*Advances in Engineering Research, volume 208*

**Proceedings of the International Conference on Innovation in Science and Technology (ICIST 2020)**

# Implementation of Deep Learning for Organic and Anorganic Waste Classification on Android Mobile

R. D. Ramadhani*
Data Science Study Program, Faculty of Informatic
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
*rima@ittelkom-pwt.ac.id

A. N. A. Thohari
Electrical Engineering Department
Politeknik Negeri Semarang
Semarang, Indonesia

C. Kartiko
Software Engineering Study Program, Faculty of Informatic
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia

A. Junaidi
Data Science Study Program, Faculty of Informatic
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia

T. G. Laksana
Informatic Engineering Study Program, Faculty of Informatic
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia

***Abstract*—In this paper, a deep learning algorithm based on convolutional neural network (CNN) is implemented using pyhon and tensorflow lite for image classification on mobile. A large number different images which contains two types of waste, namely organic and anorganic are used for classification. The first stage to make classification model is prepare a dataset such as organic and anorganic waste images. Next divide both image in the training and validation directories. The split percentage when divide image is 90 percent for training and 10 percent for validation. After get image for training and testing, the next step is image augmentation to create new data from existing data. Next pre-processing using image data generator prepare the training data that will be implemented by the model. The important step in this process is make architecture of the CNN. In this paper used four layer convolution and there are two attributes that added to increase the accuracy of the training model. The first attribute is the dropout which make model become good fit and reduces overfitting. The second is adding padding and stride attributes to speed up the step of epoch during training. So that, by using padding and stride make the training time 50 percent faster than before. After got model with accuracy more than 90 percent, the last step is testing model using image in validation directories. Based on testing step, model has been able to classify images of organic and anorganic waste correctly. Application can running smoothly and could classify waste using live camera or photo in gallery.**

***Keywords*—deep learning, organic and non-organic waste classification, and convolutional neural network (CNN)**

## I. Introduction

Waste is a material that does not have the value used regularly or specifically in the scope of production. Examples of materials that usually become waste are damaged goods during manufacturing or excessive materials that mostly come from households [1]. The waste is usually dumped carelessly in various places or burned around the residents' residences, which will damage the surrounding environment.

In Indonesia, waste is a very serious problem. This can be proven by data obtained from the Central Statistics Agency (BPS) in 2017 where Indonesians are still difficult to sort waste and do not know the benefits of discarded waste [2]. Data obtained from the Ministry of Environment and Forestry in 2019 that waste in Indonesia reached 66-67 million tons, which is higher than the previous year which reached 64 million tons. The percentage of organic waste is 60%, and plastic waste reaches 15%. According to Tiyajamorn et al. research [3] in handling and processing, waste is classified into two types, namely organic and inorganic waste.

Even though the waste has been classified, there is still disposal that is not appropriate with its type so that it makes the garbage pile increase and is not compensated with proper processing that will cause various problems again [4]. Based on the problems in managing the type of waste, education is needed in detecting the image of the type of waste. Therefore, its disposal and management are appropriate with its designation using an application.

Based on these problems, the solution offered in this study is to utilize smart system technology. This technology can identify waste imagery using the android mobile application. There are several algorithms for image classification such as K-means, Support Vector Machine (SVM), and Convolutional Neural Network (CNN) [5]. However, of these three algorithms, CNN is the most used to detect Images [6].

CNN works by receiving input in the form of imagery, the input will be trained in several layers such as softmax to produce an output that can recognize the input object [7]. CNN is a classification of imagery taken from an input image that is then processed and classified. CNN consists of neurons that have weight, bias, and activation function [8]. But what distinguishes it from ordinary neural networks is that using a regular neural network may only contain long and high scales. But CNN can contain all the information from the entire scale that can classify objects more accurately because they can use the wide-scale as well [9].

As for Sultana research [10], CNN has been used to improve waste management systems and help create smart cities. Two Convolutional Neural Networks (CNN), both based on the AlexNet network architecture, were developed to search for waste objects in imagery and separate recyclable items from waste objects. The two-stage CNN system was first trained and tested on the benchmark TrashNet indoor imagery dataset with an accuracy of 93.6%. Then the system is trained and tested on outdoor imagery taken by the author in the intended use environment using an outdoor image dataset of various kinds of waste with an accuracy ranging from 89.7% to 93.4% and overall 92%.

The study [11] detects recycling or waste imagery and classifies it into six classes consisting of glass, paper, metal, plastic, cardboard, and trash. The study also created a dataset of about 400-500 images each. Models used are Support Vector Machine (SVM) with Scale-Invariant Feature Transformations (SIFT) and Convolutional Neural Networks (CNN). The results obtained, SVM worked better than CNN because CNN was not fully trained due to the difficulty of finding the optimal hyperparameter. Besides, CNN can also be applied in the field of Natural Language Processing (NLP), which utilizes various layers with convolving filters applied to local features [12]. In the study, The CNN model has proven effective for NLP and has achieved excellent results in semantic decomposition, search, sentence modeling, and other traditional NLP tasks.

## II. Methods

The study carried out is implementation research that applies a deep learning method that is a convolutional neural network (CNN) to identify waste. Constructing a training model requires a lot of data, and it is necessary to pre-process data techniques so that the data used is even. After obtaining an accurate model, add the model to the android project to create an application that can identify waste. The steps taken in this study showed in Figure 1.

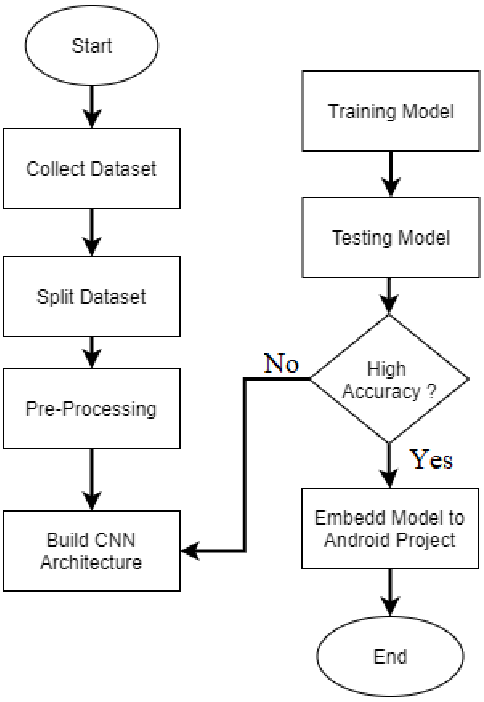


Fig. 1. Research methodology flowchart.

### A. Collect Dataset

The datasets used in this study are in the form of waste imagery. The image of waste taken is organic and inorganic waste. Images of organic and inorganic waste obtained from google image using Fatkun AI Downloader software. The total number of datasets is 172 images. The dataset is divided into two, 86 images of organic waste and 86 images of inorganic waste.

### B. Split Dataset

After obtaining a dataset of 86 images for organic waste and 86 images for inorganic, further divide the dataset into training and validation. The distribution of the dataset is done with a ratio of 80% for training and 20% for validation.

### C. Pre-Processing

At this step, dataset standardization is carried out so that all images are even. Because there are large images, some are small. At this step, image augmentation is done by rescaling, rotation, flip, etc. Therefore, the image has the same size and shape. At this step, an additional library of pythons is used, namely ImageDataGenerator.

### D. Build CNN Architecture

After pre-processing, the resulting image data has the same size of 150x150 pixels. It will then be obtained using CNN architecture that has been built. In the CNN architecture that was built, several configurations were carried out, including using the ReLU (Rectified Linear Unit) activation function

[13]. The purpose of using ReLU is to normalize the value generated convolutional layer. CNN uses a backpropagation neural network that uses the input gradient values according to equations 1 and 2 below.

$$f(x) = max(0, x) \qquad (1)$$

$$f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \qquad (2)$$

The dropout is used in the CNN architecture so that there is no overfitting. The use of dropouts is ideal for use as much as 20% [14]. The use of dropouts can also increase the speed of training. Because it will reduce the number of hidden layers used. The CNN architecture built can be seen in Figure 2.

