Road Flaw Detection using CNN and Transfer Learning Techniques

Vishakha Chourasia[1*], Satvik Tejas[2], Vishal Kakede[3]
[1]vishakhachourasia23@gmail.com (* Corresponding author),
[2]satviktejas@gmail.com, [3]vishalkhekade25@gmail.com
Department of Information Technology
Indian Institute of Information Technology Bhopal, India

Abstract:- Road cracks can have very dangerous results. Road flaw detection is a crucial aspect of ensuring road safety and maintenance. Identifying cracks and taking the necessary steps are requirements for preventing such dangers. Manual approaches are costly and time consuming. Using pricey laser equipment vehicles for operation and the associated upfront costs is also not very feasible. This paper proposes an automation-based approach to identify various types of flaws like potholes, cracks, and patchy terrain on road using machine learning techniques. The convolutional neural network (CNN) and transfer learning techniques are utilized to carry out the mentioned work. To get the precise features from the image data, first the data is preprocessed and then transfer learning is applied using different pre-trained CNN models. Thereafter, to accurately detect the road flaws, the model is fine-tuned using the transfer learning and the dataset collected by us. The proposed approach is evaluated on metrics such as precision, recall, and F1 score. The performance results show that the proposed approach performs well while achieving an F1 score of 95%. Therefore, reducing the risk of accidents and improving road safety.

Keywords: road-flaws, safety, transfer learning, CNN

1. Introduction

Roads are a critical component of any transportation infrastructure and serve as a lifeline for modern society. However, poor road conditions can lead to accidents, damage to vehicles, and even loss of life. Therefore, it is essential to ensure the proper maintenance and upkeep of roads. One way to maintain road safety is through the timely detection of road flaws such as potholes, cracks, and uneven surfaces. Traditional methods of detecting road flaws require human intervention and can be time-consuming and expensive. In recent years, machine learning (ML) algorithms have shown promising results in automated road flaw detection [1, 2].

The objective of this work is to develop a system that could detect road flaws automatically using AI and ML algorithms. The work focuses on developing the algorithm to detect road flaw which can be deployed using a robot or a mobile van with a camera to detect road flaws. We accomplished it by using convolutional neural networks (CNN) and transfer learning techniques to train the proposed model on a dataset of road flaw images collected by us. The employment of transfer learning in this research work is inspired by the fact that involves leveraging pre-trained models to solve new problems with limited data [3]. Transfer learning has recently been very popular for solving ML problems. However, it has not yet been applied for flaw or crack detection problems.

The work is motivated by the following heads: first, the automated road flaw detection approach can help to reduce the risk of accidents and improve the road safety. Second, traditional methods of detecting road flaws can be time consuming and costly. Third, ML techniques can be helpful in proving faster and more cost-effective solutions. The proposed work demonstrates the effectiveness of using transfer learning techniques in training accurate models with limited data and to develop a system that can help authorities in identifying and repairing road flaws in a timely manner, thus reducing the risk of accidents and improving road safety. The major contributions of the work are as follows:

(1) A system is developed that can detect road flaws automatically using ML algorithms.

(2) A wide range of CNN models and transfer learning techniques are used to train the model on a dataset of road flaw images.

(3) The proposed approach can help the authorities in identifying and repairing road flaws in a timely manner, thus reducing the risk of accidents and improving road safety.

2. Related Work

Automated road flaw detection has gained significant attention in recent years, with several studies proposing different approaches and techniques to address this problem. Some of the most relevant works in this area are described in this section.

Lei Zhang *et al*. in 2016, proposed a method to check road surface defects using deep CNN. The authors trained the deep CNN model on a dataset of 500 road images of size 3264 × 2448 and achieved an accuracy of around 87% [4]. An approach for classification of asphalt pavement cracks using support vector machine (SVM) and Otsu's methods was proposed by Yuslena Sari et al. [5]. Detection of potholes using you only look once (YOLO) algorithm was developed by Mohd. Omar et al. in 2020 [6]. The review article of Y. Hamishebahar et al. details about the deep learning methods used for road surface cracks detection [7]. The application of

computer vision methods for traffic analysis in urban areas was reported by Norbert Buch et al. their survey paper [8].

