input data consists of images and video frames containing Foreign Object Debris. The preprocessing operations are implemented using the **NumPy** and **OpenCV** libraries. These tools provide efficient methods for extracting useful information from visual data through operations such as image segmentation, resizing, filtering, normalization, and thresholding. The processed images serve as structured input for the CNN model, enabling more accurate feature learning and object classification.

1) Data Visualization

Data visualization is employed to represent the distribution of images within the dataset in an intuitive and easily interpretable manner. Visual representations assist in understanding class balance and identifying patterns before training the classification model.

Initially, the program retrieves the list of category folders from the dataset directory using the os.listdir() function. Each category is then assigned a numerical label based on its index position. A mapping dictionary is subsequently generated by pairing the category names with their corresponding labels using the zip() function. This process produces a label dictionary that associates each class name with its numerical representation, facilitating efficient dataset management during model training.

2) Conversion of RGB Images to Grayscale

Color images are converted into grayscale before feature extraction to reduce computational complexity while preserving the essential structural information required for object recognition. Since most image recognition algorithms rely primarily on texture, shape, and intensity rather than color information, grayscale images are generally sufficient for effective classification.

Removing color channels decreases the dimensionality of the input data, reduces memory requirements, and shortens the training time of the CNN model. Furthermore, grayscale conversion minimizes redundant information and enables more efficient extraction of discriminative image features, thereby improving the overall performance of the object detection system.

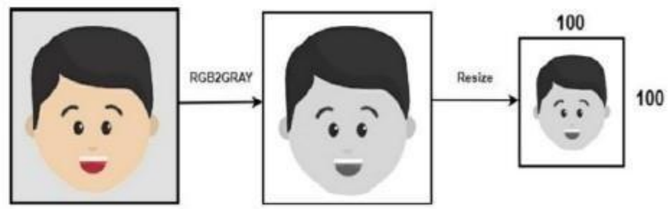

Fig. 3. The process of converting an RGB image to a grayscale image of 100x100 size.

3) Grayscale Conversion and Image Resizing

The conversion of color images into grayscale is performed using the cv2.cvtColor() function provided by the OpenCV library. This function accepts the original image and a conversion flag as input parameters. In the proposed approach, the cv2.COLOR_BGR2GRAY conversion code is applied to transform RGB images into grayscale representations. Since grayscale images contain only intensity information, they reduce computational complexity while preserving the essential features required for object recognition.

Deep Convolutional Neural Networks (CNNs) require all input images to have identical dimensions. Therefore, each grayscale image is resized to a fixed resolution of **100 × 100 pixels** using the cv2.resize() function. Standardizing image dimensions ensures compatibility with the CNN architecture and enables efficient batch processing during model training.

4) Image Scaling

Image scaling and normalization are essential preprocessing operations that prepare images for deep learning. Each image is represented as a three-dimensional tensor, where the pixel values correspond to intensity information. Although images may differ in content, all inputs supplied to the CNN must possess the same dimensional structure.

To satisfy this requirement, the pixel intensity values are normalized to the range **0–1**, improving numerical stability and accelerating the learning process. The normalized images are subsequently reshaped into four-dimensional tensors using the np.reshape() function. The resulting tensor has the format:

(Number of Images, Image Height, Image Width, 1)
where the final dimension represents the single grayscale channel.

Since the proposed classification problem involves two output classes (**High Risk** and **Low Risk**), the target labels are converted into categorical format before training. This representation enables the CNN to perform efficient multi-class prediction using an appropriate classification loss function.

B. Training the Model

1) CNN Model Construction

Convolutional Neural Networks (CNNs) have become one of the most widely adopted deep learning models for image classification and object recognition because of their ability to automatically learn discriminative visual features. In this work, Sequential **CNN architecture** is implemented to classify Foreign Object Debris images according to their risk level.

The model construction begins by initializing a Sequential CNN framework, followed by the addition of the first convolutional layer responsible for extracting low-level image features such as edges and textures. A pooling layer is then incorporated to reduce the spatial dimensions of the extracted feature maps while preserving the most significant information.

Subsequently, two additional convolutional layers are introduced to capture higher-level semantic features from the images. The resulting feature maps are flattened into a one-dimensional feature vector, which serves as the input to the fully connected (dense) layers. These dense layers perform the final classification by learning complex relationships among the extracted features. Finally, the output layer produces probability scores for the two target classes **High Risk** and **Low Risk** allowing the model to accurately categorize each input image.

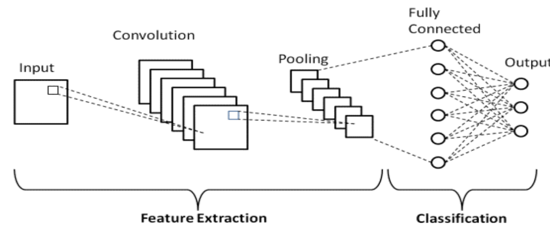

Fig. 4. CNN Architecture Model

The output shape can be calculated using the formula (input size - pooling window size + 1)/Leaps, where the default value of Leaps is (1, 1) [7].

