CALPA-NET: Channel-pruning-assisted Deep Residual Network for Steganalysis of Digital Images

Shunquan Tan, Senior Member, IEEE, Weilong Wu, Zilong Shao, Qiushi Li, Bin Li, Senior Member, IEEE, and Jiwu Huang, Fellow, IEEE

Abstract—Over the past few years, detection performance improvements of deep-learning based steganalyzers have been usually achieved through structure expansion. However, excessive expanded structure results in huge computational cost, storage overheads, and consequently difficulty in training and deployment. In this paper we propose CALPA-NET, a ChAnneL-Pruning-Assisted deep residual network architecture search approach to shrink the network structure of existing vast, over-parameterized deep-learning based steganalyzers. By providing theoretical rationale, we demonstrate that the broad inverted-pyramid structure of existing deep-learning based steganalyzers contradicts the well-established model diversity oriented philosophy and is not suitable for steganalysis. Then a hybrid criterion combined with two network pruning schemes is introduced to adaptively shrink every involved convolutional layer in a data-driven manner. The resulting network architecture presents a new height. Aiming at large-scale JPEG image steganalysis, our proposed CALPA-NET can achieve comparative detection performance with less than two percent of parameters and about one third FLOPs compared to the original steganalytic model. The new model possesses even better adaptivity, transferability, and scalability.

Index Terms—steganalysis, steganography, deep learning, convolutional neural network, network pruning.

I. INTRODUCTION

STEganalysis aims to detect covert communication established via steganography. In addition to detection performance, computational cost, model complexity, as well as adaptivity, transferability, and scalability, are all important considerations for a real-world oriented steganalytic framework.

Over the last decade, the main battleground of the war between modern steganography and steganalysis has always been in digital images [1], [2]. Most of the spatial-domain and frequency-domain image steganographic algorithms have adopted the embedding distortion minimizing framework [3], [4]. The prominent additive embedding distortion minimizing schemes include HILL [5] and MiPOD [6] in spatial domain, as well as UED [7] and UERD [8] in JPEG domain (the dominant frequency domain in practice). UNIWARD [9] is a benchmarking cross-domain scheme (noted as S-UNIWARD in spatial domain, while J-UNIWARD in JPEG domain).

In recent years, deep-learning frameworks, especially Convolutional Neural Networks (CNNs) have achieved overwhelming superiority over conventional approaches in many fields [10]. In the meanwhile, from early AlexNet [11] and VGGNet [12], to later more advanced Inception models [13] and ResNet [14], the literature witnessed deep-learning frameworks have become more and more deeper and complicated with improvements on structures, such as Batch Normalization (BN) [15] and residual connections.

The once overlord of image steganalysis is the “rich model” hand-crafted features family [16]–[20] equipped with an ensemble classifier [21]. Started from the work of Tan and Li [22], breakthroughs were made in deep-learning based steganalysis [23]–[25]. Then Ye et al. proposed a deep-learning based steganalyzer equipped with a new activation function called Truncated Linear Unit (TLU) [26], achieving significant improvement compared to those established “rich model” steganalytic features in spatial domain. In JPEG domain, deep-learning based steganalyzers have also outperformed the “rich model” features family. Chen et al. proposed a specific deep-learning based steganalyzer aware of JPEG phase [27]. Xu proposed a 20-layer much deeper framework with residual connections (named XuNet2) [28]. XuNet2 took the detection performance of deep-learning based JPEG steganalyzers to a new height. Aiming at large-scale JPEG image steganalysis, Zeng et al. proposed a generic hybrid deep-learning framework incorporating the domain knowledge behind rich steganalytic models, especially their hand-crafted convolutional kernels for residual extraction and the quantization & truncation phase [29]. On the contrary, Boroumand et al. proposed a deep residual steganalytic network designed to minimize the use of heuristic domain knowledge (named SRNet) [30]. SRNet is a cross-domain deep-learning based steganalyzer. It has shown superior detection performance for both spatial domain and JPEG domain steganography. In [31], Zeng et al. explicitly considered the correlation among color bands and proposed WISERNet, the wider separate-then-reunion network specifically designed for steganalysis of true-color images.

It has been widely acknowledged that deep-learning frameworks are over parameterized, and are therefore with huge

1Throughout this paper, for those acronyms taken from the original papers, their corresponding full names are omitted for brevity.
computational cost and storage overheads. Consequently, they are more likely to require massive amounts of training data to achieve good performance, and are hard to deploy [32], [33]. Network pruning is one of the most popular methods to reduce network complexity. It has become one important research problem even since a very early stage of the evolution of deep-learning frameworks [14]. For CNNs, state-of-the-art network pruning approaches can be classified into two categories, the non-structured weights pruning and the channel-based structured pruning methods. Non-structured weights pruning methods [35] cannot lead to complexity reduction without the support of dedicated hardware. Therefore structured pruning methods are more practical. Hu et al. proposed a channel pruning method based on the percentage of zeros in the outputs [36]. Li et al. proposed another channel pruning method based on the $l_1$ norms of the corresponding filter weights [37]. Luo et al. proposed ThiNet which greedily prunes those channels with smallest effects on the activation values of the next layer [38]. Further on, Huang and Wang used sparsity regularization to prune channels and even coarser structures, such as residual blocks [39]. The works mentioned above are just the notable ones selected from a complete collection. But, whatever the approach, the existing network pruning approaches have all followed a typical three-stage pipeline: training, pruning, and then finetuning in order to preserve a set of inherited learned important weights.