The reported ML-based studies demonstrated the effectiveness of using different ML algorithms, particularly CNN, for automated road flaw detection. They have also highlighted the potential of these techniques for improving road safety and reducing the risk of accidents. However, there is still a need for further research in this area, particularly in developing more accurate and robust models that can handle diverse road conditions and environments.

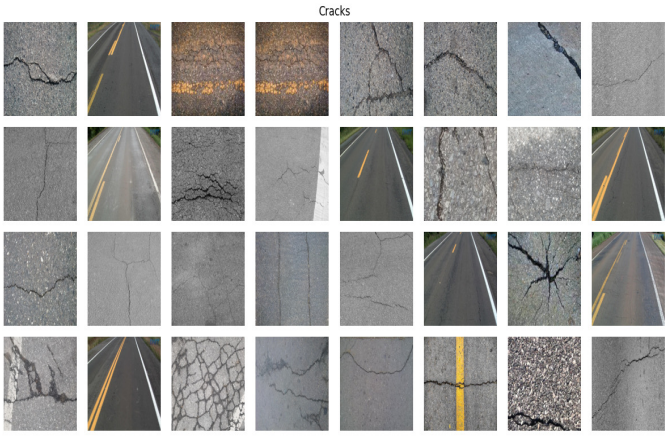
3. Materials and Methods:

This section outlines the dataset and its collection method in section 3.1. Section 3.2 presents the preprocessing and data-splitting strategy. The overall methodology of the proposed work is described in section 3.3.

3.1 Dataset

Data collection and preprocessing is a crucial step in the development of an ML model for road flaw detection. In this step, we have collected a diverse dataset of road images and prepared it for use in training the ML model. The collected dataset includes images of roads with various types of flaws such as potholes, cracks, and uneven surfaces. The dataset contains total of 4535 images of different spatial resolutions (the size is more than 224×224). The data set contains color images. It is large enough to ensure that the model is not overfitting to the training data. The images in the dataset are captured using different devices such as cameras and drones. Figure 1 below describes the sample images from the dataset.

(a)



(b)

Figure 1: (a) Sample Grooves images, and (b) cracks images collected by us for the training of road flaw detection models.

The collected images in the dataset cropped to display only the road flaw such as cracks and grooves, removing the non-essential elements such as cars, pedestrians, signs, horizon, grass, etc. Accurate cropping is done to ensure that the model learns to detect road flaws correctly.

3.2 Data preprocessing and splitting

In this step, we have prepared the dataset for use in training the model by applying various preprocessing techniques. The images were resized to a standard size of 224×224 to reduce the computational cost of training. The pixel values of the images were normalized to a standard range to improve model performance. Data augmentation techniques such as rotation, flipping, and cropping were applied to increase the size and diversity of the dataset. The preprocessed data was split into training, validation, and testing sets. As described in Figure 2, out of total 4535 images 907 (20%) were kept for test. The rest was divided in the ratio of 80% and 20% for training and validation, respectively. The training set consisted of 3628 images. The testing set was used to evaluate the performance of the trained model on unseen data.

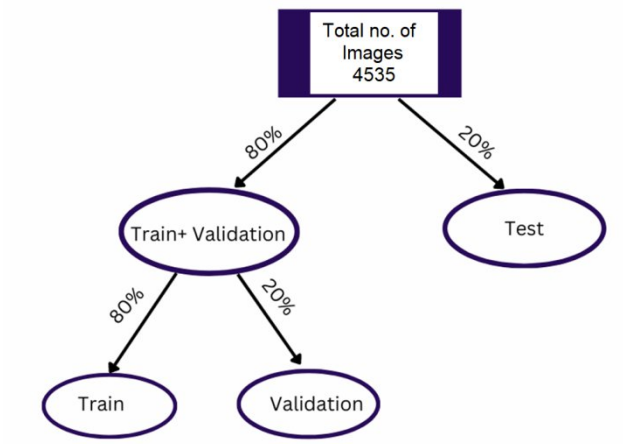


Figure 2: Data splitting.

3.3 Methodology

We have employed a suitable CNN architecture and have used transfer learning to fine-tune a pre-trained model on our dataset. A block diagram of the proposed mythology is depicted in Figure 3.

We have considered factors such as the model's size, speed, and accuracy for selecting a suitable pre-trained CNN architecture to use for our road flaw detection model. We have done experiments with different CNN architectures including ResNet152V2, Xception, InceptionV3, MobileNetV2, and InceptionResNetV2. Further, we have used transfer learning to fine-tune the pre-trained model on our road flaw detection dataset. We have frozen the weights of the pre-trained layers and have only trained the new layers added to the model for the road flaw detection task. This has helped to speed up the training process and improved the model's performance.

