AMBIGUITY ADAPTIVE INFERENCE AND SINGLE-SHOT BASED CHANNEL PRUNING FOR SATELLITE PROCESSING ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In a restricted computing environment like satellite on-board systems, running DL models has limitation on high-speed processing due to the problems such as restriction of available power to consume compared to the relatively high computational complexity. In particular, the latest GPU resources shows high computing performance but also shows relatively high power consumption, whereas in restricted environments such as satellite systems, reconfigurable resources like FPGA or low power embedded GPU are generally adopted due to their relatively low power consumption compared to computing capability. In such a constrained computing environment, in order to overcome the problem of too huge model size to fit in reconfigurable resources or limitation on high-speed processing, we propose a reconfigurable DL accelerating system where the computing complexity and size of DL model are compressed by pruning and can be adapted to the FPGA or low power GPU resources. Therefore, in this paper, we mainly address an ambiguity adaptive inference model that can enhance overall accuracy in inference step directly for mission critical task, a new method for single-shot based channel pruning that can accelerate inference of DL model through compressing the model as much as possible with maintaining accuracy performance under constrained accelerator resources. From the experimental evaluation, for the satellite image analysis model as an example application, our method can achieve up to $\times 8.53$ compression while keeping the accuracy, and verified that our method can deploy and accelerate the DL model with high computational complexity on FPGA/GPU resources.

1 INTRODUCTION

Recent advances in deep learning (DL) are largely come from the availability of massive amount of training data (Krizhevsky et al., 2012; Xia et al., 2018) and efficient parallel computation supported by modern GPUs. Accordingly, the state-of-the-art deep neural networks have continuously grown in depth and complexity, in pursue of higher performance (e.g., accuracy) and targeting more complex tasks (Devlin et al., 2018; Ren et al., 2016). In response to the increasing complexity of neural network models, advances have been made in various areas such as hardware, network computing infrastructure (Hazelwood et al., 2018), etc. to provide smooth service support in terms of inference serving. As one of these attempts to enable conducting inference for providing services in computation-constrained environments such as satellite on-board system, pruning the neural network itself by removing unnecessary parameters is introduced. Pruning the neural network can reduce the size of whole parameters in neural networks by removing the some redundant parameters, therefore, can alleviate the constraints of the memory and storage space (Hassibi et al., 1993; Han et al., 2016b; Mostafa & Wang, 2019), and accelerate computation of inference process.

However, pruning the neural network by weight-wise removing makes kernel matrices in layers sparse which can not generally shorten the inferencing time without designing a new hardware optimized for the target sparse matrix operations (Han et al., 2016a). Unlike pruning in weight-wise, channel pruning (Li et al., 2017a) removes the whole parameters that is linked to the certain output channel in each layer. As removing a output channel in a layer can be substituted with a smaller layer directly, it can achieve effect of acceleration on processing inference computation. However, remov-

ing a output channel suffers much more accuracy degradation comparing to the weight pruning, therefore, Molchanov et al. (2017) attempts to minimize the performance degradation by conducting iterative training and then pruning cycle to search the output channels to prune.

In the other way, lottery ticket hypothesis (Frankle & Carbin, 2019) introduce a new possibility of pruning that can outperform the original network model. Recent works introduce global weight pruning scheme in the framework of lottery ticket hypothesis that can prune by single-shot manner where pretraining the original model with full iterations is not required (Lee et al., 2019). However, as these single-shot pruning is mainly addressed in weight pruning scheme that can not achieve spectacular reduction on model size and computing acceleration, single-shot based channel pruning criterion is required to accelerate computation and conduct pruning more simply than conventional channel pruning schemes that pretrain the full model with full iterations and then conduct pruning.

In order to overcome the limitations, in this paper, we propose a new method of single-shot based channel pruning that can accelerate DL computations by enhancing robustness over accuracy degradation and deploy huge models into constrained accelerators. We also propose an ambiguity adaptive inference model that can enhance the accuracy in the inference step directly for the mission critical task. We evaluated our method by constructing the reconfigurable DL accelerating system, and verified that our proposed pruning method can largely compress the full model while maintaining the accuracy, and the reconfigured model achieves acceleration on DL computations under FPGA and GPU resources.

2 Related work

DL Processing on Constrained Computing Environments. Deploying or running DL based applications in constrained computing environments like satellite on-board system (Kim et al., 2021), edge server (Kim & Youn, 2020; Liu et al., 2020a), and IoT device (Zhu et al., 2020) encounters problem of too huge model size to deploy on accelerator or limitation on high-speed computing. Cloudscout (Giuffrida et al., 2020) designed very small CNN model and deployed on nanosatellite to select eligible data by cloud detection. As the available hardware resources and power budget are limited in such environment, light-weight DL model is required by constructing short and thin CNN, but the light-weight DL model inevitably results in lower performance than the deeper and wider CNNs in general (Sandler et al., 2018). Moreover, as the input size of the service requests grows up on recent practical applications, required memory occupation size for intermediate feature maps of CNN is prone to exceed the given hardware memory size (Zhao et al., 2018; Akin et al., 2019). For example, deploying Faster-RCNN (Ren et al., 2016) model with 1k x 1k input requires about 9GB resource memory occupation and about 3TFLOP of computational overhead, however, the hardware accelerators for constrained computing environments like NVIDIA TX-1 (Otterness et al., 2017) or Xilinx VC707 (Shawahna et al., 2018) can only accommodate up to 4GB memory and can only perform about 60$\sim$500GFLOPS for DL inference processing which is quite insufficient to process within seconds level.