In this paper, we propose CALPA-NET, a channel-pruning-assisted deep residual network architecture search approach to shrink the network structure of existing vast, over-parameterized deep-learning based steganalyzers. We demonstrate that the established “doubling the number of channels along with halving the size of output feature maps” rule might undermine the diversity of output features of deep steganalytic models. Therefore start from existing bloated deep steganalytic models, CALPA-NET utilizes a hybrid criterion to adaptively determine the number of channels of every involved convolutional layer. The proposed hybrid criterion, which combines the ThiNet scheme [38] and the $l_1$-norm based scheme [37], is specifically designed for residual modules widely used in deep steganalytic networks. Please note that CALPA-NET, where we abandon the training-pruning-finetuning pipeline of typical network pruning methods, is entirely different from network pruning approaches mentioned above since the proposed channel pruning criterion is just used to search efficient network architectures. The extensive experiments conducted on public datasets show that even trained from scratch, the shrunken model can still achieve state-of-the-art detection performance with merely a tiny proportion of parameters and much lower computational complexity.

The rest of the paper is organized as follows. Sect. II firstly gives a brief overview of existing representative deep steganalytic models as well as channel-based structured pruning methods. Then theoretical testimonies are provided to support the rationale of CALPA-NET. Next the procedure with our proposed hybrid channel pruning criterion used in CALPA-NET is described in detail. Results of experiments conducted on public datasets are presented in Sect. III. Finally, we make a conclusion in Sect. IV.

II. OUR PROPOSED CALPA-NET

A. Preliminaries

As far as we know, existing deep-learning based steganalyzers are all based on CNNs (Convolutional Neural Network). The principal part of CNN can be modeled as a direct graph of alternating convolutional layers and auxiliary layers (e.g. BN layers [15] and pooling layers). Convolutional layers are the central components of a CNN since they contain the overwhelming majority of learnable weights and biases. For a given convolutional layer $L_t$, it takes an after-activation input tensor $\hat{Z}^{l-1} \in \mathbb{R}^{J^{l-1} \times H^{l-1} \times W^{l-1}}$ which possesses $J^{l-1}$ input channels with height $H^{l-1}$ and width $W^{l-1}$, convolves it with $W^l \in \mathbb{R}^{J^l \times K^l \times W^l \times W^l}$, a filter tensor consisting of $J^{l-1} \times K^l$ kernels with size $W^l \times W^l$, and generates $Z^l \in \mathbb{R}^{K^l \times H^l \times W^l}$, the corresponding before-activation output tensor which has $K^l$ output channels (feature maps) with height $H^l$ and width $W^l$. The convolution operation can be modeled as (the bias is omitted for brevity):

$$Z^l_{kj} = \sum_{j=1}^J \hat{Z}^{l-1}_{kj} \cdot W^l_{jk} \cdot 1 \leq k \leq K$$

where $\hat{Z}^{l-1}_{kj}$, $W^l_{jk}$, and $Z^l_{kj}$ denotes the slides, the two dimensional sections defined by fixing all the but the last two indices, of $\hat{Z}^{l-1}$, $W^l$, and $Z^l$, respectively.

1) Interior structures of deep-learning based steganalyzers: Two popular deep-learning based steganalyzers, SRNet [30] and XuNet2 [28] are taken as examples. Their conceptual structures are illustrated in Fig. 1. The two deep-learning based steganalyzers both take spatial representation of the target image as input. For JPEG format target images, their decompressed (non-rounded and non-truncated) counterparts are used as spatial representations. All deep-learning based steganalyzers can be divided into three modules: the bottom module aiming at extracting the so-called “stego noise residuals”, the middle module which striving for learning compact representative features and the top module which is a simple binary classifier.

Actually most of the existing research works regarding to deep-learning based steganalyzers have been devoted to the bottom module. The bottom module of XuNet2 adopts a particular truncated filter bank with sixteen $4 \times 4$ DCT (Discrete Cosine Transformation) high-pass filters. Further on, the bottom module of SRNet consists of a pile of two hierarchical convolutional layers (“L1” and “L2”), following the notations in [30] and five unpooped residuals blocks with direct shortcut connections (from “L3” to “L7”) for extraction of noise residuals. Conversely, according to our best knowledge, no specific domain knowledge has ever been introduced to guide the design of the middle module as well as the top module. Researchers just simply followed the effective recipes in other research fields (e.g. computer vision). Nowadays, almost all of the state-of-the-art deep-learning based steganalyzers, including SRNet and XuNet2, have adopted shortcut connections inspired by ResNet [14] and a simple design rule—“doubling the number of channels along with halving the spatial size of feature maps” which can be traced back to VGGNet [12].
Fig. 1. From left to right, the conceptual structures of CALPA-XuNet2, XuNet2, SRNet, and CALPA-SRNet* are illustrated respectively. The number and corresponding shrinking rate of output channels of every convolutional layer is shown alongside the representing bar. For SRNet and CALPA-SRNet, blue “L1” to “L12” represent twelve composition blocks following the notations in [30].

