Journal of Applied Science, Engineering and Technology
e-ISSN: 2722-8363 p-ISSN: 2722-8371
DOI: *https://doi.org/10.47355/aset*

J. ASET

STUDY OF XCEPTION MACHINE LEARNING ARCHITECTURE IN WASTE CLASSIFICATION SYSTEM

Rian Kurniawan [1], Yessi Mulyani [1,*], Puput Budi Wintoro [1], M. Komarudin [1]

[1] Electrical Engineering Department, Engineering Faculty, University of Lampung, Jl. Sumantri Brojonegoro No.1, Bandar Lampung 35145, Indonesia

[*] Correspondence: yessi.mulyani@eng.unila.ac.id

Abstract: Garbage generated every day can be a problem because some types of waste are difficult to decompose so they can pollute the environment. Waste that can potentially be recycled and has a selling value is inorganic waste, especially cardboard, metal, paper, glass, plastic, rubber and other waste such as product packaging. Various types of waste can be classified using machine learning models. The machine learning model used for classification of waste systems is a model with the Convolutional Neural Network (CNN) method. The selection of the CNN architecture takes into account the required accuracy and computational costs. This study aims to determine the best architecture, optimizer, and learning rate in the waste classification system. The model designed using the Xception architecture with the Adam optimizer and a learning rate of 0.001 has an accuracy of 87.81%.

Keywords: Classification, Convolutional Neural Network, Machine Learning, Xception, Waste

1. Introduction

Garbage generated every day can be a problem because some types of waste are difficult to decompose so they can pollute the environment[1]. Waste consists of two types, namely organic waste and inorganic waste. Organic waste is waste that comes from the remains of living organisms, while inorganic waste comes from non-living organisms. Waste that can potentially be recycled and has a selling value is inorganic waste, especially cardboard, metal, paper, glass, plastic, rubber and other waste such as product packaging.[2]

Various types of waste can be classified using machine learning models. The machine learning model used for classification of waste systems is a model with the Convolutional Neural Network method. The model with this method will recognize garbage images by extracting image features and recognizing patterns according to the labels on the training data[3].

In the last few decades, deep learning has become a powerful tool. This is evidenced by its ability to handle large amounts of data and be able to recognize patterns from the data it manages. One of the popular algorithms for handling large amounts of data is the Convolutional Neural Network[4].

The Convolutional Neural Network architecture used to design machine learning models plays an important role. The more precise the choice of Convolutional Neural Network architecture, the better the accuracy of the model made to predict the garbage image. In addition to good accuracy, each architecture has a different size, parameters, and cost (CPU/GPU). The Xception architecture is an architecture that has high accuracy, small size, and fewer parameters than some other

architectures, so that models trained using this architecture will be used effectively and efficiently to predict images[5].

2. Materials and Methods

2.1 Machine Learning

Initially, machine learning was a term used to refer to a branch of computer science that studied algorithm design methods that were able to learn or adapt to data patterns without being explicitly programmed. Machine learning has several computational methods that can improve performance by utilizing knowledge derived from experience when learning[6].

2.2 Convolutional Neural Network

Convolutional Neural Network is one of the deep learning algorithms used to classify images. CNN operates using convolution and uses at least one layer. Artificial neural networks can generally change the input value by inserting it into a series of hidden layers. Each layer can consist of a set of neurons, where each layer will be fully related to the previous layer. Finally, the previous layers will be fully connected to generate predictions[7].

2.3 Convolutional Layer

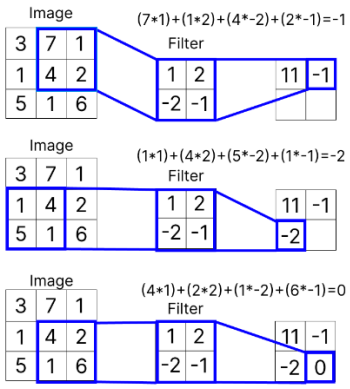


Figure 1. Convolution Operations

Convolutional Layer or convolutional layer is a layer in which there are feature essence extraction operations. The essence feature extraction operation is the dot product between the weights in the filter and the image pixels to be filtered and then summed afterwards as shown in Figure 1. Convolution Operations[3].

