

Regular Article

Jeremy Onesimus Carnagie, Aditya Rio Prabowo*, Iwan Istanto, Eko Prasetya Budiana, Ivan Kristianto Singgih, Indri Yaningsih and František Mikšík

Technical review of supervised machine learning studies and potential implementation to identify herbal plant dataset

<https://doi.org/10.1515/eng-2022-0385>

received January 20, 2022; accepted November 13, 2022

Abstract: The use of technology in everyday life is unavoidable, considering that technological advancement occurs very quickly. The current era is also known as industry 4.0. In the industry 4.0 era, there is a convergence between the industrial world and information technology. The use of modern machines in the industry makes it possible for business actors to digitize their production facilities and open up new business opportunities. One of the developments in information technology that is being widely used in its implementation is machine learning (ML) technology and its branches such as computer vision and image recognition. In this work, we propose a customized convolutional neural network-based ML model to perform image classification technique for Indonesian herb image dataset, along with the detailed review and discussion of the methods and results. In this work, we use the transfer learning method to adopt the opensource pre-trained model, namely, Xception, developed by Google.

Keywords: machine learning, computer vision, image classification, Indonesian herb, Xception

1 Introduction

The development of information technology in the world is happening very fast. This is indicated by the increasing global market demand for computer technology products in various types/forms (i.e., smartphones, personal computers (PCs), and laptops) and the number of new research works on the topic of information technology [1]. One of the main reasons for this phenomenon is that the effective use of information technology can improve performance and enable various activities to be carried out quickly, precisely, and accurately, which will ultimately increase productivity. The new industrial era that the world is currently facing is closely related to many types of technology in various fields. This new era is called the era of the industrial revolution 4.0, where the fourth industrial revolution is usually identified with a widespread digitalization, which allows the integration of various processes that will increase the efficiency of the production process [2–5]. The industrial era 4.0 is also identified by the increase in the computer technology and internet access that is spread to various circles of society in the world, as shown by research by Kaya and Aydin in Figure 1 [6].

The sustainability of the industrial revolution 4.0 era is also closely related to the potential use of artificial intelligence (AI) technology, which is the driving force of each of the pillars of the 4.0 industrial revolution. The nine pillars of industry 4.0 cannot be separated from AI technology. And within the AI circle, there is a branch that has taken many steps of advancement so far, i.e., machine learning (ML). ML itself is a branch of AI technology designed to imitate human intelligence by studying patterns from a particular dataset. The machine will be able to predict an outcome based on the dataset pattern. The difference between the programming flow in ML and traditional programming is described in Figure 2. The relationship between the industrial revolution 4.0 and AI is illustrated in Figure 3.

* **Corresponding author: Aditya Rio Prabowo**, Department of Mechanical Engineering, Universitas Sebelas Maret, Surakarta 57126, Indonesia, e-mail: aditya@ft.uns.ac.id

Jeremy Onesimus Carnagie, Eko Prasetya Budiana, Indri Yaningsih: Department of Mechanical Engineering, Universitas Sebelas Maret, Surakarta 57126, Indonesia

Iwan Istanto: Department of Electro-Mechanical, Polytechnic Institute of Nuclear Technology, Yogyakarta 55281, Indonesia, e-mail: iwan.istanto@brin.go.id

Ivan Kristianto Singgih: Department of Industrial Engineering, University of Surabaya, Surabaya 60293, Indonesia

František Mikšík: Faculty of Engineering Sciences, Kyushu University, Fukuoka 816-8580, Japan; Institute of Innovation for Future Society, Nagoya University, Aichi 464-8601, Japan

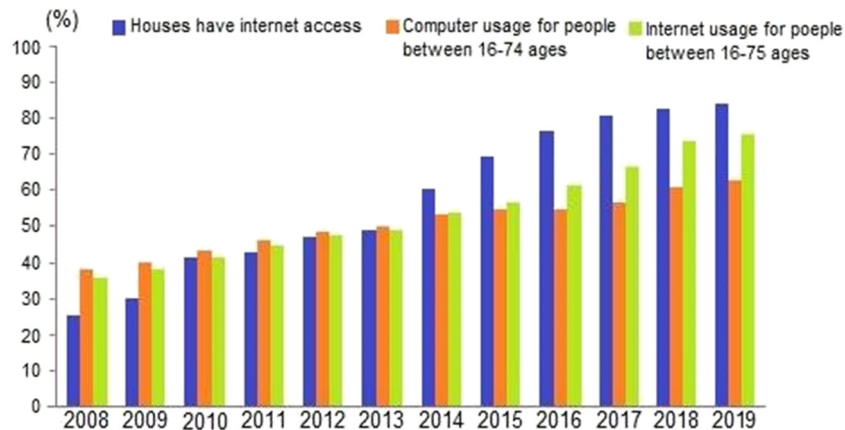


Figure 1: Percentage increase in internet and computer usage from 2008 to 2019.

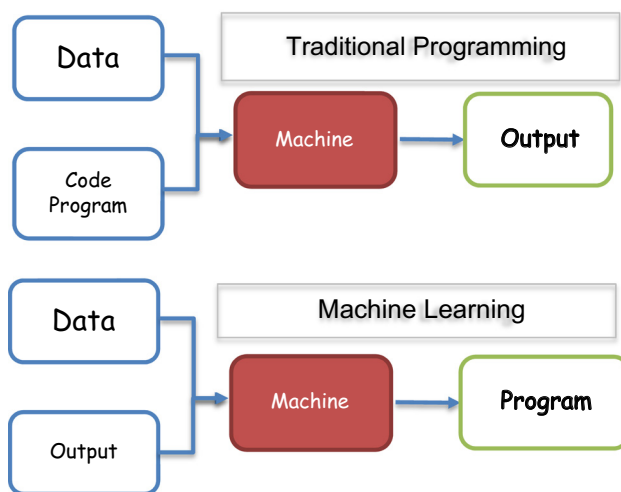


Figure 2: Flowchart of the principles of traditional programming methods (above) and ML programming (below).

In the recent times of industrial era 4.0, one of the branches of ML technology that is being widely used is the computer vision and image recognition algorithm. The AI developed by some of the latest technology companies such as Tesla's autonomous driving and facial recognition in Google Photos are both applications of computer vision and image recognition. The potential use of computer vision in industry and engineering is still considered very broad, ranging from use for quality control and sorting goods to recognizing the shape/element of material through images.

Our work in this study is building a model that adopts the algorithm of image recognition and classification to perform a classification technique using convolutional neural network (CNN) principles and supervised learning method. We have built a model that is suitable to perform image classification for particular Indonesian herbs image

dataset, which will be listed in the Section 3. The concern behind choosing Indonesian herb image dataset to be trained by our model is that this commodity is currently one of the main focuses of the development of micro, small, and medium enterprises (*Indonesian: UMKM*) for making traditional medicine, especially in the time of the global pandemic [7]. We propose this work to fill that concern in the technological aspect of performing image classification of Indonesian herbs, which potentially can help and inspire another image classification-based research on this particular topic of Indonesian herbs in the Industrial 4.0 era.

2 Review of literature

ML is becoming a superior technology that can be applied in various other fields today. Inventions and new ways of using ML continue to emerge in a fairly fast period of time. Jordan and Mitchell mentioned that ML is one of the fastest growing technical subjects today [8]. ML science has a place at the intersection of computer science and statistics and is at the core of AI and data science. Along with AI technology, ML is the main method for developing renewable technologies. Jordan and Mitchell also mentioned some examples of new developed technologies such as speech recognition, face or image recognition, object detection, natural language processing, autonomous robots, stock market price analysis, and social experimentation. These inventions and technologies have been discovered due to growing ML research works. All of the abovementioned examples of ML applications (which have only been discovered in the last few decades) open an insight into the potential of ML in the future. Mjolsness,

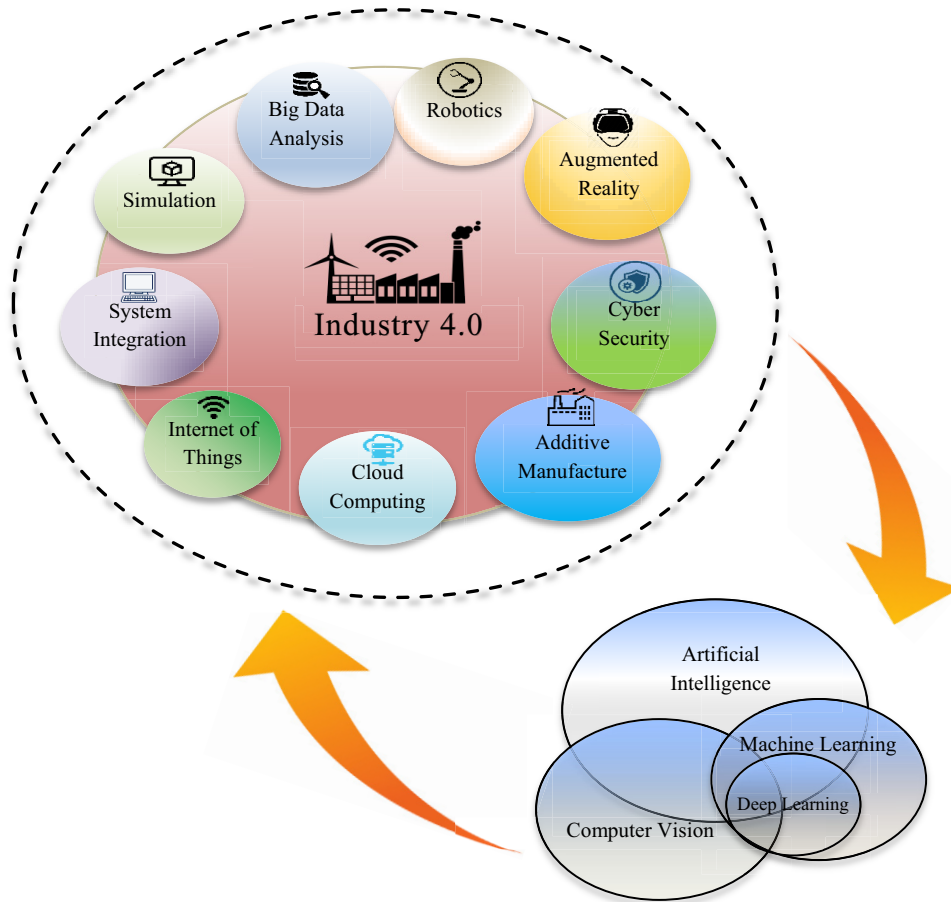


Figure 3: The relationship of the nine pillars of the industrial revolution 4.0 with AI, ML, deep learning, and computer vision.

and DeCoste stated that ML continues to develop that was proven by the deeper clarity of the mathematical approaches behind the ML processes and the increase in successful uses over the last 20 years (software or hardware) [9].

ML approaches such as data clustering, neural network classifiers, and nonlinear regression have provided very wide applications in engineering, business, and science practice. And especially from a science and research perspective, ML has the potential to strengthen the understanding and clarity in research. ML will also create an intelligent computing system with general analytic capabilities from scientific thinking. A good observation method and inferring hypotheses cannot be carried out solely by relying on human abilities when a study will involve very large volumes of data and high data acquisition rates. In this case, ML is a very powerful tool to solve those kinds of problems.

ML, according to Mahesh, is the scientific study of algorithms and statistical models that computer systems use to perform certain tasks without having to be programmed explicitly. ML is used to teach machines how

to process data more efficiently [10]. And in today's digital era, various kinds of data are available in abundance, and with the abundant availability of these datasets, the demand for using ML methods is increasing rapidly. The purpose of ML itself is to study the data, so that a program or conclusion is generated that represents the dataset. This clearly provides many breakthroughs in various activities that involve a lot of data. According to El Naqa and Murphy, ML is also a way for computers to imitate the way humans (and other living things) learn to process sensory signals/data given (input) to achieve a certain goal [11]. ML methods are seen as part of AI technology. Koza et al. also stated that ML algorithms generally build a model based on a sample of data so that the model can make predictions or decisions without having to be programmed directly and repeatedly [12].

ML relies on different algorithms to solve different data problems. ML has several types of algorithms, which, according to Ayodele, are organized into a taxonomy based on the desired results from the algorithm itself [13]. Examples of commonly known types of algorithms

include supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transduction, and learning to learn. This technology is also starting to experience rapid development and becoming one of the widely used features by modern technology-based companies, such as in Tesla cars, where the technology is used in various sides of the car in order to achieve a reasonable construction of 3D model of the environment [14], and in Apple's Siri speech recognition system which uses neural networks that converts the acoustic pattern of user's voice at each instant into a probability distribution over speech sounds [15]. In this article, we will discuss the development of image classification-based applications with supervised learning algorithms in more detail.

Supervised learning is one of the algorithms in ML where the algorithm is set to produce a function that

maps the input given to the desired output. Supervised learning is the most common technique for training neural networks and decision trees. Likewise, according to Mohammed *et al.*, the target of the supervised learning is to conclude a function or mapping from training data that have been labeled [16]. Training data consist of input vector X and output vector Y in the form of labels or tags. The label or tag of the Y vector itself is an explanation of each input example of the X input vector. This is why it is called supervised learning, which is the fact that each Y output vector already has a name or label for each training example in the training data. Each name or label has been given in advance manually by the supervisor (supervisor here can mean human or robot). This supervised learning algorithm underlies the development of image classification-based applications for herbal plants,

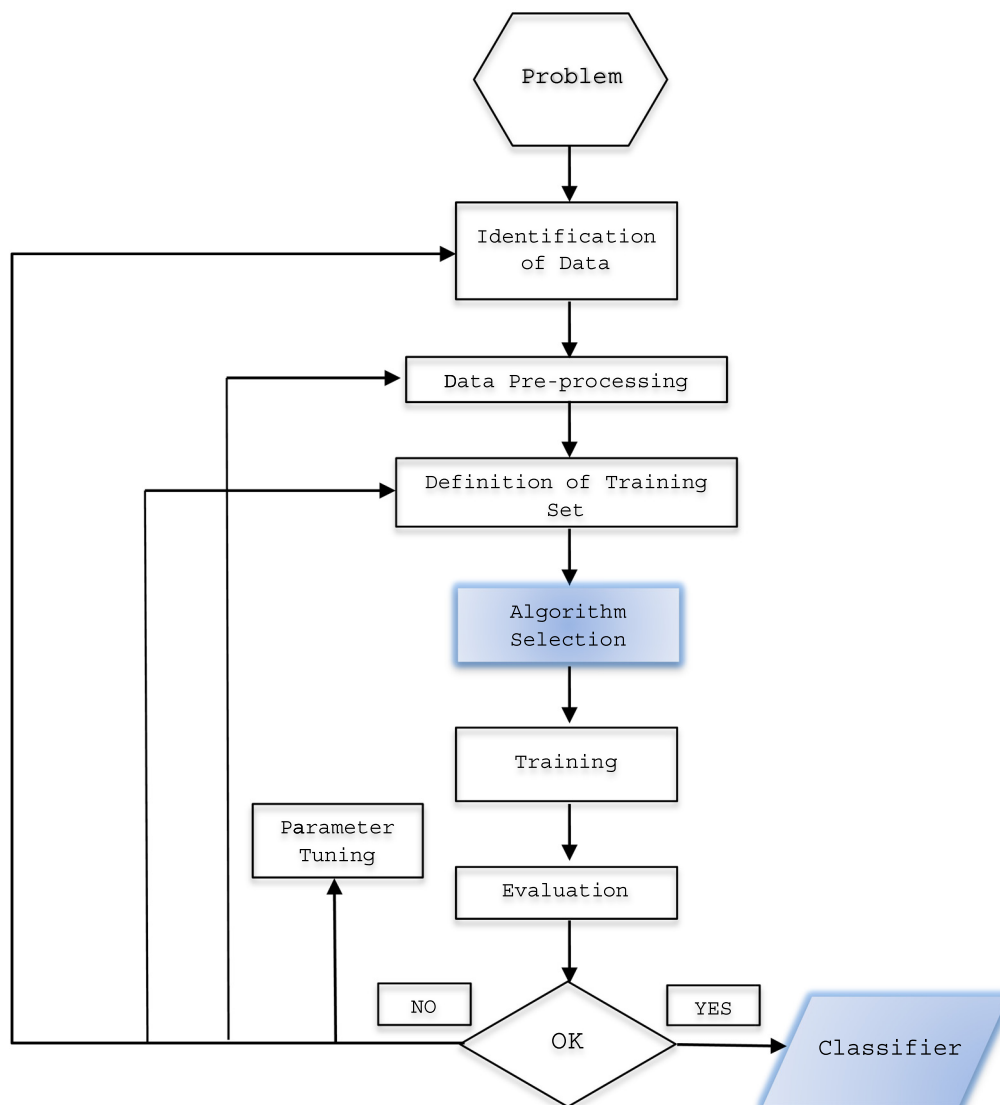


Figure 4: ML supervised process.

which will be discussed in this article. Description of a supervised learning algorithm in a flow chart is generally shown in Figure 4.

One branch of ML applications with supervised learning algorithms, which is also currently being researched and used in various fields is computer vision technology. According to Szeliski, computer vision is being widely used in very broad and diverse real-world applications, such as Optical character recognition, where machines can read handwritten letters and numbers on vehicle license plates [17]. It also can be used for inspection of certain components or parts as quality assurance, and in automotive safety system, which is used by vehicles to detect unexpected obstacles while on the road. There are many more applications of computer vision that are being intensively used in various sectors. Computer vision is a field of science that discusses how computers can understand various features that exist in an object in an image or video. Huang states, from the point of view of engineering science, that computer vision is aimed at building automated systems capable of performing tasks that the human visual system can perform [18]. A flow diagram illustration of the computer vision process is depicted in Figure 5.

Like ML principles in general, a machine/computer requires a collection of data to be able to learn the data and patterns to be analyzed, as well as computer vision. The dataset needed for computer vision techniques is in the form of a digital image of an object to be analyzed. There are several official websites available on the internet that provide various digital image datasets to train computer vision machines, such as Kaggle, ImageNet, and opensource.google. The tasks performed by machines in

performing computer vision, according to Klette, include methods comprising acquiring, processing, analyzing, and understanding a digital image, as well as extracting dimensional datasets from the real world to then produce numerical or symbolic information that the computer itself can understand [19]. Applications of the AI can also be found in road transport, especially when planning routes in autonomous driving [20–23], logistic flows [24], industrial engineering and processing [25–27], and robotization in manufacturing [28,29].

3 Methodology

The methodology used for making machine vision features of the Herbyfy application consists of three parts: (1) searching for digital image datasets, (2) creating programming scripts, and (3) training and validation processes. In training the computer to understand the dataset, the computer performs the algorithm specified in the programming script to be created. These three activities are carried out in online cloud media, namely, Google Colab and Google Cloud Platform (GCP). The step-by-step processes is explained generally in Figures 6 and 7.

3.1 Dataset creation (data harvest)

The dataset required is a collection of digital images of various kinds of herbal plants. Dataset search experiments have been carried out on open-source sites that

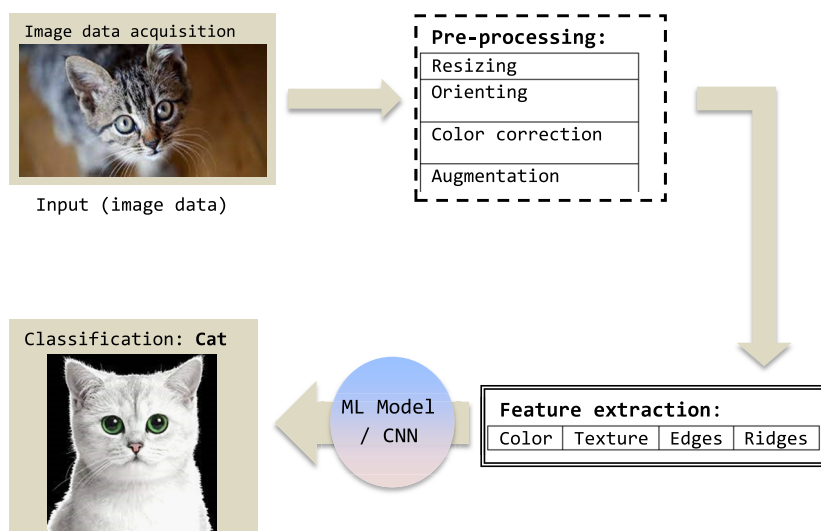


Figure 5: Illustration of the process flow of acquiring, processing, analyzing, and understanding in computer vision.

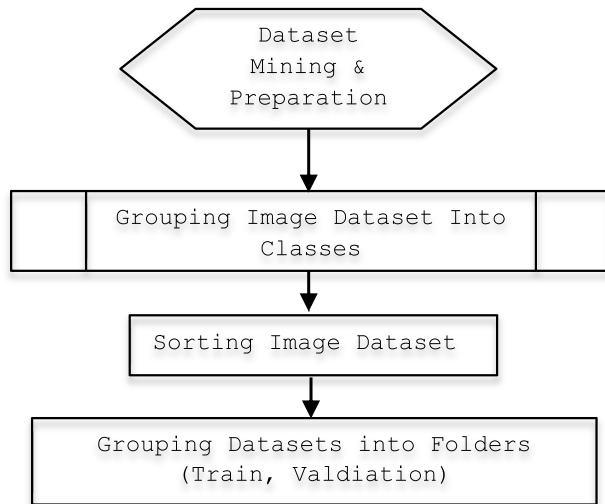


Figure 6: Flow chart of dataset making process.

provide digital image datasets (Kaggle, ImageNet, and opensource.google), but the desired results were not obtained. Thus, the search process for digital image datasets was carried out by downloading pictures of herbal plants directly from the internet, such as ginger, turmeric, garlic, etc. Downloading digital images is done using a Java programming script, which is run on an internet browser console. The script will download a text file (.txt) containing the URLs of the digital images that is available in the browser. Then, each image based on the URL that has been obtained in the .txt file will be downloaded by running the following Python programming script at the command prompt or other Python interpreter application. The snippet of the Java programming script is shown as follows [30]:

JAVASCRIPT

```

Function simulateRightClick(element) {
  var event1 = new MouseEvent('mousedown', {
    bubbles: true,
    cancelable: false,
    view: window,
    button: 2,
    buttons: 2,
    clientX: element.getBoudingClientRect().x,
    clientY: element.getBoudingClientRect().y
  });
  :
  :
  grabUrls().then(function(urls) {
    urls = urls.join( '\n');
    createDownload(urls);
  });
}

```

PYTHON

```

from imutils import paths
import argparse
import requests
import cv2
import os
import numpy as np

ap = argparse.ArgumentParser()
ap.add_argument("-u", "-urls", required = True,
                help = "path to file containing image
                URLs")
ap.add_argument("-o", "-output", required = True,
                help = "path to output directory of
                images")
:
:
if __name__ == "__main__":
  args = vars(ap.parse_args())
  output_dir = args["output"]
  is_verbose = args["verbose"]
  start = int(args["start"])

  urls = read_url_list(args["urls"])
  url_data = read_urls(urls, is_verbose)
  images = return_images(url_data, is_verbose)
  write_images(images, output_dir, start, is_verbose)

```

After the script is executed, all digital images that have been searched from the Google Image page will be automatically downloaded to the specified folder. All the downloaded images will be needed to create a computer vision dataset on the ML programming script that will be created. Digital images of herbal plants that the Herblabs.ai team was looking for to use as a ML dataset include (in Indonesian): *adas*, *akar alang-alang*, *bawang putih*, *daun katuk*, *daun salam*, *jahe*, *kumis kucing*, *lidah buaya*, *mengkudu*, *meniran*, *miana*, *pare*, *pegagan*, *rosela*, *seledri*, *temu kunci*, and *valerian*.

The images will then be categorized into folders on the cloud storage. This grouping also plays a role in ML methods that will directly categorize a dataset based on its storage folder. By managing datasets in structured folders, it is possible for the computer to directly load sample images according to the needs of the learning process. This will slow down the learning process significantly but the number of datasets that can be managed will be maximized according to the available storage capacity [31]. Under these conditions, ML can be carried out for various types of datasets. Before carrying out the ML process, the dataset must first be sorted, because the

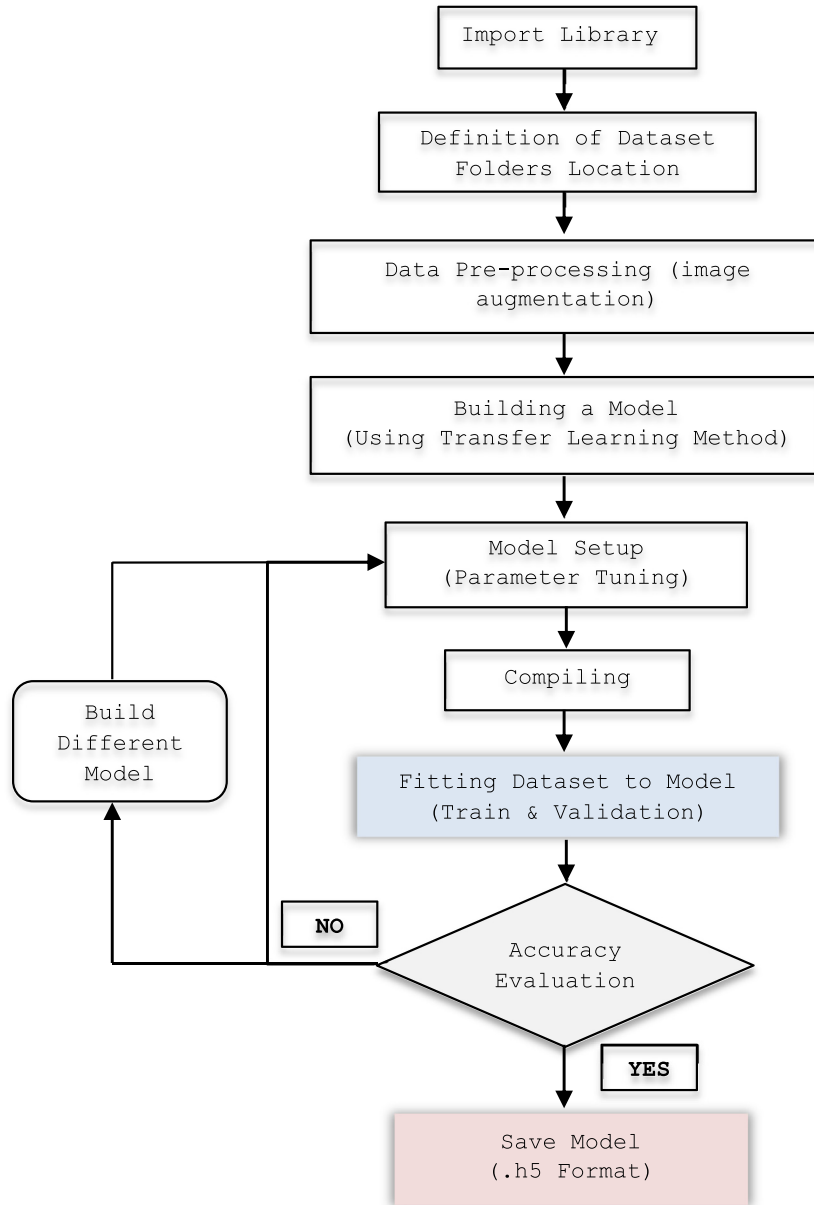


Figure 7: Flow chart of programming and training process.

image search results on Google Image also contain images that do not match the desired image. The number of datasets obtained was 8,706 digital images, with the number of datasets for each plant presented in Table 1.

3.2 ML engineering

Computer programming for ML is done using the Python programming language, with version 3.0 and above. Some interpreters that can be used to write programming

scripts include Google Colab, Jupyter Notebook, and Microsoft Visual Studio Code. Meanwhile, the dataset storage directory is placed in cloud storage on the GCP. The first programming code written is to import any libraries that we will use. The libraries used are libraries that are already available in Python version 3.0.0 and above, including: NumPy, Pandas, TensorFlow, OS, Matplotlib, and Keras. The Python programming code script to import the entire library is written as follows:

```

import numpy as np
import pandas as pd
import tensorflow as tf

```

Table 1: Number of train and validation datasets for each herb type

Herb type (class)	Train dataset	Validation dataset
Fennel	353	26
Reeds herb	389	25
Garlic	387	40
Katuk (<i>Sauropus androgynus</i>)	430	51
Bay leaf	346	28
Ginger	333	38
Cardamom	422	35
Kumis Kucing (<i>Orthosiphon aristatus</i>)	424	49
Turmeric	418	32
Aloe vera	473	55
Noni (<i>Morinda citrifolia</i>)	417	32
Meniran (<i>Phyllanthus urinaria</i>)	390	29
Coleus	416	52
Bitter melon	416	43
Gotu kola	370	23
Roselle	431	47
Celery	337	34
Lemongrass	346	13
Fingerroot	453	38
Valerian root	429	36
TOTAL	7,980	726

```
import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

```
from keras import Sequential
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import SGD, Adam
from keras.layers import Flatten, Dense, BatchNormalization, Activation, Dropout
```

```
from keras.utils import to_categorical
```

The next step is to augment image data, using the image preprocessing method. Performing this method begins by specifying the directory path of the dataset in cloud storage. In this case, the dataset is stored in the “dataset/train” directory in the cloud storage. The next step is to create a variable for the directory of the owned dataset. The directory variables are divided into two: (1) the training directory, the directory for datasets intended for training on machines, and (2) the validation directory, which is the directory containing datasets intended to validate the results of training. The datasets in each directory are augmented using the following programming script.

```
TRAINING_DIR = 'dataset/train'
training_datagen = ImageDataGenerator(
    rescale = 1/255,
```

```
    rotation_range = 2,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    zoom_range = 0.3,
    horizontal_flip = True
```

```
)
```

```
VALIDATION_DIR = 'dataset/val'
```

```
validation_datagen = ImageDataGenerator(rescale =
1/255)
```

The next step is to create a generator variable on the image data that has been augmented in the previous programming script. A data generator is needed to generate image data in the dataset folder. The generator variable is also divided into two: (1) the train generator for the training dataset and (2) the validation generator for the validation dataset. The programming script is written as follows:

```
train_generator = training_datagen.flow_from_directory(
    TRAINING_DIR,
    target_size = (299, 299),
    class_mode = 'categorical'
)
```

```
validation_generator = validation_datagen.flow_from_directory(
    VALIDATION_DIR,
    target_size = (299, 299),
    class_mode = 'categorical'
)
```

The flow from the directory method is useful when digital image datasets have been sorted and placed in their respective class/label folders. This method will identify the class automatically from the folder name. This method is also separated into two, namely, for training and validation.

The next step is to transfer learning a neural network model that has been created. Transfer learning is one of the methods in ML programming that aims to adopt a model that has been created or is commonly called a pre-trained model that is open-source or free to use by anyone. In this work, the team uses a pre-trained model called Xception, a pre-trained neural network model available in the Keras library. Xception is a CNN with 36 convolutional layers [32]. Xception has a total of 7 configurable arguments, such as include_top, weights, input_tensor, input_shape, pooling, classes, and classifier_activation. The programming script for doing transfer learning on the Xception model is written as follows:

```
from tensorflow.keras.applications.xception import
Xception
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dropout, Dense,
GlobalAveragePooling2D
```

```
from tensorflow.keras.optimizers import Adam
```

Then, adjustments are made to the available arguments in the Xception model as follows. Here a variable named base model is created as a variable containing the pre-trained Xception model and the argument settings.

```
base_model = Xception(weights = 'imagenet', input_shape = (299,299,3), include_top = False, pooling = 'avg')
```

At the end of the Xception neural network, it is necessary to add a connecting node that is activated with the “ReLU” activation function as a connecting layer from the previous neural network from Xception, and the last node as an output node according to the number of classes that have been classified, which is 20, then the script is written as follows:

```
layer = base_model.output
# adding fully-connected layer
layer = Dense(256, activation = 'relu')(layer)
# adding output layer with 20 classes
predictions = Dense(20, activation = 'softmax')(layer)
```

Next create variables for the model and optimizer. The model is the entire neural network that will be trained using the existing dataset. The optimizer used here is the “Adam” optimizer, which stands for adaptive moment estimation. The team chooses Adam optimizer for the model because adam is a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement [33]. With various and repeated trial of settings upon the methods of Model

and Adam, the model and optimizer variables are written with the following programming script.

```
model = Model(inputs = base_model.input, outputs = predictions)
opt = Adam(lr = 0.001, decay = 1e-6, beta_1 = 0.9, beta_2 = 0.999, amsgrad = False)
```

Another thing to be reminded, the first layer that must be trained later on is the top layer (Xception layer itself), since that layer is the one that has the convolution function. Therefore, the programming script for setting it up is as follows:

```
for layer in base_model.layers:
    layer.trainable = False
```

And then, it will be compiled on the model that has been created.

```
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

From here, the neural network model can be trained using the fit method, which is filled with arguments such as train generator, epoch, data validation, validation generator, and verbose, as written in the following script.

```
model.fit(train_generator, epochs = 4, validation_data = validation_generator, verbose = 1)
```

At this point, the top layer is well trained and can begin fine-tuning the convolution layer of Xception [34]. The team chose to fine-tune with the freeze layer method, which is the first 125 layers, and unfreeze the rest to train. Freezing layers in the context of a neural network is all about controlling layer weights. When a layer is frozen, its weight cannot be further modified during the training process. This freezing technique also aims to reduce computational time during the training process without

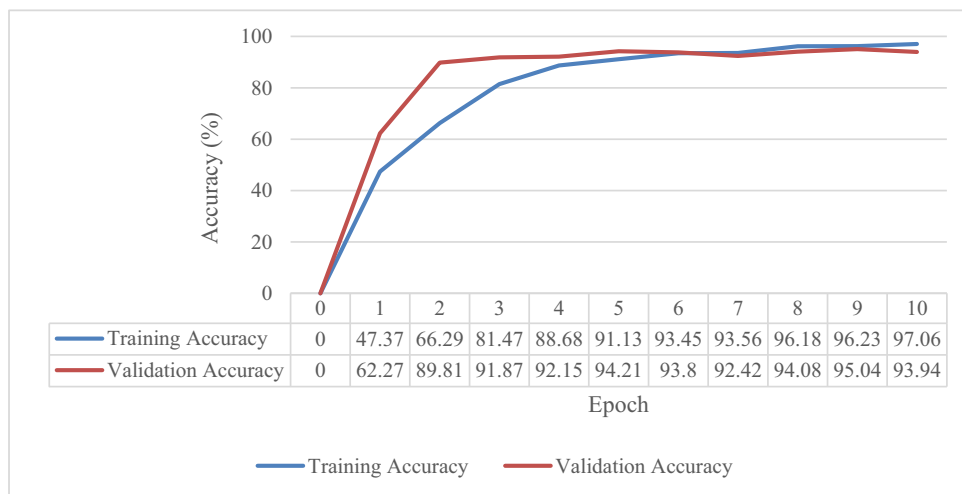


Figure 8: Plotting of the accuracy of training (training) and validation (validation) on 10 epochs using the pre-trained Xception model.

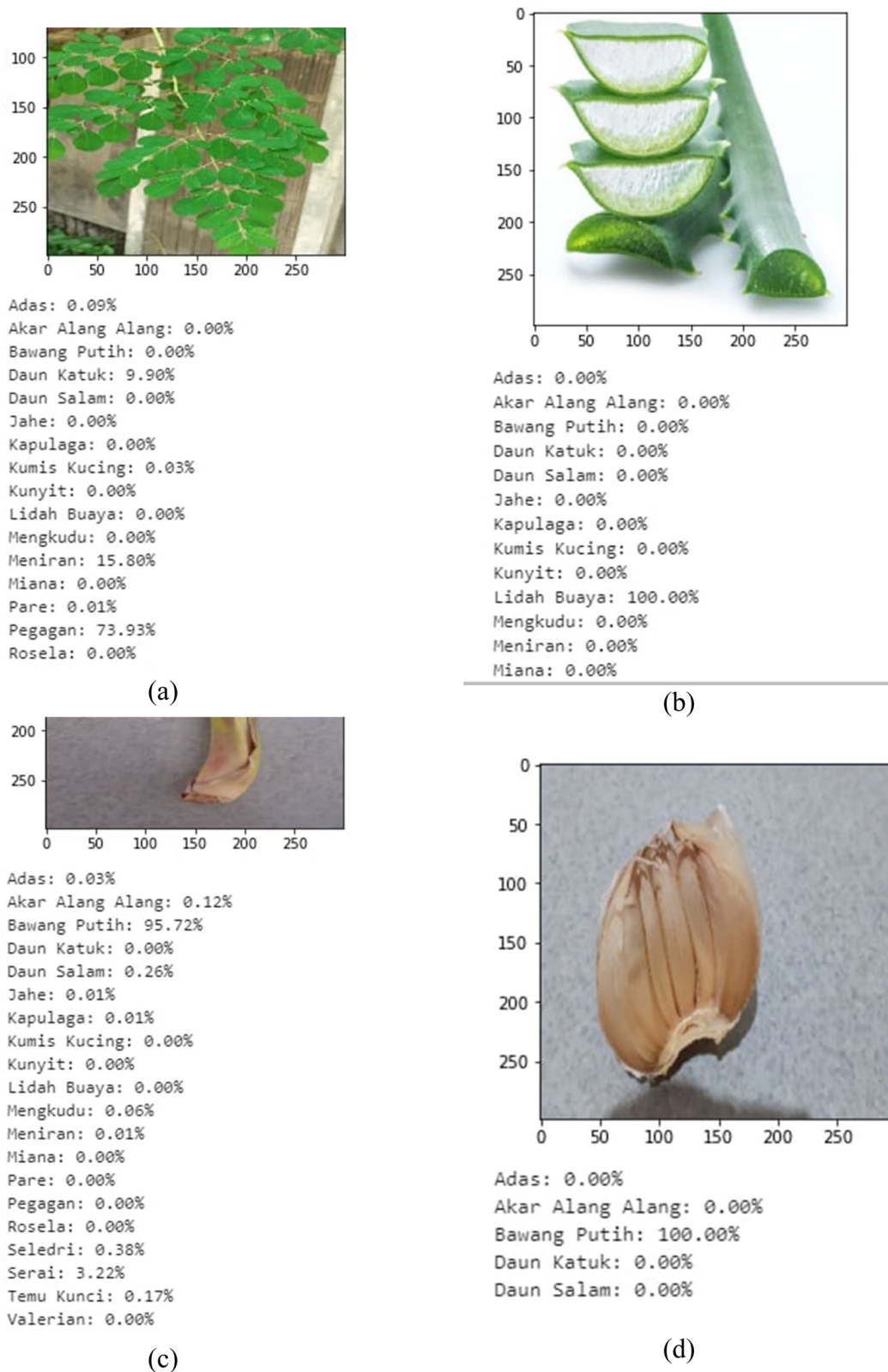


Figure 9: The results of the confidence model number with the input of testing images: (a) Pegagan leaves, (b) Aloe vera, and (c and d) Garlics.

reducing machine accuracy. The programming script to perform the freezing technique on the first 125 layers is as follows:

```
for layer in model.layers[:125]:
    layer.trainable = False
for layer in model.layers[125:]:
    layer.trainable = True
```

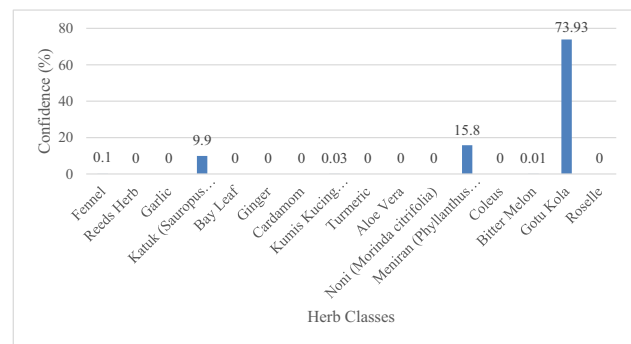
After fine-tuning with the freezing layer is done, the neural network model will be recompiled so that the modifications from the fine-tuning can be applied to the model, with the following script:

```
model.compile(optimizer = opt, loss = 'categorical_
crossentropy', metrics = ['accuracy'])
and do the training one more time for the last.
history = model.fit(train_generator, epochs = 10, vali-
dation_data = validation_generator, verbose = 1).
```

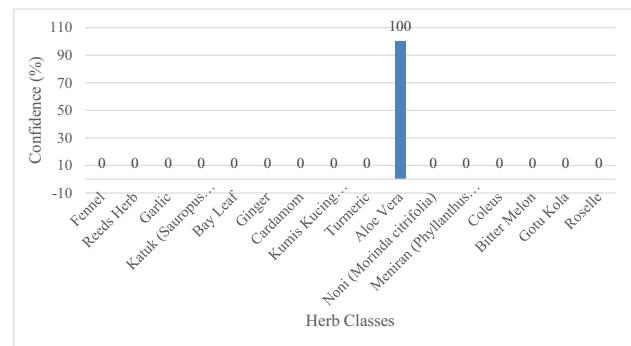
4 Results and discussion

After the programming code in the methodology section is executed, the machine/computer will enter into the training process for the specified dataset. This learning process involves two aspects of accuracy that will be used as parameters for whether the machine/computer is well trained or not, namely, training accuracy and validation accuracy. And after all the epochs have been passed (the dataset cycle enters the neural network), the team tries to plot the training accuracy and validation for each epoch. As shown in Figure 8, the training and validation results carried out using the Xception model show a good quality of accuracy. The results obtained are training accuracy of 0.97 or 97% and validation accuracy of 0.93 or 93%.

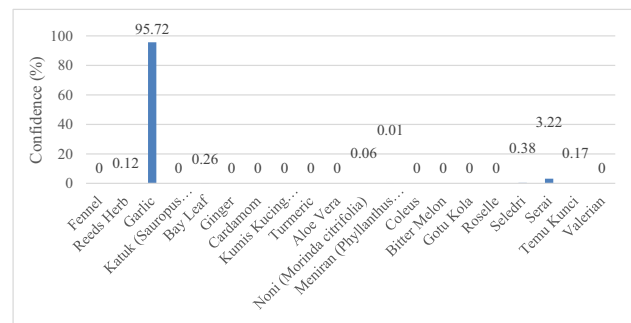
After training on the created model, the model is saved in the .h5 program format, which will then be installed on the developed Android program. The classification results of the herbal plants with the Xception model show good performance quality. The team tested the ML model by entering pictures of herbal plants at random, and then the model would classify the images as one of the 20 predefined classes. The parameters of how the model classifies images in the dataset are based on the level of confidence or the number of the model's confidence in the analyzed images. Figure 9 shows the level of confidence of the model in classifying pictures of herbal plants, including pictures of gotu kola leaves, aloe vera, and two pictures of garlic, with the spread of the confidence model figures, which can be seen in Figure 10.



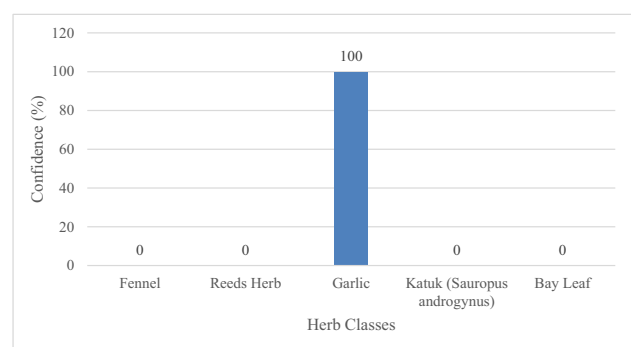
(a)



(b)



(c)



(d)

Figure 10: Graph of the percentage of the model's confidence in the analysis of images: (a) Pegagan leaves, (b) Aloe vera, and (c and d) Garlics.

It can be seen that the model has a fairly good level of confidence in classifying Pegagan leaves, although there are also confidence figures in other classes of herbal plants, namely Adas at 0.09%, Katuk Leaf 9.90%, Kumis Kucing at 0.03%, Meniran at 15.8%, and Pare 0.01%. The reason for the emergence of confidence figures in other classes of herbal plants is thought to be caused by the possibility of the similarity of colors, shapes, and background images of the leaves. This will serve as material for the team's evaluation that the digital image dataset needs to be trimmed, which in this case is done by removing the background image from the digital image dataset that will be trained on the model. Another test was carried out using an image of an aloe vera leaf, but with a digital image that has the characteristics of an image with a white background (without any other color or shape behind the object of analysis). It turns out that the perfect model confidence number is 100%, which means that the model confidently classifies the inputted digital image as Aloe vera leaf, as shown in Figure 10. Then, two different images of garlic were also tested, but with the same background, and accurate analysis results were obtained, with confidence figures for each garlic image of 95.72 and 100%. Based on these figures, the suspicion about the effect of the background image on the object to be analyzed is getting stronger. This can be a subject for further discussion, especially with the quality of the dataset currently owned by the team.

The current research can be stated to have more variations in terms of herbal plants sample, i.e., color, shape, etc., compared to pioneer works which focused on the identification of leaf type and geometry. The works were introduced by Jackulin and Murugavalli [35] who studied the detection performance of plant disease using ML and deep learning methodologies. The samples were taken to make the machine to understand the difference between healthy leaf and diseased leaf images. It was found from this work that the performance of Inception v4 (48 layered deep CNN network which is the ImageNet model extension) reached almost 100% for training accuracy and approximately 98% for the validation accuracy. On the other hand, another considered program, VGG 16, which is known as visual geometry group, produced around 83% training accuracy, and 81–82% validation accuracy. Similar research works are found which is related to plant identification by ML, e.g., Roopshree *et al.* who designed an IoT based authentication system for therapeutic herbs using ML [36], Chen *et al.* who described an easy method for identifying commonly-used Chinese herbal medicines using AutoML platforms [37], and Xu *et al.* who developed multiple attention

pyramid networks for herbal recognition, which considered Species and Classes in the Plant Kingdom of the plant samples [38]. More details based on the concept of ref. [38] can be projected as good development opportunity for the current ML platform in this research.

5 Conclusion

The results of the analysis that have been carried out and the model's achievements in learning the datasets are summarized in the following conclusions:

1. Python programming code in the methodology can be run completely without the appearance of error code.
2. The Xception model is able to study Herbify's herbal plant dataset with a fairly high training and validation accuracy, which is in the range of 97% for training accuracy and 93% for validation accuracy.
3. Although the obtained accuracy seems to show a fairly high number, yet it still needs further evaluation whether there is overfit occurrence or not.
4. The model is able to recognize digital images that are inputted to be tested according to the classes that have been categorized, with a fairly good confidence number in each test.

With this achievement, it is hoped that it can convince the reader that the training carried out using the existing methods can provide the final result of a model that can classify herbal plants according to their respective classes.

The author also wants to give recommendations to readers that there are still opportunities for further research, especially in the aspect of evaluation and improvement. Given the analytical capacity possessed by the pre-trained Xception model with some additional programming, it is not impossible to create other models that are capable of analyzing more classes of plants, or perhaps also analyzing other types of datasets, such as animals, automotive components, and so on.

Several of the challenges that can be concluded from this research are the fact that there is a need for devices with proficient capabilities to perform fast analysis and sufficient repetition (epochs), because the more the datasets to be analyzed, the greater the weight or burden of the data that the computer must analyze. Another challenge was also found in the stage of image dataset preparation, where an image dataset can be said to be good enough if the image has been trimmed from other objects surrounding the image itself. This preparation obviously will take a lot of time and effort to be done. And last but

not least, another challenge that can be concluded is in terms of parameter tuning, both at the image augmentation and modeling stages, which is expected to obtain a good result of accuracy without the occurrence of overfitting.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The authors declare that all data supporting the findings of this study are available within the article.

References

- [1] Haines VY, Lafleur G. Information technology usage and human resource roles and effectiveness. *Hum Resour Manag.* 2008;47:525–40.
- [2] Hamrol A, Gawlik J, Śladek J. Mechanical engineering in Industry 4.0. *Manag Prod Eng Rev.* 2019;10:14–28.
- [3] Prabowo AR, Tuswan T, Ridwan R. Advanced development of sensors' roles in maritime-based industry and research: from field monitoring to high-risk phenomenon measurement. *Appl Sci.* 2021;11:3954.
- [4] Kumar S, Asjad M, James AT, Suhaib M. Development of an industry 4.0 transformability index for manufacturing systems. *Ind Rob.* 2021;49:512–26.
- [5] Achouch M, Dimitrova M, Ziane K, Karganroudi SS, Dhoubi R, Ibrahim H, et al. On predictive maintenance in industry 4.0: overview, models, and challenges. *Appl Sci.* 2022;12:8081.
- [6] Kaya A, Aydin Ö. E-commerce in Turkey and SAP integrated e-commerce system. *Int J eBus eGovern Stud.* 2019;11:207–25.
- [7] Kementrian Koordinator Bidang Perekonomian Republik Indonesia; 2020. <https://www.ekon.go.id/publikasi/detail/493/konsumsi-jamu-meningkat-pemerintah-berdayakan-umkm-dengan-pendekatan-klastr-obat-tradisional>.
- [8] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science.* 2015;349:255–60.
- [9] Mjolsness E, DeCoste, D. Machine learning for science: state of the art and future prospects. *Science.* 2001;293:2051–5.
- [10] Mahesh B. Machine Learning Algorithms-A Review. *Int J Sci Res.* 2020;9:381–6.
- [11] El Naqa I, Li R, Murphy MJ. Machine learning in radiation oncology - theory and application. *Handel, Switzerland: Springer;* 2015.
- [12] Koza JR, Bennett F, Andre D, Keane A. Artificial Intelligence in Design '96. Heidelberg, Netherlands: Springer; 1996.
- [13] Ayodele TO. New advances in machine learning. Portsmouth, United Kingdom: University of Portsmouth; 2010.
- [14] Ingle S, Phute M. Tesla Autopilot: Semi Autonomous Driving, An Uptick For Future Autonomy. *Int Res J Eng Technol.* 2016;3:369–72.
- [15] Siri Team. Hey siri: An On-Device DNN-Powered Voice Trigger For Apple's Personal Assistant. *Apple Mach Learn J.* 2017, <https://machinelearning.apple.com/research/hey-siri>.
- [16] Mohammed MMZE, Khan MB, Bashier EBM. Machine learning: algorithms and applications. Florida, United States: CRC Press; 2016.
- [17] Szeliski R. Computer vision: algorithms and applications. Washington, United States: Springer Science & Business Media; 2010.
- [18] Huang TS. Computer vision: Evolution and promise. CERN European Organization for Nuclear Research-Reports-CERN; 1996. p. 21–6.
- [19] Klette R. Concise computer vision. London, United Kingdom: Springer; 2014.
- [20] Nieoczym A, Caban J, Stopka O, Krajka T, Stopková M. The planning process of transport tasks for autonomous vans. *Open Eng.* 2021;11:871–9.
- [21] Caban J, Nieoczym A, Dudziak A, Krajka T, Stopková M. The Planning Process of Transport Tasks for Autonomous Vans—Case Study. *Appl Sci.* 2022;12:2993.
- [22] Czech P, Turoń K, Barcik J. Autonomous vehicles: Basic issues. *Sci J Silesian Univ Technol Ser Transp.* 2018;100:15–22.
- [23] Misztal W. The Impact of Perturbation Mechanisms on the Operation of the Swap Heuristic. *Arch Automot Eng – Archiwum Motoryzacji.* 2019;86:27–39.
- [24] Brumerčikova E, Bukova B, Rybicka I, Drożdziel P. Measures for Increasing Performance of the Rail Freight Transport in the North-South Direction. *Komunikacie.* 2019;21:13–20.
- [25] Kavka L, Dočkalíková I, Čujan Z, Fedorko G. Technological and Economic Analysis of Logistic Activities in Interior Parts Manufacturing. *Adv Sci Technol Res J.* 2020;14:204–12.
- [26] Antosz K, Pasko L, Gola A. The use of artificial intelligence methods to assess the effectiveness of lean maintenance concept implementation in manufacturing enterprises. *Appl Sci.* 2020;10:7922.
- [27] Nieoczym A, Caban J, Marczuk A, Brumerčik F. Construction design of apple sorter. *BIO Web Conf.* 2018;10:02025.
- [28] Anczarski J, Bochen A, Głab M, Jachowicz M, Caban J, Cechowicz R. A method of verifying the robot's trajectory for goals with a shared workspace. *Appl Comput Sci.* 2022;18:37–44.
- [29] Tucki K, Oryncz O, Wasiak A, Gola A, Mieszkalski L. Potential routes to the sustainability of the food packaging industry. *Sustainability.* 2022;14:3924.
- [30] Sollami M. <https://sollami.ai/code/2017/1/3/google-image-scraper>, 2017.
- [31] DeConinck E, Bohez S, Leroux S, Verbelen T, Vankeirsbilck B, Simoens P, et al. DIANNE: A modular framework for designing, training and deploying deep neural networks on heterogeneous distributed infrastructure. *J Syst Softw.* 2018;141:52–65.
- [32] Chollet F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition.* Hawaii, United States: 2017.
- [33] Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv.* 2014. arXiv:1412.6980.
- [34] Carnagie JO, Prabowo AR, Budiana EP, Singgih IK. Essential oil plants image classification using Xception model. *Procedia Comput Sci.* 2022;204:395–402.

- [35] Jackulin C, Murugavalli S. A comprehensive review on detection of plant disease using machine learning and deep learning approaches. *Meas: Sens.* 2022;24:100441.
- [36] Roopshree S, Anitha J, Mahesh TR, Kumar VV, Viriyasitavat W, Kaur A. An IoT based authentication system for therapeutic herbs measured by local descriptors using machine learning approach. *Measurement.* 2022;200:111484.
- [37] Chen W, Tong J, He R, Lin Y, Chen P, Chen Z, et al. An easy method for identifying 315 categories of commonly-used Chinese herbal medicines based on automated image recognition using AutoML platforms. *Inf Med Unlocked.* 2021;25:100607.
- [38] Xu Y, Wen G, Hu Y, Luo M, Dai D, Zhuang Y, et al. Multiple attentional pyramid networks for Chinese herbal recognition, *Pattern Recognit.* 2021;110:107558.