# The Compression Techniques Applied on Deep Learning Model

Haoyuan He [1, *, †], Lingxuan Huang [2, †], Zisen Huang [3, †],Tiantian Yang [4, †]

[1] School of Material Science and Technology, Harbin Institute of Technology at Weihai, Weihai, Shandong 264209, PRC

[2.]School of Science and Technology, Hong Kong Metropolitan University, Hong Kong, HKG, PRC

[3] School of Computer Science, Wuhan University, Wuhan, Hubei 430072 PRC

[4.]School of Computer Science,Xijing University,Xi'an,Shaanxi,710123,PRC

* Corresponding Author Email: 2190830306@stu.hit.edu.cn

[†]These authors contributed equally.

**Abstract.** In recent years, the penetration rate of smartphones has gradually completed, artificial intelligence is the cutting-edge technology that can trigger disruptive changes. Deep learning neural networks are also starting to appear on mobile devices. In order to obtain better performance, more complex networks need to be designed, and the corresponding models, computation and storage space are increasing, however, the challenges of resource allocation and energy consumption still exist in mobile. The techniques for compressing deep learning models are quite important, and this paper studies a series of related literatures. This paper reviews deep learning-based deep neural network compression techniques and introduces the key operational points of knowledge extraction and network model on the learning performance of Resolution-Aware Knowledge Distillation. In this paper, a low-rank decomposition algorithm is evaluated based on sparse parameters and rank using the extended BIC for tuning parameter selection. This paper discusses the reduction of redundancy in the fully connected and constitutive layers of the training network model by pruning strategies.Moreover, this paper presents the quantization techniques and a neural network that quantifies weights and activations by applying differentiable nonlinear functions.

**Keywords:** Quantization, Knowledge Distillation, Low-Rank Matrix Factorization, Pruning.

## 1. Introduction

In the process of studying deep learning models and discussing the related literature, many challenges in mobile deep learning, such as the small amount of computation on mobile terminals, have been found. The network models can be optimized on the drawbacks of these mobile terminals. The focus of mobile learning is as follows: the decline in the accuracy of the overall model, and the decline in the accuracy of the overall model caused by the difference in the parameters and the accuracy of the data volume of the mobile device.

Therefore, this experiment prepares to use the technique of trimming quantization and uses the following preparation stages:

The trimming rate is set to 90%, 80%, and 70%, and the quantization degree is set to 16 bits and 8 bits to minimize the loss function and make it converge as much as possible.

Quantitative comparisons are made for running speed, accuracy, power, and CPU usage. In the experimental testing phase, in order to find out the best learning rate that can adapt to different devices, the different types of mobile devices are examined (such as cell phones, tablets, etc.), so that it is not easy to cause problems such as accuracy degradation.

A total of 17 experimental group models and 1 control group model (without quantitative pruning) were obtained, and the final number of performance index result groups was (17+1) × n. n is the number of the used devices, and the model design optimization was achieved by tuning the parameters.

In summary, in the face of frequent problems on the mobile end, due to the construction of software and hardware, there are more revealed problems. Therefore, choosing a suitable solution to design a masculine network architecture suitable for mobile terminals has become the primary task.

## 2. Knowledge Distillation

Knowledge distillation is a promising model compression solution. Model compression techniques can reduce the number of network settings and thus the storage space for the models can allow it to be ported to the embedded devices. The idea of knowledge distillation is to teach a smaller network exactly what to