1) Dividing the data into separate sets and training the CNN model: Partitioning the data into a training and testing set is a crucial step in developing reliable predictive models. To achieve this, a train-test split is performed where the dataset is split into two partitions, one for training the model and the other for evaluating its performance. In this case, the dataset is split into two parts, where 10% of the data is set aside for testing while the remaining 90% is used for training. The confirmation loss is monitored during the training process using Model Checkpoint. The Sequential model is then trained and tested on these sets, with 20% of the training data being used for validation. To avoid overfitting, the model undergoes 20 epochs during the training process, this value is chosen as a compromise between achieving high accuracy and avoiding the potential risk of overfitting. The proposed model is visually represented in Figure 5.

RESULTS AND ANALYSIS

The dataset was utilized to train, validate, and test the model, which achieved an accuracy rate of 95.77%. (as shown in Figure 7). One of the primary reasons for the model's high accuracy down-sampling is the primary reason for achieving

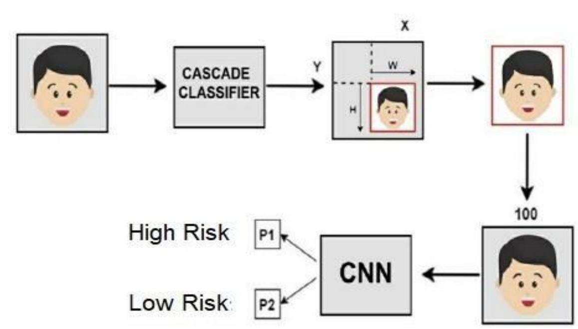

Fig. 5. Overview of the Model

high accuracy in detecting foreign objects in images. This technique reduces the dimensional space of the input image and grants translation in variance to the model's internal representation, resulting in a reduced number of parameters that the model needs to learn.

In this specific case, an ideal number of 64 neurons is used, which is not too excessive, as too many neurons and filters can negatively impact the model's performance. Additionally, optimized stride values and pool size are employed to filter out the main components of the image, allowing for precise detection of the foreign object without causing over-fitting.

Figure 6 provides evidence that achieving high accuracy in object detection significantly reduces the cost of errors. This figure may show a comparison between the cost of errors for models with different levels of accuracy or highlight the economic impact of misclassifications in this domain. By demonstrating the importance of high accuracy, the figure emphasizes the need for techniques like down-sampling, optimized stride values, and pool size, as mentioned in the paragraph, that can improve the model's performance and accuracy in detecting foreign objects. Ultimately, achieving high accuracy is vital for minimizing errors' cost in this context, where accuracy is crucial for ensuring safety and security.

The FOD system incorporates weather and light categorization inputs, and experiments were performed to evaluate their efficiency using wet and dry weather inputs. The model with two inputs was effective in distinguishing between wet and dry backgrounds in FOD-A images, and its precision rapidly improved on confirmation data. However, some unusual predictions were observed in the testing data, and it could be challenging to combine categorization and bounding box detection for advanced algorithms. Nevertheless, the weather and light inputs may have practical applications in the future.

CONCLUSION

The paper starts by providing a clear explanation of the reason behind the work, and the learning and performance objectives of the model are presented in an easy-to-understand manner. The model has achieved high accuracy using basic machine learning tools and simplified approaches. The system has various applications, including preventing accidents and rapidly detecting and eliminating foreign objects.

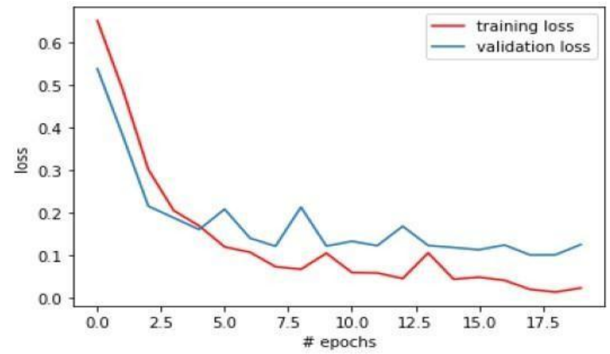

Fig. 6. This refers to a graph that shows the relationship between the number of training epochs and the loss observed in the dataset during training.

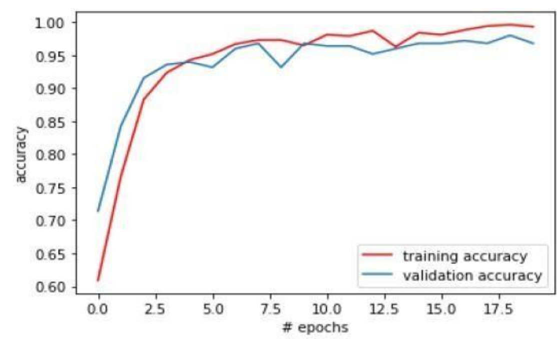

Fig. 7. This refers to a graph or chart that shows the relationship between the number of epochs used during the training process and the accuracy attained on the training dataset during the training process.