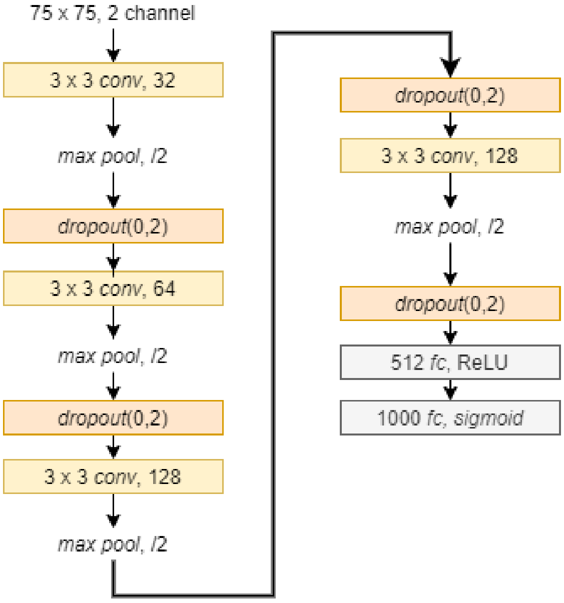


Fig. 2. CNN architecture built.

Sigmoid is also used as an activation function on the output layer in addition to using ReLU as an activation function. The use of sigmoid because the model only identified two types of waste. The equation of the softmax function showed in equation 3.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1} \qquad (3)$$

The value of S is the coefficient (weight), and x is the independent variable. The result of the sigmoid calculation will produce the shape of the letter 'S' curve. The sigmoid activation function is executed after calculating the fully connected layer on the backpropagation network [15]. The result after using the sigmoid calculation is a probability value between category 1 or positive and 0 or negative.

## E. Training Model

The training process is carried out to obtain a model based on inputs and CNN architecture built. During the training, additional parameters were added, namely binary cross-entropy and Adam's optimizer. Since the model is only used to identify organic and inorganic values, the use of binary cross-entropy is the most appropriate.

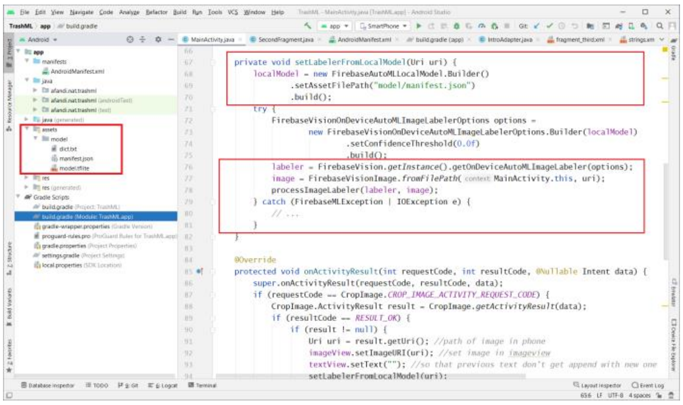
In the training process, the loss and accuracy values are obtained from the model. Based on the training results using the architecture and parameters used, the accuracy value is 94%. The training process takes 15 minutes. Increasing the dropout by 20% increases the speed so that reducing the number of parameters being processed.

## F. Testing Model

How to recognize the accuracy of the model is to do testing. Even though the training shows a validation value of 94%, it still needs to be tested using a prepared waste image. The testing process is carried out by entering data on organic and inorganic waste imagery from the training and testing directory. If the system can identify waste correctly after testing, it means that the model can be used for the waste identification process.

## G. Embedd to Android Project

The way to add a model to the android project is by creating a new directory in the res directory. Here is added a directory named model which contains the waste identification model that has been created. Figure 3 shows adding the model file to the Android project.



```
private void setLabelerFromLocalModel(Uri uri) {
    localModel = new FirebaseAutoMLLocalModel.Builder()
            .setAssetFilePath("model/manifest.json")
            .build();
    try {
        FirebaseVisionOnDeviceAutoMLImageLabelerOptions options =
                new FirebaseVisionOnDeviceAutoMLImageLabelerOptions.Builder(localModel)
                        .setConfidenceThreshold(0.0f)
                        .build();
        labeler = FirebaseVision.getInstance().getOnDeviceAutoMLImageLabeler(options);
        image = FirebaseVisionImage.fromFilePath( context: MainActivity.this, uri);
        processImageLabeler(labeler, image);
    } catch (FirebaseMLException | IOException e) {
        // ...
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        if (resultCode == RESULT_OK) {
            if (result != null) {
                Uri uri = result.getUri(); //path of image in phone
                imageView.setImageURI(uri); //set image in imageview
                textView.setText(""); //so that previous text don't get append with new one
                setLabelerFromLocalModel(uri);
```

Fig. 3. Adding models to android studio project.

Figure 3 shows the storage location for the model in the android project and how to call the model file in the java code. Therefore the model can be compiled in Java, it is necessary to add the TensorFlow lite library. To add the TFlite library, add the implementation code `'org.tensorflow:tensorflow-lite-task-vision:0.0.0-nightly'` in the project build.gradle. After that run the application to see the results of the model implementation.

## III. Implementation and Results

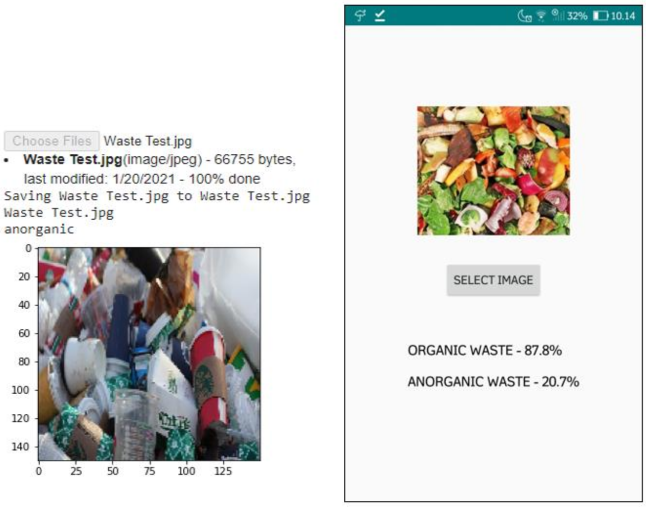
After process is carried out repeatedly using 20 iterations. The duration of the training process is about 15 minutes by using a computer with Core i7 specifications, 8 GB DDR4 RAM, and running using NVIDIA GeForce MX350. The results during the training process obtained the validation values shown in Table 1.

TABLE I. The Performance of Model Training

| Epoch | Training Loss | Training Accuracy | Validation Loss | Validation Accuracy |
|---|---|---|---|---|
| 1 | 0.8034 | 0.5400 | 0.6905 | 0.5000 |
| 2 | 0.6819 | 0.5258 | 0.6687 | 0.5000 |
| 3 | 0.4876 | 0.7400 | 0.3869 | 0.8889 |
| 4 | 0.3068 | 0.8969 | 0.2902 | 0.8333 |
| 5 | 0.2026 | 0.9500 | 0.2496 | 0.8889 |
| 6 | 0.2147 | 0.9175 | 0.3986 | 0.8889 |
| 7 | 0.2551 | 0.9278 | 0.2605 | 0.8889 |
| 8 | 0.4019 | 0.8600 | 0.3105 | 0.8889 |
| 9 | 0.2952 | 0.9072 | 0.2834 | 0.9444 |
| 10 | 0.1233 | 0.9691 | 0.2359 | 0.8889 |
| 11 | 0.2787 | 0.8866 | 0.4108 | 0.8889 |
| 12 | 0.4180 | 0.7900 | 0.3216 | 0.9444 |
| 13 | 0.2972 | 0.9000 | 0.3124 | 0.8889 |
| 14 | 0.1173 | 0.9500 | 0.2052 | 0.9444 |
| 15 | 0.2309 | 0.9278 | 0.1937 | 0.8889 |
| 16 | 0.2052 | 0.9400 | 0.2690 | 0.8889 |
| 17 | 0.1708 | 0.9200 | 0.3640 | 0.8889 |
| 18 | 0.2194 | 0.9072 | 0.2728 | 0.8889 |
| 19 | 0.1535 | 0.9485 | 0.2064 | 0.8889 |
| 20 | 0.1485 | 0.9588 | 0.1499 | 0.9444 |

