Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1
© 2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DETECTION OF FOREIGN OBJECT DEBRIS (FOD) USING CONVOLUTIONAL NEURAL NETWORK (CNN)

KUSWORO ADI[1], CATUR EDI WIDODO[1], ARIS PUJI WIDODO[2], UTAMI SRI MARGIATI[1]
[1]Department of Physics, Faculty of Science and Mathemtaics, Diponegoro University, Indonesia
[2]Department of Informatics, Faculty of Science and Mathemtaics, Diponegoro University, Indonesia
E-mail: [1]kusworoadi@lecturer.undip.ac.id

ABSTRACT

Foreign object damage (FOD) is a big problem in aviation maintenance industry that reduces the level of safety for an aircraft. Basically, FOD is known as foreign object (FO) that can cause severity and destruction to the aircraft such as engine failure and loss of human life. Nowadays there is no FOD detection system that can classify the type of FOD that is detected optimally. This study proposes the FOD classification approach using Deep Learning. The Deep Learning method that has significant results in image recognition is the Convolutional neural network (CNN). The purpose of this study is to determine the CNN algorithm model that has optimal accuracy in the FOD classification. Convolutional Neural Network (CNN) is a multilayer neural network with a supervised learning architecture that consists of two parts, namely feature reviewers and classifiers that can be trained. The research began by designing a dataset and CNN algorithm model, then proceed with Training and Testing data. Based on the research, CNN model that has optimal accuracy using a 64 x 64 input image, learning rate value of 0.001, filter size 3 x 3, number of epochs 100, number of training data of 1200 data and test data of 20 data. The number of convolution layers used is 1 layer. The algorithm created is able to classify FOD objects into 6 classes with 86.6% accuracy and provide the best classification results into 4 classes with an accuracy of up to 90%.
Keyword: *Foreign Object Debris, Deep Learning, Convolutional Neural Network.*

1. INTRODUCTION

Foreign Object Debris (FOD) is a major problem in the aviation maintenance industry that reduces the safety level for aircraft. Basically, FOD, which is known as a foreign object, can cause damage to aircraft such as engine damage and loss of human life. One example of an accident due to FOD is the crash of a French Flight 4590 in 2000 which killed 113 people. The economic loss due to this accident is estimated at 3 - 4 billion USD per year. FOD refers to objects in the vicinity of the airport, especially on the runway that can damage an airplane. Examples of FOD are bent metal strips that resulted in the French Flight 4590 accident, components detached from aircraft or vehicles, chunks of concrete from runways and plastic products [28]. Currently, in order to reduce FOD air accidents four runway inspection systems have been developed designed for airports including Tariser in the UK, FODFinder in US, FODetect in Israel and IFerret in Singapore [17].

Deep Learning technology has become a keyword because it is able to provide good results in image classification, object detection and natural language processing. This is due to the availability of large data sets and powerful Graphics Processing Units (GPU) [22]. The deep learning method that has significant results in image recognition is Convolutional Neural Network (CNN). The CNN method has been introduced into many computer vision applications, such as image classification, face verification, semantic segmentation, object detection and image annotation. The CNN algorithm has been shown to perform better in detection and recognition as it has better resolution and robustness for FOD detection [2].

Research on the CNN method to anticipate air accidents due to FOD has been carried out both experimentally, theoretically and for other applicative purposes. The introduction of foreign object debris (FOD) material based on Convolutional neural network was carried out by Xu et al in 2018 which resulted that the method used could increase the accuracy of material recognition by 39.6% on FOD objects. However, the concrete

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

background can affect the classification between metal and plastic so it requires radar and infrared to overcome this. In 2018 Cao et al also conducted a region-based CNN study for FOD detection at airports. This study uses an improved Region proposal network (RPN) with several additional rule options designed and used to produce better results. The results of this study indicate that the algorithm used is more effective and robust for FOD detection than other detection algorithms such as R-CNN and Single Shot Multibox Detector.

In this research, a review of previous research problems was carried out using the Convolutional Neural Network (CNN) method. FOD that has been detected using a FOD detector device will be classified based on its type using the CNN algorithm.

