Retail Product Classification on Distinct Distribution of Training and Evaluation Data

Jonathan[a],* and Gede Putra Kusuma[a],**

[a] *Computer Science Department, BINUS Graduate Program—Master of Computer Science, Bina Nusantara University, Jakarta, 11480 Indonesia*

** e-mail: jonathan016@binus.ac.id*

*** e-mail: inegara@binus.edu*

Abstract—Retail product classification can be beneficial in the world of commerce, take for example helping vision-disabled parties in their shopping or evaluating product placement strategy. However, the available datasets for retail product classification are few and some have very distinct distribution of training and evaluation data, thus providing a huge challenge on its own. In addition, there are only few researches on this subject which can still be improved on. This paper attempts to improve previous approaches for retail product classification on very distinct training and evaluation data distribution by utilizing convolutional neural network (CNN) models inspired by well-performing CNN models in general image classification task, which later can be fine-tuned for other computer vision tasks, namely, object detection. The results show that VGG-16 performs at 66.9167% accuracy and a new modified VGG-16 model named VGG-16-D attains 66.83% accuracy with 85% fewer parameters than VGG-16, outperforming most existing approaches considering several comparison baselines.

Keywords: image classification, retail product classification, deep learning, convolutional neural network, dilated convolution

DOI: 10.1134/S105466182104012X

INTRODUCTION

Image classification has always been one of the most popular tasks in the field of artificial intelligence. Researchers still try to incorporate human's capability of classifying objects based on visual cues such as images with ease to machines. This is proven with numerous image classification datasets, varying from simple ones such as MNIST handwriting dataset [7] to more diverse and larger ones such as ImageNet [1, 20], also added with a lengthy line of previous researches for image classification, including but not limited to scale-invariant feature transform (SIFT) [11, 12] and eventually convolutional neural network (CNN) as introduced in LeNet [6].

CNN has been the focus of researches lately with many CNN-based models for image classification such as AlexNet [5], VGG [22], and other famous architectures. One major contributing factor of CNN's fame is its simplicity of not requiring much human-related processing, thus making the development process much faster and easier as feature extraction is done autonomously by each model. Moreover, CNN-based models have been proven to yield better results compared to previous approaches, hence increasing CNN's popularity even more.

Although famous architectures have been evaluated for general datasets, these architectures are also not limited to other more specific datasets, such as retail product. Retail product classification may be beneficial for vision-disabled parties in shopping and also for assuring correct product placement in retail stores as planned. Retail product classification is also quite challenging on its own as the available datasets are small compared to those used by famous architectures. As have been observed along CNN's development, more data tend to be beneficial for a model as the model learns from many examples and thus increases its generalization capability. Furthermore, the biggest challenge lies in the distinction between training and evaluation's data as retail product classification datasets are often comprised of retail product images in ideal condition as their training set, while the evaluation set contains retail product images in a very different condition due to lighting and other environmental issues. This is clearly seen in GroZi-120 [14] as this paper's used dataset, which is shown in the following sections.

There have been several approaches for retail product classification, albeit only a few. Santra et al. proposed deterministic dropout and composite random forest on a modified AlexNet [21]. Srivastava used ResNeXt-101_32×8d [26] pretrained on Instagram with local-concepts-accumulation layer and maximum entropy loss [23]. These approaches provide

Received December 23, 2020; revised May 1, 2021; accepted July 8, 2021

Table 1. Classification results on retail product classification from existing approaches

Technique	Dataset	Classification accuracy, %
Deterministic dropout, composite random forest (CRF), AlexNet [21]	GroZi-120	45.15
ResNeXt-WSL [23]	GroZi-120	60.4
ResNeXt-WSL, local-concepts-accumulation (LCA), maximum entropy [23]	GroZi-120	72.3
ResNeXt-WSL, local-concepts-accumulation (LCA), maximum entropy [23]	Grocery products	81.62
Guidance learning [8]	Products-90	71.4

room for improvement in terms of network accuracy. Moreover, existing approaches have not been found to be fine-tuned for other computer vision tasks such as object detection, which limits the aforementioned approaches' applications.

This paper attempts to improve existing results on retail product classification. The experiments in this paper use CNN models for retail product classification with few modifications. More specifically, this paper uses VGG-16 [22] and Darknet models [16], namely Darknet-19 [18] and Darknet-53 [19]. These models have been proven to yield good accuracy on ImageNet dataset. They also serve as backbones for YOLO [17–19] and SSD [10], two of the fastest and most accurate single-stage object detector models. Thus, using these backbones would also be beneficial as these backbones can be fine-tuned for detection tasks.

This paper continues with a brief review of existing approaches for retail product classification. Then, the used models are elaborated in detail, continued with this paper's dataset, experiments, and results explanation and discussion. This paper's conclusion and future works are then given.

LITERATURE REVIEW

There have been several researches, albeit few, on retail product classification on varying datasets. The datasets include GroZi-120 dataset [14] of 120 retail products with very distinct training and evaluation set distribution, Grocery Products dataset [2] (also known as GroZi-3.2k) for multilabel classification, and Products-90 dataset [8] containing noisy labels of 90 retail products.

Santra et al. [21] proposed deterministic dropout as a refinement of vanilla dropout [4, 24]. They believed that dropout can be refined to dropping only the unimportant connections instead of being stochastic. To identify the unimportant connections, a composite random forest (CRF) is proposed and integrated to AlexNet. While using CRF makes training time slower due to the construction of the CRF, in inference there is no CRF construction at all. This shows a trade-off between increased accuracy with training time for that deterministic dropout using CRF. They evaluated their proposed approach on multiple datasets, all of which gain increase by 0.04 to 9.25% in accuracy compared to other dropout variants and network without dropout. For GroZi-120 dataset itself, the accuracy on the evaluation set outperforms vanilla dropout by 3.85%, reaching 45.15% accuracy. On Grocery Products dataset, this approach attained 81.62% accuracy.

Srivastava proposed the combination of Instagram-pretrained ResNeXt-101_32x8d model [26] with a new type of layer coined as local-concepts-accumulation (LCA) with maximum entropy auxiliary loss for retail product classification [23]. It is argued that using Instagram-pretrained model shows a better performance on ImageNet, thus increasing the model's capability on a set of very diverse objects. LCA layer on its own works by averaging the local concepts contained in an image to be used by the classifying layer. LCA layer is proposed to be put as the penultimate layer of any CNN for training and/or fine-tuning purposes. To boost the model's performance, maximum entropy loss is added as an additional loss to be weight-averaged with negative likelihood loss. ResNeXt-101_32×8d on its own obtained 60.4% accuracy, while with the proposed combination, an accuracy boost of 11.9 to 72.3% on GroZi-120's evaluation set is obtained.

Li et al. proposed guidance learning, in which a teacher network helps a student network learn to classify retail products with noisy labels [8]. In addition, Li et al. also proposed Products-90 dataset, of which there are approximately 8 thousand correctly labeled training and testing images, respectively, with 124 thousand noisy training data. The teacher network is trained first prior to training the student network with all data including the noisy ones. Then, the student network is trained separately on the correctly labelled training images and fine-tuned on noisily labelled images with the teacher network's help. The best accuracy on Products-90 dataset is at 71.4% after fine-tuning the student network. Table 1 lists the existing approaches on retail product classification.

The researches highlighted in Table 1 used CNN for retail product classification with some modifications and attained good but still improvable accuracy scores. The used CNN models vary from CNN's early iterations, latest models, to custom architectures. Although their respective performances for retail product classification are good, these approaches have not been found to be fine-tuned for other computer vision tasks and thus limit their usability for other use cases.

Table 2. VGG-16 architecture for retail product classification

Layer	Input size	Output size	Kernel	Stride	Parameters
Convolution + ReLU	3	64	3 × 3	1	1792
	64	64	3 × 3	1	36928
Max pooling	64	64	2 × 2	2	0
Convolution + ReLU	128	128	3 × 3	1	73856
	128	128	3 × 3	1	147584
Max pooling	128	128	2 × 2	2	0
Convolution + ReLU	128	256	3 × 3	1	295168
	256	256	3 × 3	1	590080
	256	256	3 × 3	1	590080
Max pooling	256	256	2 × 2	2	0
Convolution + ReLU	256	512	3 × 3	1	1180160
	512	512	3 × 3	1	2359808
	512	512	3 × 3	1	2359808
Max pooling	512	512	2 × 2	2	0
Convolution + ReLU	512	512	3 × 3	1	2359808
	512	512	3 × 3	1	2359808
	512	512	3 × 3	1	2359808
Max pooling	512	512	2 × 2	2	0
Flatten	512	25088	–	–	0
Fully connected + ReLU	25088	4096	–	–	102794544
Dropout (0.5)	4096	4096	–	–	0
Fully connected + ReLU	4096	4096	–	–	16781312
Dropout (0.5)	4096	4096	–	–	0
Fully connected	4096	120	–	–	491640
Total parameters					134752184

EXISTING CLASSIFICATION MODELS

Several models have been proposed to be performant in classification and can be fine-tuned to other computer vision tasks as well. Two among these models are VGG-16 as used in SSD and Darknet models (Darknet-19 and Darknet-53) for YOLO; both of which (SSD and YOLO) are considered performant and fast single-stage object detectors.

VGG-16

VGG-16 [22] was the very first model which pushes the limit of network depth. By using 3 × 3 convolution with ReLU activation function, max pooling, and fully connected layers, the model achieves state-of-the-art performances on ImageNet and other datasets. VGG-16 has been used for numerous tasks, such as fine-tuning for detection task as done by SSD [10] or even for other classification tasks [15], proving its capability to be retrained for other tasks while achieving good results as well. For this reason, VGG-16 is adapted for this paper's experiments.

The adaptation includes increasing the input resolution and changing the output channel of the last fully connected layer according to the total available classes in the used dataset. Increasing VGG-16's input resolution is believed to give better results as more detailed features may be extracted, which may increase classification accuracy as well. The new input resolution is 300 × 300 pixels as opposed to VGG-16's original input resolution at 224 × 224 pixels. Another reason as to why the input resolution is increased is to ease fine-tuning for other computer vision tasks, namely object detection using SSD. The final fully connected layer output size is also modified to force VGG-16 to predict the total number of classes in the used dataset. VGG-16's modified architecture for this paper's experiments is shown in Table 2.

Table 3. Darknet-19 architecture for retail product classification

Layer	Input size	Output size	Kernel	Stride	Parameters
Convolution + batch normalization + leaky ReLU	3	32	3 × 3	1	960
Max Pooling	32	32	2 × 2	2	0
Convolution + batch normalization + leaky ReLU	32	64	3 × 3	1	18624
Max pooling	64	64	2 × 2	2	0
Convolution + batch normalization + leaky ReLU	64	128	3 × 3	1	74112
	128	64	1 × 1	1	8384
	64	128	3 × 3	1	74112
Max pooling	128	128	2 × 2	2	0
Convolution + batch normalization + leaky ReLU	128	256	3 × 3	1	295680
	256	128	1 × 1	1	33152
	128	256	3 × 3	1	295680
Max Pooling	256	256	2 × 2	2	0
Convolution + batch normalization + leaky ReLU	256	512	3 × 3	1	1181184
	512	256	1 × 1	1	131840
	256	512	3 × 3	1	1181184
	512	256	1 × 1	1	131840
	256	512	3 × 3	1	1181184
Max pooling	512	512	2 × 2	2	0
Convolution + batch normalization + leaky ReLU	512	1024	3 × 3	1	4721664
	1024	512	1 × 1	1	525824
	512	1024	3 × 3	1	4721664
	1024	512	1 × 1	1	525824
	512	1024	3 × 3	1	4721664
Convolution	1024	120	1 × 1	1	123000
Global average pooling	120	120	–	–	0
Total parameters					19947576

Darknet-19 and Darknet-53

Darknet-19 and Darknet-53 were first proposed by Redmon et al. [18, 19] as backbones for You Only Look Once (YOLO) single-stage object detector model. Darknet-19 is a 19-layer deep fully convolutional neural network. While being 19-layer deep, Darknet-19 is much lighter than VGG-16 as it uses 1 × 1 convolutions and attains comparable classification accuracy at 224 × 224 input resolution for ImageNet.

Darknet-53 is another upgrade for Darknet-19, where it increases the model's depth to 53 layers, uses residual connections [3], and replaces pooling layers with convolutional layers to achieve higher classification accuracy—2.1% increase than Darknet-19 at 448 × 448 input resolution for ImageNet—at the cost of slower forward pass and being much heavier than Darknet-19.

Both models use convolution layers with batch normalization and leaky ReLU activation function [13], unless for the predictor convolution layer. Global average pooling [9] is also used in both models. Both models' architectures are given in Tables 3 and 4.

PROPOSED MODEL—VGG-16-D

VGG-16's core principle of stacking convolution layers has greatly influenced many CNN architectures, such as Darknet-19, Darknet-53, ResNeXt [26], and ResNet [3]. This shows the robustness of VGG-16's core principle, even combined with other modifications of CNN, be that more complex ones such as in ResNeXt or simpler ones such as in Darknet models and ResNet. This implies that modifying VGG-16 itself should achieve better, if not at least equivalent, results compared to the vanilla VGG-16.

Table 4. Darknet-53 architecture for retail product classification

Layer	Input size	Output size	Kernel	Stride	Repetition	Parameters × repetition
Convolution + batch normalization + leaky ReLU	3	32	3 × 3	1	1	960 × 1
Convolution + batch normalization + leaky ReLU	32	64	3 × 3	2	1	18624 × 1
Residual + convolution + batch normalization + leaky ReLU	64	32	1 × 1	1	1	2144 × 1
	32	64	3 × 3	1		18624 × 1
Convolution + batch normalization + leaky ReLU	64	128	3 × 3	2	1	74112 × 1
Residual + convolution + batch normalization + leaky ReLU	128	64	1 × 1	1	2	8384 × 2
	64	128	3 × 3	1		74112 × 2
Convolution + batch normalization + leaky ReLU	128	256	3 × 3	2	1	295680 × 1
Residual + convolution + batch normalization + leaky ReLU	256	128	1 × 1	1	8	33152 × 8
	128	256	3 × 3	1		295680 × 8
Convolution + batch normalization + leaky ReLU	256	512	3 × 3	2	1	1181184 × 1
Residual + convolution + batch normalization + leaky ReLU	512	256	1 × 1	1	8	131840 × 8
	256	512	3 × 3	1		1181184 × 8
Convolution + batch normalization + leaky ReLU	512	1024	3 × 3	2	1	4721664 × 1
Residual + convolution + batch normalization + leaky ReLU	1024	512	1 × 1	1	4	525824 × 4
	512	1024	3 × 3	1		4721664 × 4
Global average pooling	1024	1024	–	–	1	0
Convolution	1024	120	1 × 1	1	1	123000 × 1
Total parameters						40725784

An improvement in VGG-16 can come from changing its classifier module (fully connected layers) to convolution layers. The reason behind this change is convolution layers have been shown capable of performing classification task [18, 19] with simpler model design as measured with the number of required parameters. Instead of flattening the extracted feature vectors/maps from CNN's earlier parts and computing the classification scores globally, using convolution layers allows us to process the feature vectors/maps locally. Replacing the fully connected layers with convolution layers is a viable option, but, to maintain the learned weights of the fully connected layers, a subsample of the fully connected layers' parameters can be taken to serve as the convolution layers' parameters. However, doing so could lead to worse results if the subsampling was incorrectly done.

To avoid worse results, the authors took inspiration from SSD and followed SSD's design of transforming the fully connected layers to convolution layers. The convolution layers are composed of dilated convolution [27] with dilation of 6 and vanilla convolution with 1×1 kernel sizes. This design ensures that the dilated convolution layer processes feature maps with context and thus produces more meaningful feature maps, and, afterwards, the feature maps are convolved again with 1×1 convolution to map the features to new dimensions. In addition to the convolution layers, global average pooling is also added to the classifier module as shown in [18, 19] to give an aggregated value of confidence in the form of the average of each feature map. All this leads to a more localized processing of image for classification as opposed to using fully connected layers which propagate information globally.

As for the activation function used in VGG-16-D, the authors opted to use ReLU in the feature extractor module, which is the equivalent of the original VGG-16's feature extractor module. This is done as it is hypothesized that changing the activation function in that module could lead to worse results as the training process will try to adapt the feature extractor module's parameters to the new activation function instead of

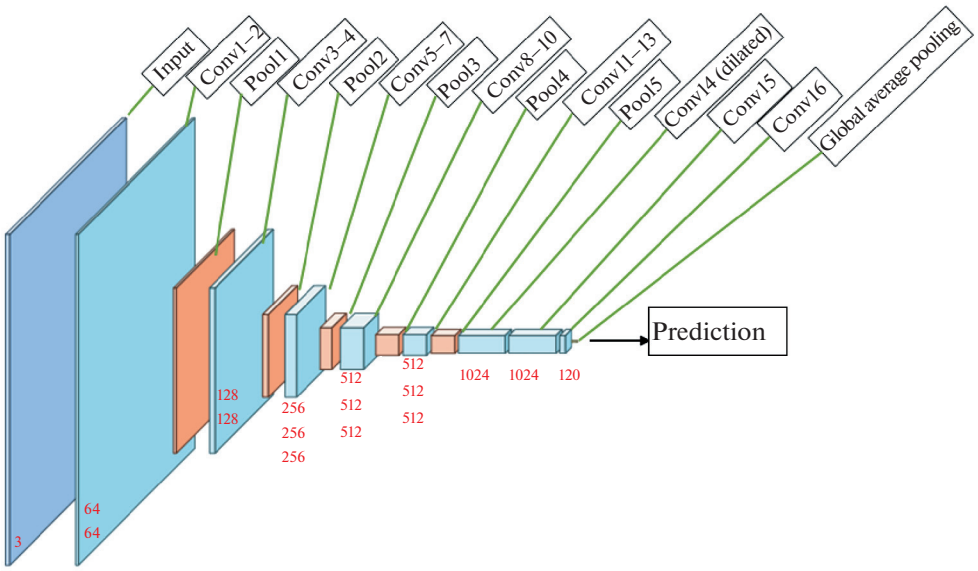

Fig. 1. VGG-16-D architecture diagram for retail product classification.

Fig. 2. Sample in vitro (training) data for three products in GroZi-120 dataset.

Fig. 3. Sample cropped in situ (testing) data for three products in GroZi-120 dataset.

focusing on improving the classification accuracy. On the other hand, the transformed classification module is designed to use Leaky ReLU activation function as used in Darknet models implementations. The network design of VGG-16-D is given in Fig. 1 and Table 5.

EXPERIMENTS

The experiments on this paper were conducted on GroZi-120 dataset using three existing CNN models. In this section, the dataset is discussed in detail, continued with how the experiments are designed, and closed with the results and discussions from the conducted experiments.

GroZi-120 Dataset

GroZi-120 dataset [14] contains images and videos of 120 retail store products with provided training and evaluation sets, respectively, known as in vitro and in situ sets. The distinction between the training and evaluation sets is contrast, where the training set contains individual product images taken from web search and is in ideal condition, while the evaluation set is from video of shelves in a retail store, where each video is taken with limited lighting and resolution.

The training set has 676 total images, while the evaluation set has 29 videos of more than 50000 frames. GroZi-120 dataset is also imbalanced, where the total number of images per class in the training set varies from 2 to 14 images only. The evaluation set is usually annotated per 5 frames, and from the annotations, a cropped version of the evaluation set is provided, where each crop contains only a specific product without the presence of other products. Sample images of in vitro and the cropped in situ data from GroZi-120 dataset are given in Figs. 2 and 3.

GroZi-120 dataset also has its own evaluation protocol. For each product, there should be 10 cropped in situ images of the corresponding product, with a total of 1200 images for classification evaluation. Unfortunately, there is no given list of used images for evaluation in previous researches, including in the dataset's proposal's research [14]; hence, the comparison between each research result cannot be done equally. However, a fairer comparison can be obtained if the

Table 5. VGG-16-D architecture for retail product classification

Layer	Input size	Output size	Kernel	Stride	Dilation	Parameters
Convolution + ReLU	3	64	3 × 3	1	1	1792
	64	64	3 × 3	1	1	36928
Max pooling	64	64	2 × 2	2	1	0
Convolution + ReLU	64	128	3 × 3	1	1	73856
	128	128	3 × 3	1	1	147584
Max pooling	128	128	2 × 2	2	1	0
Convolution + ReLU	128	256	3 × 3	1	1	295168
	256	256	3 × 3	1	1	590080
	256	256	3 × 3	1	1	590080
Max pooling	256	256	2 × 2	2	1	0
Convolution + ReLU	256	512	3 × 3	1	1	1180160
	512	512	3 × 3	1	1	2359808
	512	512	3 × 3	1	1	2359808
Max pooling	512	512	2 × 2	2	1	0
Convolution + ReLU	512	512	3 × 3	1	1	2359808
	512	512	3 × 3	1	1	2359808
	512	512	3 × 3	1	1	2359808
Max pooling	512	512	3 × 3	1	1	0
Dilated convolution + ReLU	512	1024	3 × 3	1	6	4719616
Convolution + ReLU	1024	1024	1 × 1	1	1	1049600
Convolution	1024	120	1 × 1	1	1	123000
Global average pooling	120	120	–	–	1	0
Total parameters						20606904

comparison is done on approaches following GroZi-120's evaluation protocol.

It is also noteworthy that the distinction between training and evaluation sets are very contrast. The employed techniques for classification on this dataset must be very robust to heavy change of color schemes, orientation, and other conditions present in the evaluation set. In addition, the limited number of training data is challenging, especially for CNN-based approaches as using CNN often requires large training data to make CNN more sensitive to features of each class present in the dataset. Lastly, GroZi-120 dataset is found to be the most used dataset for retail product-related researches, either for classification or for detection. This shows that the dataset is challenging and interesting to be used in researches. These three observations from GroZi-120 dataset are the reasons as to why this research uses GroZi-120 dataset.

Experimental Designs

GroZi-120 dataset has very few training data and being imbalanced on each class as well. The authors hypothesized that training on only these data without balancing the data would yield poor results. To avoid this, the authors balanced the training data by using image augmentations and specifying how many data should be present per class. Extensive augmentation techniques such as blurring, color jitter, random rotation, random perspective, random crop, and random erasing are also employed to present variations of the training data. In addition, the classification will be done on grayscale images only to simplify the model learning.

In addition to balancing the dataset and performing augmentation on the training images, as stated before, there are no provided list of images for the evaluation set, be that the test set or validation set. To solve this, the authors randomly select 10 images from each class in the cropped in situ images of GroZi-120 dataset as our test set as specified in GroZi-120's evaluation protocol. The unused images are then selected as our validation set while also considering the total image limit per class as specified in GroZi-120's evaluation protocol.

The used optimizers vary between stochastic gradient descent (SGD) with momentum and Adam optimizer considering Wilson et al.'s work [25]. SGD has

Table 6. Recapitulated performances on GroZi-120 dataset

Model/Technique	Image resolution	Pretrained weight	Classification accuracy, %
Deterministic dropout, CRF, AlexNet [21]	224 × 224	ImageNet	45.15
ResNeXt-101_32×8d [23]	Unknown	ImageNet	60.4
ResNeXt-101_32×8d, LCA, maximum entropy [23]	Unknown	ImageNet	72.3
VGG-16	300 × 300	ImageNet	66.9167
VGG-16-D	300 × 300	VGG-16 (above)	66.833
Darknet-19	224 × 224	None	59.833
	448 × 448	None	58.33
Darknet-53	224 × 224	None	54.4167
	448 × 448	None	49

been reported for its better generalization capability on unseen training data despite being slow. On the other hand, Adam is reported to provide faster convergence and training although not being as generalized as SGD. As for the used loss function, the authors used cross entropy loss with mean loss reduction. PyTorch is selected as the used library for conducting this research's experiments.

For VGG-16, the authors used PyTorch's ImageNet-pretrained VGG-16 to be fine-tuned as fine-tuning existing weights for a model could help the model's learning progress, especially on small datasets. VGG-16 training uses SGD optimizer with 0.001 learning rate, 0.9 momentum, and 0.0005 weight decay. Training will go for 75 epochs and batch size of 8 is used. VGG-16's input image resolution is changed from the original 224 × 224 to 300 × 300 to provide fine-grained features.

Similar to VGG-16, VGG-16-D will process input resolution of 300 × 300 pixels and will be trained with SGD optimizer with 0.001 learning rate, 0.9 momentum, 0.0005 weight decay, and batch size of 8. The distinction is VGG-16-D will use the best result obtained from VGG-16 experiment to be fine-tuned for 30 epochs. This is to simplify VGG-16 without sacrificing much of its performance.

All Darknet models were trained following their respective implementation details, although these models were trained without utilizing ImageNet-pretrained weights. First, the authors trained Darknet models on images with lower resolution; 224 × 224 resolution for Darknet-19 and 256 × 256 resolution for Darknet-53. Afterwards, the authors fine-tuned these models on images with 448 × 448 resolution to enrich the model with more fine-grained features. The authors opted to use Adam optimizer in training as several implementations of Darknet models on PyTorch reports that SGD cannot help Darknet converge whereas Adam could. For all training, a learning rate of 0.0001 with weight decay of 0.0005 is used. Training will last for 100 epochs with batch size of 16.

RESULTS AND DISCUSSION

Overall, there are several variations during our experiments: model type, image resolution, and using ImageNet-pretrained weights. Each experiment provided unique results as can be seen on Table 6 and the best results are charted in Fig. 4.

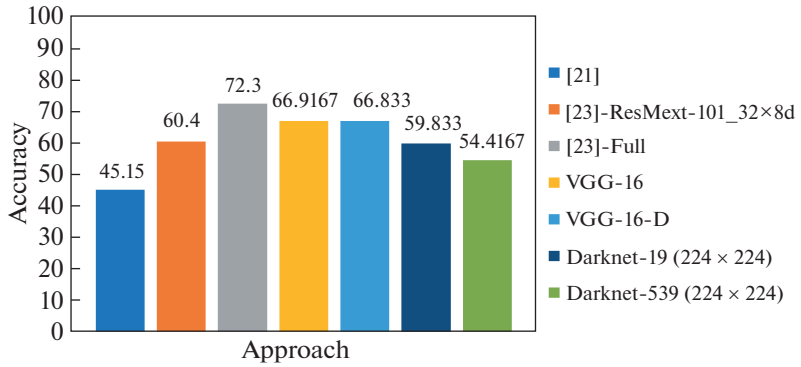

Fig. 4. Accuracy comparison between existing approaches and best experiment results.

From Table 6 and Fig. 4, several insights can be derived. The first is on equivalent image resolution at 224 × 224, all Darknet models exceed the accuracy obtained by Santra et al. [21] by a significant margin, even though Darknet models were trained from scratch without using ImageNet-pretrained weights. This shows the robustness of those models in generalizing data on GroZi-120 dataset. The authors argue that the modifications implemented in Darknet models contributes to their better performances. 1 × 1 convolution helps filtering and processing existing features to be the most important ones. Moreover, new feature map dimensions/channels can also be derived by using 1 × 1 convolutions, thus enriching the processed feature maps. Batch normalization as another modification for Darknet models also contributes positively to their respective performances, which helps the models to be more stable during training while also getting faster training time as opposed to using dropout. Leaky ReLU also helps in avoiding dying ReLU problem as negative values are permitted.

The second insight is comparing performances from ImageNet-pretrained models, VGG-16 on its own outperforms approaches by [21] and is quite comparable to the full proposed solution by [23]. The authors believe that operating on grayscale images helps boost VGG-16's performance. Also, using SGD optimizer for VGG-16 seems to help VGG-16 in its learning and having better generalization capability on unseen data. Note that VGG-16 alone performs much better than plain fine-tuned ResNeXt-101_32×8d by 6.51% margin, although ResNeXt-101_32×8d is much more complex and deeper than VGG-16 with aggregated residual connection block and being 101-layer deep. This means using simpler and shallower model which has been proven to yield good results on benchmark datasets such as VGG-16 is adequate, if not better, at classifying retail products from images. And although using plain VGG-16 did not result in higher accuracy than 72.3% as obtained by utilizing ResNeXt-101_32×8d, LCA layer, and maximum entropy auxiliary loss, it managed to achieve a comparable accuracy. The authors had considered on implementing LCA layer and maximum entropy loss on our experiments, but due to the limited information of the implementation details, the authors are unable to implement such modifications.

The third insight is VGG-16-D is capable of operating at very competitive accuracy at 66.833% compared with VGG-16's 66.9167% despite having much simpler design with only 20 million parameters compared with VGG-16's 134 million parameters for classification on GroZi-120 dataset. This shows that fine-tuning existing CNN model which utilizes fully connected layers and replacing such layers with convolution layers could yield a somewhat comparable performance with efficient and simple model design. In addition, dilated convolution manages to help give comparable performance despite such few number of parameters as it processes feature maps with considering context. This alone has been demonstrated in [9] with different CNN model and shows the robustness of dilated convolution to be used by other CNN models. In addition, using global average pooling is assumed to greatly impact classification performance as all feature maps are forced to be representative of the final confidence score.

The last insight is although operating on higher image resolution, Darknet models are found to be incapable of matching VGG-16 and VGG-16-D's respective performances, despite Darknet-19 and Darknet-53, respectively, being comparable and beating VGG-16's performance on ImageNet dataset as reported in their respective publications. One reason behind this is VGG-16's result is obtained by fine-tuning VGG-16's ImageNet pretrained weights and VGG-16-D is obtained by fine-tuning the VGG-16's weights, whereas Darknet-19 and Darknet-53 did not utilize such weights. This factor may contribute to the anomaly of Darknet-19 and Darknet-53 of not obtaining better results than that of pretrained VGG-16 and VGG-16-D.

In addition to those insights, the authors noticed several hypothesis confirmation and unique observations during our experiments. The first is training on imbalanced data yielded very poor results. All Darknet models attained accuracy of 2%, whereas on using balanced data Darknet models managed to achieve results as provided in Table 6. Second, although this has been shown in various implementations, using SGD optimizer on Darknet models resulted in worse performances at around 45–48% accuracy only. Using Adam with lower learning rate is found to be the best combination for Darknet models implemented in PyTorch.

CONCLUSIONS

In this paper, classification experiments on GroZi-120 dataset have been presented and discussed. By utilizing CNN from existing well-known models such as VGG-16 and Darknet, generally better results can be obtained considering varying comparison baselines.

The best accuracy this paper's experiments can obtain on GroZi-120, which has very distinct data distribution on training and evaluation sets, is 66.9167% by VGG-16 operating at 300 × 300 input image resolution. A new model named VGG-16-D which transforms VGG-16's fully connected layers to dilated convolution and vanilla convolution layers combined with global average pooling performs competitively with 66.833% accuracy while having much lower number of parameters. Other results from Darknet models show adequate performance despite being trained from scratch.

Some future works may include investigating other CNN models which can be fine-tuned for other computer vision tasks to enable more diverse use cases for such CNN models. Implementing LCA layer and maximum entropy loss on the proposed solutions is another possible work to be done in the future. Lastly, fine-tun-

ing these models for detection tasks may be beneficial for other use cases in retail stores, such as helping vision-disabled parties in shopping or evaluating product placement implementation of a specified strategy.

COMPLIANCE WITH ETHICAL STANDARDS

This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the *PRIA* Editorial Board decides not to accept it for publication.

Conflict of Interest

The process of writing and the content of the article do not give grounds for raising the issue of a conflict of interest.

REFERENCES

1. J. Deng, W. Dong, R. Socher, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, 2009* (IEEE, 2019), pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848
2. M. George and C. Floerkemeier, "Recognizing products: A per-exemplar multi-label image classification approach," in *Computer Vision–ECCV 2014*, Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Lecture Notes in Computer Science, vol. 8690 (Springer, Cham, 2014), pp. 440–455. https://doi.org/10.1007/978-3-319-10605-2_29
3. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016* (IEEE, 2016), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
4. G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature." arXiv:1207.0580 [cs.NE]
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS'12: Proc. 25th Int. Conf. on Neural Information Processing Systems, Lake Tahoe, Nev., 2012*, Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Red Hook, N.Y., 2012), Vol. 1, pp. 1097–1105.
6. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**, 2278–2234 (1998). https://doi.org/10.1109/5.726791
7. Y. Lecun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs.* http://yann.lecun.com/exdb/mnist
8. Q. Li, X. Peng, L. Cao, W. Du, H. Xing, and Y. Qiao, "Product image recognition with guidance learning and noisy supervision," Comput. Vision Image Understanding **196**, 102963 (2019). https://doi.org/10.1016/j.cviu.2020.102963
9. M. Lin, Q. Chen, and S. Yan, "Network in network." arXiv:1312.4400v3 [cs.NE]
10. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multiBox detector," in *Computer Vision–ECCV 2016*, Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling, Lecture Notes in Computer Science, vol. 9905 (Springer, Cham, 2016), pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
11. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th Int. Conf. on Computer Vision, Corfu, 1999* (IEEE, 1999), Vol. 2, pp. 1150–1157. https://doi.org/10.1109/ICCV.1999.790410
12. D. G. Lowe, "Distinctive image features from scale-invariant key points," Int. J. Comput. Vision **60**, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
13. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proceedings of the 30th Int. Conf. on Machine Learning, Atlanta, 2013.*
14. M. Merler, C. Galleguillos, and S. Belongie, "Recognizing groceries in situ using in vitro training data," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Minneapolis, 2007* (IEEE, 2007), pp. 1–8. https://doi.org/10.1109/CVPR.2007.383486
15. R. Rajagopal, "Comparative analysis of COVID-19 X-ray images classification using convolutional neural network, transfer learning, and machine learning classifiers using deep features," Pattern Recognit. Image Anal. **31**, 313–322 (2021). https://doi.org/10.1134/S1054661821020140
16. J. Redmon, Darknet: Open source neural networks in C (2013). https://pjreddie.com/darknet/
17. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2015* (IEEE, 2015), pp. 779–788. https://doi.org/10.1109/CVPR.2016.91
18. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 2017* (IEEE, 2017), vol. 1, pp. 6517–6525. https://doi.org/10.1109/CVPR.2017.690
19. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." arXiv:1804.02767 [cs.CV]
20. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," Int. J. Comput. Vision **115**, 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
21. B. Santra, A. Paul, and D. P. Mukherjee, "Deterministic Dropout for Deep Neural Networks Using Composite Random Forest," Pattern Recognit. Lett. **131**, 205–212 (2020). https://doi.org/10.1016/j.patrec.2019.12.023
22. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations, 2015.* arXiv:1409.1556 [cs.CV]
23. M. M. Srivastava, "Bag of tricks for retail product image classification," *Image Analysis and Recognition. ICIAR 2020*, Ed. by A. Campilho, F. Karray, and

Z. Wang, Lecture Notes in Computer Science, vol. 12131 (Springer, Cham, 2020), pp. 71–82. https://doi.org/10.1007/978-3-030-50347-5_8

24. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learning Res. **15**, 1929–1958 (2014).

25. A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," in *31st Conf. on Neural Information Processing Systems (NIPS 2017), Long Beach, Calif., 2017*, Ed. by U. von Luxburg, I. Guyon, S. Bengio, H. Wallach, and R. Fergus (Curran Associates, Red Hook, N. Y., 2017), pp. 4151–4161.

26. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, Hawaii, 2017* (IEEE, 2017), pp. 5987–5995. https://doi.org/10.1109/CVPR.2017.634

27. F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, 2017* (IEEE, 2017), pp. 636–644. https://doi.org/10.1109/CVPR.2017.75

Jonathan is a candidate of Master of Computer Science at Bina Nusantara University's BINUS Graduate Program. He has published several researches on computer vision, pattern recognition, and software engineering on various international journals. His research interests include computer vision, pattern recognition, deep learning, and software engineering.

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Research Coordinator in Computer Science Department, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R–A*STAR, Singapore. His research interests include pattern recognition, machine learning, face recognition, appearance-based object recognition, mobile learning, and gamification of learning.