The table 1 shows that the loss value is inversely proportional to the accuracy value. The greater the loss, the smaller accuracy will be. As much as possible, the resulting model has a high accuracy value so that the system can accurately recognize waste in the training process. At the beginning of the epoch, the value of loss tends to be greater than accuracy. This result is natural because the new system learns to recognize the given pattern. After that, the more iterations (epoch), the loss will tend to decrease, and the accuracy value will increase. One of the proofs that the system learns to the pattern given is declining the value of loss and increasing the accuracy. Finally, in the 20th epoch obtained an accuracy value of about 94%.

Further tests were carried out on the models obtained. Testing is done by randomly retrieving waste imagery. Before being implemented on android smartphones, testing is first done in jupyter notebooks. After the testing process in jupyter notebook successfully identified the waste as shown in Figure 4.a. Further models are implemented into android applications. Previously, the Android project has been created using android studio and java language. Then inserted the waste pattern recognition model into the project, then installed the build apk on android smartphones. The results of the waste recognition process on android applications can be seen in Figure 4.b. In Figure 4, the application can detect the type of waste accurately.



```
Choose Files  Waste Test.jpg
• Waste Test.jpg(image/jpeg) - 66755 bytes, last modified: 1/20/2021 - 100% done
Saving Waste Test.jpg to Waste Test.jpg
Waste Test.jpg
anorganic
```

ORGANIC WASTE - 87.8%

ANORGANIC WASTE - 20.7%

(a) (b)

Fig. 4. Model test results (a) desktops, (b) android apps.

## IV. Discussion

The addition of hyperparameters, such as padding, stride, and dropout values, is carried out to get good accuracy. The padding parameter is used to set the output dimension to remain the same as the input dimension or at least not drastically reduced. In architecture, the value of the padding parameter is set to "same". Then the strides parameter is used to determine the number of filter shifts. In architecture, the stride value is set to 2x2, which means a shift every 4 pixels. Finally, dropout parameters are used to prevent overfitting and speed up the learning process. Dropout works by temporarily removing neurons in the form of hidden layers, then using them randomly. The dropout value used in the network architecture is 20%, which means it will eliminate 20% of the neuron value.

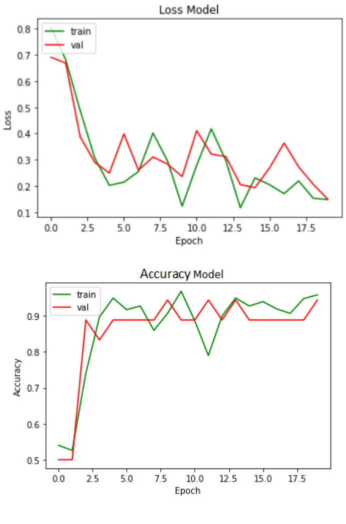


Fig. 5. Model training process graphic.

The result of adding the hyperparameter increases the accuracy value and prevents overfitting. This result can be seen in Figure 5, showing the sloping loss and accuracy model values. In the loss model chart, the loss value is quite high at the beginning of the epoch but then shrinks. Although there is a very significant fluctuation in the loss value, it is not overfitting because the loss value tends to decline. Whereas in the model accuracy graph, the resulting value is good because the resulting accuracy tends to increase.

Based on the results of training and model training obtained accuracy and loss value shown in Table 2. The value is already very good for identifying organic or inorganic waste. The value in Table 2 might be better if added to other hyperparameters such as learning rate and call-back. But still, have to be tested the addition of this hyperparameter because it can also make the model become overfitting or underfitting.