The shrinking rates of CALPA-XuNet2 and CALPA-SRNet are aiming at deflecting J-UNIWARD stego images with 0.4 bps AC payload and QC 75.

Take SRNet for instance, as shown in Fig. [1] from “L3” up to “L11” are with shortcut connections. From its “L9” up to “L12”, the doubling numbers of output channels (64, 128, 256, and 512) correspond respectively to the halving sizes of feature maps (128 \times 128, 64 \times 64, 32 \times 32, and 16 \times 16).

2) Channel pruning methods: ThiNet and l1-norm based:
Since the training-pruning-finetuning pipeline of typical network pruning methods is completed abandoned in CALPA-NET, only the pruning criteria in ThiNet [38] and the l1-norm based scheme [37] are addressed.

ThiNet [38] is a data-driven channel selection algorithm. For a pre-defined pruning rate \( \gamma \), a subset of m images are randomly selected from training set. For a trained CNN model, Every image in the subset is fed to it, and the input and output tensors of every convolutional layer of the CNN model are obtained in a forward propagation. For a given convolutional layer \( L_{i+1} \), n elements are randomly sampled from \( Z_{i+1} \). Therefore a set with \( \bar{m} = m \cdot n \) samples is obtained. For those \( \bar{m} \) elements, denote the corresponding filters in \( W^{i+1} \) and receptive fields in \( Z^{i+1} \) as \( \bar{W}^{i+1} \) and \( \bar{Z}^{i+1} \), 1 \leq t \leq \bar{m} \), and \( \bar{W}^{i+1}_{j} \) and \( \bar{Z}^{i+1}_{j} \), 1 \leq t \leq \bar{m} \), respectively. What proposed in ThiNet is actual a greedy solution of the following optimization problem:

\[
\arg\min_{T} \sum_{t=1}^{\bar{m}} \left( \sum_{j \in T} \bar{Z}^{i+1}_{j} \cdot \bar{W}^{i+1}_{j} \right)^{2} \\
\text{s.t.} \quad |T| = J \cdot (1 - \gamma), \ T \subset \{1, 2, \cdots, J \} (2)
\]

in which \( |T| \) denotes the number of selected indexes in \( T \), and \( \bar{Z}^{i+1}_{j} \), \( \bar{W}^{i+1}_{j} \) denote the \( j \)-th slide of \( \bar{Z}^{i+1} \) and \( \bar{W}^{i+1} \). Those \( \bar{Z}^{i+1}_{j} \) with indexes fall in \( T \) are categorized as “weak channels” and are pruned. The corresponding before-activation slides \( \{ \bar{Z}^{i+1}_{j}, j \in T \} \) are implicated, and are pruned as well. Consequently, the corresponding filters \( \{ W^{i+1}_{j}, j \in T \} \) which generate \( \{ \bar{Z}^{i+1}_{j}, j \in T \} \) are also pruned. Please note that as three dimensional sub-tensors, the pruning of \( \{ W^{i+1}_{j}, j \in T \} \) results in cutting down of numerous parameters and FLOPs (FLoating-point OPerations) in the corresponding convolutions.

On the contrary, l1-norm based scheme [37] only considers the statistical properties of the filters themselves. For a given convolutional layer \( L_{i} \), its \( K^{i} \) output channels are traversed. The \( l_{1} \) norms of the corresponding filters \( \{ W^{i}_{k}, 1 \leq k \leq K^{i} \} \) are calculated and sorted from lowest to highest. For a pre-defined pruning rate \( \gamma \), the first \( K^{i}(1 - \gamma) \) filters in the sorted list are pruned. Those output channels corresponding to the pruned filters are removed as well.
B. Rationale of our proposed CALPA-NET

In the field of steganalysis, a notable philosophy is established since the reign of “rich model+ensemble classifier” solution [16]: model diversity is crucial to success of steganalytic detectors. The adoption of ResNet-style shortcut connections in latest deep-learning based steganalyzers actually supports the above philosophy since ResNet-style shortcuts resemble ensemble classifier, as pointed out in [40]. However, in this section we provide a theoretical demonstration to show that the broad inverted-pyramid structure adopted in the upper modules of state-of-the-art deep-learning based steganalyzers might contradict the well established philosophy. In the CNN pipeline, given a intermediate convolutional layer $L_l$, let us start from (1). Let $O_{jk} = Z_{jk}^l \ast W_{jk}^l$, then $Z_{jk}^l = \sum_{j=1}^J O_{jk}$, $1 \leq k \leq K$. Assume that $O_{jk}$ is a discrete sample drawn from a continuous latent distribution $O_{jk}$, with PDF (Probability Distribution Function) $o_{jk}$. Accordingly, let $Z_{jk}^l$ be a sample of the latent distribution $Z_{jk}$ with PDF $z_{jk}$. Please note that the PDF of the aggregation is equal to the convolution of the PDF of the aggregated terms:

$$ z_k = o_{1k} \ast o_{2k} \ast \cdots \ast o_{jk} \ast \cdots \ast o_{jk} \quad (3) $$

Denote the Fourier transform of $o_{jk}$ and $z_k$ as $O_{jk}$ and $Z_k$, respectively. Therefore in frequency domain, we can get:

$$ Z_k(\omega) = \prod_{j=1}^J O_{jk}(\omega) \quad (4) $$

According to [16], an ideal steganalytic model should be a union of diverse sub-models. Get back to the intermediate convolutional layer $L_l$. It can be regarded as a features extractor. Every output channel (feature map) $Z_{jk}^l$, $1 \leq k \leq K$ of $L_l$ can be viewed as a generated sub-model and should contains diverse statistical patterns. From the perspective of time-frequency analysis, those ideal diverse statistical patterns correspond to sparse intensity fluctuations in nearly non-overlapping frequency subbands. Conversely, those strong intensities in the most common frequency subbands (usually those low-frequency subbands) usually represent the underlying universal patterns and contribute little to model diversity. Given two frequencies $\omega_1$ and $\omega_2$, assume that for a fixed k, $O_{jk}$ in $\omega_1$ contains sparse intensity fluctuations, while in $\omega_2$ contains strong intensities. Therefore for most $j$, $|O_{jk}(\omega_1)| < |O_{jk}(\omega_2)|$. Obviously, for large enough $J$:

$$ \frac{|Z_k(\omega_1)|}{|Z_k(\omega_2)|} = \frac{\prod_{j=1}^J O_{jk}(\omega_1)}{\prod_{j=1}^J O_{jk}(\omega_2)} = \frac{\prod_{j=1}^J |O_{jk}(\omega_1)|}{\prod_{j=1}^J |O_{jk}(\omega_2)|} \rightarrow 0 \quad (5) $$

From (5) we can see the aggregation in (1) with large enough $J$ suppresses sparse intensity fluctuations as well as heights strong densities in low-frequency subbands of $Z_{jk}^l$, $1 \leq k \leq K$. As a result, the broad convolutional layers in the inverted-pyramid style upper modules of existing deep-learning based steganalyzers actually violate the model diversity oriented philosophy.

C. Detailed algorithm of our proposed CALPA-NET

1) The overall procedure: Our proposed approach is named CALPA-NET, the ChAnneL-Pruning-Assisted deep residual NETwork for image steganalysis. “CALPA” is homophonic with “KALPA”, a Sanskrit word for a never ending loop in which the universe circularly expands and shrinks. The demonstration in Sect. II.B has revealed that the broad inverted-pyramid structure might be a bloated solution for deep-learning based steganalyzers. Therefore our target is to shrink the excessive expanded structure of existing successful deep-learning based steganalyzers to make them more cost-effective, as the word “KALPA” implies.

Recently, an interesting phenomenon that network pruning can be regarded as one sort of network architecture searching approach is reported in the field of computer vision [41]. An effective pruned network trained from scratch can show the performance as good as the one with traditional three-stage training-pruning-finetuning pipeline. This phenomenon inspires us to use effective network pruning criteria to determine the most cost-effective structure for deep-learning based steganalyzers. The diagram of the overall procedure is shown in Fig. 2. Given a deep-learning based steganalytic model $N$, it is trained first using the original training protocol. At the same time, the structure of $N$ is analyzed. A specific pruning scheme is determined for every inclusive convolutional layer in the pipeline according to its position in the pipeline (see Sect. II.C2). Then the well-trained $N$ is traversed from bottom to top and the shrinking rate for every inclusive convolutional layer is determined in a data-driven manner. A brand new model structure is obtained via shrinking the channels of every inclusive layer with corresponding shrinking rate (see Sect. II.C3). The resulting model is referred to as CALPA-$N$, and is reset and trained from scratch. In this paper, two representative deep-learning based steganalytic models are involved: SRNet [30] and XuNet2 [28]. We call the corresponding CALPA-NET as CALPA-SRNet and CALPA-
XuNet2.

2) The proposed hybrid channel pruning criterion: Unfortunately, there is no known channel pruning schemes for shortcut connections yet. Our preferred channel pruning criterion is the one used in ThiNet [38] due to its simplicity and efficiency. However, the ThiNet scheme cannot be used in shortcut connections.

An example from SRNet is given in Fig. 3. Two types of residual blocks are picked out from Fig. 1: Blocks “L3” and “L4” with blue background contain direct shortcut connections, while blocks “L9” and “L10” with pink background contain transformed shortcut connections. Please note that for direct shortcut connections, the element-wise addition of the two relevant convolutional layers should be with the same output tensor size. From Fig. 3 it is clear that the size of $Z^2$, $Z^4$ and $Z^6$ must be the same. Further, all of the relevant convolutional layers linked via shortcut connections should be with the same output tensor size. Given a main-branch output tensor $Z^l$, the input tensor in the corresponding direct shortcut connections as well. Also take SRNet for example, $Z^4$, $Z^6$, $Z^{16}$ and even $Z^4$, $Z^6$, $Z^{10}$ (Not appeared in Fig. 3, please refer to Fig. 2) are linked via direct shortcut connections. $l_1$-norm based scheme is applied to $Z^2$, the lowest one. Then the obtained shrinking rate is assigned to all the linked output tensors.

For those transformed shortcut connections, the two convolutional layers relevant to the element-wise addition in transformed shortcut connections are handled together. Given a main-branch output tensor $Z^l$, its shrinking rate is determined via $l_1$-norm based scheme. The same shrinking rate is assigned to $Z^l$, the output tensor in the corresponding transformed shortcut connection as well. Also take SRNet for example, $l_1$-norm based scheme is used to determine the shrinking rate of $Z^{16}$ and $Z^{18}$, and then the obtained shrinking rate is assigned to $Z^6$ and $Z^8$, respectively.

3) Channel-pruning-assisted architecture search: Please note that all of the existing channel pruning approaches, including ThiNet [38] and $l_1$-norm based scheme [37], aim at screening out the most useful channels given a pre-defined pruning rate. On the contrary, in CALPA-NET, channel pruning approaches is used to assist the determination of the shrinking rate of every involved convolutional layer. A new term “shrinking rate” is introduced in order to highlight the difference between our proposed CALPA-NET and existing channel pruning approaches. Given a convolutional layer $L_l$ with a determined shrinking rate $\zeta_l$, the corresponding pruning rate used in channel pruning approaches is $\gamma_l = 1 - \zeta_l$.

As mentioned in Sect. 2.1, the well-trained model $N$ is traversed from bottom to top, for every inclusive convolutional layer $L_l$ with shrinking rate undetermined, the detailed algorithm is shown in Algorithm 1.

After the bottom-up traversal, every inclusive convolutional layer $L_l$ is assigned a determined shrinking rate $\zeta_l$. The resulting CALPA-N model is obtained via shrinking the volume of $Z^l$, the before-activation output tensor of every inclusive layer $L_l$ to $R^{K_l\cdot w_l\cdot h_l \cdot \hat{c}'_l}$, in which $K_l = K_l \cdot \zeta_l$. The obtained CALPA-N is reset and trained from scratch.
Algorithm 1 Shrinking rate determination algorithm for $L_l$ and all the relevant convolutional layers.

Require: A well-trained model $N$, a standalone validation dataset $D_v$, a pre-defined step $\epsilon$ (set to 5% in our experiments), and a tolerable accuracy loss $\varsigma$ (set to 5% in our experiments).

1: Initialize the pruning rate $\eta_1 = 0$.
2: Validate $N$ in $D_v$. Denote the obtained validation accuracy as $Acc_0$. Let $Acc_p = Acc_0$.
3: loop
4:    Set $\eta_l = \eta_1 + \epsilon$.
5:    if $L_l$ is not in shortcut connections then
6:        use ThiNet to prune $L_l$.
7:    else
8:        use $l_1$-norm based scheme to prune $L_l$ and all the relevant convolutional layers linked with it via shortcut connections.
9:    end if
10:   Let $N_p$ denotes the pruned model. Validate $N_p$ in $D_v$ and denote the validation accuracy as $Acc_{p_l}$.
11:   if $Acc_{p_l} - Acc_{p_{l+1}} > \varsigma$ then
12:       an obvious accuracy decline is observed.
13:       break
14:   else if $Acc_{\gamma_0} - Acc_{\gamma_1} > \varsigma$ then
15:       the detection accuracy has gradually declined to beyond the tolerance threshold.
16:       break
17:   end if
18:   Let $Acc_{p_{l+1}} = Acc_{p_l}$.
19: end loop
20: set $\eta_l = \eta_l - \epsilon$, namely roll back to prior pruning rate.
21: Set the corresponding shrinking rate $\zeta_l = 100\% - \eta_l$. If $L_l$ is in shortcut connections, the shrinking rates of all the relevant convolutional layers linked with it via shortcut connections are set to $\zeta_l$ as well.

### III. Experiments

#### A. Experiment setup

The primary image dataset used in our experiments is the union of BOSSBase v1.01 and BOWS2, each of which contains 10,000 512 × 512 grayscale spatial images. All of the images were resized to 256 × 256 using Matlab® function 

For JPEG images, the corresponding JPEG images were further generated with QFs (Quality Factors) 75 and 95. In every experiments, 10,000 BOWS2 images and 4,000 randomly selected BOSSBase images were used for training. Another 1,000 randomly selected BOSSBase images were for validation. The remaining 5,000 BOSSBase images were retained for testing. Furthermore, following the generation pipeline mentioned in [30], a subset of ImageNet CLSLOC dataset with 250,000 grayscale 256 × 256 JPEG images was also introduced to demonstrate the performance of CALPA-SRN on large-scale dataset.

Four representative steganographic schemes, UERD [8] and J-UNIW ARD [9] for JPEG domain, and HILL [5] and S-UNIW ARD [9] for spatial domain, were our attacking targets in the experiments. The default parameters of the four steganographic schemes were adopted in our experiments. For JPEG steganographic algorithms, the embedding payloads were set to 0.2 and 0.4 bpp (bits per non-zero cover AC DCT coefficient). For spatial domain steganographic algorithms, the embedding payloads were set to 0.2 and 0.4 bpp (bits per pixel).

XuNet2 [28] and SRNet [30] were selected as the two initial architectures of our proposed CALPA-NET. Our implementation of CALPA-NET and its corresponding initial architectures are based on TensorFlow [45]. Unless otherwise specified, the two initial architectures were trained with the hyper-parameters mentioned in the corresponding original papers. The batch size in the training procedure was set to 32 (namely

Since the traditional training-pruning-finetuning pipeline is not followed in this step, $N_p$ is directly validated without finetuning procedure.

For a fair comparison, we tried our best to adopt almost the same experiment setup as which SRNet [30] was evaluated on.
16 cover-stego pairs). Data augmentation with random mirroring and rotation of images by 90 degrees was applied to the images in every batch. The maximum number of iterations was set to $30 \times 10^4$ for XuNet2, and $50 \times 10^4$ for SRNet, as in the corresponding original papers. CALPA-XuNet2 and CALPA-SRNet adopted the same maximum number of iterations as their corresponding initial architectures.

In each experiment, the model was validated and saved every one epoch (in primary dataset, $14,000/16 = 875$ iterations). The one with the best validation accuracy was evaluated on the corresponding testing set. All of the experiments were conducted on a GPU cluster with sixteen NVIDIA® Tesla® P100 GPU cards. Bounded by computational resources, every experiment was repeated three times, and the mean of the results on testing set were reported. The source codes and auxiliary materials are available for download from GitHub.

B. Compactness of CALPA-NET

In this section, we take CALPA-SRNet as example to analyze its compactness and effectiveness.

1) Compactness of CALPA-NET with different $\xi$: In Tab. I, we compare the effect of different tolerable accuracy lost $\xi$ on the model parameters, FLOPs, and detection accuracies of CALPA-SRNet. It has been highlighted in Sect. II-C that every obtained CALPA-SRNet model is reset and trained from scratch. Here our adopted “trained from scratch” scenario is also compared with the traditional “training-pruning-finetuning” pipeline. In “training-pruning-finetuning” pipeline, the weights in non-pruned kernels of the model were kept. Then the pruned model was re-trained/finetuned for another $30 \times 10^4$ iterations. From Tab. I we can see that with the rise of tolerable accuracy lost $\xi$, the parameters and FLOPs required by CALPA-SRNet significantly decrease. However, with a mild $\xi$ (2% ~ 20%), the detection accuracy of the shrunk model does not decline, but instead improves compared to the original SRNet. Another notable thing is that the detection performances achieved by the shrunk models trained from scratch are almost equivalent to those with “training-pruning-finetuning” pipeline. Such a phenomenon indicates that in CALPA-Net, the efficient network architecture obtained via channel-pruning-assisted search is critical. On the other hands, the preserved kernel weights in traditional network pruning approach are inessential.

By contrasting the benefits of the reduction of parameters and FLOPs with the losses of detection performance, $\xi$ is set to 5% in our final proposed framework. Please note that when $\xi = 5\%$, CALPA-SRNet outperforms original SRNet by a clear margin with mere 1.34% parameters (6.37$\times$10$^4$ vs. 476.87$\times$10$^4$) and 31.3% FLOPs (2.00$\times$10$^9$ vs. 6.40$\times$10$^9$). The corresponding structure of CALPA-SRNet can be referred back to Fig. 1. From Fig. 4 we can see that the upper module of CALPA-SRNet no longer presents the broad inverted-pyramid structure. The top-most convolutional layer is only with 5% kernels compared with original SRNet (25 vs. 512). Either ThiNet criterion or $l_1$-norm based criterion is effective in the shrinking procedure. However from “L3”—“L9” blocks we can see ThiNet criterion usually obtains more aggressive shrinking rate compared with $l_1$-norm based criterion. As a result, CALPA-SRNet presents a slender bottleneck-like structure in contrast to traditional deep-learning based steganalyzers.

2) Tolerance to pruning rate in our approach: In Fig. 2, we show how the validation accuracy changes with successive growing pruning rates when determining the shrinking rate of every inclusive convolutional layer of CALPA-SRNet. From Fig. 2, no particular patterns can be observed. However, no matter for ThiNet scheme (as shown in Fig. 2(a)), or for $l_1$-norm based scheme (as shown in Fig. 2(b)), convolutional layers in top blocks always present high tolerance to severe pruning rates.

3) Effectiveness of CALPA-NET compared to traditional “training-pruning-finetuning” pipeline: In Fig. 3 we show how the training accuracy, validation accuracy, and testing accuracy change with successive training iterations for the original SRNet, the corresponding CALPA-SRNet trained from scratch, and the pruned SRNet with “training-pruning-finetuning” pipeline. When trained to detect J-UNIWARD stego images with 0.4 bppzAC payload and QF 75, SRNet achieved the best validation accuracy in $4.95 \times 10^4$ iterations. The corresponding model was selected out to apply channel-pruning-assisted search to get CALPA-SRNet. CALPA-SRNet was reset and trained from scratch. As shown in Fig. 3, the validation accuracies and testing accuracies of CALPA-SRNet trained from scratch had become stable even before $4.95 \times 10^4$ iterations. For comparison, training procedure of the pruned SRNet was resumed. It was further finetuned till $80 \times 10^4$ iterations. But it can be observed that the validation accuracies and testing accuracies of pruned SRNet during the finetuning procedure was unstable. Furthermore, its best performance never surpassed CALPA-SRNet trained from scratch, which demonstrates the effectiveness of CALPA-SRNet model structure.

