

# Classification of C2C e-Commerce Product Images using Deep Learning Algorithm

Herdian<sup>1</sup>, Gede Putra Kusuma<sup>2</sup>, Suharjito<sup>3</sup>

Computer Science Department  
BINUS Graduate Program-Master of Computer Science  
Bina Nusantara University, Jakarta, Indonesia, 11480

**Abstract**—C2C (consumer-to-consumer) is a business model where two individuals transact or conduct business with each other using a platform. A consumer act as a seller put their product in a platform later will be displayed to another consumer act as a buyer. This condition encourages platform to maintain high quality product information especially image that is provided by the seller. Product images need to be relevant to the product itself. It can be controlled automatically using image classification. In this paper, we carried out a research to find out the best deep learning model in image classification for e-commerce products. A dataset of 12,500 product images is collected from various web sources to be used in training and testing process. Five models are selected and fine-tuned using a uniform hyperparameter set-up. Those hyperparameters are found by using a manual process by trying a lot of hyperparameters. The testing result from every model is presented and evaluated. The result shows that NASNetLarge yield the best performance among all evaluated models with 84% testing accuracy.

**Keywords**—Image classification; e-commerce; product images; deep learning; hyperparameter tuning

## I. INTRODUCTION

The current adoption of e-commerce in Indonesia is high. Das et al. [1] mentioned in 2016, 78% or more than 80 million users had made online purchases. The online transactions that occur in Indonesia, one of which occurs in the C2C business model, where two individuals seller and buyer transact with each other [2].

In the C2C business model, there is a platform that mediates between sellers and buyers. The seller advertises their products on a platform, which will then be seen by the buyers. This causes weak control of the information contained in the products displayed on the platform to the buyers [2]. Product information, such as product image is an important factor for successful transaction. Several researches show that the image of the product is very important in buying interest [3], [4], [5]. Therefore, a mechanism is needed to maintain the quality of product images uploaded by the seller. An automatic approach using image classification method can be used to achieve it.

There are many algorithms that can be used for image classification. One algorithm that is currently popular to solve image classification problems is Convolutional Neural Network (CNN) [6], [7], which is one of the deep learning algorithms. Deep learning itself in recent years has received a

lot of attention from researchers, communities and industry. Deep learning is able to provide excellent results for various tasks such as the traffic signs identification [8], mandarin letter writing identification [9], etc.

Deep learning implementation from scratch requires huge amount of dataset. This can be an obstacle because the huge data collection requires a lot of resource. Besides that, in various cases, the data needed is difficult to collect. Under these conditions, deep learning can still be applied by transfer learning method. Transfer learning refer to the situation where what has been learned in one setting is exploited to improve generalization in another setting [10]. In case of image classification using deep learning, this can be done by using a previously trained model that is often called the pre-trained model and fine-tuned that model using a target dataset.

The use of pre-trained models is very helpful because it can save time and costs of the training process. In this study several pre-trained models will be re-trained using fine-tuning method on a 12,500 product images dataset. Then their performances will be compared based on the testing accuracy. Although there are previous studies related to image classification using deep learning, it is relatively difficult to find study on the image classification process that addresses everything from data collection to the use of several models and comparing the results, specifically in the area of e-commerce.

The rest of this paper is organized as follows. Previous works on related topics are presented in Section II. Background theories of the pre-trained models, which are used in our research, are described in Section III. We describe our datasets in Section IV. Experiments and Results are presented in Sections V and VI, respectively. Discussion on the experimental results is presented in Section VII. Finally, we conclude our paper in Section VIII.

## II. RELATED WORK

There are several papers related to the importance of the product information's quality on e-commerce, such as [11], [12] and [13]. Those papers discuss the quality of information, which includes content accuracy, completeness, and relevance. The accuracy, which is one of the dimensions of information quality, represents the perception of consumers that the information presented for a product or other content on the platform is true [13]. One of the product information dimension that will help provide a buyer understanding of a

product is image. In the process of creating product information, there may be a mistake that causes a decrease in the quality of product information. So to prevent this, a mechanism to maintain the quality of information or product images remain relevant and accurate is needed. One way is to automatically classify which images are good and which of them are not good.

Regarding the image classification itself, paper [14] provides an overview of the process of image classification. Generally, it covers image pre-processing, feature extraction, and classification. Image pre-processing is needed before an image is analysed. It can be in the form of image normalization. The feature extraction than do the image transformation to understand the image. In the end, the classification process is done to identify an image as a class from a group of classes.

Various deep learning algorithms have been used for image classification, such as MobileNet, NASNet, and DenseNet. MobileNet is a small-sized model optimized for use on mobile devices [15], [16]. Although the size of the model is small but its performance on Imagenet outperforms GoogleNet, which was the winner of the 2014 ILSVRC. There is also NASNet, which managed to match the performance of SENet, which is the winner of ILSVRC 2017 [17]. There is also DenseNet, which performed as good as ResNet, which is the winner of ILSVRC 2015 [18].

### III. PRE-TRAINED MODELS

Imagenet pre-trained models are used for image classification. The experiments are performed using Keras library. All models are available in the Keras library. The pre-trained models are MobileNetV1 [15], MobileNetV2 [16], NASNetMobile [17], NASNetLarge [17], and DenseNet121 [18]. Transfer learning process will be done using fine-tuning method for the 5 models. The last layers that are related to the classification layer will not be included but replaced by new layers.

#### A. MobileNetV1

MobileNet is the smallest model among the other 4 models. It contains around 4 million parameters. It has a total of 87 layers without top layers that are related to classification layers. The last layers related to classification are removed and replaced using new layers that is suitable for the category in the target dataset. For MobileNetV1 the first 75 layers will be frozen and will use 224x224 as input dimension. It's layer architecture generally consist of several block of depthwise separable convolution as shown in Fig. 1.

#### B. MobileNetV2

This is an improvement to MobileNetV1. For the ImageNet dataset, this newly MobileNet architecture improves the state of the art for wide range of performance points [16]. Using Keras library, total layer of MobileNetV2 is 155 layers without top layers. The first 135 layers will be frozen and will use 224x224 as input size. In MobileNetV2, there are two types of blocks. One is residual block with stride of 1. Another one is

block with stride of 2. Those two blocks are stacked to form MobileNetV2 as shown in Fig. 2.

#### C. NASNetMobile

NASNetMobile is one of the variants of NASNet architecture for mobile platforms. Based on the Keras version 2.2.4, NASNetMobile has a total of 769 layers. It has more layers than MobileNet models. The number of layers that will be frozen is 724 and will use input dimension similar with MobileNet that is 224x224. NASNet architecture consist of several cell stacked together: normal cell and reduction cell. For NASNetMobile, every 4 normal cell stacked together followed by a reduction cell, as shown in Fig. 3.

#### D. NASNetLarge

This is the largest model from NASNet. This model achieved top-1 accuracy for ImageNet at 82.7%. This performance is similar to SENet as the winner of ILSVRC 2017. Based on Keras library, this model has 1039 layers. The first 950 layers will be frozen. This model will use 331x331 input size. NASNetLarge has similar architecture with NASNetMobile it is consist of several cells stacked together.

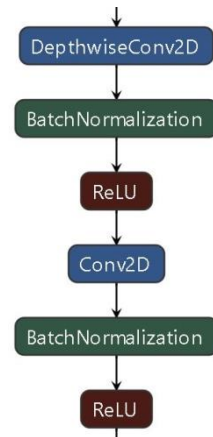


Fig. 1. Depthwise Separable Convolution Block.

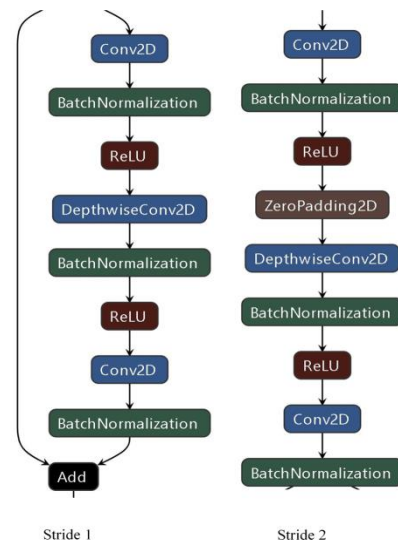


Fig. 2. MobileNetV2 Architecture Blocks.

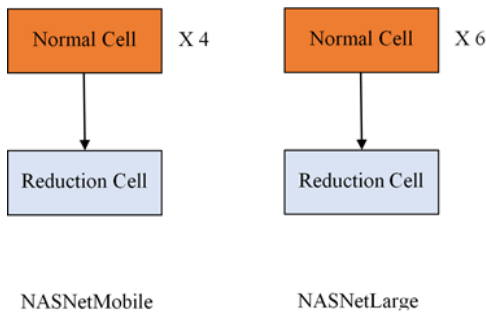


Fig. 3. NASNet Cells.

E. DenseNet121

The smallest variant of DenseNet available in Keras library is chosen. It has been shown to yield performance similar to ResNet101, but with less parameters [18]. Based on Keras library, it has a total of 427 layers. The first 411 layers will be frozen and will use input size of 224x224. DenseNet121 consist of 4 dense blocks. Every dense block consists of several convolution block. Fig. 4 shows the convolution block that is stacked together to construct a dense block.