Single-shot Weight Pruning. The pruning schemes (Molchanov et al., 2017; Han et al., 2016b) that conduct pruning from the pretrained neural network and then fine tunes the pruned network inevitably suffer performance (e.g., accuracy) degradation from the original pretrained network even though further fine tuning is conducted. However, a recent study observes the lottery ticket hypothesis (Frankle & Carbin, 2019) that a randomly initialized dense neural network contains a subnetwork that has same initialization state can match the test accuracy of the original full network after training with at most the same number of iterations. The identification of lottery ticket hypothesis implies that if the original network itself is too large, it can be easier to fall into local optima, and rather pruning into a smaller network can present the possibility of performance improvement.

Developed from the lottery ticket hypothesis, single-shot weight pruning schemes attempt to find the wining ticket effectively by observing dataset only once (not pretraining with full iterations). As one of the study, SNIP (Lee et al., 2019) quantify the effect of the loss on each masking at parameters, and introduce single-shot weight pruning scheme without training the original network model. Others (Tanaka et al., 2020) propose a weight pruning scheme that can find which channels to prune even without any training data by quantifying synaptic strength on total flow on the whole network. All of these works attempts to globally search which links between neurons to prune in a whole network model at once. In terms of first motivation on pruning, pruning weight-wise in a whole network has

advantage of reducing memory and storage size. However, due to the characteristics of the neural network operations, unless a special hardware optimized for the sparsity of each pruned network is designed to be applied (Han et al., 2016a), the computational acceleration can not be achieved on accelerator resources.

Channel pruning. To cope with the computational constrained environments such as satellite on-board system, channel pruning (Li et al., 2017a; He et al., 2017) can be one of the solutions by removing whole operations related to the target output channel in a layer which results in reduction on both the amount of required computation and occupying parameter size. However, as it removes whole links related to the target output channel at once, performance degradation of the network model can be more vulnerable to be affected compared to the weight pruning. Due to this issue, previous studies (Han et al., 2016b; Molchanov et al., 2017) try to suppress performance degradation mainly through conducting training and pruning alternatively with iterative cycles, which burdens the additional computation overhead for training compared to the single-shot based weight pruning method.

To overcome these limitations, in this paper, we aim to address accurate inference model and efficient channel pruning scheme for restricted computational environments, therefore, we analyze how to adapt the single-shot based weight pruning criterion to the channel pruning scheme, and finally propose a single-shot based channel pruning scheme that can identifies wining ticket with largely compressed size. Based on the proposed pruning scheme, in order to realize the acceleration for DL computation, we also developed resource adaptation middleware that practically reconfigure the full model into deployable form for target accelerators as shown in Figure 1.

3 Ambiguity Adaptive Inference and DL Accelerating System for Restricted Computing Environments

In this section, we firstly introduce an ambiguity adaptive inference model that attempt to improve overall accuracy at inference step directly. In order to deploy and accelerate the DL model, we propose a new single-shot based channel pruning method that is robust to accuracy degradation and therefore can further compress the model. In particular, we mainly address how to adapt single-shot based weight pruning criterion to channel pruning scheme with theoretical analysis and how to overcome the vulnerability of removing a certain whole layer in single-shot based channel pruning scheme to enhance robustness of accuracy degradation. On the basis of the proposed single-shot channel pruning scheme, we developed resource adaptation layer that practically reconfigures the DL models to heterogeneous accelerators (FPGA, GPU) and integrated as a reconfigurable DL accelerating system.

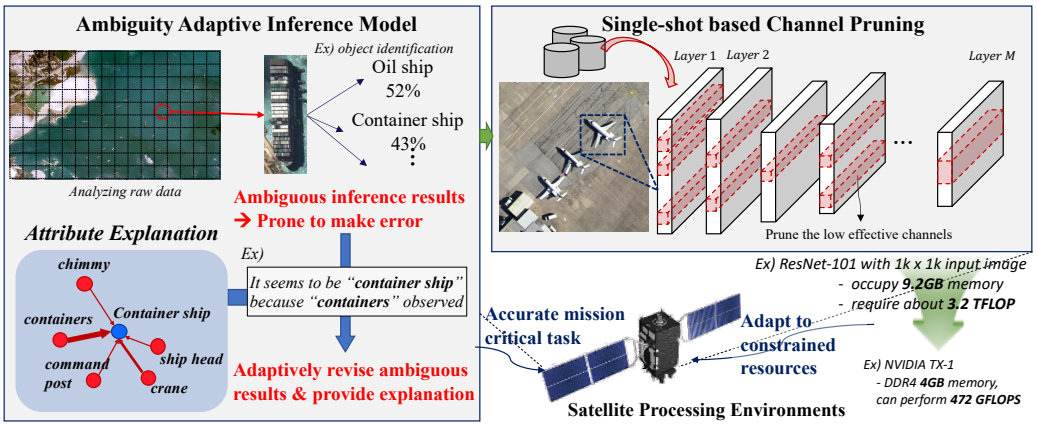

Figure 1: Illustration of ambiguity adaptive inference and single-shot based channel pruning for satellite processing environments

3.1 AMBIGUITY ADAPTIVE INFERENCE MODEL

As the task under noisy or low resolution data makes the DL model hard to achieve the high task performance (i.e., accuracy), the recent studies (Zhang et al., 2019; Mo et al., 2021) attempt to overcome it by mitigating on efficient training process. However, unless the model can not guarantee the 100% accuracy, the model still can invoke false alarm which is critical to certain environments such as remote sensing (Li et al., 2020). Therefore, as an attempt to enhance the performance in the inference step directly, we propose a hypothesis that ambiguity on inference result of DL model can represent the error. From the hypothesis, we can attempt to improve the performance of the DL model in inference step by adaptively revising the inference results with high ambiguity from external knowledge.

In this paper, we focus on the classification task, and we try to discriminate the ambiguity by the maximum class probability ($\max_{c \in C} P(y = c|\mathbf{x})$, where C denotes a set of classes, and $\mathbf{x}$ denotes input) of inference result which is a general indicator to make decision in classification task (Krizhevsky et al., 2012). Evolved from the work of Corbière et al. (2019), we derive the property that discriminating the ambiguity by maximum class probability can approximately discriminate the error of inference result as follows.

Lemma 1. *If* $\max_{c \in C} P(y = c|\boldsymbol{x}) > \sigma_l, 1/2 \leq \sigma_l < 1$, *then the estimation result is probably approximately correct. Otherwise, if* $\max_{c \in C} P(y = c|\boldsymbol{x}) \leq \sigma_h, 0 < \sigma_h \leq 1/2$, *then the estimation result can be probably approximately wrong.*

The proof of the lemma is presented in detail at Appendix. From the fundamentals of the property, we can try to revise the inference result with high ambiguity ($\max_{c \in C} P(y = c|\mathbf{x}) \leq \sigma_h$) that is likely to be wrong to improve the overall performance of DL model directly in inference step adaptively as shown in Figure 2.

For example, we introduce a knowledge graph based revising step to revise ambiguous results (suspected to be wrong prediction) in ship classification from satellite imagery data. The several studies (Fang et al., 2017; Liu et al., 2020b) try to use knowledge graph of co-occurrence between objects to improve the task performance. However, for the object identification on satellite imagery, not the co-occurrence between objects but the existence of distinct subordinate attribute (e.g., container, tank dome, crane, etc. for ship classification) can represent the certain object class. Accordingly, in order to extract the existence information of subordinate attributes, we introduce a multi-attribute classifier which is more light-weight and have smaller training complexity than the detector which is generally used in similar studies of scene graph generation (Yang et al., 2018; Li et al., 2017b). For the knowledge graph, as an example, we apply the term frequency-inverse document frequency (TF-IDF) as an edge value ($e(c, a)$) to quantify the co-occurrence between attribute ($a \in A$, where A denotes a set of attributes) and object classes ($c \in C$), which is widely used for image classification tasks (Chanti & Caplier, 2018). Therefore, on the inference results with high ambiguity (i.e.,

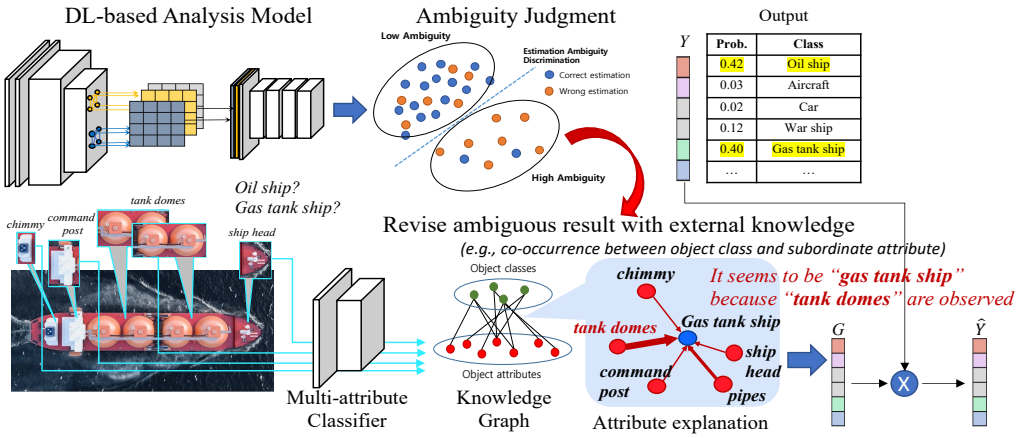

Figure 2: An example illustration of ambiguity adaptive inference model

samples with $\max_{c \in C} P(y=c|x) \leq \sigma$, where σ denotes the certain ambiguity threshold value), the prediction (P_{rev}) is revised by multiplying the inference result with prediction from the knowledge graph as:

$$P_{rev}(y=c|x) = P(y=c|x, \theta_{task}) \cdot \frac{\exp(\sum_{a \in A} P_{a,\theta_{mac}} \cdot e(c,a))}{\sum_{c \in C} \exp(\sum_{a \in A} P_{a,\theta_{mac}} \cdot e(c,a))}, \tag{1}$$