C. Detection performance of CALPA-NET

In Fig. 6 we compare the detection performance of CALPA-SRNet and the corresponding original SRNet for JPEG-domain steganographic algorithms J-UNIWARD and UERD, and spatial-domain steganographic algorithm HILL. From Fig. 6(a) and Fig. 6(b) we can see that CALPA-SRNet obtains comparable detection performance when aiming at detecting stego images with QF 75. The detection performance of CALPA-SRNet is slightly worse than original SRNet when stego images are with QF 95, as shown in Fig. 6(c). However, Fig. 6(d) indicates that when used to detect 0.4 bpp HILL spatial-domain stego images, CALPA-SRNet can obtain significant performance improvement. Please note that all the above high performance of CALPA-SRNet are achieved with mere 1%~3% parameters compared to original SRNet.

In Fig. 7 we show the corresponding shrinking rates of CALPA-SRNet used in Fig. 6. From Fig. 7 we can observe that though no obvious patterns can be found, the shrinking rate of every convolutional layer adapts to the target steganographic scheme, the embedding payload, the

https://github.com/tansq/CALPA-NET
Fig. 4. Validation accuracies vs. growing pruning rates when determining the shrinking rate of every inclusive convolutional layer of CALPA-SRNet. The trained models are aiming at detecting J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75. \( \varsigma = 5 \). (a) is for ThiNet scheme in blocks “L3”—“L12”. (b) is for \( l_1 \)-norm based scheme in transformed shortcut connections of blocks “L8”—“L12”.

Fig. 5. Comparison of the training accuracies, validation accuracies and testing accuracies vs. training iterations for the original SRNet, CALPA-SRNet trained from scratch, and CALPA-SRNet with “training-pruning-finetuning” pipeline. The trained models are aiming at detecting J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75. \( \varsigma = 5 \).
embedding domain and even the actual quality factor used in JPEG target images. In general, the shrinking rate rises as the corresponding convolutional layer gets higher and higher. Significantly high shrinking rates can always be observed in top convolutional layers.

In Tab. II we compare the testing accuracy of CALPA-XuNet2 and original XuNet2. We adopt an alternative performance measure used in [30]: $P_{FA}(0.5)$ and $P_{FA}(0.3)$, the false-alarm rates for stego-image detection probability 0.5 and 0.3. Two targets, J-UNIWARD and UERD, are involved. The results for two representative payloads, 0.2 bpnzAC and 0.4 bpnzAC are given. From Tab. II we can see no clear performance margin between CALPA-XuNet2 and the corresponding original XuNet2. In some cases CALPA-XuNet2 even performs better than original XuNet2. The results demonstrate the amazing detection performance of CALPA-XuNet2 since it is quite a lightweight model compared to original XuNet2. The structure of one of the CALPA-XuNet2 models aiming at detecting J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75 can be found in Fig. 1, in which we can see CALPA-XuNet2 also presents a slender bottleneck-like structure.

D. Adaptivity, Transferability, and Scalability of CALPA-NET

As demonstrated in prior experiments, the structure of CALPA-NET is adaptive to the actual targets it aims at due to the fact that CALPA-NET is data-driven. Firstly in Fig. 8 we evaluate the effectiveness of the adaptive data-driven scheme used in CALPA-NET taking CALPA-SRNet as example. Here two less adaptive alternative schemes are introduced:

- AGGR scheme: The global shrinking rate of all the inclusive convolutional layers is aggressively set to the minimal one determined in the CALPA-NET bottom-up traversal.
- AVG scheme: The global shrinking rate of all the inclusive convolutional layers is set to the average of all the determined shrinking rates.

Refer back to the CALPA-SRNet illustrated in Fig. 1, the global shrinking rate can be obtained as 5% and 52% for AGGR scheme and AVG scheme respectively. From Fig. 8 we can see that both for JPEG-domain J-UNIWARD stego images and spatial-domain HILL stego images, CALPA-SRNet outperforms other two alternative schemes. Though AGGR scheme has the least parameters and FLOPs, its detection performance degrades remarkably. AVG scheme possesses similar scale of parameters and FLOPs as CALPA-SRNet,
but no longer has bottleneck-like structure. It performs better than AGGR scheme but is inferior to CALPA-SRNet when aiming at J-UNIWARD stego images. It is interesting that either CALAP-SRNet or less adaptive AGGR/AVG schemes all perform better than the much more giant SRNet, especially aiming at special-domain HILL steganographic algorithm.

Next, we inspect the transferability of the obtained structure of CALPA-NET. Take CALPA-SRNet structure, which aims at J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75, as example. Fig. 9 shows its detection performance for original J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75, as example. Fig. 9a shows its detection performance for original J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75, as example.

Fig. 8. Comparison of testing accuracy of CALPA-SRNet with two less adaptive alternative schemes, the AGGR scheme and the AVG scheme. (a) For J-UNIWARD stego images with 0.4 bpnzAC payload and QF 75. (b) For HILL stego images with 0.4 bpp payload.

Fig. 7. The corresponding shrinking rates of CALPA-SRNets used in Fig. 6. (a) is for ThiNet scheme in blocks “L3”—“L12”. (b) is for $l_1$-norm based scheme in transformed shortcut connections of blocks “L8”—“L12”.

### TABLE II

Comparison of testing accuracy of CALPA-XuNet2 and the corresponding original XuNet2. Results for J-UNIWARD/UERD stego images with 0.2/0.4 bpnzAC payload and QF 75/95 are included. Aiming at each target, the better result of two models, CALPA-XuNet2 or XuNet2, is underlined.

| Quality Factors | Targets    | CALPA-XuNet2 | XuNet2 |
|-----------------|------------|--------------|--------|
|                 |            | $F_{PA}(0.5)$ | $F_{PA}(0.3)$ | $F_{PA}(0.5)$ | $F_{PA}(0.3)$ |
| QF75            | J-UNIWARD  | 0.0030       | 0.0012  | 0.0013 | 0  |
|                 | 0.2 bpnzAC |              |         |        |        |
|                 | 0.4 bpnzAC |              |         |        |        |
|                 | UERD       | 0.0002       | 0.0001  | 0.0004 | 0.0002 |
|                 | 0.2 bpnzAC |              |         |        |        |
|                 | 0.4 bpnzAC |              |         |        |        |
| QF95            | J-UNIWARD  | 0.1764       | 0.0328  | 0.2269 | 0.0706 |
|                 | 0.2 bpnzAC |              |         |        |        |
|                 | 0.4 bpnzAC |              |         |        |        |
|                 | UERD       | 0.0006       | 0        | 0.0006 | 0  |
|                 | 0.2 bpnzAC |              |         |        |        |
|                 | 0.4 bpnzAC |              |         |        |        |

Fig. 8. Comparison of testing accuracy of CALPA-XuNet2 and the corresponding original XuNet2. Results for J-UNIWARD/UERD stego images with 0.2/0.4 bpnzAC payload and QF 75/95 are included. Aiming at each target, the better result of two models, CALPA-XuNet2 or XuNet2, is underlined.
bnpzAC payload, J-UNIWARD stego images with alternative QF 95, and HILL stego images (spatial domain) with 0.4 bpp payload, respectively. It is clear that the CALPA-SRNet structure obtains good detection performance in the above three mismatched scenarios. Furthermore, compared to other two likely less mismatched scenarios, the specific JPEG-oriented CALPA-SRNet structure remains effective for spatial-domain HILL stego images, implying that HILL and J-UNIWARD share some intrinsic characteristics although they are in completely different domain.

We then inspect the transferability of the trained CALPA-NET model. In Table III, we compare the detection performance of CALPA-SRNet and original SRNet trained on one target and tested on another target. The experiments were conducted on two major steganographic embedding domains: spatial domain and JPEG domain. In spatial domain, two representative steganographic algorithms, HILL and S-UNIWARD, were involved. In JPEG domain, J-UNIWARD and WOW were involved. In order to make a fair comparison, we used the following performance measurement (as in Tab. II of [30]):

\[ P_T = \min_{P_{FA}} \left( \frac{1}{2} (P_{FA} + P_{MD}) \right) \]

where \( P_{FA} \) and \( P_{MD} \) are the false alarm rate and miss detection rate. From Table III, it can be clearly observed that when the payload of the targets are the same, CALPA-SRNet achieves similar, and even better transferability compared with original SRNet.

At last the scalability of CALPA-NET is inspected. Take CALPA-SRNet and original SRNet as example, we demonstrate the performance of CALPA-NET on the subset of ImageNet CLS-LOC dataset. From Fig. 10, we can see even in a tenfold larger dataset, CALPA-SRNet still shows similar validation and testing performance compared to original SRNet. Please note that the gap between training accuracies and validation/testing accuracies for original SRNet is much bigger than that for CALPA-SRNet, indicating that CALPA-SRNet may be less vulnerable to overfitting the training set.

IV. CONCLUDING REMARKS

It may become difficult to make progress with the approach relying only on structure expansion to improve detection performance of deep-learning based steganalyzers. In this paper we propose CALPA-NET, which is aiming at adaptively search efficient network structure on top of existing over-parameterized and vast deep-learning based steganalyzers. The major contributions of this work are as follows:

- We have developed a theoretical rationale to explain why the broad inverted-pyramid structure of existing deep-learning based steganalyzers contradicts the well-established model diversity oriented philosophy.
- We have proposed a channel-pruning-assisted deep residual network architecture search approach which uses a hybrid criterion combined with ThiNet and \( l_1 \)-norm based network pruning scheme. In the proposed network architecture, the traditional training-pruning-finetuning network pruning pipeline is completed abandoned.
- The extensive experiments conducted on de-facto benchmarking image datasets show that CALPA-NET can achieve comparative and even better detection performance with just a few percent of the model size and a small proportion of computational cost.

Our future work will focus on two aspects: (1) development of a fast adaptive structural adjustment algorithm to make deep-learning based steganalyzers self adapt to targets without the introduction of redundant parameters/components; (2) further exploration of the feasibility of completely automatic deep-learning based steganalytic framework generation.

REFERENCES

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.
[2] R. Böhme, Advanced Statistical Steganalysis, 1st ed. Springer Publishing Company, Incorporated, 2010.
[3] J. Fridrich and T. Filler, “Practical methods for minimizing embedding impact in steganography,” in Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, 2007, pp. 650502–1–650502–15.
[4] T. Filler and J. Fridrich, “Gibbs construction in steganography,” IEEE Transactions on Information Forensics and Security, vol. 5, no. 4, pp. 705–720, 2010.
[5] B. Li, M. Wang, J. Huang, and X. Li, “A new cost function for spatial image steganography,” in Proc. IEEE 2014 International Conference on Image Processing, (ICIP’2014), 2014, pp. 4206–4210.
[6] V. Sedighi, R. Cogranne, and J. