Model Compression and Acceleration: Lip Recognition Based on Channel-Level Structured Pruning

Yuanyao Lu *, Ran Ni and Jing Wen

School of Information Science and Technology, North China University of Technology, Beijing 100144, China
* Correspondence: luyy@ncut.edu.cn

Abstract: In recent years, with the rapid development of deep learning, the requirements for the performance of the corresponding real-time recognition system are getting higher and higher. However, the rapid expansion of data volume means that time delay, power consumption, and cost have become problems that cannot be ignored. In this case, the traditional neural network is almost impossible to use to achieve productization. In order to improve the potential problems of a neural network facing a huge number of datasets without affecting the recognition effect, the model compression method has gradually entered people’s vision. However, the existing model compression methods still have some shortcomings in some aspects, such as low rank decomposition, transfer/compact convolution filter, knowledge distillation, etc. These problems enable the traditional model compression to cope with the huge amount of computation brought by large datasets to a certain extent, but also make the results unstable on some datasets, and the system performance has not been improved satisfactorily. To address this, we proposed a structured network compression and acceleration method for the convolutional neural network, which integrates the pruned convolutional neural network and the recurrent neural network, and applied it to the lip-recognition system in this paper.

Keywords: lip recognition; structured network pruning; network compression; Bi-LSTM

1. Introduction

With the arrival of the big data era, deep learning architecture has been widely regarded as one of the most important tools for the development of AI products. Deep learning has a very clear application direction, and compared with it, in the face of massive data, the traditional machine learning methods can be used in even more limited scenarios [1].

However, if we want to make AI products based on deep learning architecture widely used in daily life, it is not only required that the system performances such as big data and cloud computing can handle the use scenarios, but also the hardware requirements of the product cannot be ignored. As the equipment needs to be commercialized, we should consider both performance and consumed resources. Therefore, aiming at this problem, this paper proposed a compression and acceleration of the network model and applied the method to the lip-language recognition system. Compared with existing models, our model showed plenty of obvious advantages.

Voice interaction is commonly used in the part of human–computer interaction, which is effective and convenient. However, as human activities become more and more abundant, the accuracy of voice recognition declines drastically in some special scenes [2], such as the public place with mixed sounds and so on [3]. As a result of it, the lip-recognition system attracts our interest. Lip-reading methods are divided into two categories according to different feature extraction methods: one is the lip-recognition method based on traditional manual feature extraction and such methods include, for instance, hidden Markov models (HMMs) [4], the other is based on deep learning, which includes feed-forward networks, autoencoders, and convolutional neural networks (CNN) [5]. However, the construction of
the lip-recognition system tends to depend on the performance of the convolutional neural network [6]. Networks with high recognition accuracy have many problems, such as a large parameter scale, high storage requirements, and complex computation at the same time. Therefore, in the case of limited equipment resources, how to build convolutional neural networks with the highest accuracy has become a research hotspot.

An important factor affecting the network complexity is the size of the model. The Visual Geometry Group 16 (VGG16) has 140 million trainable parameters approximately, and if these parameters are stored in single-precision floating-point type, the model needs to occupy 530 MB of storage space. The model not only handles billions of floating points calculations, but also needs to store the results of intermediate procedures; the complexity of the model is obvious. The consumption of a large amount of computing resources accelerates the consumption of device hardware and reduces the system performance [7]. Therefore, it has become an important problem to compress convolutional neural networks and remove parameter redundancy while maintaining performance as much as possible. Model compression has been proposed to solve these problems, but traditional model compression still has many defects.

This paper is mainly divided into the following parts. Firstly, we compared some classical network models and proposed that the VGG network used for pruning and the bidirectional long short-term memory (Bi-LSTM) [8] network used for processing the network. Secondly, we compressed and accelerated the VGG16 network. Then, we used the pruned network and the Bi-LSTM network to extract lip features [9]. Lastly, we introduced the datasets and verified the feasibility and correctness of our method from the three aspects of algorithm convergence, algorithm recognition, and speed algorithm accuracy. Moreover, the main contribution of this paper is that we applied the structured pruning to the lip recognition, which achieved good results.

2. Framework

The development of deep learning has been a pivotal contribution to the improvement of the field of human–computer interaction. Although the model that possesses the networks with deep layers has a strong learning ability, the calculation amount, resource consumption, and the complexity of the model are so high that it is greatly difficult to apply to all kinds of hardware platforms widely. Thus, according to the problem, this paper proposed a solution to the parameters of the network model and the number of network layers so that it could reduce the problem of high model consumption and high memory usage.

Currently, there is redundancy in many deep model structures. As a result of it, if we train the networks after pruning a part of the unimportant weight parameters, the recognition accuracy can be approximate to what it was before pruning. Several network parameters are shown in Table 1. Moreover, network compression can reduce the complexity, calculation amount, and resource consumption of the lip-reading model which is proposed in the paper and speed up the time to train the model to a great extent.

| Years | Model | Network Layers | Parameters (Million) | FLOP (Billion) |
|-------|-------|----------------|---------------------|--------------|
| 2012  | AlexNet [10] | 8              | 233                | 4.4          |
| 2014  | VGG16 | 16             | 552                | 1.33         |
| 2014  | GoogleNet [11] | 22 | 51            | 8.9          |
| 2015  | ResNet-50 | 50            | 98                 | 17.9         |
| 2015  | ResNet-101 | 101           | 319                | 71.4         |
| 2015  | ResNet-152 [12] | 152         | 230                | 48.1         |
| 2016  | DenseNet-121 [13] | 121         | 31                 | 13.4         |
2.1. Network Pruning

A network pruning strategy is utilized to delete unimportant parameters and expand the sparsity of the network. However, the pruning performance of some primary methods which are based on unstructured pruning is far from perfect and is difficult to improve. Thus, in recent years, researchers have focused on structured pruning, which is used to thin high-density connections and parameters of the network, such as introducing regularization terms, which can make all model parameters tend to zero during the training. The stochastic gradient descent method is included in the loss function to sparse structure. Furthermore, the method of judging the network model channel or filter by a threshold can also prune the network effectively, which prunes the input characteristic of the channel shown in Figure 1 to reduce the memory consumption of the model.

![Figure 1. Schematic diagram of pruning filter method.](image)

For the proposed pruned network, we trained and tested it on the CIFAR [14] dataset first, and the result is shown in Table 2. As a result of it, although the network parameters declined obviously, the recognition accuracy was not reduced and there was even some improvement in the result, which proved that there are some redundant parameters in the network.

| Model       | Baseline -CIFAR10 | Prune0.6 -CIFAR10 | Baseline -CIFAR100 | Prune0.6 -CIFAR100 |
|-------------|-------------------|-------------------|--------------------|--------------------|
| Top1-acc (%)| 93.77             | 93.78             | 72.12              | 73.32              |
| Parameters (M)| 20.04            | 2.25              | 20.04              | 4.93               |

2.2. Network Model Compression Based on VGG16

With the development of deep learning theory, convolutional neural networks possessing powerful performance emerged in an endless stream, such as LeNet, AlexNet, GoogLeNet, VGG, ResNet, and so on. According to achievements published by the study field, to extract more abstract feathers, much deeper networks are utilized to be trained and more accurate recognition results are obtained simultaneously. However, improvement of the results’ accuracy also brings some problems, which cannot be ignored, such as massive datasets and large memory devices. As a result of it, we decided to compress the VGG16 network, and the compressed network was used to extract feathers in the field of lip pictures. VGG is a deep network with a more perfect performance developed based on convolutional neural networks, which was born in the Computer Vision Group of Oxford University, and is a major research achievement. First of all, the accuracy of the VGG which won the championship in the 2014 ImageNet competition is better than the other models. Then, VGG possesses strong transfer learning capabilities, high scalability, and generalization capabilities. Moreover, VGG16 is a simple network model formed by a pure convolutional layer simultaneously. From the experience of AlexNet, a smaller convolution kernel and more network layers can improve accuracy. The network structure diagram of VGG16 is shown in Figure 2 (Conv, convolutional layer; Relu, rectified linear unit; FC, fully connected layer).
Furthermore, a $3 \times 3$ size fixed convolution kernel and $2 \times 2$ size pooling layer are utilized to build a network model with a depth of 16–19, which is the foremost feather of VGG. Its design makes the VGG possess an advantage in the part of parameters and calculations. This paper used its powerful feature extraction capabilities to extract visual features of lip language, and Figure 3 shows the ratio of time occupied by the convolutional layer involved in feature extraction during the extraction process.

![Figure 2. VGG16 network structure diagram.](image)

**3. Lip-Recognition Model Structure and Prune**

In this paper, network training can be divided into three steps, which are the acceleration and compression of the VGG16 network, the lip-feature extraction of the pruned network, and the lip temporal sequence extraction based on the Bi-LSTM network. Furthermore, the memory size of the model and the waste of calculation resources are reduced without reducing the accuracy of recognition by compressing the convolutional neural networks in this paper. The pruned network is used to extract lip features. Then, it is combined with an RNN (recurrent neural network), which is utilized to learn the temporal features of the lip. The network structure of the paper is shown in Figure 4.

![Figure 3. Percentage of running time of convolutional layer.](image)
Figure 4. Network structure diagram.

The overall structure is divided into the following four parts.

First, we needed to process the video dataset and lip-language image. A semirandom video frame sequence extraction algorithm was used to extract lip-language frame sequences randomly from the video, where the video frame was a fixed length, and then, we could obtain the location of the face and lips.

Secondly, the CNN was compressed based on a channel level. The channel pruning method was utilized to prune the VGG16 network with excellent performance. Parameter redundancy was removed, and the model was accelerated without reducing its performance as much as possible.

Then, the Bi-LSTM network was used to extract the temporal features of lip movement image sequences so that it could learn the contextual semantic information contained in lip language effectively and improve the recognition accuracy.

Lastly, the extracted spatial and temporal features were input into the fully connected layer for classification. Since the classification was a reverse event, it was more suitable to use Softmax as the activation function.

Pruning Network Training and Testing

The commonly used pruning algorithms generally follow the flow shown in Figure 5. Although the details of each algorithm may differ, the processes are similar [15].
The first step is the most important part of the pruning process. The importance of a pruned unit can be a weight parameter or an entire network layer, depending on the granularity of the pruning. The methods for measuring importance range from basic optimization-based algorithms to structure-based methods, with their own complexity and accuracy, and they can be chosen according to the needs. The second step is to cut out the unimportant connection parameters based on the results measured in the previous step. We can set the corresponding thresholds or rank them according to their importance to decide which parameters to cut. The third step is to fine-tune the network, i.e., to retrain the network. Retraining the pruned network can improve the accuracy and performance of the network, and the degree of fine-tuning can be adjusted according to your needs.

The purpose of compression and acceleration is to transform the original network with high-density parameters into a sparse network, without sacrificing its recognition accuracy [16]. To maintain its accuracy, we need to prune the unimportant parameters in the model, so the core of the pruning problem is to set evaluation criteria for the importance of the parameters. There are two classical pruning methods, one is to set a threshold value to judge the weights according to the importance of the weights, the weights below the threshold value are unimportant, and the other is based on the Hessian matrix of the loss function on the weights, i.e., [17], the higher the value of the matrix, the more important the parameters are.

Furthermore, the overall process of model channel pruning is shown in Figure 6, which is applied to the sparse channel. Channel-level based sparse pruning has good results in lip recognition. The application of sparsification can be implemented at different levels, including fine-strength-based sparsification, but fine-strength sparsification relies on special hardware and software equipment to perform sparsification, otherwise, the model runs slowly and the results are not satisfactory.

**Figure 5.** Pruning flow chart.

**Figure 6.** The pruning process.
When applying channelization-based sparsification to a deep network (e.g., VGG16 in this paper), pruning the number of layers also achieves the desired effects, balancing speed and implementation. Furthermore, by adjusting the parameter size and channel weights in the model channels for joint training, channel-level sparsity can be achieved. Firstly, we trained the network sparsely to adjust the parameters. Secondly, we pruned the channels whose parameters were close to 0 after adjustment, and then we fine-tuned the network and retrained it. Through several iterations, we could obtain a network with low computational cost that requires little memory and retains the accuracy without cutting down on the parameters.

The batch normalization layer enables the subsequent training of the input data to reach a steady state [18]. BN layers are widely used in convolutional neural networks for their ability to improve network convergence speed and learning performance. Thus, we proposed a channel-level network pruning method by borrowing the batch normalization layer (BN) concept to discriminate channels, the process of which is shown in Figure 7.

![Figure 7. Schematic diagram of channel-level pruning process.](image)

During training, the update of parameters can constantly reset the proportion intensity of each layer. In other words, layers affect each other, while the pattern of input information to each layer is constantly changing. As a result, each layer needs to adapt continuously to the distribution of information during training, which may reduce training speed and lead to slower convergence. Moreover, this phenomenon occurs more frequently in deep neural networks, and it is called internal covariate shift [19]. To tackle this, we proposed batch normalization. The batch normalization algorithm not only improves the performance of deep networks, but also boosts the performance of lightweight small networks. At present, batch normalization has become the standard configuration of almost all convolutional neural networks.

Since BN is based on minibatch, we normalized the expectation and variance of the m training examples in each batch of training. The normalized data was transformed and reconstructed to enhance the expressing ability of the network. The formula for transformation reconstruction is listed as follows.

We could regard the channel adjustment parameters as parameters used to measure the significance of a channel. They were multiplied with the channel weights and then processed by the sparse constraint immediately, which would not increase the computational burden on the model. The parameter quantity of the BN layer was small before training, and then the parameters were normalized.

Furthermore, after introducing the adjustment parameters, the value of some parameters in the model was approximately equal to 0 and we needed to prune the channels...
represented by these close to 0 parameters. For instance, the output dimension of the convolutional layer was a feature map of $H \times W \times C$, where $H$ and $W$ represent the weight and height the feature map and $C$ is the number of channels. If we input it to the BN layer, then each of the $C$ feature maps had a set of corresponding adjustment parameters. By setting a global threshold for the network, we could judge the magnitude of the adjustment parameters, and when the threshold was 0.2, we pruned 20% of the network channels, and then, the first 20% of the adjustment parameters taken from the smallest to the largest order would be pruned, resulting in a smaller network.

4. Results and Analysis

In this paper, we used the VGG16 network, a structure commonly used in VGG Net, with a 13 + 3 network layer structure, i.e., a convolutional layer plus a fully connected layer. Our experiments were based on channel-level sparsity training, and the inputs and outputs of channels with adjustment parameters close to 0 and their corresponding weights were pruned in the trained model. We used global thresholds as a metric. The use of global thresholds avoids too many iterations and too much retraining.

According to the evaluation criteria, the number of parameters and FLOPs were selected as the criteria after pruning the model; the loss function was used as the important criteria when we trained the model and the accuracy rate was utilized as the criteria of the lip-recognition result. Additionally, the accuracy rate meant that the number of correctly identified samples was divided by the total number of samples.

The network was pruned with global pruning rates of 20%, 40%, and 60%, and the number of training rounds was the same for each pruning rate during training. It was verified that pruning was effective in removing the redundancy of network parameters, FLOPs were compressed, and memory and running memory were reduced. The results are shown in Table 3 (where FLOP means the times of floating-point operations performed by the computer per second, memory means computer disk memory, and Mem R + W means running memory occupied by reading and writing), which lists the original parameters of the VGG16 network and the changes in the parameters under each pruning rate. We can see that the value of each index decreased significantly after pruning.

| Model       | Parameters | FLOP  | Memory (MB) | Mem R + W (MB) |
|-------------|------------|-------|-------------|----------------|
| Baseline    | 20.03      | 19.56 | 175.71      | 428.02         |
| Pruned 0.2  | 11.6       | 16.16 | 168.61      | 382.04         |
| Pruned 0.4  | 6.19       | 13.4  | 161.66      | 347.15         |
| Pruned 0.6  | 2.96       | 11.2  | 152.3       | 316.14         |

After compressing the network, experiments compared algorithm convergence, algorithm recognition speed, and recognition accuracy between the Pruned-VGG16 and the original network under the three pruning rates on the self-made dataset.

4.1. Algorithm Convergence

The convergence of the algorithm can indicate how well the model fits the input data, which can reflect the learning ability of the Pruned-VGG16+Bi-LSTM model on our homemade dataset and is an important metric to judge how good the model is. Improper setting of model parameters will lead to difficulty in the convergence of the model and affect the smooth operation of model identification, so we first analyzed the convergence of the model. In this paper, we recorded the change of the comparison curve between the pruning threshold (20%, 40%, 60%) and the loss function of the original model, and we used the magnitude of the loss function to analyze the convergence of the model, which is shown in Figure 8.
Figure 8. Loss function comparison chart.

As is shown in the chart, as the number of iterations gradually increased, the loss function of the original model decreased faster because of the large number of network parameters, and when the number of iterations reached 20, the loss function changed more slowly and gradually stabilized. This indicates that the model can fit the training data distribution well and the network is optimal at this point. It can be seen that the pruned model gradually converges with the increase of training times, which indicates that the model performs well in lip recognition and there is no unusual problem in the homemade dataset.

4.2. Algorithm Recognition Speed

After building the Pruned-VGG16-Bi-LSTM model, we compared the time of recognition on the self-made dataset (0–9 English independent pronunciations), the result of which is shown in Figure 9.

Figure 9. Time consumed in recognition.

The times obtained from training were summed over the time of recognition (in milliseconds) for ten English pronunciations. In order to compare the performance im-
provement of the model, we recorded the recognition time for each test, and the total number of tests was 1000, which shows that as the pruning rate increased, the recognition speed increased and the recognition time decreased. The networks with pruning rates of 20% and 40% stabilized in the later stage of the test, and the network with the pruning rate of 60% had better recognition performance, and the speed stabilized after 500 tests and was better than the original network.

For different pronunciations, the models differed in the recognition time of different digits, influenced by the number of iterations. In this paper, we compared the average recognition times of different network models (original network, pruned 20%, pruned 40%, and pruned 60%) for recognizing ten digit pronunciations, as shown in Figure 10. It can be seen that compared with the original model, the average recognition time after pruning is significantly reduced in all cases.

Figure 10. Average recognition time.

4.3. Algorithm Accuracy

After comparing the convergence and speed of the model, we verified the accuracy of the lip-recognition model based on the self-made dataset and analyzed the pattern of experimental data in each epoch. Figure 11 shows the accuracy curve of the original network and different pruned networks on the training set.

Figure 11. Comparison curve of accuracy.
As is shown in the picture, the accuracy of the original model was lower than that of the pruned model, because the parameters of the original model were so large that it led to overfitting during the training process. As the number of iterations increased, the accuracy curve gradually rose. Furthermore, when the number of iterations reached about 20 times, the value of recognition accuracy reached an excellent level, which indicated that the network model fit the distribution of training data well and reached an optimal level.

In order to compare the performance of different models after pruning, we compared the accuracy of ResNet-18, ResNet-34, and VGG16 with a pruning rate of 0.6 and baseline, which is shown in Figure 12. We found that different models with a large number of parameters could reach convergence well under the self-made dataset after pruning. Moreover, because the VGG16 network had a large number of parameters, the overall performance of the model was greatly improved compared with the other models after pruning.

![Figure 12. Accuracy comparison between different models at a pruning rate of 0.6.](image)

Furthermore, the main factors affecting the accuracy rate in lip recognition are as follows. Firstly, the lip region is small and there is little difference between lip movements. Secondly, there are not enough researchers and insufficient datasets for lip recognition. We made recall statistics for each English number pronunciation prediction result, as shown in Figure 13.

![Figure 13. Recall statistics of the English independent pronunciations.](image)
5. Conclusions

The existing networks with superior performance had the disadvantages of deep network layers and a large number of parameters. Therefore, this article presented the application of structured pruning in deep learning.

Furthermore, in order to verify the effectiveness of the proposed model, we also performed three experiments. Firstly, by increasing the number of iterations, the convergence of the pruned model was similar to the original model. Secondly, we compared the average recognition time on the different network models (original network, the networks with 20% pruning rate, 40% pruning rate, and 60% pruning rate), and we concluded that the average recognition time of the pruned network was shorter. Lastly, we compared the accuracy on different pruning rates. As a result of the above experiments, our model was found to be effective.

Future research should be devoted to study methods to find the optimal pruning rates for different networks. Finally, we should also balance between appropriate pruning rate and recognition accuracy.

Author Contributions: Data curation, R.N.; Software, Y.L. and R.N.; Supervision, Y.L.; Visualization, J.W.; Writing—review & editing, R.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (61971007 and 61571013).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schmidhuber, J. Deep learning in neural networks. *Neural Netw.* 2015, 61, 85–117. [CrossRef] [PubMed]
2. Chavan, K.; Gawande, U. Speech Recognition in Noisy Environment, Issues and Challenges: A Review. In Proceedings of the International Conference on Soft-Computing & Networks Security IEEE, Coimbatore, India, 25–27 February 2015; pp. 1–5.
3. Jeon, S.; Kim, M.S. End-to-End Lip-Reading Open Cloud-Based Speech Architecture. *Sensors* 2022, 22, 2938. [CrossRef] [PubMed]
4. Fenghour, S.; Chen, D.; Guo, K.; Li, B.; Xiao, P. Deep learning-based automated lip-reading: A survey. *IEEE Access* 2021, 9, 121184–121205. [CrossRef]
5. Li, X.; Zhang, T.; Zhao, X.; Yi, Z. Guided autoencoder for dimensionality reduction of pedestrian features. *Int. J. Speech Technol.* 2020, 50, 4557–4567. [CrossRef]
6. Hara, K.; Kataoka, H.; Satob, Y. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA, 18–23 June 2018; pp. 6546–6555.
7. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 2014, arXiv:1409.1556.
8. Huang, Z.; Wei, X.; Kai, Y. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* 2015, arXiv:1508.01991.
9. Hossain, M.S.; Yousuf, A. Real time facial expression recognition for nonverbal communication. *Int. Arab. J. Inf. Technol.* 2018, 15, 278–288.
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
11. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* 2015, arXiv:1512.03385.
13. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
14. Zhang, X. The AlexNet, LeNet-5 and VGG NET applied to CIFAR-10. In Proceedings of the 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhubai, China, 24–26 September 2021; pp. 414–419. [CrossRef]
15. Lazarevic, A.; Obradovic, Z. Effective pruning of neural network classifier ensembles. In Proceedings of the IJCNN’01, International Joint Conference on Neural Networks, Washington, DC, USA, 15–19 July 2001; pp. 796–801.
16. Zhang, C.; Hu, T.; Guan, Y.; Ye, Z. Accelerating Convolutional Neural Networks with Dynamic Channel Pruning. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; p. 563. [CrossRef]
17. Hao, D.; Tian, J.; Yongpeng, D.; Zhuo, X. A compact human activity classification model based on transfer learned network pruning. In Proceedings of the IET International Radar Conference (IET IRC 2020), Chongqing, China, 4–6 November 2020; pp. 1488–1492. [CrossRef]

18. Thakkar, V.; Tewary, S.; Chakraborty, C. Batch Normalization in Convolutional Neural Networks—A comparative study with CIFAR-10 data. In Proceedings of the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), Howrah, India, 12–13 January 2018; pp. 1–5. [CrossRef]

19. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv 2015, arXiv:1502.03167.