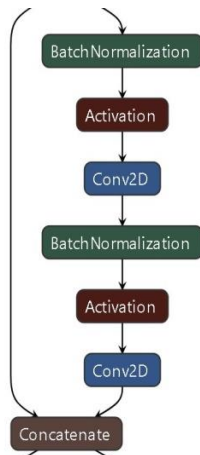


Fig. 4. Convolution Block in Densenet.

IV. DATASETS

A total of 12,500 product images are collected from several C2C e-commerce websites. The collection process is done manually by visiting a web page that contains images related to a category. For example, to collect images for trouser category, a webpage containing all trouser images is visited and then by using a chrome extension all images are downloaded and saved to a local folder. This process is repeated until a category contains desired total number of images. Fig. 5 shows the process of collecting the product images for the dataset.

The dataset is organized into two sub-groups: 10,000 training data and 2,500 testing data. Each sub-group has 10 balance categories with 1,000 images for each category in training dataset and 250 images for each category in testing dataset. Validation dataset will be obtained by performing a random split on the training dataset with split ratio of 0.2. Fig. 6 shows the dataset distribution across all categories.

Meanwhile, Fig. 7 shows sample images from the collected dataset.

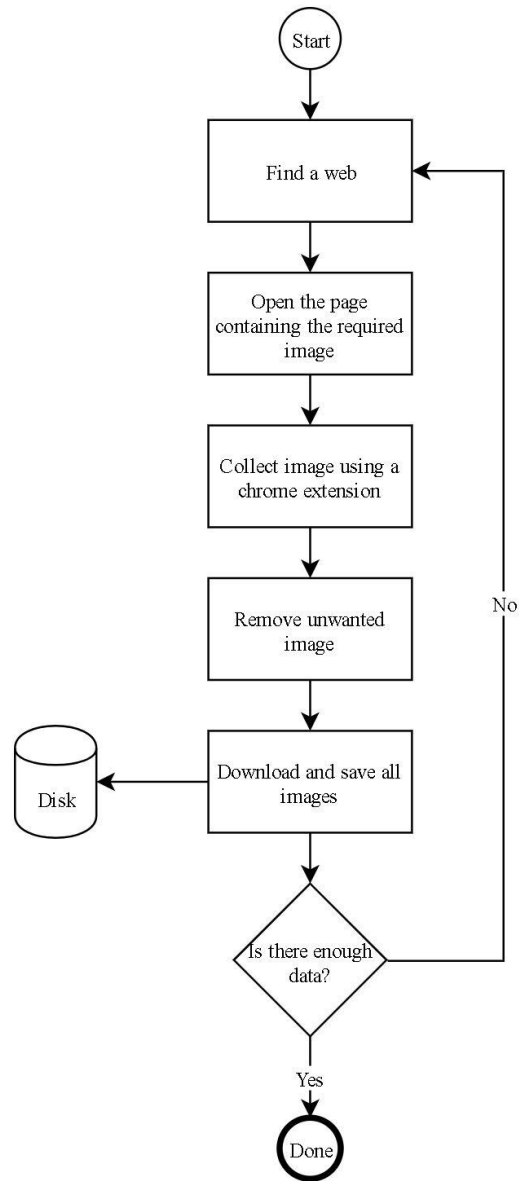


Fig. 5. Images Dataset Collection Process.

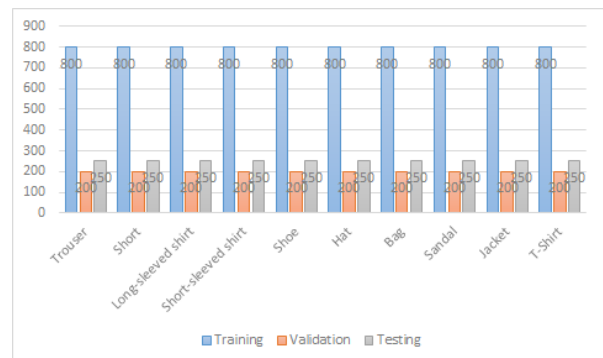


Fig. 6. The Distribution of 12,500 Product Images Dataset.

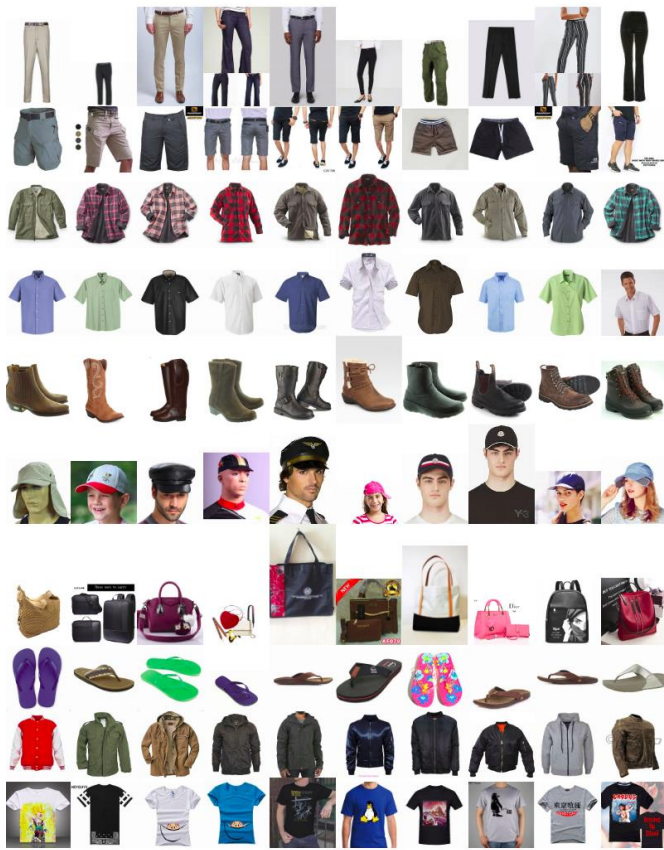


Fig. 7. Sample of 12,500 Product Images Dataset.

## V. EXPERIMENTS

Training for each model is done using a uniform hyperparameter set-up. Those hyperparameters is found by using a manual process by trying a lot of hyperparameter and training a model using each hyperparameter. The number of epoch is set to 100. It is chosen due to the fact that some models beginning to show overfitting at the end of 100 epoch. Also, NASNetLarge requires more time for training.

The experiment for every model is performed according to the following steps:

- 1) *Finding hyperparameter*: This is a set of activities to find suitable hyperparameter to train all models.
- 2) *Data augmentation*: This is performed to the training and validation data.
- 3) *Model training*: It uses Keras' `fit_generator` to feed sample data using batch to the model.
- 4) *Model testing*: The model then tested using 2,500 testing sub-group from the dataset to find out testing accuracy.

These steps are applied similarly to all 5 models. The differences are just in the input dimension and the total number of frozen layers.

### A. Finding Hyperparameter

Model training is done using transfer learning. A fine-tuning process is performed for all models using the same

hyperparameter. Those hyperparameter are learning rate, validation split, and the number of bottleneck layers from the original model being trained. It can be per block or per cell. Validation split is the number of validation data ratio from the training data. Meanwhile, number of bottleneck layer refers to how many layers can be re-trained from the original model architecture. These values are obtained by experimenting with several different hyperparameter values to one model and the results are compared. After the optimal values are obtained, the values are implemented for the other models. The model chosen for training to find hyperparameter values is MobileNet. This is due to the fact that MobileNet requires relatively faster training time compared to the other models.

### B. Data Augmentation

The image dataset will be loaded batch by batch, then for every batch of images, data augmentation process is performed. It is used to introduce variation and noise to the model. The input image is rotated, shifted, shear, zoomed, and flipped horizontally. Augmentation process will not be performed for the testing data. Fig. 8 shows the data after augmentation process.

### C. Model Training

The training process uses transfer learning method for 5 pre-trained models and uses the Tensorflow and Keras library. The 5 pre-trained models are available in the Keras library. The training process is done one-by-one for each model. The original weight of the model is used, and the classification layer of the original model is not included, so that it is possible to add new classification layer that matches the 12,500 product images dataset characteristic. Training process is performed for 100 epochs using the same hyperparameter values for all 5 models.

### D. Model Testing

After the training process is complete, it continues with the testing process using 2,500 testing images from the dataset. The result of testing accuracy from this process will be evaluated and compared to find out which model has the best performance.



Fig. 8. Sample Data after Augmentation Process.



## VI. RESULTS

This section provides the results from every experiments, which includes validation and testing performances.

### A. Finding Hyperparameter

The result of this process is optimal hyperparameter values for learning rate, validation split, and also number of bottleneck layer included in the training.

There are three values compared for every hyperparameter. Each value is then used to train MobileNet and the most optimal hyperparameter, based on validation accuracy, will be used to train another models. Learning rate are compared at 0.001, 0.0001, and 0.00001. The result shows that learning rate of 0.0001 gives optimal result based on validation accuracy. Validation split are evaluated at 0.20, 0.25, and 0.30. The most optimal result is validation split ratio of 0.20. Different numbers of bottleneck layer can be included in re-training process are also tested. The first experiment uses 0 layer, which means all layer from original model architecture is frozen. The second training is started from the last 1 block, and the third one is from the last 2 blocks. The result shows that the training from the last 2 blocks gives the most optimal result. It yields the highest validation accuracy after 100 epochs.