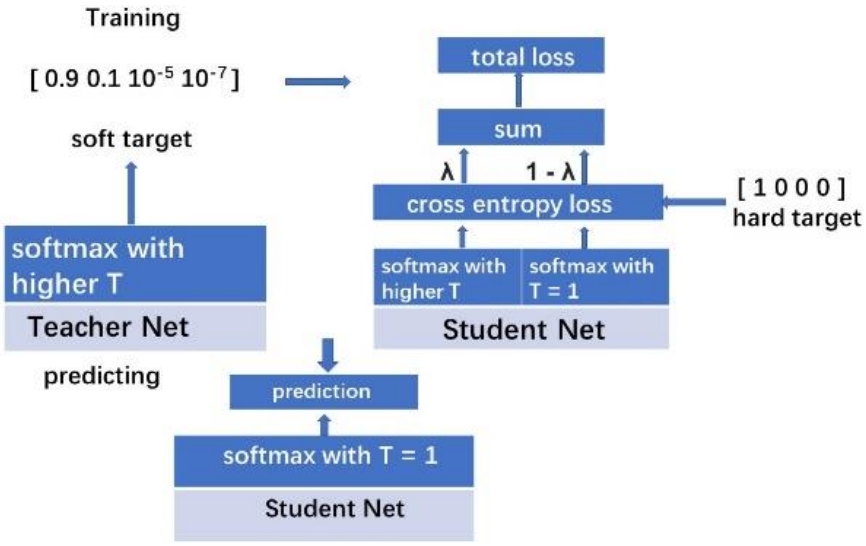


**Figure 1.** Knowledge distillation operation diagram [2]

do by incrementally using a large trained network. Hinton proposed a technique [1]. The robustness and generalization of the Student Network (SNet) can be continuously enhanced by using transfer learning. SNet continuously leans the complicated Teacher Net (TNet), in order to close and fit the TNet, it is necessary to train highly complex TNet individually using the full dataset. The correspondence between SNet and TNet intermediate outputs should be established. Figure 1[2] illustrates the process of knowledge distillation. SNet is trained in two parts. Firstly, the SNet is trained using the same transfer set as the temperature T applied in the TNet training process, and the cross-entropy loss is used as the first part of the total loss for SNet training. Secondly, the cross-entropy loss between softmax prediction output (T = 1) and hard labels will be minimized by training SNet. It can be observed that T is the key component during this operation.The different forms of T can alter SNet training. Distillation is the final step in the process. If the weighting of soft target cross-entropy is higher, it suggests that the SNet will rely on the TNet. The SNet will get a bigger performance boost. Moon and his team [3] showed that depth scaling is more critical, compared to width when using smaller networks. The ability to learn based on distillation can be affected by this factor. Learning performance will be affected by considering the adjustment of a student architecture. In Yuan's study [4], Two Stage Knowledge Distillation (TSKD) outperformed with 99.21% accuracy of 2.2M parameters, while MobileNet V1 achieved 98.64% accuracy of 3.2M parameters. Moreover, TSDK is compared with traditional methods for face recognition. The results showed that the model size of TSDK is only 7.5MB, compared to other models. The experiment also tested that the error rate and compression ratio can be determined by the depth level and the residual block type. In addition, computation complexity is significant for the popularization of deep networks in practical applications. Feng and his team [5] investigated how to improve deep network performance and relieve the computation burden. To address the appearance changes among resolutions and limit the significant performance decrease in extracting features from LR inputs, they developed the Resolution-Aware Knowledge Distillation (RKD) system. The proposed method can extract the useful information from the HR domain and apply it to the LR domain. The final experimental results show that RKD can speed up the deep network while maintaining the discriminative power, which is critical for practical applications of computational resource allocation. Moreover, RKD outperforms other frameworks at the level of knowledge transfer across the different resolutions.

## 3. Low-Rank Matrix Factorization

In recent years, deep learning model shines brightly. In such case, low-rank decomposition is a worthwhile algorithm for researchers to study and apply in the experiment. In Sigurdsson's paper[6], the sparse and low rank model are given. The sparse and low rank hyperspectral model is given by Formula 1 as follows:

$$Y = SA^T + X + \varepsilon\,. \tag{1}$$

The sparse and low rank unmixing problem is given by formula (2):

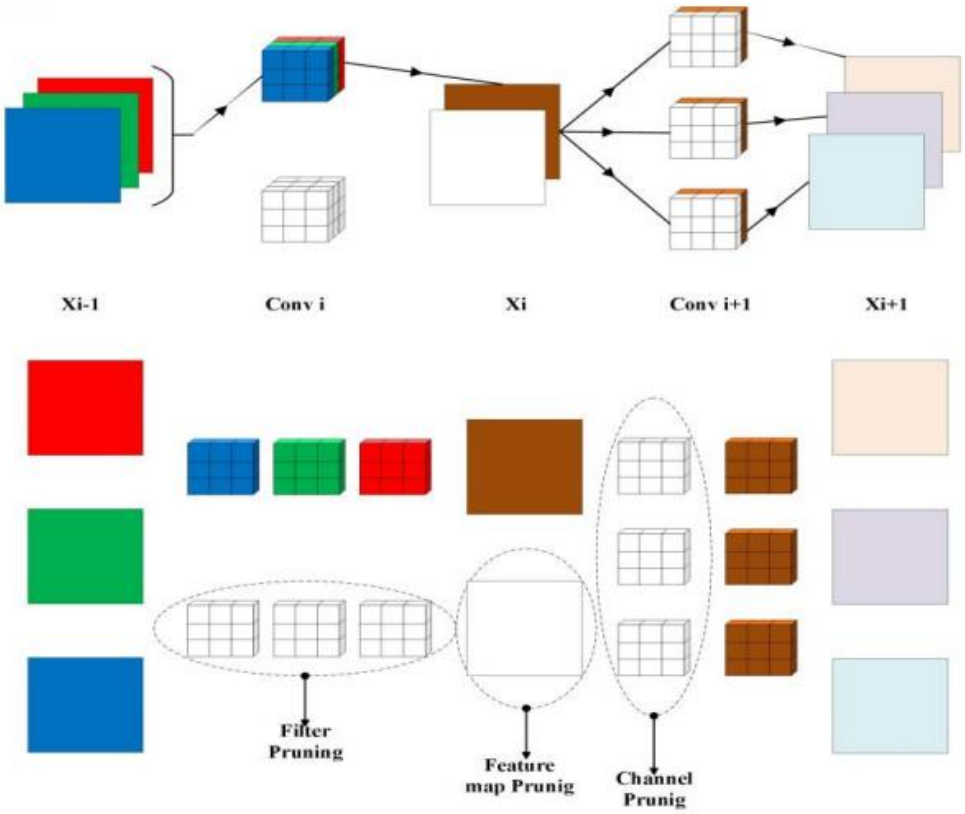


**Figure 2.** CNN optimization

$$\min_{\substack{A\in\Gamma^{M\times r}\\ X\in R^{P\times M}\\ S\in R_+^{P\times r}}} \frac{1}{2}\left|\left|Y - SA^T - X\right|\right|_F^2 + h_s\left|\left|S\right|\right|_{1,1} + h_x\left|\left|X\right|\right|_{1,1} \tag{2}$$

Further understanding the decomposition, Ong' research [7] proposed a multiscale low rank modelling that represents a data matrix as a sum of block-wise low rank matrices with increasing scales of block sizes.

Similarly in Ulfarsson's paper [8], the high dimensional data is often modeled as a linear combination of a sparse component, a low-rank component, and noise. Sparse and low rank (SLR) matrix decomposition is a recent method that estimates those components. As the sparse and low rank model is given in formula (3):

$$Y = SA^T + X + \varepsilon\,. \tag{3}$$

This model is treated, in which the robust Principal Component Pursuit (RPCP) method can be established.

In Shang's paper [9], they developed an l0 based on SLR method and an associated tuning parameter selection method based on the extended Bayesian information criterion (EBIC) method. In simulations, the performance of the new method was evaluated under various settings and shown to outperform both the RPCP method and a CD-SLR1 method. As in Kaloorazi's paper [10], it

developed an efficient rank-revealing decomposition algorithm aided by randomization, which provides information about the singular subspaces and singular values of a given data matrix.

Generally, the algorithm employs a cyclic descent method for estimation and the extended BIC for the tuning parameter selection, which is based on the sparsity parameter and the rank. In the simulations, the performance of the new method can be evaluated under different settings. It has shown that this method can outperform both the RPCP method and a CD-SLR1 method.

## 4. Pruning and CNN optimization

It is necessary to compress the convolutional neural network (CNN) for resource-constrained devices deployment, resulting in the accelerated model training and inference. Soft filter pruning (SFP) [11] conducts dynamic pruning operations during training time, allowing updates of the prunes to compensate for any possible miso. This paper finds a method to reduce the redundancy in the fully connected and constitutional layers of the trained network [12] model. With the rapid development of deep convolutional network the deep learning model reasoning has to consume a large amount of computational resources. Because of these problems, the CNN optimization, shown in Figure 2, has become our primary task.

