Seoul National University of Science and Technology

Department of Applied Artificial Intelligence

Machine Learning I

Spring 2023

Paper defect type classification model

BeomJune Kim (18101968), Soyoung Park (20102140) and Sihyeon Jeon (21101974)

Paper defect type classification model

BeomJune Kim (18101968), Soyoung Park (20102140) and Sihyeon Jeon (21101974)

May 6, 2023

Abstract

Create a learning model that classifies defect types for the wallpaper defect image dataset.

1 Introduction

Papering is an essential construction process not only to protect structural materials such as walls and ceilings of a house, but also to block drafts and achieve aesthetic satisfaction. There are various types of defects in wallpapering work, such as tearing, scratches, lifting, surface defects, contamination, poor bonding, poor cutting, bursting, twisting, different colors, unpainted, discolored, and damaged. And, depending on the type of defect, the required repair method is different.

This project focused on determining how it can be helpful in establishing a detailed process plan for work related to defects by analyzing the types of defects with images of plastering defects.

하자유형	원 인	대 책
찢김	- 타공정 기능공들의 부주의 - 공사 후 보양미흡 - 부적절한 도배지 선정	- 기능공들의 사전교육 및 관리감독 철저 - 도배공사 완료 후 보양재 설치 - 내구성이 강한 도배지의 사용
이음면 불량	- 미숙련공에 의한 시공 - 도배용 풀의 접착력 부족	- 숙련된 기능공의 투입 - 도배용 풀의 자재품질 확인
들뜸	- 바탕면 품질 불량 - 부적절한 시공환경 - 미숙련공에 의한 시공	- 시공전 바탕면 평활도 검사 및 청소 철저 - 동절기 시공 금지 - 숙련된 기능공의 투입
접착불량/ 마감불량	- 도배용 풀의 접착력 부족 - 미숙련공에 의한 시공	- 도배용 풀의 자재품질 확인 - 숙련된 기능공의 투입

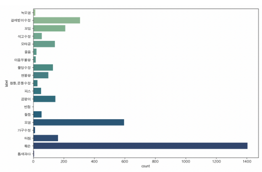
2 Dataset

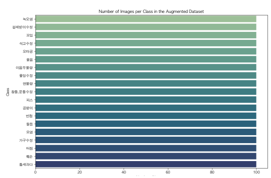
Hansol Deco provides Dacon with wallpaper defect image type data for the development of an AI model for paper defect type classification. The training data includes 19 defect type data, and the test data includes 791 wallpaper defect images.

3 Data Processing

3.0.1 Augmentation

When checking the number of data by defect type, some types, such as 'damaged' and 'contaminated', have excessively large weights, so the weights of other types can be ignored. Therefore, data with a large weight is randomly extracted and only a portion is used, and the rest is augmented by methods such as inverting, skewing, and adjusting brightness. This is especially useful when available data is limited or there is class imbalance where certain classes have fewer examples than others. Additionally, by applying random rotations during augmentation, the model can learn to recognize objects from all angles, allowing it to generalize to new, more robust and unseen data. As a result of the augmentation, the 19 species had 100 data each and were able to balance the class distribution.





3.0.2 Train/Val/Test Split

In addition, before proceeding with model evaluation on the test data set for the model learned through train data, 0.2 of the training data set is separated as a validation set to improve the performance of the model.

4 Data Training Model

4.0.1 Multiple Image Classification Model

First of all, to find the learning model with the highest accuracy for our dataset, we trained the following representative image classification learning model by 1 epoch. DenseNet121, MobileNetV2, DenseNet201, EfficientNetB0, EfficienNetB1, InceptionV3, MobileNetV3Large, ResNet152V2, ResNet50V2, VGG19, VGG16, Xception

	model	val_accuracy	accuracy	Training time (sec)
0	ResNet152V2	0.6913	0.6416	197.53
1	ResNet50V2	0.7235	0.6329	78.21
2	DenseNet121	0.6495	0.6272	101.30
3	DenseNet201	0.6527	0.6243	160.51
4	InceptionV3	0.6688	0.6069	56.60
5	MobileNetV2	0.6495	0.6012	40.77
6	Xception	0.6688	0.5983	88.86
7	VGG16	0.5434	0.4624	167.30
8	VGG19	0.4887	0.4480	204.89
9	ResNet50	0.4598	0.4133	94.97
10	EfficientNetB0	0.4341	0.4075	57.55
11	EfficientNetB1	0.4341	0.4075	77.92
12	MobileNetV3Large	0.4405	0.4075	39.24

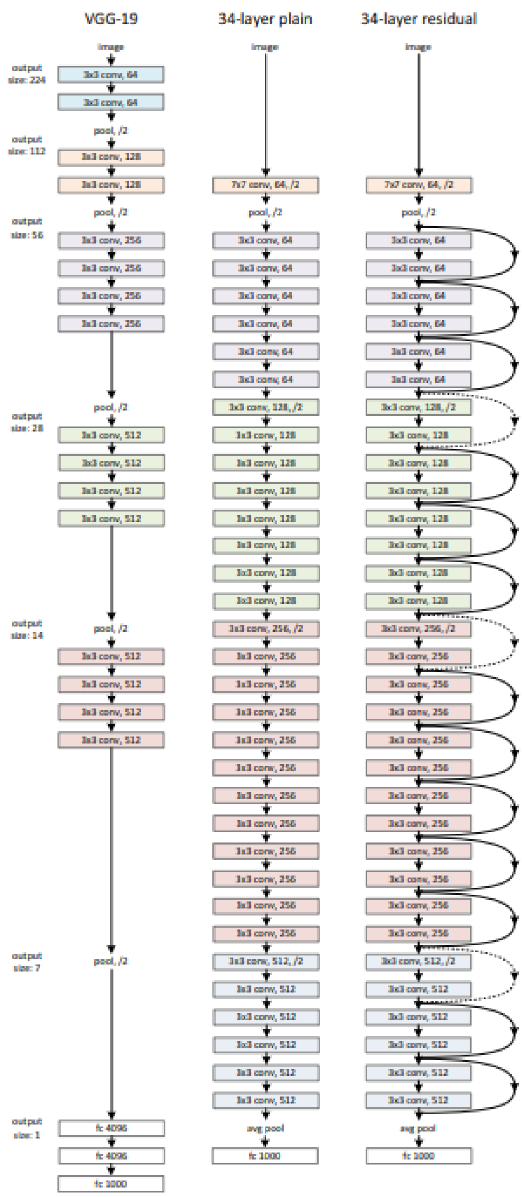
4.0.2 Model Selection

After training several image classification training models, DenseNet201, ResNet152V2, and ResNet50V2 took the top three. As a result of learning DenseNet201 Model, ResNet152V2 Model, and ResNet50V2 Model individually, accuracies of 0.6850, 0.6994, and 0.7023 were derived, respectively. It was confirmed that ResNet152v2 and ResNet50v2 had the highest accuracy.

4.0.3 ResNet Model

ResNet is a CNN model with increased layers based on VGGNet. However, according to the paper Deep Residual Learning for Image Recognition, ResNet solves the vanishing gradient problem with significantly increased depth, but with a significant accuracy improvement, but with less complexity than VGGNet. The vanishing gradient problem arises because as the gradients are back-propagated through the layers, they can get very small and eventually vanish, making it difficult for the network to learn. To solve this problem, ResNet introduces the concept of residual blocks, which are building blocks that allow the training of very deep networks by allowing gradients to more easily propagate through the layers. A residual block consists of two convolutional layers with skip connections that add the block's inputs to its outputs. This skip connection allows the network to learn a residual mapping, which is the difference between input and output.

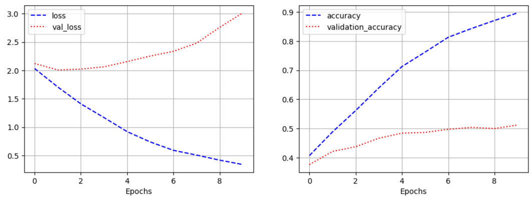
These characteristics can explain the excellent performance of ResNet shown above.



5 Improvement of the model

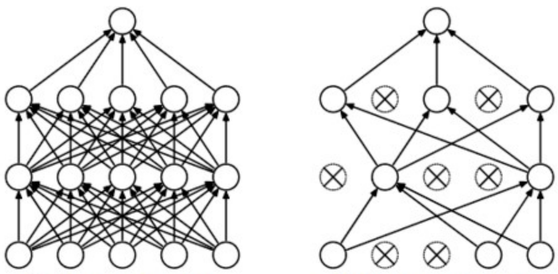
5.0.1 overfitting problem

Overfitting indicates that the model is too complex for the problem it is trying to solve. Filters for convolutional neural networks, too many layers for full deep learning models. This causes the model to recognize the training data well, but perform poorly on the test data.



5.0.2 Dropout

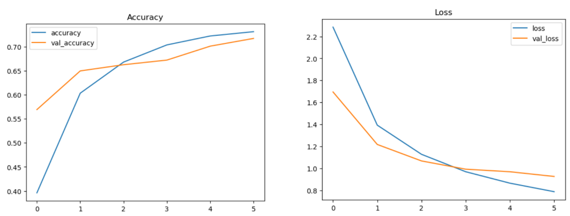
Dropout regularization ignores a random subset of units in the layer while setting the weights to zero in that training step. According to the referenced paper, the dropout ratio of the input and hidden layers is set to 0.4.



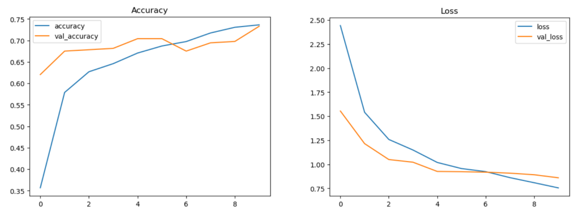
(a) Standard Neural Net (b) After applying dropout.

5.0.3 result

As a result of evaluating the accuracy of ResNet152v2 and ResNet50v2 by F1score, ResNet50v2 performed better on the trained dataset with 0.70 and 0.74, respectively.



accuracy			0.70	346
macro avg	0.63	0.49	0.45	346
weighted avg	0.71	0.70	0.69	346



accuracy			0.74	346
macro avg	0.82	0.49	0.50	346
weighted avg	0.75	0.74	0.72	346

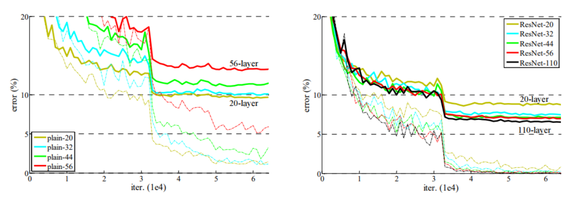
5.0.4 Why ResNet50v2 is better

ResNet152 has more layers than ResNet50v2, but it doesn't necessarily guarantee better performance. The following two reasons explain why ResNet50v2 performs better than ResNet152v2

First, larger ResNet152 models have more layers and parameters. This allows the model to learn more complex features on the training data, but when the data set is small, noisy, or contains irrelevant data, it becomes over-adaptable and hard to generalize to the training data. In other words, overfitting occurs when the model learns patterns specific to the training data and becomes unpredictable on new data.

Second, the larger ResNet152 model has higher computational complexity than the smaller ResNet50v2 model. This slows down the learning rate and efficiency and can prevent the optimization algorithm from converging. In addition, memory usage increases as the size of the model increases, and the possibility of numerical problems such as overflow or underflow increases. These issues can negatively affect the model's performance.

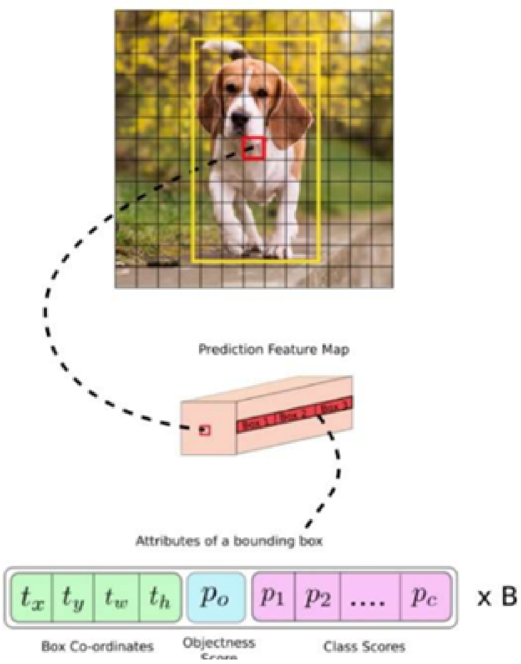
Thus, we can see that the 152-layer network is unnecessarily large compared to the small size of the dataset we have for training. In this case, using the smaller ResNet50v2 model has better performance.



6 Challenging YOLO

YOLO will have better performance as it adds clarity to the objects you want to classify by creating a bounding box. Also, unlike the existing R-CNN, YOLO will greatly reduce the speed because the location of the bounding box and the class classifier are performed simultaneously in the final output stage using the feature map. Therefore, we tried to synthesize the trained ResNet classifier with YOLO. However, since YOLO simultaneously outputs the probability that an object is included in a bounding box through a feature map and the probability that an object identified in a bounding box belongs to a specific class at the same time in the final output step, the labeled data had to be implemented separately as a txt file. YOLO also calculates the relative feature values of each object to distinguish them, but we did not know whether it could generate a suitable bounding box because the defects in our data do not have clear boundaries.

Therefore, programming for YOLO was discontinued because it appeared to be more time consuming than expected performance gains. Still, if you have enough labeled data files, I think it's a challenge worth trying.



7 Contributions

7.0.1 Overfitting problem

How to Treat Overfitting in Convolutional Neural Networks Published On September 7, 2020 and Last Modified On September 8th, 2020

7.0.2 ResNet Model

Deep Residual Networks (ResNet, ResNet50) – 2023 Guide

7.0.3 Why ResNet50v2 is better

Deep Residual Learning for Image Recognition Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun