

Letter

# Empirical Remarks on the Translational Equivariance of Convolutional Layers

Kyung Joo Cheoi <sup>1</sup>, Hyeonyeong Choi <sup>2</sup> and Jaepil Ko <sup>3,\*</sup>

<sup>1</sup> Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea; kjcheoi@chungbuk.ac.kr

<sup>2</sup> ICT-CRC, Kumoh National Institute of Technology, Gumi 39177, Korea; hychoi3609@kumoh.ac.kr

<sup>3</sup> Department of Computer Engineering, Kumoh National Institute of Technology, Gumi 39177, Korea

\* Correspondence: nonezero@kumoh.ac.kr; Tel.: +82-54-478-7529

Received: 30 March 2020; Accepted: 29 April 2020; Published: 1 May 2020



**Abstract:** In general, convolutional neural networks (CNNs) maintain some level of translational invariance. However, the convolutional layer itself is translational-equivariant. The pooling layers provide some level of invariance. In object recognition, invariance is more important than equivariance. In this paper, we investigate how vulnerable CNNs without pooling or augmentation are to translation in object recognition. For CNNs that are specialized in learning local textures but vulnerable to learning global geometric information, we propose a method to explicitly transform an image into a global feature image and then provide it as an input to neural networks. In our experiments on a modified MNIST dataset, we demonstrate that the recognition accuracy of a conventional baseline network significantly decreases from 98% to less than 60% even in the case of 2-pixel translation. We also demonstrate that the proposed method is far superior to the baseline network in terms of performance improvement.

**Keywords:** translational equivariant; translational invariant; complementary information; object recognition; convolutional neural networks

## 1. Introduction

Convolutional neural networks (CNNs) [1] have been successfully applied to computer vision applications such as object detection, recognition, segmentation and tracking. In particular, CNNs' success in object detection is due to their property of translational equivariance which allows multiple cats to be detected in a single image. In object recognition, invariance is more important than equivariance. The pooling layers, such as max-pooling and global average pooling [2], provide some level of invariance. However, data augmentation [3] is essential to ensure a practical level of invariance.

Recently, several methods have been introduced focusing on obtaining scaling and rotational equivariance as well as translational equivariance [4–6]. The capsule network [4] pointed out the problem of max-pooling and proposed the dynamic routing to replace it. The main concept of the capsule network is the grouping of a few convolutions. The concept of group convolution has been successfully adopted for efficiency in many convolutional networks [7–10]. There are still several open issues with the appropriate number of filter groups, overlapping filter groups, and group size.

Several methods on facial image analysis have reported that feature images tailored to a given specific application domain are effective as input [11–14]. As additional information, they use the local binary pattern (LBP) mapped coded image [11,12], the neighbor-centered difference image (NCDI) [13], and Gabor response maps [14]. The success of these methods is largely due to the capability of CNNs to effectively learn texture information. However, CNNs have been reported to be generally vulnerable when it comes to learning geometry information. To overcome this drawback, Geirhos et al. [15]

proposed the use of the style transfer network [16] for texture augmentation while preserving the main geometric shape. The success of this method indicates that CNNs are biased towards learning from local textures. Learning local texture does not require invariance of networks. Instead, equivariance is sufficient.

Geometric information as well as texture is important for object recognition. Therefore, the feature representation schemes have to obtain geometrical invariance. Several papers have applied the polar coordinate system to mitigate variations coming from geometric transforms [17–19]. This transform converts the rotational variant problem into the translational variant problem so that it can be easily learned by convolutional layers. Jaderberg et al. [20] proposed the localization and grid generator layers to cope with geometric transformations, but it requires mandatory augmentation to train the proposed layers.

In this paper, we investigate how vulnerable CNNs without pooling or augmentation are to translation in object recognition. In addition, motivated by the use of additional information as input to CNNs [7–11], we propose a method to explicitly transform an image into a global feature image and then provide it as input to the neural network since CNNs are specialized in learning local textures, but encounter difficulties when learning global geometric information. For this purpose, we firstly adopt the 2D-discrete Fourier transform (2D-DFT) [21] which transforms spatial images into the global invariant magnitude response images. We can use the fast Fourier transform (FFT) [22] algorithm to estimate 2D-DFT. For additional invariance, we can consider a mixture model [23] based on a statistical transform such as independent component analysis (ICA) [24]. Our experimental results clearly show that the convolution layers are extremely vulnerable to a few pixels of translation, and the proposed method is far superior to the baseline network in terms of the performance improvement.

This paper is organized as follows. Section 2 briefly reviews equivariance and invariance. In Section 3, we describe the properties of frequency images and three network structures for using frequency images as additional input. Section 4 presents an experimental setup to demonstrate the weaknesses of equivariant convolutional networks for translation and shows that the utilization of frequency images improves translational invariance. We give conclusion remarks in Section 5.

## 2. Equivariance versus Invariance

Formally, a function  $f$  is equivariant with respect to a transform  $T$  if  $f(T(x)) = T(f(x))$ . This means that applying the transformation  $T$  to  $x$  is equivalent to apply it to the result  $f(x)$ . Meanwhile, a function  $f$  is invariant with respect to a transform  $T$  if  $f(T(x)) = f(x)$ . In other words, the result by the function  $f$  does not change when you apply the transformation  $T$  to the input  $x$ . Figure 1 modified from [6] illustrates the translational equivariance. A transform  $T$  translates the star object of the input image  $I(x)$  into right side resulting in the transformed image  $T[I]$ . The function  $f$  is a feature representation of the input. The result  $T[f(I)]$  by applying the transform  $T$  to the feature representation  $f(I)$  is the same to the feature representation of the transformed image  $f(T[I])$ .

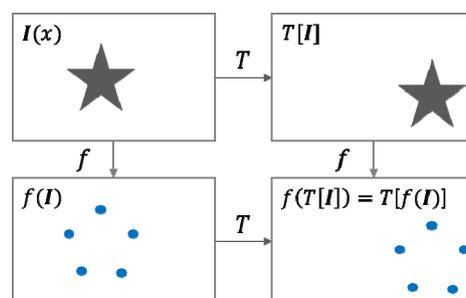


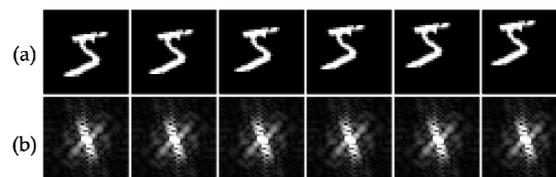
Figure 1. Illustration of a translational equivariance.

The CNNs achieve the property of translational equivariance by the concept of weight sharing, i.e., convolutional layers [10]. This equivariant property is effective for multiple occurrences of objects in an

image. Meanwhile, invariance is more important than equivariance in object recognition. The CNNs have some level of invariance by pooling layers, especially max-pooling layers, but it is not sufficient to obtain strong invariance. That is generally achieved by a proper data augmentation [2].

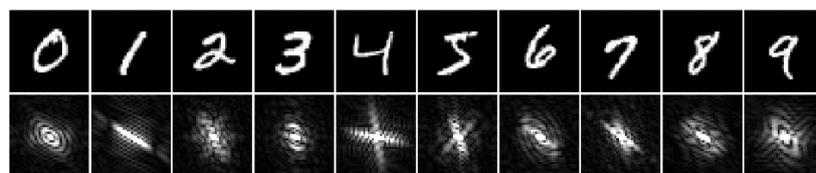
### 3. Fusion Strategy for Frequency Image as Complementary Information

According to the translational property of the Fourier transform known as the shift theorem [21], two images have the same magnitude when they differ by translation, i.e., the magnitude has a translational invariant property. Figure 2 shows spatial images and their corresponding frequency images (the log-scaled magnitude) with different translations where all the frequency images are the same, while the spatial images are shifted by 0 to 5 pixels, respectively. Here, we can easily get the frequency images by FFT [22].



**Figure 2.** Spatial images and their corresponding frequency images. (a) spatial images of a zero-pixel shift to 5-pixel shifts from the center, (b) their corresponding frequency images having the same appearance.