where θ_{task} denotes parameters of original model for task, θ_{mac} denotes parameters of multi-attribute classifier, $P_{a,\theta_{mac}}$ denotes derived probability of a attribute occurrence on input x from multi-attribute classifier θ_{mac}, and $P(y = c|x, \theta_{task})$ is the probability of class c for input x derived from task model with θ_{task}. Otherwise, on low ambiguity cases (i.e., samples with $\max_{c \in C} P(y=c|x) > \sigma$), just the derived inference result from the task model is used to make the final decision for the task. As observed in Lemma 1, if we can set the appropriate threshold for discriminating high ambiguity case that can screen out the wrong inference results with high confidence, overall task accuracy can be improved by trying to correct wrong predictions in further adaptive revision step.

However, as discussed in the previous section, the recent DL model itself contain high computation complexity, and the model compression is required to deploy onto restricted computing environments. Accordingly, we propose a new single-shot base channel pruning scheme that is robust to performance degradation in the following subsection.

3.2 Single-shot based Channel Pruning

Let n_i, h_i, and w_i denote the number of output channels, height and width of output feature map in ith layer, respectively. From the input $\mathbf{x}_{i-1} \in \mathbb{R}^{n_{i-1} \times h_{i-1} \times w_{i-1}}$, a convolution layer conducts $n_i \cdot n_{i-1}$ convolution operations by 2D kernel to output feature map $\mathbf{x}_i \in \mathbb{R}^{n_i \times h_i \times w_i}$. Each kernel parameter $\theta^i_{p,q} \in \mathbb{R}^{k \times k}$ for linking pth output channel and qth input channel in ith layer constructs a filter matrix $\mathcal{K}_i \in \mathbb{R}^{n_i \times n_{i-1} \times k \times k}$ for ith layer (Let $\mathcal{K}$ denote a set of whole filter parameters in the network). Pruning the jth output channel in ith layer can be represented as conduct masking to the filter matrix denoted by $\mathcal{M}_i \odot \mathcal{K}_i$, where $\odot$ denotes Hadamard product (element-wise product), $\mathcal{M}_i \in \{0,1\}^{n_i \times n_{i-1} \times k \times k}$ is a masking matrix for ith layer where each element in the matrix represents the connectivity of parameters. We denote $\mathcal{M}$ as a set of whole masking matrices in the network, and denote $\mathcal{M} \odot \mathcal{K}$ as masking in each layer by $\mathcal{M}_i \odot \mathcal{K}_i$. For convenience, let $m^i_j \in \{0,1\}$ denotes controllable masking indicator for the parameters that are related to jth output channel in ith layer where $m^i_j = 0$ corresponds to making all elements in $\mathcal{M}_i$, $\mathcal{M}_{i+1}$ with jth index at first, second dimension respectively as zero. When jth output channel in ith layer is pruned, 3D filter $\mathcal{K}_{i,j} \in \mathbb{R}^{n_{i-1} \times k \times k}$ in jth index at first dimension on the (4-dimensional) filter matrix $\mathcal{K}_i$ is masked by $\mathcal{M}_i|_{m^i_j=0}$, and also the filters $(\theta^{i+1}_{j,q}, \forall q)$ in jth index at second dimension on the filter matrix $\mathcal{K}_{i+1}$ in next layer is masked by $\mathcal{M}_{i+1}$.

Problems of applying single-shot weight pruning criterion to channel pruning directly. When applying the criterion of single-shot weight pruning (Lee et al., 2019) directly to channel pruning as like in (Li et al., 2017a), we observe that pruning by the sum of each weight masking effect can not guarantee equal to empirical risk minimization problem. From the pruning criterion in SNIP (Lee et al., 2019), $s^i_{p,q} = \frac{|g^i_{p,q}(\mathcal{K};\mathcal{D})|}{\sum_{i'}\sum_{p'}\sum_{q'}|g^{i'}_{p',q'}(\mathcal{K};\mathcal{D})|}$ where $g^i_{p,q} = \frac{\partial L(\mathcal{M} \odot \mathcal{K};\mathcal{D})}{\partial m^i_{p,q}}|_{\mathcal{M}=1} \approx L(\mathcal{K};\mathcal{D}) - L(\mathcal{K};\theta^i_{p,q} = 0,\mathcal{D})$, and i, p, q denote index of layer, output channel, input channel respectively. Pruning a channel by conducting element-wise product on filters with masking matrix ($\mathcal{M}_i \odot \mathcal{K}_i$) can be considered as equal to finding $\mathcal{K}_i$ with which channel to be pruned ($\theta^i_{j,q} = 0, \forall q$) in the finite hypothesis space H_i where the space consists of all possible channel pruning cases. Therefore, the problem that choose a channel to prune by using sum of sensitivity score can be written as $\arg\min_{j \in [0,n_i)} \sum_q s^i_{j,q} = \arg\min_{\mathcal{K}|_{\theta^i_{j,q}=0,\forall q} \in H_i} \sum_q s^i_{j,q}$. From $\sum_q s^i_{j,q} = \frac{\sum_q |L(\mathcal{K};\mathcal{D}) - L(\mathcal{K};\theta^i_{p,q}=0,\mathcal{D})|}{\sum_{i'}\sum_{p'}\sum_{q'}|g^{i'}_{p',q'}(\mathcal{K};\mathcal{D})|}$, the denominator term and $L(\mathcal{K};\mathcal{D})$ term in numerator are constant with regard controlling variable j (wrapped as $\mathcal{K}_i|_{\theta^i_{j,q}=0,\forall q}$). As the cross entropy is usually used as loss function, when assume that $L(\cdot) > 0$, the problem in the equation becomes

$\arg\min_{\mathcal{K}|_{\theta^i_{j,q}=0,\forall q}\in H_i} \frac{\sum_q |\alpha - L(\mathcal{K};\theta^i_{p,q}=0,\mathcal{D})|}{\beta}$, where α and β are positive constants. The derived problem equation does not guarantee equivalent problem form of empirical risk minimization on the network ($\neq \arg\min_{\mathcal{K}'\in H'} L(\mathcal{K}';\mathcal{D})$). Therefore, applying single-shot weight pruning criterion to the channel pruning by summing all related weight-wise scores (Li et al., 2017a) can not show its efficiency in terms of minimizing empirical loss on the network.

Global channel sensitivity. Alternatively, we transformed the single-shot based weight pruning criterion in SNIP (Lee et al., 2019) for channel pruning by defining global channel sensitivity as:

$$CS^i_j = \frac{|g^i_j(\mathcal{M}\odot\mathcal{K};\mathcal{D})|}{\sum_{i'}\sum_{j'}|g^{i'}_{j'}(\mathcal{M}\odot\mathcal{K};\mathcal{D})|}, \tag{2}$$

where $g^i_j(\mathcal{M}\odot\mathcal{K};\mathcal{D}) = \frac{\partial L(\mathcal{M}\odot\mathcal{K};\mathcal{D})}{\partial m^i_j}|_{\mathcal{M}=1} \approx L(\mathcal{M}\odot\mathcal{K};\mathcal{D}) - L(\mathcal{M}|_{m^i_j=0}\odot\mathcal{K};\mathcal{D})$. By observing the output channel masking-wise sensitivity, not the sum of weight-wise sensitivity, pruning a channel with this criterion (CS^i_j) can be stated as an equivalent problem of optimizing the performance (minimizing the empirical loss) of the network, as shown in the following theorem and its corollary.

Theorem 1. *If $L(\mathcal{M}\odot\mathcal{K};m^i_j=0,\mathcal{D}) \geq L(\mathcal{M}\odot\mathcal{K};\mathcal{D}) \geq 0, \forall(i,j)\in\{(i,j)|m^i_j=1,\forall m^i_j\in\mathcal{M}\}$, pruning a channel by CS^i_j is equal to solving empirical risk minimization (ERM) problem of the neural network in the finite hypothesis space of pruning.*

Corollary 1. *As solving ERM guarantees probably approximately correct (PAC) bound, under the same condition in Theorem 1., pruning a channel by CS^i_j also guarantees PAC bound and its estimation error is upper bounded.*

Detailed proofs for the theorem and its corollary are presented in Appendix A.1. The above theorem and its corollary represent that channel pruning based on CS^i_j explores in the direction of reducing error and show its validity. For the exception case, $0 < L(\mathcal{M}\odot\mathcal{K};m^i_j=0,\mathcal{D}) < L(\mathcal{M}\odot\mathcal{K};\mathcal{D})$ is the case where pruning reduces the loss than when pruning is not conducted, in which case pruning can be interpreted as a direct solution of ERM problem.

However, when conducting channel pruning by CS^i_j, the summed score of eliminating i-th layer equals $\sum_j CS^i_j$. The effect of layer deletion in score ($\sum_j CS^i_j$) just occupy a fraction of the overall score ($\sum_i\sum_j CS^i_j$) of the original network. In other words, when exploring which channels to prune on the basis of CS^i_j, there exist a risk that can be dropped into a local optimal where pruning criterion deletes a whole particular layer. This vulnerability of removing a layer on channel pruning scheme can be observed in Figure 5, and the detail will be discussed in Section 4.

Layer-wise sensitivity. To address the vulnerability of removing a certain layer, in the methods of pruning from pretrained state, layer-wise accuracy degradation curves with regard to the pruning ratio can be obtained by profiling, and it can be used to regulate pruning a particular layer excessively (Li et al., 2017a). However, the layer-wise performance degradation curves can not be practically obtained in single-shot pruning scheme as it requires training with full iterations. Instead, when exploring j-th output channel to be pruned in i-th layer ($\mathcal{M};m^i_j=0$), we define a new layer-wise sensitivity for single-shot pruning scheme in the inverse form of a total score sum on the remaining channels (at the point after pruning the candidate channel) of that layer as follows:

$$LS_i(\mathcal{M};m^i_j=0) = \frac{1}{\sum_{j'=0}^{n_i-1} CS^i_{j'} - \sum_{j'\in\{j'|CS^i_{j'}=0,\forall c^i_{j'}\in\mathcal{M}_i\}} CS^i_{j'}}. \tag{3}$$

By the arithmetical property of the layer-wise sensitivity, it regulates to remove a particular layer at all by computing numerator term as 0 to make layer-wise sensitivity as infinity when try to prune the last remaining output channel in any layer.

Therefore, based on the aforementioned properties, we propose a single-shot based channel pruning scheme with layer-wise sensitivity as described in Algorithm 1. The proposed scheme searches channels to prune in a whole network globally by selecting a channel that shows minimum $CS^i_j \cdot$

$LS_i(\mathcal{M}; m_j^i = 0)$ score, iteratively updating layer-wise sensitivity, where layer-wise sensitivity term on score suppresses to choose a channel in excessively pruned layer on remaining searching space.

Algorithm 1 Single-shot channel pruning scheme with layer-wise sensitivity

Input: Target overall pruning ratio pr

$m_j^i \leftarrow 1$ for $\forall i, j$ ▷ Initialize $\mathcal{M}$
calculate CS_j^i for $\forall i, j$ ▷ Obtain channel sensitivity score in single-shot manner
while $(\sum_i \sum_j m_j^i > \sum_i n_i \cdot (1 - pr))$ **do** ▷ Check target overall pruning ratio
    update $LS_i(\mathcal{M})$ for $\forall i$ or i^* of previous step
    $s_{min}^* \leftarrow \infty$
    $(i^*, j^*) \leftarrow \emptyset$
    for i **in** range(K) **do** ▷ Search a best channel to prune in current iteration
        $j \leftarrow \arg\min_{j \in \{j | m_j^i = 1\}} CS_j^i \cdot LS_i(\mathcal{M}; m_j^i = 0)$
        $s_{min} \leftarrow CS_j^i \cdot LS_i(\mathcal{M}; m_j^i = 0)$
        if $(s_{min}^* > s_{min})$ **then**
            $s_{min}^* \leftarrow s_{min}$
            $(i^*, j^*) \leftarrow (i, j)$
        end if
    end for
    $\mathcal{M} \leftarrow \mathcal{M}|_{c_{j^*}^{i^*} = 0}$ ▷ Apply the new searched channel to prune, update $\mathcal{M}$
end while
return $\mathcal{M}$ ▷ Return the searched pruning channels

4 Evaluation

Feasibility of ambiguity adaptive inference model. First, we evaluate the feasibility of our ambiguity adaptive inference model. We evaluate on ship classification from xView satellite imagery dataset (Lam et al., 2018) with ResNet-18 network. From the trained model, we measure correct detecting accuracy (the detected correct ratio from the total correct inference results) for the samples discriminated as low ambiguity (samples with $\max_{i \in C} P(y = i|x) > \sigma$ on test set) and wrong detecting accuracy for the samples discriminated as high ambiguity (samples with $\max_{i \in C} P(y = i|x) \leq \sigma$ on test set) with regard to various threshold (σ) levels. As shown in Figure 3, high ambiguity case can detect more wrong samples with higher threshold, whereas the low ambiguity case can detect more correct samples with lower threshold. Therefore, there exist the optimal threshold according to the revising performance that can improve overall accuracy performance by applying the further revision step on high ambiguity cases. In order to observe the practical feasibility of our ambiguity adaptive inference model, as an example, we test on applying knowledge graph of co-occurrence between object class and subordinate attribute as revising step for high ambiguity cases. For the revision step, we multiply the prediction result from the knowledge graph with the prediction re-

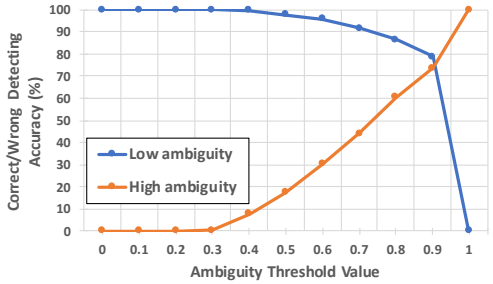

Figure 3: Correct/wrong detecting accuracy for discriminated low/high ambiguity cases over various threshold levels

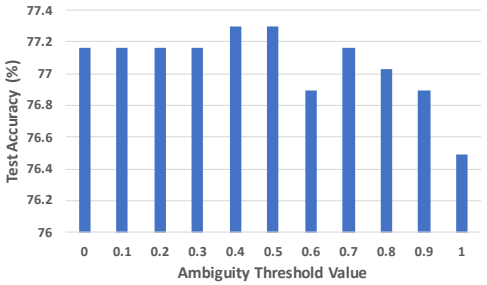

Figure 4: Top-1 test accuracy achieved by ambiguity adaptive inference model over various threshold levels

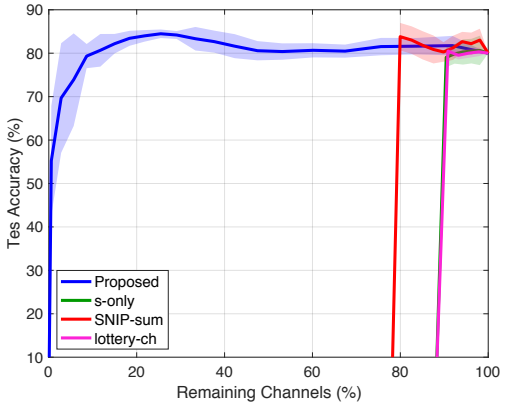

Figure 5: Test accuracy with respect to the percentage of remaining channels over pruning methods