According to the number of weights and positions of one-time pruning, pruning methods can be divided into the unstructured pruning 26 and structured pruning 3[13]. The unstructured pruning applies a single weight at one time and there is no restriction on the position of the pruning weight. It can obtain a relatively high compression ratio and is not sensitive to the decrease in accuracy. Irregular pruning changes the network structure [14] and results in irregular parameter connections and irregular memory access conditions. It makes the network unable to work properly [15] and is not conducive to GPU acceleration. Therefore, online testing can adversely affect the efficiency of the stage. The structured pruning has restrictions on the weight position of pruning. By pruning a row of weight vector, evolution kernel, filter, etc., the model structure after pruning will not be changed and it can be implemented in the existing framework. However, its expression effect is relatively limited. Compared with the unstructured pruning, the structured pruning is more sensitive to the accuracy changes. Pruning in the model compression method is the only one of the efficient strategies and does not conflict with other methods. Therefore, it's combined with other methods, such as knowledge distillation and quality.

## 5. During-training Quantizationand Post-training Quantization

Once the quantization methods were proposed, many researchers started to contrive new quantization methods. The quantization methods now can be categorized by lots of ways. As for the period of the quantization procedure, the quantization methods are classified as post-training quantization during-training quantization. In Yang's paper [16] and Zhu's paper [17], the researchers proposed their during-training quantization methods and others came up with their post-training methods in Moura's research [18]and Liu's research [19].

According to the Yang's research [16], the writer proposed a neural network that quantized the weights and the activation by applying a differentiable non-linear function. The researchers used the combination of the sigmoid function to form the function which was used for the quantization. Some parameters in these functions can be altered in the following training procedures. Afterwards, these parameters can be trained by using the loss function produced in the training procedures of CNN model. When these parameters were completely trained, the researchers put these parameters in a new step function and made this step function as the final function of quantization [16]. The mathematical expression of that function is presented as Formula 4:

$$\mathcal{Y} = \alpha(\sum_{i=1}^{n} s_i \mathcal{A}(\beta x - b_i) - \mathcal{O}) \quad (4)$$

In Formula 4,$\mathcal{A}$is a step function. $\mathcal{Y}$is the integers set and also the quantization output.x is the input of quantization.βis the scale factor of the input xwhile αis the scale factor of the output $\mathcal{Y}.s_i$and $\mathcal{O}$ both can be determined by quantization integer set.

In the Zhu's paper [17], the researcher proposed a new ternary quantization method which created two full-precision scaling coefficient and it was different from the old ternary quantization which set the quantization coefficient as {-1,0,1} or {-E,0, E}. What's more, the two coefficients used in this section are learnable, and eventually the absolute value of these coefficients can be different.

The post-training quantization also has its advantages over the during-training quantization. Some researchers develop some post-training quantization methods, such as the researchers in Moura's research [18] and Liu's research [19]. In Moura's paper [18], the researchers trained a CNN models and then quantized them, after which they used these models to discriminate the different sort of heartbeats. In their research, researchers used the Tensorflow Lite to quantize the model produced after training. They mainly quantized the float point weights to 8-bits integers by using the TFL converter, which can work in the corresponding Android platform. However, their models' input and output are the float point numbers. Moura's experiment showed that the size of their models was reduced by 90% with small accuracy degradation. In Liu's research [19], the researchers contrived an Adaptive Floating Point (AFP) to quantize the FP32 models. The AFP is a flexible low-bits variant which contains some flexible exponent bits and mantissa bits in the new created float point numbers. The amount of the flexible segments can be altered by the following training. To decide the bit-width of the AFP used in the quantization, the researchers apply the Bayesian Optimization to delete the redundant bits in the initial 32-bits Float Point numbers and determine the final number of bits by several times' iteration. Liu's experiment proved that their methods can get a higher-accuracy model than that of other methods with less iteration. Figure3 shows that the AFP method has higher accuracy and lower hardware cost.

Moreover, in Pouransari'sresearch [20], some researchers firstly contrived the rank-1 quantization method to transform the matrix of weights' or the matrix of activation to the flattened vector, and then they used k-bits quantization method to reduce the difference of the accuracy between the novel weights and the full-precision weights. In the functions of quantization, the

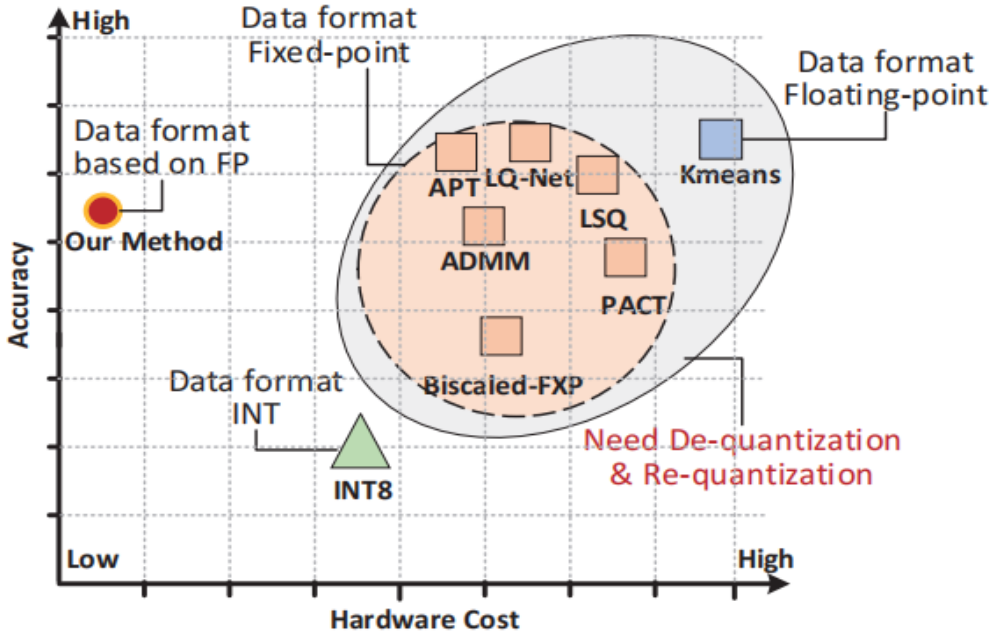


**Figure 3** The performance of the AFP method [19]

researchers applied the sign function and this function became an important part of the quantization. The researchers also found the most suitable bit-width quantization method by applying the least square on computing the loss of the accuracy of the novel weights and activation and using the greedy k-bits algorithm, which revealed that the 2-bits quantization can perform well.

## 6. Conclusion

For the fast development of the microelectronic manufacture techniques,many mobile devices have higher performance when they run the neural network models.However,some neural network models are still too large for the mainstream mobile devices.Since the rapid development of the compression technique applying on the deep learning models,the energy consumption and the computational resources can be largely saved when the mobile devices run the deep learning models.The knowledge distillation techniques create a student net. The student net has compact structure and its weights can be obtained by learning from the teacher net softmax output and the hard label.These techniques increase the student net's generalizing capacity and robustness. The low-rank matrix factorization decomposes the complex matrices into a set of simpler matrices. During this procedure, the kernels of the CNN models become smaller. This technique effectively reduces the number of the weights and increases the speed of running the CNN models. Pruning technologies can decrease the number of weights without changing the models structures. The quantization techniques reduce the number of bits in the weights and activations.To decrease the accuracy degradation, some researchers developed ternary quantization method and AFP method. The conclusion of this article is that the future trend of these techniques' development is making less accuracy loss of the processed models and at the same time the learning ability of the processed models will also be enhanced.Therefore,these outstanding features can guarantee us to enjoy more precise models while the models become lighter and more easy for mobile devices to run. The more compression methods for deep learning models will be developed with the wide usage of these models on mobile devices.

## References

[1] G. Hinton, O. Vinyals, J. Dean. Distilling the Knowledge in a Neural Network[J]. Computer Science, 2015

[2] X. Chen, Z. Q. Xing and Y. Y. Cheng, "Introduction to Model Compression Knowledge Distillation," 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), 2021, pp. 1464-1467, doi: 10.1109/ICSP51882.2021.9408881.

[3] I. -H. Shin, Y. -H. Moon and Y. -J. Lee, "Towards Understanding Architectural Effects on Knowledge Distillation," 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, pp. 1144-1146, doi: 10.1109/ICTC49870.2020.9289630.

[4] H. Ni, J. Shen and C. Yuan, "Enhanced Knowledge Distillation for Face Recognition," 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 1441-1444, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00207.

[5] Z. Feng, J. Lai and X. Xie, "Resolution-Aware Knowledge Distillation for Efficient Inference," in IEEE Transactions on Image Processing, vol. 30, pp. 6985-6996, 2021, doi: 10.1109/TIP.2021.3101158.

[6] J. Sigurdsson, M. O. Ulfarsson and J. R. Sveinsson, "Sparse and low rank hyperspectral unmixing," 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 229-232, doi: 10.1109/IGARSS.2017.8126936.

[7] F. Ong and M. Lustig, "Beyond Low Rank + Sparse: Multiscale Low Rank Matrix Decomposition," in IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 4, pp. 672-687, June 2016, doi: 10.1109/JSTSP.2016.2545518.

[8] M. O. Ulfarsson, V. Solo and G. Marjanovic, "Sparse and low rank decomposition using l0 penalty," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 3312-3316, doi: 10.1109/ICASSP.2015.7178584.

[9] H. Cao, X. Shang, C. Yu, M. Song and C. -I. Chang, "Hyperspectral Classification Using Low Rank and Sparsity Matrices Decomposition," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 477-480, doi: 10.1109/IGARSS39084.2020.9324009.

[10] M. F. Kaloorazi and R. C. de Lamare, "Low-rank and sparse matrix recovery based on a randomized rank-revealing decomposition," 2017 22nd International Conference on Digital Signal Processing (DSP), 2017, pp. 1-5, doi: 10.1109/ICDSP.2017.8096137.

[11] L.H.Guo,D. Chen,K. Jia.Knowledge transferred adaptive filter pruning for CNN compression and acceleration[J/OL].ScienceChina(InformationSciences):1-2[2022-04-06].

[12] Y.Fang,C.Li,P.C.Wang,C.Q.Han,R.Huang,X. Huang. EasiEdge: A Novel Global Deep Neural Networks Pruning Method for Efficient Edge Computing[J]. IEEE INTERNET OF THINGS JOURNAL,2021,8(3).

[13] Y. S. Ki, S. Philipp, L. Sebastian, B. Alexander,W. Simon, M. K. Robert, S. Wojciech. Pruning by explaining: A novel criterion for deep neural network pruning[J]. Pattern Recognition,2021,115(prepublish).

[14] K. KAMMA, Y. ISODA, S. INOUE, T. WADA. Neural Behavior-Based Approach for Neural Network Pruning[J]. IEICE Transactions on Information and Systems,2020,E103.D(5).

[15] G.Li,F.Liu,Y.P.Xia.Overview of Deep Convolutional Neural Network Pruning[J]. 2020 INTERNATIONAL CONFERENCE ON IMAGE, VIDEO PROCESSING AND ARTIFICIAL INTELLIGENCE,2020,11584.

[16] J.Yang,X.Shen,J.Xing, et al. Quantization networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7308-7316.

[17] C.Zhu, S.Han, H.Mao, et al. Trained ternary quantization[J]. arXiv preprint arXiv:1612.01064, 2016.

[18] V.Moura, V.Almeida, D.B.S.Santos, et al. Mobile Device ECG Classification using quantized Neural Networks[J]. 2020.

[19] F.Liu, W.Zhao, Z.He, et al. Improving neural network efficiency via post-training quantization with adaptive floating-point[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 5281-5290.

[20] H.Pouransari, Z.Tu, O.Tuzel. Least squares binary quantization of neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 698-699.