### B. MobileNetV1

MobileNetV1 gives validation accuracy of 89.34% and testing accuracy of 82%. Fig. 9 shows the confusion matrix of the testing results for MobileNetV1.

Table I summarizes the classification report on testing dataset for MobileNetV1. As shown in the table, it gives an average precision of 79%, average recall of 77%, and average f1-score of 77%.

### C. MobileNetV2

MobileNetV2 gives validation accuracy of 78.15% and testing accuracy of 75%. Fig. 10 shows the confusion matrix of the testing results for MobileNetV2.

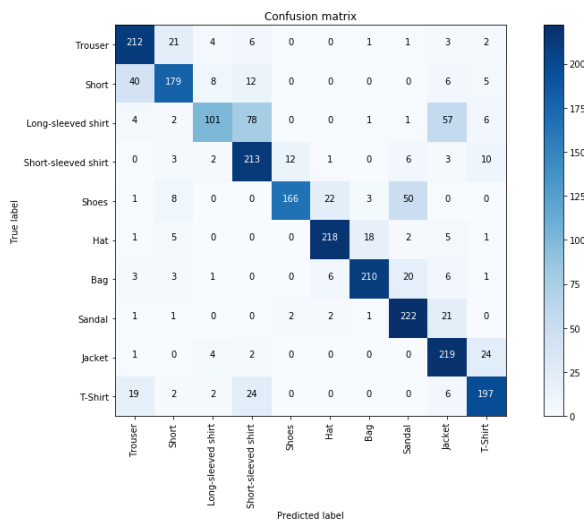


Fig. 9. Confusion Matrix for MobileNetV1.

TABLE. I. CLASSIFICATION REPORT OF MOBILENETV1

| Category            | Precision  | Recall     | F1-Score   |
|---------------------|------------|------------|------------|
| Trouser             | 75%        | 85%        | 80%        |
| Short               | 80%        | 72%        | 76%        |
| Long-sleeved shirt  | 83%        | 40%        | 54%        |
| Short-sleeved shirt | 64%        | 85%        | 73%        |
| Shoes               | 92%        | 66%        | 77%        |
| Hat                 | 88%        | 87%        | 87%        |
| Bag                 | 90%        | 84%        | 87%        |
| Sandal              | 74%        | 89%        | 80%        |
| Jacket              | 67%        | 88%        | 76%        |
| T-Shirt             | 80%        | 79%        | 79%        |
| <b>Average:</b>     | <b>79%</b> | <b>77%</b> | <b>77%</b> |

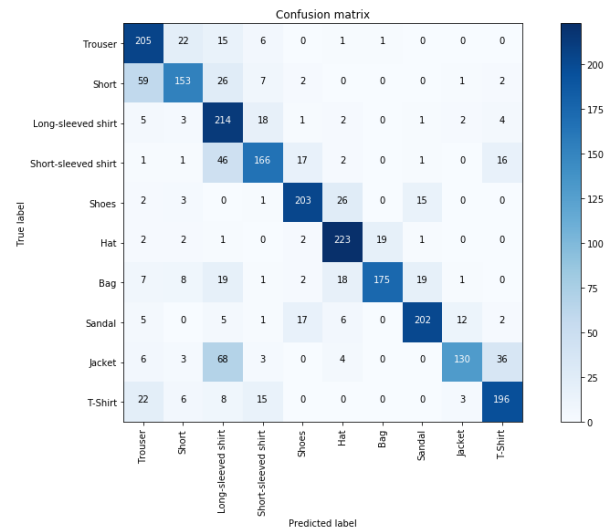


Fig. 10. Confusion Matrix for MobileNetV2.

Table II summarizes the classification report on testing dataset for MobileNetV2. As shown in the table, it gives an average precision of 77%, average recall of 75%, and average f1-score of 75%.

TABLE. II. CLASSIFICATION REPORT OF MOBILENETV2

| Category            | Precision  | Recall     | F1-Score   |
|---------------------|------------|------------|------------|
| Trouser             | 65%        | 82%        | 73%        |
| Short               | 76%        | 61%        | 68%        |
| Long-sleeved shirt  | 53%        | 86%        | 66%        |
| Short-sleeved shirt | 76%        | 66%        | 71%        |
| Shoes               | 83%        | 81%        | 82%        |
| Hat                 | 79%        | 89%        | 84%        |
| Bag                 | 90%        | 70%        | 79%        |
| Sandal              | 85%        | 81%        | 83%        |
| Jacket              | 87%        | 52%        | 65%        |
| T-Shirt             | 77%        | 78%        | 77%        |
| <b>Average:</b>     | <b>77%</b> | <b>75%</b> | <b>75%</b> |

D. NASNetMobile

NASNetMobile gives validation accuracy of 84% and testing accuracy of 78%. Fig. 11 shows the confusion matrix of the testing results for NASNetMobile.

Table III summarizes the classification report on testing dataset for NASNetMobile. As shown in the table, it gives an average precision of 79%, average recall of 78%, and average f1-score of 78%.

E. NASNetLarge

NASNetLarge is the biggest model in term of architecture. It gives validation accuracy of 90.69% and testing accuracy of 84%. Fig. 12 shows the confusion matrix of the testing results for NASNetLarge.

Table IV summarizes the classification report on testing dataset for NASNetLarge. As shown in the table, it gives an average precision of 84%, average recall of 84%, and average f1-score of 84%.

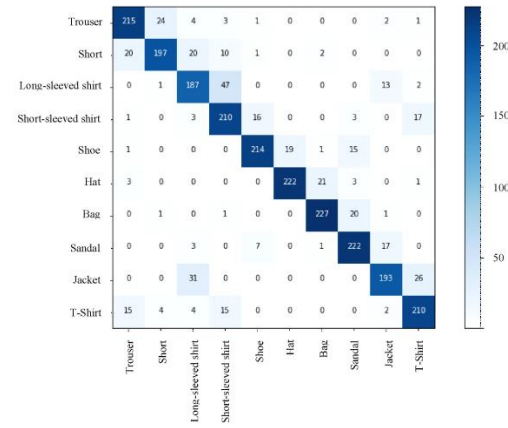


Fig. 12. Confusion Matrix for NASNetLarge.

TABLE IV. CLASSIFICATION REPORT OF NASNETLARGE

| Category            | Precision  | Recall     | F1-Score   |
|---------------------|------------|------------|------------|
| Trouser             | 84%        | 86%        | 85%        |
| Short               | 87%        | 79%        | 83%        |
| Long-sleeved shirt  | 74%        | 75%        | 75%        |
| Short-sleeved shirt | 73%        | 84%        | 78%        |
| Shoes               | 90%        | 86%        | 88%        |
| Hat                 | 92%        | 89%        | 90%        |
| Bag                 | 90%        | 91%        | 90%        |
| Sandal              | 84%        | 89%        | 87%        |
| Jacket              | 85%        | 77%        | 81%        |
| T-Shirt             | 82%        | 84%        | 83%        |
| <b>Average:</b>     | <b>84%</b> | <b>84%</b> | <b>84%</b> |

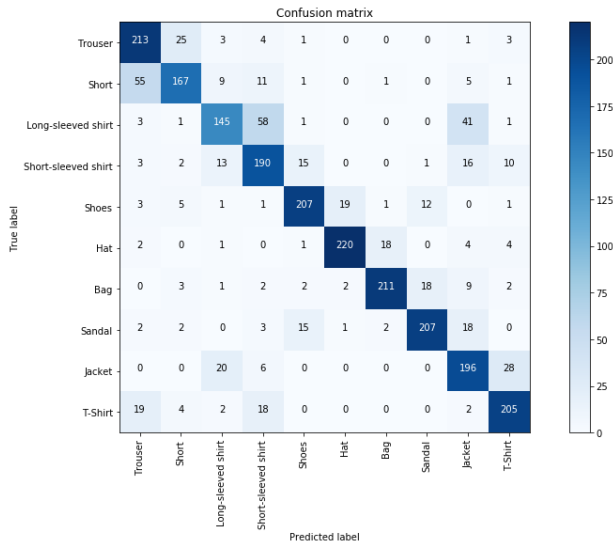


Fig. 11. Confusion Matrix for NASNetMobile.

TABLE III. CLASSIFICATION REPORT OF NASNETMOBILE

| Category            | Precision  | Recall     | F1-Score   |
|---------------------|------------|------------|------------|
| Trouser             | 71%        | 85%        | 77%        |
| Short               | 80%        | 67%        | 73%        |
| Long-sleeved shirt  | 74%        | 58%        | 65%        |
| Short-sleeved shirt | 65%        | 76%        | 70%        |
| Shoes               | 85%        | 83%        | 84%        |
| Hat                 | 91%        | 88%        | 89%        |
| Bag                 | 91%        | 84%        | 87%        |
| Sandal              | 87%        | 83%        | 85%        |
| Jacket              | 67%        | 78%        | 72%        |
| T-Shirt             | 80%        | 82%        | 81%        |
| <b>Average:</b>     | <b>79%</b> | <b>78%</b> | <b>78%</b> |

F. DenseNet121

DenseNet121 is the smallest variant from DenseNet model. It is comparable to other mobile models. DenseNet121 gives validation accuracy of 84.09% and testing accuracy of 75%. Fig. 13 shows the confusion matrix of the testing results for DenseNet121.