2.5 Xception

Xtreme of Inception (Xception) is a Convolutional Neural Network architecture that uses the depthwise separable convolution method. This architecture is the result of the development of the Inception architecture which has 36 convolution layers that form the basis of the feature extraction network[8].

2.6 Optimizer

Optimizer is an algorithm or method used to change the existing attributes on artificial neural networks. The attributes that can be changed using this algorithm are the value of weight and learning rate with the aim of reducing the value of loss during the training process[9].

2.6.1 Adaptive Moment Estimation (Adam)

Adam is an algorithm developed from the classic Stochastic Gradient Descent (SGD) algorithm where the network weight values have been updated. The use of this algorithm requires setting the learning rate first before carrying out the training process. Learning rate is a training parameter that is used to calculate the connection weight value when the training process is carried out. The size of the learning rate used will affect the effectiveness and efficiency of the training process[10].

2.7 Methods

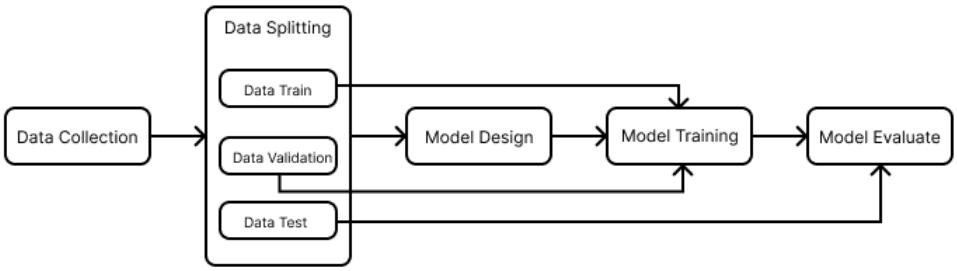


Figure 2. Research Stages

Research materials are in the form of Kaggle datasets, photo datasets obtained independently, datasets sourced from the internet, books, theses, journals, and other scientific sources obtained from various websites. The stages of this research are starting with data collection, data splitting, model design, model training, and model evaluate.

3. Results

3.1 Data Collection

Table 1. Dataset Description

Name	Files Type	Source	Download Link
Garbage Classification	.jpg	Kaggle	https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification
Independent Datasets	.jpg	Obtained independently	

The data collected consisted of 2,527 data sourced from Kaggle and 120 data produced independently by the author by taking photos directly and searching for them from the internet, so that the total data used was 2,647 data. Each data in the dataset is an image with a .jpg extension (Joint Photographic Experts Group). The following table shows the description of the dataset used.



Figure 3. Kaggle Datasets

The image is some data for cardboard, glass, metal, paper, plastic, and trash classes obtained from the kaggle site. There are 403 data in the cardboard class, 501 data in the glass class, 410 data in the metal class, 594 data in the paper class, 482 data in the plastic class, and 137 data in the trash class.



Figure 4. Independent Datasets

The images are some data for cardboard, glass, metal, paper, plastic, and trash classes that were obtained independently. There are 20 data for each class, a total of 120 data.

3.2 Data Splitting

```
import splitfolders
splitfolders.ratio('/content/datasets/', output="datasplit", seed=1337, ratio=(0.7, 0.15,0.15))
```

Figure 5. Data Splitting

At this stage the author divides the collected dataset into several parts, namely training data, validation data, and testing data, with respective ratios of 0.7, 0.15, and 0.15 of the total data which has a ratio of 1 The distribution ratio aims to distribute the validation data and test data obtained independently, which is distributed more, namely 3 data for each class. If the ratio is reduced to 0.1 for validation data and 0.1 for test data, then there are only 2 scattered data for each class. The distribution ratio of 0.15 for the validation data and test data makes the data distributed 50% more. From dividing by this ratio, 1,850 training data, 402 validation data, and 395 test data are obtained.

The distribution of training data, test data, and previous validation data was carried out randomly using the library available in Python. The author installs and uses a library called split folder to separate the data set into different directories according to the division of the data. The training data will be separated into a directory named "train", the test data will be separated into a directory named "test", and the validation data will be separated into a directory named "val."

```
train_dir = '../content/datasplit/train/'
test_dir = '../content/datasplit/test/'
val_dir = '../content/datasplit/val/'
```

Figure 6. Data Variable

Next, the location of the "train" directory is assigned to the train_dir variable, the "val" directory location is assigned to the val_dir variable, and the "test" directory location is assigned to the test_dir variable where these variables will be processed using the ImageDataGenerator class.

```
train_generator = train_datagen.flow_from_directory(
    train_dir,
    batch_size = BATCH_SIZE,
    class_mode = 'categorical',
    target_size = TARGET_SIZE,
    subset = 'training',
    save_to_dir = save_to
    )

test_generator = datagen.flow_from_directory(
    test_dir,
    batch_size = BATCH_SIZE,
    class_mode = 'categorical',
    target_size = TARGET_SIZE
    )

validation_generator = datagen.flow_from_directory(
    val_dir,
    batch_size = BATCH_SIZE,
    class_mode = 'categorical',
    target_size = TARGET_SIZE
    )
```

```
Found 1850 images belonging to 6 classes.
Found 402 images belonging to 6 classes.
Found 395 images belonging to 6 classes.
```

Figure 7. Target and Batch Data

After dividing, we create the TARGET_SIZE and BATCH_SIZE variables where these variables are used to store pixel size values and batch data sizes for training, validation and testing processes. Then use the flow_from_directory function of the ImageDataGenerator class to change the image data to the size according to the TARGET_SIZE variable and put it into a data batch where each data batch has a size equal to the value of the BATCH_SIZE variable. After this process, the resulting data is 1,850 training data, 402 validation data, and 395 test data.

3.3 Model Design

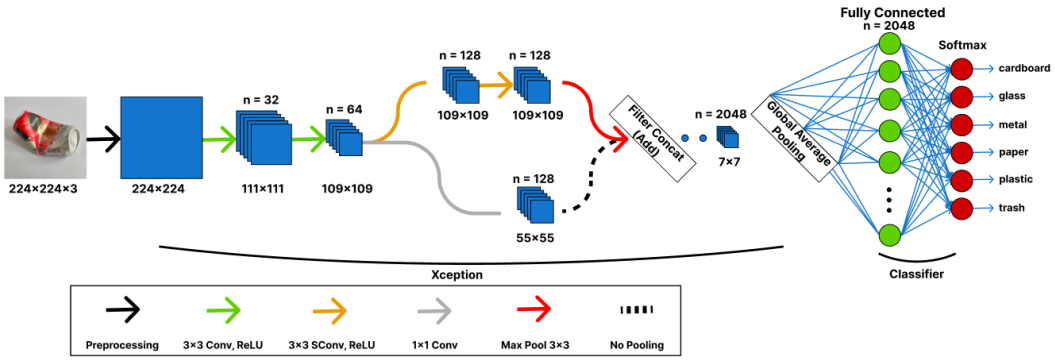


Figure 8. Xception

The image is the design of the Xception model used. there are 2048 neurons in the Fully-Connected layer before entering the prediction layer which will predict 6 classes with the Softmax activation function. From the image, initially the input size was 224x224 and then convolution was carried out to become 7x7 in size, but the depth increased.

3.3.1 Optimizer and Learning Rate

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics = ['accuracy'])
```

Figure 9. Optimizer and Learning Rate

The model was compiled using the Adam optimizer, learning rate of 0.001, loss using categorical crossentropy, and metrics accuracy. So that the model tends to be trained long enough to get optimal prediction accuracy for classifying each category.

3.4 Model Training

```
[17] history = model.fit(
        train_generator,
        validation_data = validation_generator,
        epochs = 100,
        verbose = 1,
        validation_steps = len(validation_generator),
        callbacks=[model_checkpoint,early_stop]
        )

     58/58 [==============================] - 53s 920ms/step - loss: 0.0947 - accuracy: 0.9627 - val_loss: 0.6481 - val_accuracy: 0.8481
     Epoch 36/100
     58/58 [==============================] - ETA: 0s - loss: 0.0776 - accuracy: 0.9724
     Epoch 36: val_loss did not improve from 0.47318
     58/58 [==============================] - 53s 917ms/step - loss: 0.0776 - accuracy: 0.9724 - val_loss: 0.5395 - val_accuracy: 0.8709
```

Figure 10. Model Training

At this stage training is carried out on each model with different optimizers and learning rates. The epoch value here does not have a significant effect because during the training process the callbacks function is used to stop training when the validation value does not increase, the model accuracy has become optimal.

3.5 Model Evaluate

```
model.evaluate(test_generator, verbose=1)

13/13 [==============================] - 3s 245ms/step - loss: 0.5010 - accuracy: 0.8781
[0.50103759765625, 0.8781094551086426]
```

Figure 11. Model Evaluate

At this stage, the evaluation of the model is carried out using test data and the evaluate method provided by the TensorFlow framework with the aim of obtaining a model accuracy value. Based on the figure above, the accuracy for the Xception model is 87.81%, where the model uses the Adam optimizer with a learning rate of 0.001.

4. Discussion

The results of the evaluation of the model using the evaluation method are that the optimal accuracy value is obtained in the Xception model with the Adam optimizer setting and a learning rate of 0.001. Based on these results, the optimizer setting and learning rate have an effect on the accuracy model.

5. Conclusion

The machine learning model of the garbage classification system designed using the Xception architecture with the Adam optimizer and a learning rate of 0.001 has a better accuracy of 87.81% than the models designed using the optimizer and other learning rates.

References

[1] Mengenal Berbagai Jenis Sampah yang Sulit Terurai dan Waktunya! Available online: https://www.gramedia.com/literasi/sampah-yang-sulit-terurai-dan-waktunya/ (accessed on 25 Jan 2023).

[2] Kusumaningsari, D., PEMANFAATAN DAN PENGOLAHAN SAMPAH ORGANIK DAN NON-ORGANIK UTILIZATION AND PROCESSING OF ORGANIC AND NON-ORGANIC WASTE. *J. Fisika, F. Matematika, D. Ilmu, and P. Alam*.

[3] Payana, M.D., TB, D. R. Y., Musliyana, Z., and Wibawa., 2022. M.B. DETEKSI MASKER WAJAH MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) MENINGKATKAN NILAI AKURASI MELALUI ARSITEKTUR LAYER KONVOLUSI. *JOURNAL OF INFORMATICS AND COMPUTER SCIENCE*, 8(1), 30–35. doi: 10.33143/JICS.VOL8.ISS1.2123.

[4] Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN) - Trivusi. Available online: https://www.trivusi.web.id/2022/04/algoritma-cnn.html (accessed on 25 Jan 2023).

[5] Keras API reference. Available online: https://keras.io/api/. (accessed on 25 Jan 2023).

[6] Heryadi, Y. and Wahyono, T., 2020. Machine Learning Konsep dan Implementasi. Gava Media.

[7] Memahami Apa Itu Convolutional Neural Network! Yuk Simak Penjelasannya di Sini. Available online: https://idmetafora.com/news/read/1114/Memahami-Apa-Itu-Convolutional-Neural-Network-Yuk-Simak-Penjelasannya-di-Sini.html (accesed on 25 Jan 2023)

[8] Sandhopi, Lukman Zaman, P.C.S.W., and Kristian, Y., 2020. Identifikasi Motif Jepara pada Ukiran dengan Memanfaatkan Convolutional Neural Network. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 9(4), 403–413. doi: 10.22146/JNTETI.V9I4.541.

[9] Various Optimization Algorithms For Training Neural Network. Available online: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6 (accessed on 25 Jan 2023).

[10] Rochmawati, N., Hidayati, H.B., Yamasari, Y., Tjahyaningtijas, H.P.A., Yustanti, W., and Prihanto, A., 2021. Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology*, 5(2), 44–48. doi: 10.26740/JIEET.V5N2.P44-48.