FUTURE WORK

In the future, this model could potentially be integrated with robots, which could prioritize the time taken to clean materials based on the level of threat posed by foreign object debris.

Additionally, this technology could be useful in space missions to prevent foreign objects from obstructing the path of rockets. However, this would be a challenging task, as it

TABLE I ACHIEVED SCORING INDICATORS

Dataset	Confusion Matrix		
	Precision	**Detection Rate**	**Harmonic mean**
High Risk Objects	1.00	0.99	0.99

Low Risk Objects	0.99	1.00	0.99

would require the integration of dedicated servers to support the system.

REFERENCES

[1] Munyer, T., Huang, P. C., Huang, C., & Zhong, X. (2021, October 6). FOD-A: A Dataset for Foreign Object Debris in Airports.

[2] S. Srinivasan, R. Rujula Singh, R. R. Biradar and S. Revathi, "COVID-19 Monitoring System using Social Distancing and Face Mask Detec-tion on Surveillance video datasets," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021.

[3] N. Saurabh et al., "Efficient Face Mask Detection For Different Surveil-lance Applications Using Python",

[4] A. Das, M. Wasif Ansari and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV".

[5] Dey, S.K., Howlader, A., Deb, C. (2021). MobileNet Mask: A Multi-phase Face Mask Detection Model to Prevent Person-To-Person Trans-mission of SARS-CoV-2.

[6] H. Adusumalli, D. Kalyani, R. K. Sri, M. Pratapteja and P. V. R. D. P.

Rao, "Face Mask Detection Using OpenCV",

[7] N. Yonemoto, S. Futatsumori, A. Kohmura, K. Morioka and N. Miyazaki, "Multi-static Millimeter Wave Radar Connected by Radio over Fiber for Foreign Object Detection on Runways".

[8] Annie Silviya S H, Pathmanaban J, Kingsley S, Sriman B, Suryaa Narayanan K and Santhosh Kumar E, "Prediction and Prevention Analysis Using Machine Learning Algorithms for Detecting the Crime Data".

[9] G. Harriat Christa, J. J, A. K and K. M. Sagayam, "CNN-based Mask Detection System Using OpenCV and MobileNetV2,".

[10] Y. Wang, Q. Song, J. Wang and H. Yu, "Airport Runway Foreign Object Debris Detection System Based on Arc-Scanning SAR Technology,".

[11] Yadav, Shashi. (2020). Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence. International Journal for Research in Applied Science and Engineering Technology.

[12] M. Ote, S. Jeong and M. M. Tentzeris, "Foreign Object Detection for Wireless Power Transfer Based on Machine Learning,".

[13] Sneha Sen and Khushboo Sawant 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1070 012061. DOI 10.1088/1757-899X/1070/1/012061

[14] K. Suresh, M. Palangappa and S. Bhuvan, "Face Mask Detection by using Optimistic Convolutional Neural Network,".

[15] I. B. Venkateswarlu, J. Kakarla and S. Prakash, "Face mask detection using MobileNet and Global Pooling Block,".

[16] W. Vijitkunsawat and P. Chantngarm, "Study of the Performance of Machine Learning Algorithms for Face Mask Detection,".

[17] N. Yonemoto, S. Futatsumori, A. Kohmura, K. Morioka and N. Miyazaki, "Multi-static Millimeter Wave Radar Connected by Radio over Fiber for Foreign Object Detection on Runways,".

[18] Federal Aviation Administration,"Airport foreign object debris manage-ment,".

[19] Federal Aviation Administration, "Airport foreign object debris detection equipment," Federal Aviation Administration, Tech. Rep., 09 2009, advisory circular: 150/5220-24

[20] C. M'Sila, R. Ayad and N. Ait-Oufroukh, "Automated Foreign Object Debris Detection System based on UAV,".

[21] J. Wang, S. Yan, L. Zhou, R. Fan, X. Han and X. Wang, "Research on Application of New Technology in Foreign Object Debris Detection Field,".

[22] W. Wang, B. Wu, S. Yang and Z. Wang, "Road Damage Detection and Classification with Faster R-CNN,".

[23] D. Arya et al., "Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022),".

[24] X. Wang, Z. Lan, Z. Li, Y. Lin and Q. Xiao, "A Practical Application of Runway Foreign Object Debris Detection System at the Airport,".

[25] S. M. Zainab, K. Khan, A. Fazil and M. Zakwan, "Foreign Object Debris (FOD) Classification Through Material Recognition Using Deep Convolutional Neural Network With Focus on Metal,".

[26] H. Ali, M. Khursheed, S. K. Fatima, S. M. Shuja and S. Noor, "Object Recognition for Dental Instrument

Using SSD-MobileNet,”.