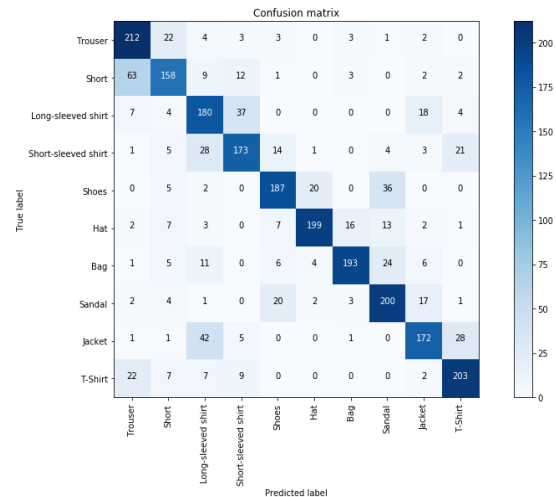


Fig. 13. Confusion Matrix for DenseNet121.

Table V summarizes the classification report on testing dataset for DenseNet121. As shown in the table, it gives an average precision of 76%, average recall of 75%, and average f1-score of 75%.

Based on the experimental results of the 5 pre-trained models above, we can see that NASNetLarge has the best performance with testing accuracy of 84%. NASNetLarge also gives highest precision, recall, and f1-score at 84%. The validation and testing performances of the 5 evaluated models are summarized in Table VI.

During training, at the last epoch, NASNetLarge also yields the lowest validation loss among all evaluated models at 0.2910 and the highest validation accuracy of 90.69%. The NASNetLarge results have correlation with its original performance on ImageNet. It is the best among the evaluated models in this paper. It is also yield jointly best performance with winner of ILSVRC 2017.

Fig. 14 shows the validation accuracy graphs and Fig. 15 shows the validation loss graph for all 5 models. From these figures, we can see that NASNetLarge performance is also better than the other models during training. It consistently achieves the highest validation accuracy and also the lowest validation loss at every epoch. However, the main drawback of NASNetLarge is its training time. It requires much more time to train compared to the other models.

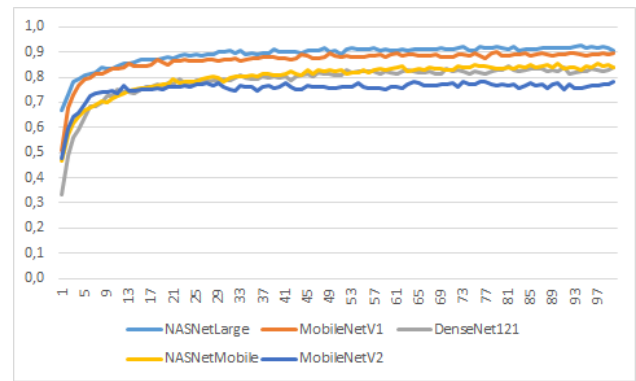


Fig. 14. Validation Accuracy for 100 Epochs.

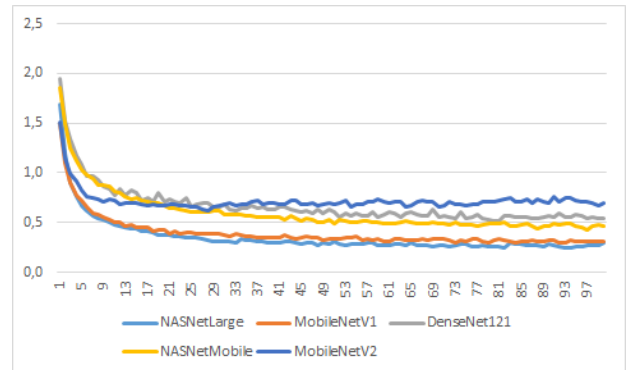


Fig. 15. Validation Loss for 100 Epochs.

TABLE. V. CLASSIFICATION REPORT OF DENSENET121

| Category            | Precision  | Recall     | F1-Score   |
|---------------------|------------|------------|------------|
| Trouser             | 68%        | 85%        | 76%        |
| Short               | 72%        | 63%        | 68%        |
| Long-sleeved shirt  | 63%        | 72%        | 67%        |
| Short-sleeved shirt | 72%        | 69%        | 71%        |
| Shoes               | 79%        | 75%        | 77%        |
| Hat                 | 88%        | 80%        | 84%        |
| Bag                 | 88%        | 77%        | 82%        |
| Sandal              | 72%        | 80%        | 76%        |
| Jacket              | 77%        | 69%        | 73%        |
| T-Shirt             | 78%        | 81%        | 80%        |
| <b>Average:</b>     | <b>76%</b> | <b>75%</b> | <b>75%</b> |

TABLE. VI. MODELS PERFORMANCE RESULTS

| No. | Models       | Validation Accuracy | Validation Loss | Testing Accuracy |
|-----|--------------|---------------------|-----------------|------------------|
| 1   | NASNetLarge  | 90.69%              | 0.2910          | 84%              |
| 2   | MobileNetV1  | 89.34%              | 0.3121          | 82%              |
| 3   | NASNetMobile | 84.00%              | 0.4653          | 78%              |
| 4   | DenseNet121  | 84.09%              | 0.5373          | 75%              |
| 5   | MobileNetV2  | 78.15%              | 0.6939          | 75%              |

## VII. DISCUSSION

A clothing dataset has been collected from various sources to be used for experiments on 5 models, which consist of training and testing processes. The training process is carried out to obtain an optimal model. It will then be used in the testing process to find out how the model's performance against the clothing dataset.

The classification performance of each model for the long-sleeved shirt class is the lowest compared to other classes. Many images of long-sleeved shirts are incorrectly classified as short-sleeved shirt or jacket. This can be caused by the three classes having similar characteristics. Several images of long-sleeved shirts and short-sleeved shirts are folded. Furthermore, long-sleeved shirts and jackets have the same characteristic, which is long sleeved.

NASNetLarge provided the best performance on ImageNet according to [17]. This also matches the experimental results in this study. NASNetLarge provides the best performance compared to the other 4 models. Another finding in this study is that MobileNetV1 provides a fairly good performance, which is ranked second under NASNetLarge, but with a much smaller model size [15]. This model can be considered during the implementation process due to its light weight.

## VIII. CONCLUSIONS

In this work, e-commerce product images classification has been demonstrated using deep learning algorithm. We have collected and labelled a total of 12,500 product images dataset. The images were crawled from several C2C e-commerce websites. Five deep learning models have been evaluated on the dataset, which include MobileNetV1, MobileNetV2, NASNetMobile, NASNetLarge, and DenseNet121.

Based on the experimental results, NASNetLarge achieves the best performance for image classification with testing accuracy of 84%. Also, it shows the best performance during training with validation accuracy of 90.69%. It outperforms the other four models that are trained using similar hyperparameter. However, this performance comes with a cost of larger architecture and longer training time compared to the other models.

Further research related to this study can be done by making variations to the dataset, using imbalanced dataset, or increasing the number of images. Another research can also be done by conducting experiments using different sets of hyperparameters.

## ACKNOWLEDGMENT

This work was supported by the Directorate General of Strengthening for Research and Development, Ministry of Research, Technology, and Higher Education, Republic of Indonesia, through research grant number 23/AKM/MONOPNT/2019.

## REFERENCES

- [1] K. Das, M. Gryseels, P. Sudhir, and K. T. Tan, "Unlocking Indonesia's digital opportunity," no. October, pp. 1–28, 2016.
- [2] C. Dan, "Consumer-To-Consumer (C2C) Electronic Commerce: The Recent Picture," *Int. J. Networks Commun.*, 2014.
- [3] X. Li, M. Wang, and Y. Chen, "the Impact of Product Photo on Online Consumer Purchase Intention: an Image-Processing Enabled Empirical Study," *PACIS 2014 Proc.*, 2014.
- [4] C. Lumb, "The Customer Decision Process and User Interaction in E-commerce," 2014.
- [5] E. Huang and C.-C. Liu, "A Study on Trust Building and Its Derived Value in C2C E-Commerce," *J. Glob. Bus. Manag.*, 2010.
- [6] A. KrizhKrizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9. Sutskever, I., Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, 2012.
- [7] J. Patterson and A. Gibson, *Deep Learning A Practitioner's Approach*, 1st ed. boston: oreilly, 2016.
- [8] D. Cireşan, U. Meier, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proceedings of the International Joint Conference on Neural Networks*, 2011.
- [9] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *Int. Conf. Pattern Recognit.*, 2012.
- [10] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*. 2016.
- [11] T. Singh Chhikara, "Information Quality -Crucial Aspect of E-Commerce," *IOSR J. VLSI Signal Process. Ver. II*, 2015.
- [12] W. H. DeLone and E. R. Mclean, "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update," *J. Manag. Inf. Syst. / Spring*, vol. 19, no. 4, pp. 9–30, 2003.
- [13] B. H. Wixom and P. A. Todd, "A Theoretical Integration of User Satisfaction and Technology Acceptance," *Info. Sys. Res.*, 2005.
- [14] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *Int. J. Adv. ...*, 2013.
- [15] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *arxiv*, 2017.
- [16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [17] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.
- [18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.