TABLE II. ACCURACY AND LOSS OF THE CLASSIFICATION MODEL

| **Training Stage** | | **Testing Stage** | |
|---|---|---|---|
| ***Accuracy*** | ***Loss*** | ***Accuracy*** | ***Loss*** |
| 4,12% | 95,88% | 94,44% | 5,56% |

The supported version of android to run this waste recognition app is android 5.0 (Lollipop) or more. Under android 5.0 or below API level 21, the application cannot be installed or run. Testing has been carried out on android 5.0 and android 10 (API level 29), and the application can run properly.

## V. CONCLUSION

Identification is carried out by entering the image of waste, then the computer will recognize the type of waste accurately. So that computers can identify waste, it is necessary to make a model for pattern recognition of waste. The modeling method is carried out using a deep learning algorithm convolutional neural network (CNN). In building CNN architecture that has high accuracy need to add a hyperparameter. One of the additions of hyperparameters carried out in this study was the addition of dropouts. By adding dropouts, the resulting model is more accurate and prevents overfitting. Also, the addition of padding and stride hyperparameters makes the training process would be faster. Based on the test results, the model has been able to identify waste well to have an accuracy value of 94%. The model then inputs into the android project to be able to identify waste anytime and anywhere.

## REFERENCES

[1] D. Hoornweg, P. Bhada-Tata, and C. Kennedy, “Waste production must peak this century,” Nature, vol. 502, no. 7473, pp. 615–617, 2013.

[2] BPS, “Statistik Lingkungan Hidup Indonesia 2017,” 2017. [Online]. Retrieved from: http://www.un-ilibrary.org/economic-and-social-development/the-sustainable-development-goals-report-2017_4d038e1e-en

[3] P. Tiyajamorn, P. Lorprasertkul, R. Assabumrungrat, W. Poomarin, and R. Chancharoen, “Automatic Trash Classification using Convolutional Neural Network Machine Learning,” Proc. IEEE 2019 9th Int. Conf. Cybern. Intell. Syst. Robot. Autom. Mechatronics, CIS RAM 2019, pp. 71–76, 2019.

[4] K. Fatmawati, E. Sabna, and Y. Irawan, “Rancang Bangun Tempat Sampah Pintar Menggunakan Sensor Jarak Berbasis Mikrokontroler Arduino,” Riau J. Comput. Sci., vol. 6, no. 2, pp. 124–134, 2020.

[5] C. Chen, O. Li, C. Tao, A.J. Barnett, J. Su, and C. Rudin, “This looks like that: Deep learning for interpretable image recognition,” arXiv, no. NeurIPS, pp. 1–12, 2018.

[6] H. Yanagisawa, T. Yamashita, and H. Watanabe, “A study on object detection method from manga images using CNN,” 2018 Int. Work. Adv. Image Technol. IWAIT 2018, pp. 1–4, 2018.

[7] S. Albawi, T.A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” Proc. 2017 Int. Conf. Eng. Technol. ICET 2017, vol. 2018-Janua, pp. 1–6, 2018.

[8] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” arXiv, pp. 1–11, 2015.

[9] W. Mo, X. Luo, Y. Zhong, and W. Jiang, “Image recognition using convolutional neural network combined with ensemble learning algorithm,” J. Phys. Conf. Ser., vol. 1237, no. 2, 2019.

[10] R. Sultana, Trash and Recyclable Material Identification Using Convolutional. Carolina: Western Carolina University, 2020.

[11] G. Thung and M. Yang, “Classification of Trash for Recyclability Status Data Collection &amp; Dataset,” 2017. [Online]. Retrieved from: http://cs229.stanford.edu/proj2016/poster/ThungYang-ClassificationOfTrashForRecyclabilityStatus-poster.pdf

[12] Y. Kim, “Convolutional neural networks for sentence classification,” in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1746–1751, 2014.

[13] H. Ide and T. Kurita, “Improvement of learning for CNN with ReLU activation by sparse regularization,” in Proceedings of the International Joint Conference on Neural Networks, vol. 2017-May, pp. 2684–2691, 2017.

[14] S. Wang, T. Zhou, and J.A. Bilmes, “Jumpout: Improved dropout for deep neural networks with RELus,” in 36th International Conference on Machine Learning, ICML 2019, pp. 11565–11573, 2019.

[15] C.E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” arXiv preprint arXiv:1811.03378, 2018.