The hyper-parameters of the models were tuned to further improve the performance. We have experimented with different values of hyper-parameters such as learning rate, batch size, and the number of epochs to find the best combination that results in the highest accuracy.

4. Model Description

4.1 Transfer Learning

Transfer learning is a technique in ML where a pre-trained model, trained on a large and general dataset, is fine-tuned for a specific task. The pre-trained model already has learned a set of features that are relevant for a wide range of related tasks. The idea is to take advantage of the knowledge and insights gained while solving one problem and apply it to a different but related problem. Instead of training a new model from scratch, transfer learning allows us to use a pre-trained model as a starting point and then adapt it to our specific problem by adjusting the final layers of the network. This can save a lot of time and computational resources, especially when the dataset available for the new task is relatively small. The process of transfer learning typically involves several steps such as choosing a suitable model, modifying the model, and training the model. The steps are described in brief in this section.

A. Choosing a pre-trained model: The first step is to choose a pre-trained model that is relevant to the task. The pre-trained model should have been trained on a large and diverse dataset and have learned a set of task-relevant features.

B. Modifying the model: The next step is to modify the pre-trained model by adding new layers or adjusting the existing layers to adapt the model to the task. This is also called fine-tuning. Fine-tuning involves training the new layers on the new task while keeping the pre-trained layers frozen.

C. Training the model: After the model has been modified, it is trained on the new dataset. The training process involves feeding the model with inputs and evaluating its performance. This process continues for several iterations until the model has learned the optimal set of parameters to solve the new task.

Transfer learning has been successfully applied in various domains such as computer vision, natural language processing, and speech recognition [9]. In this work, transfer learning is used over pre-trained ResNet152V2, Xception, InceptionV3, MobileNetV2, and InceptionResNetV2 CNN models. These models are described in this section.

4.2 CNN Models used for Transfer Learning:

4.2.1 ResNet152V2:

ResNet-152V2 is an improved version of the original ResNet-152, which was introduced by He et al. in 2016 [10]. The ResNet architecture uses residual connections, which allow information to flow more easily through the network, and reduce the vanishing gradient problem that can occur in very deep neural networks. This is accomplished by adding skip connections that bypass one or more layers of the network and allow the gradient to propagate more easily back through the network during training.

ResNet-152V2 consists of 152 layers and uses a combination of convolutional layers, pooling layers, and fully

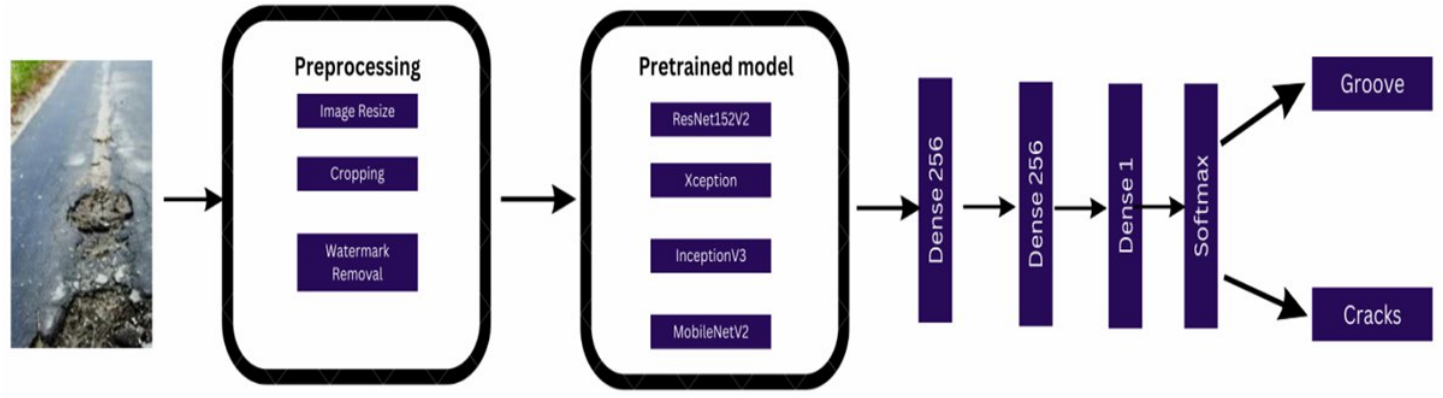


Figure 3: The block diagram of the proposed methodology

connected layers. The model uses batch normalization and ReLU activation functions to speed up the training process and improve the accuracy of the model. ResNet-152V2 has been trained on a very large ImageNet dataset [11], which includes a diverse range of images from many different categories.

4.2.2 Xception:

Xception or extreme inception was introduced by Google in 2016 as an extension of the inception architecture. Xception uses depthwise separable convolutions [12]. The depthwise separable convolution is an operation that is performed separately on each channel of the input tensor. This operation can be thought of as a spatial convolution followed by a channel-wise convolution. By performing these operations separately, the network can reduce the number of parameters and computations required, making it more efficient. Xception also uses residual connections. This helps to prevent overfitting and allows the model to learn more generalizable features from the data.

One advantage of Xception over other networks is its ability to capture fine-grained features from the input images, which can be useful in tasks such as object recognition and image classification. Xception has been shown to perform well on a variety of benchmark datasets, including ImageNet, CIFAR-10, and CIFAR-100. In the context of road flaw detection, Xception has been shown to be a powerful tool for identifying cracks and grooves in road surfaces. By leveraging the efficient depthwise separable convolutions and residual connections, Xception can learn to detect even subtle patterns in the images, making it a valuable addition to the toolset of road maintenance professionals.

4.2.3 Inception V3

InceptionV3 is developed by Google in 2015 [13]. It is designed to improve upon the original inception architecture by using a series of smaller convolutions rather than one large convolution. This allows the network to learn more complex features while reducing the number of parameters and computations needed. The network consists of multiple inception modules, each containing several parallel convolutions of different sizes. The outputs of these convolutions are concatenated and fed into the next module. This architecture allows the network to learn features at different scales, which improves its ability to recognize objects of varying sizes. Inception V3 also includes several auxiliary classifiers that are inserted at intermediate layers of the network. These classifiers help to mitigate the problem of vanishing gradients by providing additional supervision signals to the network during training.

In terms of performance, inception V3 has been shown to achieve state-of-the-art results on several image classification tasks, including the ImageNet large scale visual recognition challenge. Inception V3 is a powerful and efficient neural network architecture that is well-suited for a wide range of computer vision applications.

4.2.4 MobileNetV2

MobileNetV2 is a lightweight architecture designed for efficient on-device image classification tasks on mobile and embedded devices. It was developed and released by Google in 2018 as an improved version of the original MobileNet architecture [14]. The MobileNetV2 uses depthwise separable convolutions instead of traditional convolutions to reduce the number of parameters and computational complexity of the model. Depthwise separable convolutions consist of two layers: depthwise convolution and pointwise convolution. The depthwise convolution applies a single filter per input channel, while the pointwise convolution applies 1×1 filters to combine the output of the depthwise convolution. MobileNetV2 also uses linear bottlenecks with shortcut connections to further reduce the number of parameters and computation.

MobileNetV2 is pre-trained on the ImageNet dataset and

can be fine-tuned on specific classification task or used as a feature extractor for transfer learning. It has achieved state-of-the-art performance on various image classification benchmarks while maintaining a small model size and low computational requirements.

4.2.5 InceptionResnetV2:

Inception ResNetV2 architecture combines ideas from two influential models, Inception and ResNet. It was proposed by Christian Szegedy et al. from Google in 2016 as an extension to the original Inception architecture. The ResNetV2 improves the accuracy and performance of deep CNN for image classification tasks. It addresses the challenge of training very deep networks by introducing residual connections and Inception modules. Residual connections, inspired by the ResNet architecture, allow the network to learn residual functions by adding skip connections that bypass one or more layers. These connections enable the gradient to flow more easily during training, alleviating the vanishing gradient problem and enabling the training of deeper networks.

The Inception modules in Inception ResNetV2 consist of multiple parallel convolutional operations with different kernel sizes, which capture features at different scales. By concatenating the outputs of these parallel branches, the network can capture both local and global contextual information, enhancing its representational power.

5. Results and Discussion

5.1 Experimental Results:

We have evaluated the performance of the trained model on a testing dataset. Metrics such as accuracy, precision, recall, and F1 score have been used to evaluate the model's performance. By following these subtopics, it ensured that the model used for road flaw detection is accurate, reliable, and suitable for the task. We have also experimented with different architectures and hyper-parameters to achieve the results for our dataset. The confusing matrices obtained for different models, fine-tuned using transfer learning are presented in table 1.

Table 1: Confusion matrices obtained for different CNN architectures fine-tuned using transfer learning.

ResNet152V2		Xception		InceptionV3		MobileNetV2		Inception ResnetV2	
373	7	362	18	371	9	373	7	355	23
29	370	14	385	34	365	28	371	30	371

The accuracy and F1 scores obtained after training the dataset images in five different models are reported in Table 2. Figure 4 presents the loss metric value obtained for the different models.

Table 2: Accuracy and F1 scores for different CNN models fine-tuned using transfer learning.

Model Name	**Accuracy**	**F1 score Crack**	**F1 score Groove**
ResNet152V2:	95.37869%	0.9539642	0.9536082
Xception:	95.89217%	0.9576723	0.9600998
InceptionV3:	94.48010%	0.9452229	0.9443726
MobileNetV2:	95.50706%	0.9551857	0.9549550
InceptionResnetV2	93.20000%	0.9300000	0.9300000

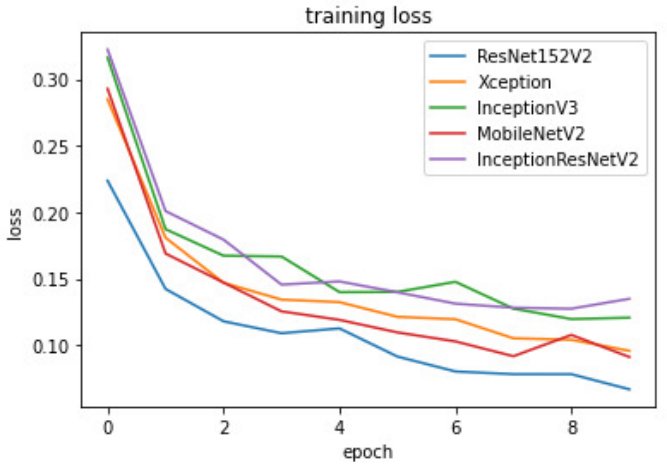


Figure 4. The summary history plot of all the models.

5.2 Discussions:

It can be observed from Table 2 that the highest accuracy and F1 score are obtained for ResNet152V2 model. Moreover, ResNet152V2 has produced lowest loss among the different models. Therefore, based on the results, the best model for the road flaw detection for our data set is ResNet152V2. We have found the following reasons for the phenomena:

(i) ResNet152V2 is deeper and more complex neural network architecture than the other transfer learning models used in this research. This allows ResNet152V2 to learn more complex and abstract features, which is especially important when working with a complex and varied dataset like road cracks and road grooves. If we closely look at the data set the noise is significant in the dataset which can be a car, blue sky, grass, pedestrian, scooter, or special marking on the road, hence more the noise more deeper the neural network requirement to produce better results. In cases where the dataset is less noisy, or where the task requires less complex features to be extracted, other models may perform just as well

or even better than ResNet. For example, in a medical image classification task like melanoma detection, the dataset is typically less noisy and may require a different type of feature extraction than in the case of road flaw detection. In this case, other models such as VGG or DenseNet may perform just as well or even better than ResNet.

(ii) ResNet152V2 uses residual connections, which allow information to flow more easily through the network, and reduce the vanishing gradient problem. This helps to prevent overfitting and allows the model to learn more generalizable features from the data. Residual connections also have the benefit of making it easier to train very deep networks. In a traditional neural network without residual connections, adding more layers can actually make it more difficult to train the network, as gradients can become very small and lead to slow convergence or even stagnation. With residual connections, however, the gradients can be more easily propagated through the network, allowing deeper networks to be trained more effectively.

(iii) ResNet152V2 has been pre-trained on a large dataset of images called ImageNet. This pre-training helps the model to learn general features that can be useful for a wide range of image recognition tasks. Because the ImageNet dataset contains a diverse range of images from many different categories, ResNet152V2 has learned to recognize many different types of features that are relevant to image recognition tasks, including the detection of road cracks and road grooves. By leveraging the pre-trained ResNet152V2 model, we can save a significant amount of time and computational resources. Further, we can fine-tune the pre-trained model on our dataset, allowing it to adapt to the specific characteristics of our data while still retaining the general features that it has learned from ImageNet.

Optimization of the models:

We have optimized the model by removing the duplicate images in the dataset which was done through the use of image hash algorithm in Python. Removing duplicate images has significantly optimized the models in the following ways:

(i) Faster training: Removing duplicates reduces the number of images that need to be processed, speeding up training.

(ii) Better accuracy: Duplicate images can bias the model and make it more likely to over-fit. By removing duplicates, we can ensure that the model is trained on a diverse set of images, improving its ability to generalize.

(iii) More efficient use of resources: By reducing the size of the dataset, we can optimize the use of computing resources such as memory and storage.

(iv) Easier data management: A smaller dataset is easier to manage and work with.

Post-Processing for maintenance and update:

Maintenance and updates (including data, model, code, and hardware updates) are critical for the long-term success of a ML model used for road flaw detection. *Data update:* As new data becomes available, we may update the model to ensure that it can detect new types of road flaws. *Model update:* As new pre-trained CNN architectures become available, we may need to update the model architecture. We can experiment with new architectures and fine-tune the model on the latest data to achieve better results. *Code update:* As new versions of libraries and frameworks are released, we may update the code to ensure that the model runs smoothly and efficiently. *Hardware updates:* As the size of the dataset or the complexity of the model increases, we may upgrade the hardware used to train and run the model.

5. Conclusion

This paper has proposed and developed a road flaw detection system using CNN and transfer learning. The research goal was to create a ML model that could accurately detect road cracks and road grooves from images, which could be used to support road maintenance and safety efforts. The collected dataset of road images contained both road flaws and other non-relevant features. The data was pre-processed and the duplicate and low-quality images were removed. We have used transfer learning with several popular CNN models, including ResNet152V2, Xception, InceptionV3, MobileNetV2, and InceptionResNetV2, to train on the dataset. The experimental results showed that ResNet152V2 performed the best among the tested models, likely due to its ability to learn generalizable features from the large ImageNet dataset and its use of residual connections to reduce the vanishing gradient problem. We achieved a high accuracy of 95.37869% on the test set, demonstrating the effectiveness of our approach. Future work could explore additional improvements to the model, such as the use of more advanced image processing techniques or the incorporation of other types of data, such as sensor data from vehicles.

References

[1] Kalfarisi, Rony, Zheng Yi Wu, and Ken Soh. "Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization." *Journal of Computing in Civil Engineering* 34.3 (2020): 04020010.

[2] Czimmermann, Tamás, et al. "Visual-based defect detection and classification approaches for industrial applications—A survey." *Sensors* 20.5 (2020): 1459.

[3] Zhuang, Fuzhen, et al. "A comprehensive survey on transfer

learning." *Proceedings of the IEEE* 109.1 (2020): 43-76.
[4] Zhang, Lei, et al. "Road crack detection using deep convolutional neural network." 2016 IEEE international conference on image processing (ICIP). IEEE, 2016.
[5] Sari, Yuslena, Puguh Budi Prakoso, and Andreyan Rezky Baskara. "Road crack detection using support vector machine (SVM) and OTSU algorithm." 2019 6th International Conference on Electric Vehicular Technology (ICEVT). IEEE, 2019.
[6] Omar, Mohd, and Pradeep Kumar. "Detection of roads potholes using YOLOv4." 2020 International Conference on Information Science and Communications Technologies (ICISCT). IEEE, 2020.
[7] Hamishebahar, Younes, et al. "A comprehensive review of deep learning-based crack detection approaches." Applied Sciences 12.3 (2022): 1374.
[8] Buch, Norbert, Sergio A. Velastin, and James Orwell. "A review of computer vision techniques for the analysis of urban traffic." IEEE Transactions on intelligent transportation systems 12.3 (2011): 920-939.
[9] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." Journal of Big data 3.1 (2016): 1-40.
[10] He, Kaiming, et al. "Identity mappings in deep residual networks." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer International Publishing, 2016.
[11] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009.
[12] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[13] Xia, Xiaoling, Cui Xu, and Bing Nan. "Inception-v3 for flower classification." 2017 2nd international conference on image, vision and computing (ICIVC). IEEE, 2017.
[14] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.