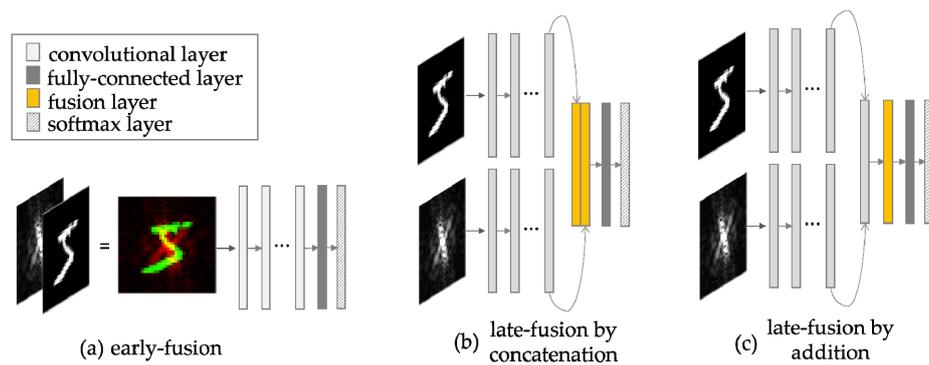
The frequency images are translational-invariant, so they are not affected by pixel shifts in spatial images. Therefore, we can regard that frequency images are appropriate for translational invariance. However, we should note that the discrimination power of frequency images is not strong because local information is lost in frequency images. The phase spectrum represents local information, but this information is initially excluded from frequency images. In Figure 3, we can notice that all the frequency images share the bright center area. This property of frequency images can reduce discrimination power.



**Figure 3.** Digit images in the first row and their corresponding frequency images in the second row where they share the bright center area.

Therefore, we propose to use both spatial images and frequency images as complementary information to CNNs for translation invariance. To utilize both images, we propose one early-fusion strategy and two late-fusion strategies. Figure 4 illustrates these strategies. The early-fusion strategy feeds CNNs the concatenation of spatial and frequency images. The late-fusion strategy creates a two-stream model with convolutional networks of the same structure, and then fuses the features of each stream followed by fully-connected layers. For the late-fusion strategies, we simply design a fusion layer for addition and concatenation respectively.

The early-fusion strategy learns features by feeding two heterogeneous sets of information together as the input. Meanwhile, the late-fusion strategy learns features by feeding each heterogeneous input image separately.



**Figure 4.** The fusion strategy. (a) early fusion by concatenating raw image and its frequency image, (b) late fusion by concatenation, (c) late fusion by addition.

#### 4. Experimental Results

In this section, we demonstrate how vulnerable convolutional layers are to translation, and that the proposed late-fusion strategy can compensate for convolutional layers of translational equivariant. Thus, we design the networks that consist of convolutional layers only for feature representation. Note that we do not adopt any max-pooling layers and we do not take any data augmentation. Such a simple architecture is advantageous to prove our goal instead of achieving a state-of-the-art performance.

##### 4.1. Dataset

The MNIST dataset [25] consists of 60,000 and 10,000 centered images for training and testing, respectively. The size of the images is  $28 \times 28$ . For analyzing translational invariance, we generate noncentered test images by applying  $N$ -pixel shifts. To prevent the digits from leaving image boundary, we convert the images to be  $38 \times 38$  size by padding 10 pixels. We use the 60,000 MNIST training dataset to train the networks. For test evaluation, we use  $N$ -pixel shifted 10,000 images generated from the original 10,000 test images in which we vary  $N$  from 0 to 5.

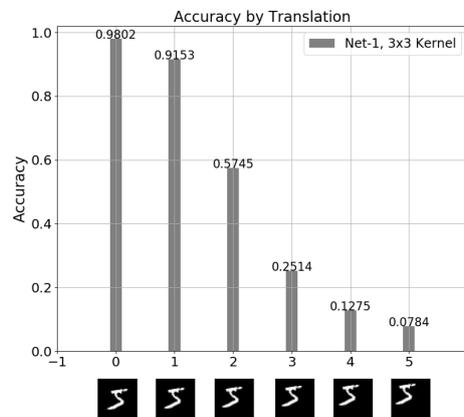
##### 4.2. Experimental Setup

We perform the comparative analysis for the four networks; the network trained with spatial images only, and three networks for the proposed fusion strategies. All the networks consist of convolutional layers followed by one fully connected layer and a softmax layer. For each network, we vary the number of convolutional layers by  $K$  times for deep architecture. For simplicity, we named the repeated  $K$  convolutional layers *Net-K*. For the maximum number of convolutional layers,  $K$  varies from 1 to 17 and the kernel size varies from  $3 \times 3$  to  $37 \times 37$ . The number of nodes for the fully connected layer is 32, and 10 nodes for the softmax layer. Without max-pooling and preserving the size of feature maps, a convolutional layer of  $3 \times 3$  kernel reduces the size of feature maps by 2 pixels, which results in 17 layers at maximum for  $38 \times 38$  input images. We focused on the representation capability of convolutional layers, so we set the number of nodes of the fully connected layer mainly in charge of classification to 32, the minimum number that can obtain 99% training accuracy. Finally, the number of nodes of the softmax layer is set to 10 considering 10 numbers to be classified.

We set training hyperparameters as follows: The RMSprop optimizer with an initial learning rate of 0.001 is used. The batch size is 5000, and the training epoch is 100. For a fair comparison, we fixed 100 epochs for all the experiments, where all the networks are sufficiently trained to achieve more than 99% training accuracy. We do not provide a loss-by-epoch graph since the MNIST dataset is well known for easy training, even by a simple network.

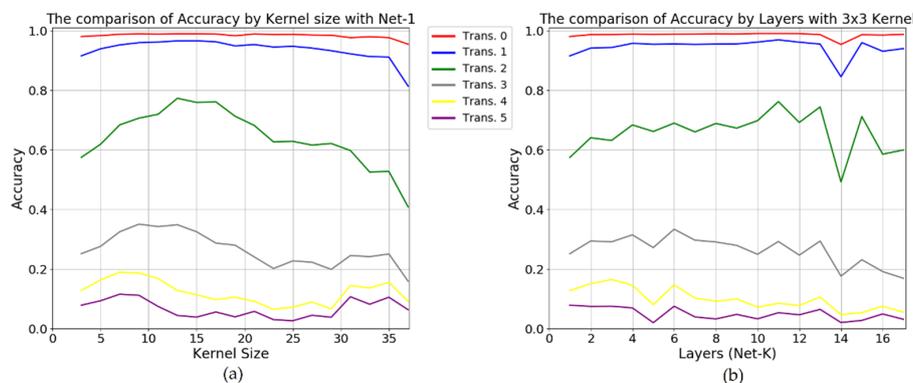
### 4.3. On the Translational Invariance of Spatial Image Trained Networks

In this experiment, the network took spatial images only as inputs where the input images are shifted from 0 to 5 pixels. Even if images are shifted by just 2-pixel, the accuracy of the Net-1 network drops dramatically to less than 60% as shown in Figure 5.



**Figure 5.** The test accuracy for the Net-1 network of  $3 \times 3$  kernels trained with spatial images only.

Even utilizing a deep network and a large receptive field, we obtained similar results. For deep networks, we trained the *Net-K* network with  $3 \times 3$  kernels by varying *K* from 1 to 17. For large receptive fields, we trained the Net-1 network of varying kernel sizes from  $3 \times 3$  to  $37 \times 37$ . The receptive fields of the network cover the entire image at 17 layers and 37 kernel size, respectively. Only 2-pixel translation (green line) significantly decreases accuracy regardless of the larger number of layers or larger kernel size as shown in Figure 6.



**Figure 6.** The accuracy of the networks tested with *N*-pixel translated (depicted by Trans. *N*) images. (a) for the deep networks with varying convolutional layers of up to 17 with the same  $3 \times 3$  kernel size, (b) for the Net-1 network with varying kernel sizes of up to 37.

From the experimental results in this section, we can remark that convolutional layers alone are extremely vulnerable to translation, even on larger kernel sizes and deeper networks.

In addition, we conducted experiments on all networks that can be combined with up to 17 layers and up to  $37 \times 37$  kernel size. All the results for these networks are almost the same as Figure 6, so they are not included in this paper.

### 4.4. On the Translational Invariance of Fusion Strategy

In this section, we compared the four networks by varying translation from 0 to 5 pixels. We selected three results; one for a shallow network with a small kernel size, another for a deep network with a

small kernel size, and the other for a shallow network with a large kernel size. The test error rates are shown in Tables 1–3, respectively.

**Table 1.** Test error rate for a shallow network with a small kernel size (Net-1 with  $3 \times 3$  kernel).

Translation by Pixel	No Fusion Baseline	Early Fusion		Late-Fusion Concatenation		Late-Fusion Addition	
0	1.98	1.66	(19.28)	1.74	(13.79)	1.56	(26.92)
1	8.47	6.04	(40.23)	5.08	(66.73)	5.03	(68.39)
2	42.55	30.63	(38.92)	25.59	(66.28)	26.56	(60.20)
3	74.86	66.62	(12.37)	58.06	(28.94)	60.66	(23.41)
4	87.25	81.06	(7.64)	73.08	(19.39)	79.48	(9.78)
5	92.16	85.75	(7.48)	79.14	(16.45)	86.05	(7.10)

No fusion denotes the network trained with spatial images only. Fusion denotes the network trained with both spatial and frequency images. The numbers in brackets denote performance improvement over the baseline.

**Table 2.** Test error rate for a deep network with a small kernel size (Net-8 with  $3 \times 3$  kernel).

Translation by Pixel	No Fusion Baseline	Early Fusion		Late-Fusion Concatenation		Late-Fusion Addition	
0	1.08	1.19	(−9.24)	1	(8.00)	0.99	(9.09)
1	4.51	3.67	(22.89)	2.95	(52.88)	4.07	(10.81)
2	31.16	26.67	(16.84)	19.17	(62.55)	29.62	(5.20)
3	70.93	71.15	(−0.31)	53.18	(33.38)	68.14	(4.09)
4	90.83	89.70	(1.26)	72.36	(25.53)	88.07	(3.13)
5	96.77	96.01	(0.79)	79.03	(22.45)	94.33	(2.59)

No fusion denotes the network trained with spatial images only. Fusion denotes the network trained with both spatial and frequency images. The numbers in brackets denote performance improvement over the baseline.

**Table 3.** Test error rate for a shallow network with a large kernel size (Net-1 with  $15 \times 15$  kernel).

Translation by Pixel	No Fusion Baseline	Early Fusion		Late-Fusion Concatenation		Late-Fusion Addition	
0	1.07	0.93	(15.05)	0.99	(8.08)	1.03	(3.88)
1	3.44	3.22	(6.83)	2.74	(25.55)	2.87	(19.86)
2	24.1	20.14	(19.66)	13.86	(73.88)	17.77	(35.62)
3	67.51	64.02	(5.45)	45.1	(49.69)	54.79	(23.22)
4	88.67	89.07	(−0.45)	69.23	(28.08)	79.05	(12.17)
5	96.15	95.69	(0.48)	79.18	(21.43)	87.12	(10.37)

No fusion denotes the network trained with spatial images only. Fusion denotes the network trained with both spatial and frequency images. The numbers in brackets denote the performance improvement over the baseline.

From Table 1, all the networks obtained over 98% in accuracy in the case of zero translation. This indicates that all the networks were successfully trained with the original centered training dataset. For all the networks, the error rates significantly increase as the pixel translation increases, especially after 2-pixel translation. However, focusing on relative performance improvement, we can note that all the fusion networks outperform the baseline network regardless of pixel translation. The late-fusion networks achieved at least 13.79% to 26.29% performance improvement in the absence of pixel translation. In addition, the concatenation network achieved a 66.28% performance improvement in the case of 2-pixel translation. We can remark that the fusion networks are far superior to the baseline network in terms of translational invariance.

Experimental results for a typical deep CNNs with a  $3 \times 3$  kernel size are shown in Table 2. The error rate was reduced compared to Table 1 for all cases. In the case of early fusion, there are cases where the performance improvement is lower than the baseline network. Overall, all the fusion networks are superior to the baseline network. For late fusion by concatenation, it is still far superior to

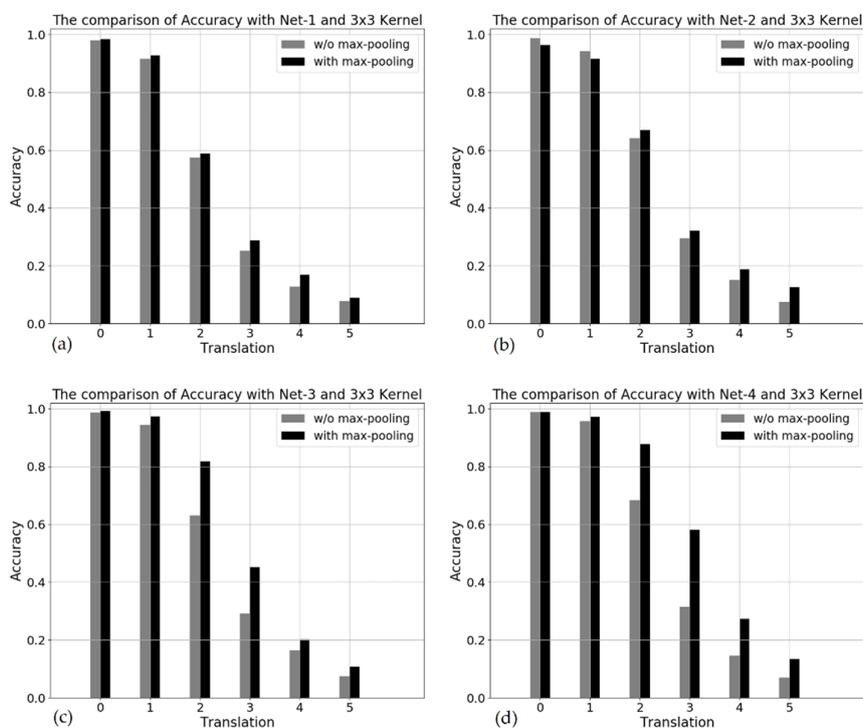
the baseline network by 62.55% in the case of 2-pixel translation. We can remark that the concatenation strategy consistently achieves high performance improvements over the baseline network.

Table 3 shows the experimental results for a very large kernel size, but a shallow network. Compared to Table 2, the error rates were reduced in all cases. This indicates that the kernel size is more effective for translational invariance than for network depth. Early fusion performs relatively well in the absence of pixel translation. Overall, the concatenation strategy is superior to all the other networks. The performance improvement of this strategy compared to the baseline network is 20.35% for 1-pixel translation and 73.88% for 2-pixel translation. Table 3 remarks that it is intrinsic to increase the kernel size to improve translational invariance when the network depth is shallow. We note that the concatenation strategy still achieves high performance improvements.

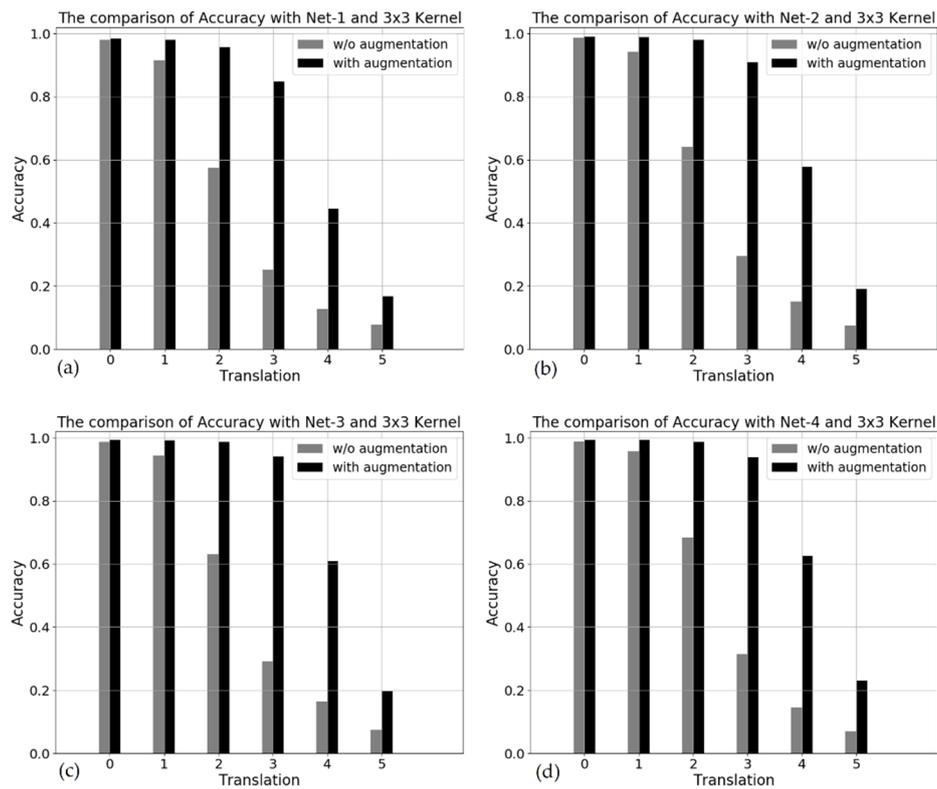
#### 4.5. On the Translational Invariance of Max-Pooling and Augmentation

This section demonstrates the well-known property of max-pooling layers and augmentation in terms of translational invariance. For this purpose, we provide three comparative results. One is for max-pooling, another is for augmentation within 2-pixel translation to partially cover the test images of 2-pixel translation, and the last one is for augmentation within 5-pixel translation to fully cover the test images of 5-pixel translation. For the network with max-pooling, we repeat a pair of convolutional and max-pooling layers  $K$  times, and name it *Net-K* for simplicity. We obtained all the test results by varying translation from 0 to 5 pixels. We note that the Net-4 network with max-pooling is the deepest one for  $38 \times 38$  input images.

Figure 7 shows that the networks with max-pooling are always superior to the networks without max-pooling. The deeper the network, the greater their difference, but they suffer from the large-pixel translation. However, augmentation is much more effective for translation as shown in Figure 8. The augmentation within 2-pixel translation partially covers test images. In the case of 3-pixel translation, they achieve more than a 90% accuracy regardless of the network depth.

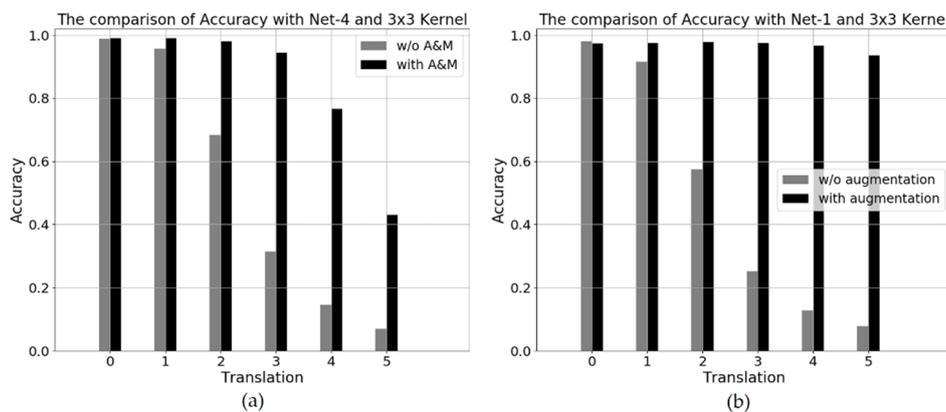


**Figure 7.** The comparison of networks with and without max-pooling by varying network depth with the same  $3 \times 3$  kernel size. (a–d) for the Net-1, Net-2, Net-3 and Net-4 networks, respectively.



**Figure 8.** The comparison of networks with and without augmentation within 2-pixel translation by varying network depth with the same 3 × 3 kernel size. (a–d) for the Net-1, Net-2, Net-3 and Net-4 networks, respectively.

We also investigated how effective the models with max-pooling and augmentation within 2-pixel translation, as shown in Figure 9a. In the case of 4-pixel translation, the accuracy is almost 80%. It is about 60% by augmentation only, as shown in Figure 8d. This means that using them together is synergetic when augmentation does not fully cover test images. Figure 9b demonstrates that augmentation that fully covers test images is almost translational invariant. However, we should note that it is not cost-effective or intractable to do such augmentation to cover the entire test images in real-world applications.



**Figure 9.** The comparison of networks with the same 3 × 3 kernel size. (a) for the Net-4 network with and without both max-pooling and augmentation (A&M) within 2-pixel translation. (b) for the Net-1 network with and without augmentation within 5-pixel translation.

In Figures 7–9, we demonstrated that max-pooling provides some level of invariance in deep layers, and that augmentation is crucial to obtain translational invariance.

## 5. Conclusions

The convolution layers are intrinsically translational-equivariant. The CNNs generally achieve small translational invariance by max-pooling. In this paper, we demonstrated that CNNs without pooling or augmentation are extremely vulnerable to translation, even on larger kernel sizes and deeper networks in our experiment on the modified MNIST dataset. From this result, we can recommend that the augmentation for translational invariance should be performed by 2-pixel interval or less.

In this paper, we proposed the use of frequency images which are global and translational invariant features. We demonstrated that the fusion strategies outperform the conventional baseline network, and that late fusion is superior to the early-fusion strategy. We remark that the concatenation strategy consistently achieves high performance improvement over the baseline network. For example, late-fusion by concatenation is far superior to the baseline network by 62.55% in the case of 2-pixel translation. We also demonstrated that augmentation is crucial to obtain translational invariance.

We conclude that the fusion strategy can compensate for convolution layers that are vulnerable to translation. It is also valuable to note that kernel size is more effective for translational invariance than network depth. Finally, we expect that our fusion strategies using frequency images can be applied to conventional networks to achieve better performance.

As future work, we will extend the proposed approach by applying the Fourier–Mellin transform [26] which is scale- and rotational-invariant as well as translational-invariant.