2. THEORY

2.1 Foreign Object Debris (FOD)

Foreign Object Debris or what is known as FOD is a substance, debris, or foreign object that is in the system that can cause damage [3]. According to the ATSB about 11% of the damage was caused by FOD. This damage also has a financial impact on the organization, either directly or indirectly [7].

2.2 Deep Learning

The application of artificial intelligence has now entered human life. Artificial intelligence or Artificial Intelligence (AI) is a field of science that studies how to build computer systems that apply intelligence in several ways. Currently, there are many studies on artificial intelligence, one of which is machine learning [27]. Machine Learning itself is a branch of Artificial Intelligence which focuses on systems that allow learning from data [4]. Deep learning is a branch of machine learning where information is processed in a hierarchical layer to understand the representation and features of data in increasing the level of complexity [26]. The most well-known and used group of deep learning algorithms is the convolutional neural network because of its powerful application in recognizing different patterns, especially in detecting objects in digital images [11]. Figure 1 is the relationship between artificial intelligence, machine learning, deep learning and convolutional neural networks.

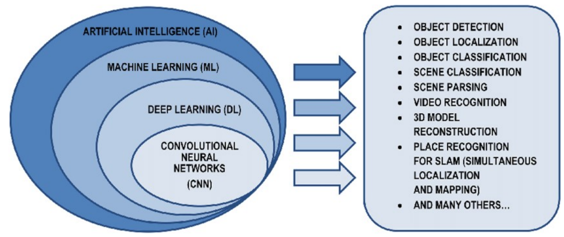

Figure 1 : Deep Learning

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a method of deep learning that is designed to cover the weaknesses of the previous method. The CNN model can reduce a number of free parameters and input image deformation such as translation, rotation and scale can be handled [24]. Convolutional Neural Network is a multilayer neural network with a supervised learning architecture which consists of two parts, namely extractor features and classifier that can be trained [16]. CNN has a high network depth and is often applied to image data so that it is included in the Deep Neural Network [20].

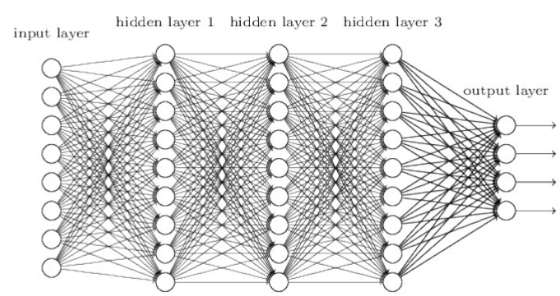

Figure 2 : Network of CNN

In CNN, the data propagated by the network is two-dimensional, so the linear operation and weight parameters on the CNN are different. In CNN linear operation uses a convolutional operation. Then the weight network will be trained so that the output can identify the image correctly [18]. The workings of CNN are the same as MLPs, but in CNN, each neuron is represented in two dimensions, whereas the MLP of each neuron is only one dimension. The base CNN network contains one input layer, one output layer, one convolution layer, one maxpooling layer, and one fully connected layer [15]. The input layer receives the signal from outside, then passes it to the first hidden layer, which will be forwarded until it reaches the output layer [1].

2.4 Backpropagation Algorithm

Backpropagation is a learning algorithm that is used to change each weight associated with neurons in hidden layers in the reverse direction using an error. This training includes three stages, namely, a forward stage (forward), a backward stage (backward) and a stage that will modify the weights to reduce errors [21].

3. METHODS

The tools needed in the research include a PC computer, python software, the OpenCV library, the Keras library, and the Tensorflow library. While the materials used are bolt image, glass image, key image, and nail image. Research on the classification of FOD objects using the CNN method as a whole includes 3 stages, namely the data set design, the Convolutional Neural Network (CNN) algorithm model design and the test design.

3.1 Dataset Design

The dataset design is done by collecting the data used in the study. Data taken as much as 1220 data consisting of 305 images of each type of FOD. The data is divided into two parts, namely 1200 data as training data and 20 data as test data.

3.2 Training Model

After the dataset design process, the model training process is carried out. This process is done first by making a CNN model with the name ShallowNet.py. CNN model script as shown in Figure 3.

```
class ShallowNet:
      @staticmethod
      def build(width, height, depth, classes):

            model = Sequential()
            inputShape = (height, width, depth)

            if K.image_data_format() == "channels_first":
                  inputShape = (depth, height, width)

            model.add(Conv2D(32, (5, 5), padding="same",
                   input_shape=inputShape))
            model.add(Activation("relu"))

            model.add(Flatten())
            model.add(Dense(classes))
            model.add(Activation("softmax"))

            return model
```

Figure 3 : CNN Model Script

The CNN model used has an architecture with two convolution layers. In each convolution process, the RELU activation function is used to change the minus value in a matrix. This function performs the threshold from zero to infinity. For each convolution process, zero padding is used, namely pixels containing zero values that are added to each side of the input matrix. The result of maxpooling in the form of a multidimensional array will be converted into a vector using flatten. There is a dense layer with 256 vectors and the RELU activation function and the softmax activation function. Softmax activation function is used to classify into many classes.

In the preprocessing stage as shown in Figure 4, SimplePreprocessor is used to resize the input image to 32 x 32. ImageToArrayPrepocessor is used to handle image arrays. Both preprocessing is combined in SimpleDatasetLoader. After preprocessing the image is loaded with the label and converted to a scale [0, 1]. The data will be separated into 2 parts, 75% as training data and 25% as validation data. The next stage is to create a model based on the data using ShallowNet that has been created. In the process of compiling the model, cross entropy is used as a loss function, SGD optimizer, learning rate of 0.001, the input image size is 32 x 32 pixels with 3 color channels and consists of 4 classes. In the training process, a batch size of 32 is used and the number of epochs is 100. Then the training model is stored in the HDF5 extension.

```
# initialize the image preprocessors
sp = SimplePreprocessor(64, 64)
iap = ImageToArrayPreprocessor()

# load the dataset from disk then scale the raw pixel intensities
# to the range [0, 1]
sdl = SimpleDatasetLoader(preprocessors=[sp, iap])
(data, labels) = sdl.load(imagePaths, verbose=500)
data = data.astype("float") / 255.0

# partition the data into training and testing splits using 75% of
# the data for training and the remaining 25% for testing
(trainX, testX, trainY, testY) = train_test_split(data, labels,
     test_size=0.25, random_state=42)

# convert the labels from integers to vectors
trainY = LabelBinarizer().fit_transform(trainY)
testY = LabelBinarizer().fit_transform(testY)

# initialize the optimizer and model
print("[INFO] compiling model...")
opt = SGD(lr=0.001)
model = ShallowNet.build(width=64, height=64, depth=3, classes=6)
model.compile(loss="categorical_crossentropy", optimizer=opt,
     metrics=["accuracy"])

# train the network
print("[INFO] training network...")
H = model.fit(trainX, trainY, validation_data=(testX, testY),
     batch_size=32, epochs=100, verbose=1)

# save the model to disk
print("[INFO] serializing network...")
model.save(args["model"])
```

Figure 4 : Preprocessing and Training Process Script

3.3 Testing Model

The data training process aims to train the CNN algorithm in recognizing the dataset and forming a model based on the training. The model testing process is used to test a model of training or training results. The first process in testing is to call the model that has been formed in the training process

Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1
© 2021 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and then call the image to be tested. The ext process is preprocessing and prediction of the test image. Prediction results will be displayed with the test image. The model testing script can be seen in Figure 5.

```
# load the trained convolutional neural network
print("[INFO] loading network...")
model = load_model(args["model"])

# load the image
image = cv2.imread(args["image"])
orig = image.copy()

# pre-process the image for classification
image = cv2.resize(image, (32, 32))
image = image.astype("float") / 255.0
image = img_to_array(image)
image = np.expand_dims(image, axis=0)

# make predictions on the sample of testing data
print("[INFO] predicting on testing data...")
probs = model.predict(image)
prediction = probs.argmax(axis=1)[0]
lbl = gtLabels[prediction]
print(lbl)

#draw the label on the image
output = imutils.resize(orig, width=400)
cv2.putText(output, lbl, (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,
      0.7, (0, 255, 0), 2)

# show the output image
cv2.imshow("Output", output)
cv2.waitKey(0)
```

Figure 5 : Testing Process Script

The prediction results at the model testing stage are displayed with the type label of the test image in the upper left corner. Prediction results can be seen as Figure 6.

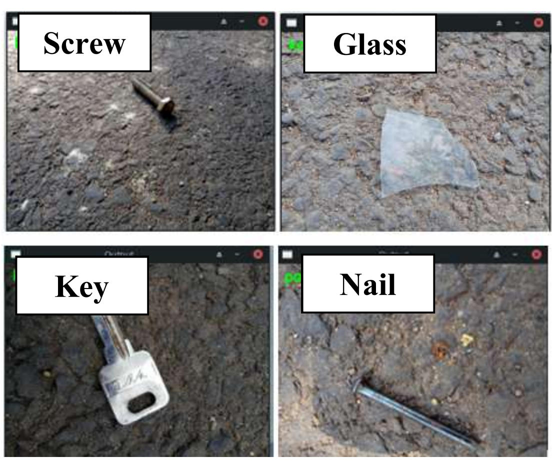

Figure 6 : Predicted Image

4. RESULT AND ANALYSIS

4.1 CNN process

In the CNN algorithm the formation of network architecture can affect the results and accuracy of the model. Enter the image on CNN using an image with a size of 32x32x3. The input image will go through a feature learning stage in the form of a convolution and pooling process. Then the image will go through the fully connected layer stage in the form of flatten and activation of the Softmax function. The convolution process carried out at this stage is 2 times with the first convolution process using 32 filters and a 3x3 kernel. The filter will shift stride 2 from a horizontal and then a vertical direction to all parts of the input image. So that the size of the output image from the convolution process has the same size of 32x32 pixels, so that it does not reduce the information in the image, a padding of 1. The activation function used is ReLU (Rectified Linear Unit). The value of the convolution result which is negative will be changed by this activation to zero and the others to infinity. The second convolution process used 64 filters with a kernel size of 3x3 and a ReLU activation function.

The second process in the feature learning stage is the pooling process. Pooling is a reduction in the size of the matrix using pooling operations. This study uses max-pooling to obtain a new matrix size. Furthermore, the fully connected layer process is used to transform data so that it can be classified linearly. This process uses flatten to convert the results of feature learning in the form of multidimmensional arrays into vectors. In the classification process, the Softmax activation function is used.

4.2 Determination of Model Parameters

The purpose of determining model parameters is to compare the models that have the best results by taking into account the parameter values. Based on the parameter of the number of train data, it can be concluded that the more the number of train data used, the greater the accuracy obtained. This shows that computers are more in understanding image patterns so that the classification process can be carried out appropriately.

Based on the learning rate parameter, it shows that the use of a learning rate of 0.001 results in high accuracy compared to the learning rate of 0.0001. This is due to the slow convergence process of loss values during the training process, so the results of loss validation are very large. Based on the input image size parameters, the results show that the 64x64 image size has the best accuracy. From the convolutional kernel size parameters, it can be concluded that the smaller the kernel size, the higher the accuracy value because the observation of the image is more detailed. The greater the number of layers used can slow down the running of training data. This is due to the number of image extraction steps performed. The effect of the number of classes on the accuracy of the classification results is inversely proportional to the number of classes, the smaller the accuracy will be. This is due to the increasingly complex separation of data due to the increasing number of classes.

4.2.1 Effect of Total Training Data

In this research using a different amount of training data. The training data used are 320 and 1200 data. From each of the training data as much as 25% is used as validation data. Table 1 is the result of a comparison of the effect of the amount of training data.

Table 1 : Accuracy Based on Data Training

Training Data	Accuracy Validation	Loss Validation	Testing
320	65 %	1,1977	74 %
1200	65,33 %	0,7698	78 %

Based on Table 3, it can be concluded that the greater the amount of training data used for model training, the greater the accuracy obtained. This shows that computers are more able to understand image patterns so that the classification process can be carried out correctly.

4.2.2 Effect of Learning Rate Value

The determination of the learning rate value is usually determined by the researcher and greatly influences the accuracy performance. In the image classification process, a learning rate value of 0.1 to 0.0001 is widely used. In this study, two learning rate values were used, namely 0.001 and 0.0001 with accuracy results as shown in Table 2.

Table 2 : Accuracy Based on Learning Rate

Learning Rate	Accuracy Validation	Loss Validation	Testing
0,001	80,33 %	0,4929	85 %
0,0001	73,67 %	0,8215	74 %

Table 2 shows the use of a learning rate value of 0.001 resulting in high accuracy compared to a learning rate value of 0.0001. This is due to the slow convergence of the loss value during the training process, so the loss validation results are very large.

4.2.3 Effect of Image Size Input

The image sizes input used in this research are 32 x 32 pixels and 64 x 64 pixels. After training the data, the results are shown in Table 3. Based on the research that has been done, it can be seen that the best accuracy value occurs at the input image size of 64 x 64 pixels.

Table 3 : Accuracy Based on Input Image

Input Shape	Accuracy Validation	Loss Validation	**Testing**
32 x 32	66,67 %	0,7917	72 %
64 x 64	80,33 %	0,4929	8**5 %**

4.2.4 Effect of Convolution Kernel Size

In this study, different convolution kernel sizes were used. The size of the CNN algorithm convolution kernel in this study is 3x3 and 5x5. Table 4 shows the results of the research with different convolution kernel sizes.

Table 4 : Accuracy Based on Convolution Kernel Size

Convolutional Size	Accuracy Validation	Loss Validation	Testing
3 x 3	80,33 %	0,4929	85 %
5 x 5	79,33 %	0,5101	82 %

Based on Table 4, it can be seen that the validation accuracy value is greater using a 3 x 3 kernel. The value of loss validation which has a small value occurs in the CNN model which has a convolution kernel size of 3 x 3. It can be concluded that the smaller the kernel size, the smaller the value. The accuracy is higher because the observation of the image is getting more detailed.

4.2.5 Effect of Number Convolution Layer

In the CNN algorithm, the convolution layer is used in the feature extraction process in the image. This layer is very important in convolutional neural networks. Table 5 shows the number of convolution layers used in a CNN algorithm that can affect the accuracy of the model.

Table 5 : Accuracy Based on Convolution Layers

Convolution Layers	Accuracy Validation	Loss Validation	Testing
1	66,67 %	0,7917	72 %
2	65,33 %	0,7698	78 %

Table 5 shows that the number of convolution layers to produce a higher validation accuracy value is 1 layer. The more the number of layers used can slow

Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1
© 2021 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

down the training data. This is due to the many stages of image extraction carried out.

4.5.6 Effect of Number of Class

In this research, the number of different classes used. The first number of FOD classes is 6 classes, namely stone, bolt, bird, glass, key, and nail. The second number of classes is 4 classes, namely bolts, glass, keys, and nails. The results of the comparison of the accuracy of each number of classes in each scenario can be seen in Table 6.

Tabel 6 : Accuracy Based on Number of Class

Number of Class	Accuracy
Scenario 1:	
6 Classes	43 %
4 Classes	65 %
Scenario 2 :	
6 Classes	86 %
4 Classes	**90 %**

The effect of the number of classes on the accuracy of the classification results is inversely proportional, namely the more the number of classes, the smaller the accuracy obtained. This is due to the increasingly complex data separation due to the increasing number of classes.

4.3 Results of Fist Sekanrio Training and Testing

Based on the Convolutional Neural Network (CNN) network architecture that has been created, training and testing results are obtained. The parameters used in model testing can be seen in Table 7.

Table 7 : First Scenario CNN Parameters

Parameters	Value
Size Image	32 x 32
Size Filter	3 x 3
Size pooling	2 x 2
Number data training	320
Number data testing	20
Epoch	100
Learning Rate	0,001

The graph of the training and validation results can be seen in Figure 6. The number of train data used in the first scenario is small, so that the network experiences overfitting. Overfitting Neural Networks do not learn data effectively and do not classify new data well [6]. From the graph of validation loss, which is still high, it is possible that overfitting has occurred on the network. In the second scenario several parameters will be changed to reduce overfitting [25].

Figure 6 : Training Progress First Scenario

Based on testing the models that have been made in the training process, the classification results are obtained with testing accuracy of 65% for 4 classes and 43% for 6 FOD classes.

4.4 Results of Second Sekanrio Training and Testing

The parameters used in the second scenario can be seen in Table 8.

Table 8 : CNN Second Scenario Parameters

Parameters	Value
Size Image	64 x 64
Size filter	3 x 3
Number of convolution layers	1
Number data training	1200
Number data testing	20
Epoch	100
Learning Rate	0,001

The graph of the training and validation results can be seen in Figure 7. Based on the small validation loss value, this network is not overfitting. This shows that the CNN algorithm is capable of performing classification well.

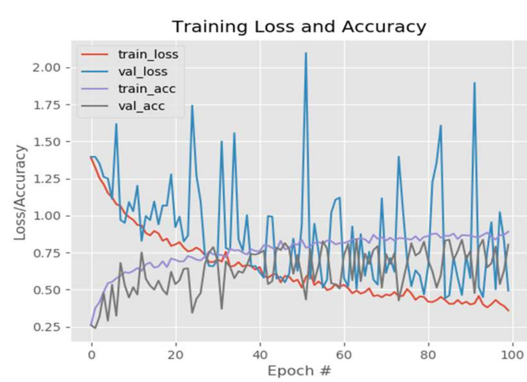

Figure 7 : Training Progress Scenario 2

© 2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Based on testing the models that have been made in the training process, the classification results are obtained with testing accuracy of 90% for 4 classes and 86% for 6 FOD classes. In this research, an interface is also made so that it is easy and practical to use in the introduction of FOD types as shown in Figure 8.

Figure 8 : FOD System Interface

Based on the test results with 2 scenarios, then the second scenario is tested with 4 classes. The test results are as in Table 9, in this test, the observed class is compared with the actual class. In this test there are 2 classes that are wrong, namely in the bolt class and the nail class. The validation process of the system uses calculations with the equation :

$$Accuracy = \left(100\,\% - \left(\frac{|Observed\ Class - Actual\ Class|}{Actual\ Class} x100\%\right)\right)$$

$$Accuracy = \left(100\,\% - \left(\frac{|18-20|}{20} x100\%\right)\right) = 90\%$$

Table 9 : Results of the Second Scenario FOD Classification for 4 Classes

No	Image Code	Observed Class	Actual Class	Detection
1.	Bolt (1)	Bolt	Bolt	True
2.	**Bolt (2)**	**Key**	**Bolt**	**False**
3.	Bolt (3)	Bolt	Bolt	True
4	Bolt (4)	Bolt	Bolt	True
5	Bolt (5)	Bolt	Bolt	True
6	Glass (1)	Glass	Glass	True
7	Glass (2)	Glass	Glass	True
8	Glass (3)	Glass	Glass	True
9	Glass (4)	Glass	Glass	True
10	Glass (5)	Glass	Glass	True
11	Key (1)	Key	Key	True
12	Key (2)	Key	Key	True
13	Key (3)	Key	Key	True
14	Key (4)	Key	Key	True
15	Key (5)	Key	Key	True
16	Nail (1)	Nail	Nail	True
17	Nail (2)	Nail	Nail	True
18	Nail (3)	Nail	Nail	True
19	Nail (4)	Nail	Nail	True
20	**Nail (5)**	**Bolt**	**Nail**	**False**

Comparison with other methods as shown in Table 10, it appears that based on the table the method offered gives promising results. The method offered has better accuracy than the Faster R-CNN, SSD (Single Shot Detector) and Selective Search methods. However, it is still lower with the Region Proposal Network (RPN) + FOD Detector method, the Region Proposal Network (RPN) + FOD Detector method classifies 2 classes, namely screw and stone. While the method that we offer classifies 4 classes, so this method is quite promising for further development [2].

Tabel 10 : Comparison With Other Methods

No	Methods	Accuracy
1	Faster R-CNN	89.43%
2	SSD (Single Shot Detector)	89.92%
3	Selective Search	85.37%
4	Region Proposal Network (RPN) + FOD Detector	98.41%
5	**Proposed Method Using CNN + Tensorflow**	**90.00%**

5. CONCLUSION

From the results of the research conducted, the following conclusions were obtained: The best CNN model uses an input image of 64 x 64 size, a learning rate of 0.001, a filter size of 3 x 3, the number of epochs of 100, the number of training data 1200 and 20 test data. The number of convolution layers used is 1 layer. The Convolutional Neural Network (CNN) algorithm in this study is able to classify FOD objects into 4 classes, namely bolts, glass, locks, and nails with testing accuracy reaching 90%. The advantage of this method is that it can immediately recognize FOD objects captured by the camera more quickly. While the weakness of this method for changes to each class to be identified accuracy will decrease. However, it can be overcome by adding training data so that it can recognize objects with more classes.

ACKNOWLEDGMENTS
This research was funded by the Indonesian Directorate General of Higher Education Program in 2020.

REFFERENCES

[1] Fuyong Xing, Yuanpu Xie, Hai Su, Fujun Liu and Lin Yang, "Deep Learning in Microscopy Image Analysis: A Survey", *IEEE Transactions On Neural Networks And Learning Systems*, Vol.29, No.10, 2018, pp.4550-4568.

[2] Cao, X., Wang, P., Meng, C., Bai, X., Gong, G., Liu, M., and Qi, J, "Region based CNN for foreign object debris detection on airfield pavement", *Sensors (Switzerland)*, Vol.18, No.3, 2018, Pp.1–14.

[3] Z.D. Yuan, J.Q. Li, Z.N. Qiu, and Y. Zhang, "Research on FOD detection system of airport runway based on artificial intelligence", *Journal of Physics: Conference Series*, Vol. 1635, 2020, p. 012065.

[5] Gonzalez, Rafael and Woods, R. E, "*Digital Image Processing Second Edition* (second)", 2002, New Jersey: Prentice Hall.

[7] Hussin, R., Ismail, N., and Mustapa, S., "A study of foreign object damage (FOD) and prevention method at the airport and aircraft maintenance area", *IOP Conference Series: Materials Science and Engineering*, Vol.152, No.1, 2016, Pp.1-9.

[9] Jain, A, 2016, "*Fundamentals of Digital Image Processing"*,2016, Prentince-Hall International.

[11] Khoma, V., Pelc, M., Khoma, Y., and Sabodashko, D., "Outlier Correction in ECG - Based Human Identification", *Proceedings of the 3rd International Scientific Conference on Brain-Computer Interfaces,* 2018, Pp. 11 - 21.

[13] Laganiere, R., "*OpenCV 2 Computer Vision Application Programming Cookbook",* 2011, Packt Publishing, Brimingham.

[15] Leskovec, J., 2018, Representation Learning with Convolutional Neural Networks, *Proquest.*

[16] Liang, W., Zhang, H., Zhang, G., and Cao, H., "Rice Blast Disease Recognition Using a Deep Convolutional Neural Network", *Scientific Reports*, Vol.9, No.1, 2019, Pp.2869.

[17] Luo, M., Wu, C., Sun, H., Xie, X., and Wu, X., "Demonstration of Airport Runway FOD Detection System Based On Vehicle SAR", *IOP Conference Series: Materials Science and Engineering*, 2018, Vol.452.

[18] Mehling, B., "Artificial neural networks", In *Studies in Systems, Decision and Control*, Vol.131, 2019, Pp.11-35.

[20] Nielsen, M, "Neural Networks and Deep Learning", 2018, In *Artificial Intelligence*.

[22] Pathak, A. R., Pandey, M., and Rautaray, S, "Application of Deep Learning for Object Detection", *Procedia Computer Science*, Vol.132, 2018, Pp.1706–1717.

[23] Rosebrock, A, "*Deep Learning for Computer Vision with Python : Starter Bundle",* 2017, PyImageSearch.

[25] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, Vol.15, 2014, Pp.1929–1958.

[26] Vasilev, I, "*Python deep learning: exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow* (Second)", 2019, Birmingham-Mumbai: Packt Publishing Ltd.

[27] W. J. Teahan, "*Artificial Intelligence - Agents and Environments*, 2010, Bookboon.

[28] Xu, H., Han, Z., Feng, S., Zhou, H., and Fang, Y, "Foreign object debris material recognition based on convolutional neural networks", *Eurasip Journal on Image and Video Processing*, Vol.2018, No.1, 2018, Pp.1-10.