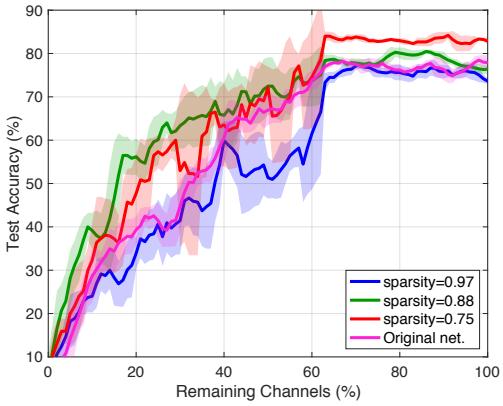

Figure 6: Convergence of test accuracy on training the model pruned by the proposed scheme over various sparsity levels

sult from the DL model as like Liu et al. (2020b). The prediction from the knowledge graph only just achieved about 30% accuracy. Figure 4 shows the top-1 test accuracy of the ambiguity adaptive inference model with regard to the ambiguity threshold (σ). The result shows that our ambiguity adaptive inference model can achieve accuracy enhancement at certain threshold values (0.4 or 0.5) although the accuracy of the revision step only is lower than the original inference step. For the high threshold values, as more correct samples are detected as the high ambiguity cases and the accuracy of revision step only is lower than the original step, the overall accuracy fall down by wrong revision on correct samples.

Performance of the proposed pruning scheme. In addition, we evaluate the effectiveness of the proposed pruning algorithm and its accelerating effect on computing system empirically. For the network model, as an example application of satellite on-board processing, we mainly evaluate on ResNet-101 (He et al., 2016) with UC Merced land use satellite imagery dataset (Yang & Newsam, 2010). The further evaluations on various models that show similar tendency are also presented in Appendix A.2. The proposed pruning method is evaluated by comparing two conventional methods as follows:

- `SNIP-sum`: Adapting single-shot weight pruning method of SNIP (Lee et al., 2019) to channel pruning scheme by summing up weight-wise scores linked to target channel (Li et al., 2017a).
- `lottery-ch`: Adapting weight pruning method on evaluation of lottery ticket hypothesis (Frankle & Carbin, 2019) to channel pruning scheme by scoring sum on magnitude of weight parameters linked to output channel.

First, we evaluated robustness of accuracy degradation among the proposed pruning scheme and the other comparing methods (`SNIP-sum`, `lottery-ch`). For all comparing methods, after re-initializing the pruned model, we train the pruned model with 160 epoch and observe the best top-1 test accuracy as the performance of the model. We test on 20 pruning ratios, and observe the accuracy results with regard to the percentage of overall remaining channels. As shown in Figure 5, in the case of `SNIP-sum` or `lottery-ch`, as we discussed in the previous section, by the scale difference on criterion score over layers, layer removing occur from near 10%$\sim$20% pruning, which shows that global searching scheme in channel pruning is vulnerable to removing a whole certain layer. As we prune the channels in layers linked to the residual link together, this vulnerability appear more remarkably. Unlike the other comparing methods, the proposed scheme shows improvement on robustness of the accuracy degradation by regulating any layer to be excessively pruned via layer-wise sensitivity term, and can achieve up to x8.53 compression while maintaining the accuracy of the original network model.

As a brief ablation study, we also observe accuracy result of pruning with only the proposed channel sensitivity (denoted as `s-only` in the result graph). Likewise to `SNIP-sum` and `lottery-ch`,

Table 1: Effect of acceleration for inference serving on each accelerator (FPGA/GPU)

	Proposed	SNIP-sum	lottery-ch	Original
Accuracy (%)	81.43	83.81	80.48	80.00
Mem. occ. (MiB)	2,911	5,681	4,983	5,767
Latency on GPU (ms)	81.38	374.63	379.12	523.43
Speed up on GPU	$\times$6.43	$\times$1.40	$\times$1.38	$\times$1.00
Latency on FPGA (ms)	5.20	15.55	16.96	23.75
Speed up on FPGA	$\times$4.56	$\times$1.53	$\times$1.40	$\times$1.00

Table 2: Effect of throughput improvement on GPU deployed with maximum available batch size

	Proposed	SNIP-sum	lottery-ch	Original
Max. batch size	87	32	38	31
Throughput (Req/s)	198.2	42.66	42.08	30.7
Improvement	$\times$6.45	$\times$1.39	$\times$1.37	$\times$1.00

pruning with the proposed channel sensitivity only also shows vulnerability of removing a certain layer, and this problem is alleviated by additionally considering layer-wise sensitivity.

In addition, in order to observe whether the proposed scheme can find wining ticket or not, we also examine the test accuracy convergence trend with regard to training epochs. As shown in Figure 6, the proposed scheme is able to find the wining ticket that can achieve the accuracy of the original network below the certain sparsity level (that will not excessively pruning over the capacity of the network model). In the results, until about sparsity with 0.88, proposed scheme can maintain to find wining ticket.

Accelerating effect on the system. Finally, we evaluate accelerating effect on the practical computing system where the full model is reconfigured to the deployable form for target accelerators. We developed and test the system under the computing environment with FPGA and GPU accelerators where hardware consists of Intel(R) Xeon(R) Silver 4214R CPU @2.40GHz, NVIDIA RTX 3080 GPU, and Xilinx Alveo U200 FPGA. Inference serving latency and throughput are measured over applied pruning methods where the model with pruned as much as possible to maintain accuracy of original network is deployed to each accelerator. We measure in 100 trials, and observe the averaged value for each test case.

Table 1 shows accelerating effect on inference serving by setting same batch size (i.e., 16 for GPU and 1 for FPGA) over applied pruning methods. The model reconfigured by the proposed pruning scheme can achieve smaller GPU memory occupation than comparing pruning methods and original full model, and also achieve the highest accelerating enhancement ($\times$6.43 on GPU and $\times$4.56 on FPGA) on latency at each accelerators while maintaining the accuracy of original model.

In particular, for the GPU resource, the pruned model can also enlarge the available maximum batch size to deploy, which can improve the throughput of inference serving accordingly. As shown in Table 2, the model reconfigured from the proposed pruning method can achieve the highest throughput improvement ($\times$6.45) among the other pruning methods and original network by enlarging the maximum available batch size to 87.

5 Conclusion

In this paper, we propose an ambiguity adaptive inference model that can improve the accuracy for mission critical task at inference step directly and a new single-shot channel pruning scheme that can provide feasibility of accelerating DL computations. We see the validity of fundamentals on our methods theoretically and develop the practical accelerating system for empirical evaluation. From the empirical evaluation, our ambiguity adaptive inference model shows feasibility to improve accuracy, and our pruning method shows higher accelerating effect than the conventional pruning methods under the developed system.

REFERENCES

Berkin Akin, Zeshan A Chishti, and Alaa R Alameldeen. Zcomp: Reducing dnn cross-layer memory footprint using vector extensions. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 126–138, 2019.

Dawood Al Chanti and Alice Caplier. Improving bag-of-visual-words towards effective facial expressive image classification. *arXiv preprint arXiv:1810.00360*, 2018.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *NeurIPS*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object detection meets knowledge graphs.(2017). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence: Melbourne, Australia, August 19*, volume 25, pp. 1661–1667, 2017.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR*, 2019.

Gianluca Giuffrida, Lorenzo Diana, Francesco de Gioia, Gionata Benelli, Gabriele Meoni, Massimiliano Donati, and Luca Fanucci. Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images. *Remote Sensing*, 12(14):2205, 2020.

Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, 2016a.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016b.

Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pp. 293–299. IEEE, 1993.

Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, et al. Applied machine learning at facebook: A datacenter infrastructure perspective. In *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 620–629. IEEE, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1389–1397, 2017.

Heejae Kim, Kyungchae Lee, Changha Lee, Sanghyun Hwang, and Chan-Hyun Youn. An alternating training method of attention-based adapters for visual explanation of multi-domain satellite images. *IEEE Access*, 9:62332–62346, 2021.

Woo-Joong Kim and Chan-Hyun Youn. Lightweight online profiling-based configuration adaptation for video analytics system in edge computing. *IEEE Access*, 8:116881–116899, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *ICLR*, 2019.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *ICLR*, 2017a.

Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020.

Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE international conference on computer vision*, pp. 1261–1270, 2017b.

Ruixuan Liu, Yang Cao, Hong Chen, Ruoyang Guo, and Masatoshi Yoshikawa. Flame: Differentially private federated learning in the shuffle model. In *AAAI*, 2020a.

Zheng Liu, Zidong Jiang, Wei Feng, and Hui Feng. Od-gcn: Object detection boosted by knowledge gcn. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6. IEEE, 2020b.

Sangwoo Mo, Hyunwoo Kang, Kihyuk Sohn, Chun-Liang Li, and Jinwoo Shin. Object-aware contrastive learning for debiased scene representation. *arXiv preprint arXiv:2108.00049*, 2021.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *ICLR*, 2017.

Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655. PMLR, 2019.

Nathan Otterness, Ming Yang, Sarah Rust, Eunbyung Park, James H Anderson, F Donelson Smith, Alex Berg, and Shige Wang. An evaluation of the nvidia tx1 for supporting real-time computer-vision workloads. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 353–364. IEEE, 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Ahmad Shawahna, Sadiq M Sait, and Aiman El-Maleh. Fpga-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access*, 7:7823–7859, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *NeurIPS*, 2020.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pp. 270–279, 2010.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Shuo Zhang, Guanghui He, Hai-Bao Chen, Naifeng Jing, and Qin Wang. Scale adaptive proposal network for object detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 16(6):864–868, 2019.

Zhuoran Zhao, Kamyar Mirzazad Barijough, and Andreas Gerstlauer. Deepthings: Distributed adaptive deep learning inference on resource-constrained iot edge clusters. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2348–2359, 2018.

Dixian Zhu, Dongjin Song, Yuncong Chen, Cristian Lumezanu, Wei Cheng, Bo Zong, Jingchao Ni, Takehiko Mizoguchi, Tianbao Yang, and Haifeng Chen. Deep unsupervised binary coding networks for multivariate time series retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1403–1411, 2020.