**Author Contributions:** Conceptualization, J.K. and K.J.C.; software, H.C.; validation, K.J.C. and J.K.; investigation, H.C. and J.K.; writing—original draft preparation, J.K.; writing—review and editing, K.J.C.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This paper was supported by Kumoh National Institute of Technology.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Pdf ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
2. Chang, J.-R.; Chen, Y.-S. Batch-normalized Maxout Network in Network. *arXiv* **2015**, arXiv:1511.02583.
3. Van Dyk, D.; Meng, X.-L. The Art of Data Augmentation. *J. Comput. Graph. Stat.* **2001**, *10*, 1–50. [[CrossRef](#)]
4. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic Routing Between Capsules. In *Neural Information Processing System NIPS*; 2017; pp. 3859–3869.
5. Chidester, B.; Do, M.N.; Ma, J. Rotation Equivariance and Invariance in Convolutional Neural Networks. *arXiv* **2018**, arXiv:1805.12301.
6. Worrall, D.E.; Garbin, S.J.; Turmukhambeto, D.; Brostow, G.J. Harmonic Networks: Deep Translation and Rotation Equivariance. *arXiv* **2016**, arXiv:1612.04642.
7. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
8. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
9. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
10. Cohen, T.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the Machine Learning Research 48 (PMLR), New York, NY, USA, 20–22 June 2016; pp. 2990–2999.

11. Levi, G.; Hassner, T.; Zhang, Z.; Cohen, P.; Bohus, D.; Horaud, R.; Meng, H. Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction—ICMI '15, Seattle, WA, USA, 9–13 November 2015; pp. 503–510.
12. Muhammad Anwer, R.; Khan, F.S.; van de Weijer, J.; Laaksonen, J. Tex-nets: Binary Patterns Encoded Convolutional Neural Networks for Texture Recognition. In Proceedings of the ACM on International Conference on Multimedia Retrieval, Bucharest, Romania, 6–9 June 2017; pp. 125–132.
13. Van Hoai, D.P.; Hoang, V.T. Feeding Convolutional Neural Network by hand-crafted features based on Enhanced Neighbor-Center Different Image for color texture classification. In Proceedings of the 2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Ho Chi Minh City, Vietnam, 9–10 May 2019; pp. 1–6.
14. Hosseini, S.; Lee, S.H.; Cho, N.I. Feeding Hand-Crafted Features for Enhancing the Performance of Convolutional Neural Networks. *arXiv* **2018**, arXiv:1801.07848.
15. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-Trained CNNs Are Biased Towards Texture. Increasing Shape Bias Improves Accuracy and Robustness. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
16. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
17. Jiang, R.; Mei, S. Polar Coordinate Convolutional Neural Network: From Rotation-Invariance to Translation-Invariance. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 355–359.
18. Esteves, C.; Allen-Blanchette, C.; Zhou, X.; Daniilidis, K. Polar Transformer Networks. *arXiv* **2017**, arXiv:1709.01889.
19. Henriques, J.F.; Vedaldi, A. Warped convolutions: Efficient Invariance to Spatial Transformations. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, NSW, Australia, 6–11 August 2017; pp. 1461–1469.
20. Xu, C.; Makihara, Y.; Li, X.; Yagi, Y.; Lu, J. Cross-View Gait Recognition using Pairwise Spatial Transformer Networks. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *1*. [[CrossRef](#)]
21. Reddy, B.; Chatterji, B. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Process.* **1996**, *5*, 1266–1271. [[CrossRef](#)] [[PubMed](#)]
22. Nussbaumer, H.J. The Fast Fourier Transform. In *Fast Fourier Transform and Convolution Algorithms*; Springer: New York, NY, USA, 1981; pp. 80–111.
23. Salazar, A.; Igual, J.; Safont, G.; Vergara, L.; Vidal, A. Image Applications of Agglomerative Clustering Using Mixtures of Non-Gaussian Distributions. In Proceedings of the 2015 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 7–9 December 2015; pp. 459–463.
24. Comon, P.; Jutten, C. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*; Academic Press: New York, NY, USA, 2010.
25. LeCun, Y.; Cortes, C.; Burges, C.J. The MNIST Database. Available online: <http://yann.lecun.com/exdb/mnist> (accessed on 1 January 2020).
26. Derrode, S.; Ghorbel, F. Robust and Efficient Fourier–Mellin Transform Approximations for Gray-Level Image Reconstruction and Complete Invariant Description. *Comput. Vis. Image Underst.* **2001**, *83*, 57–78. [[CrossRef](#)]

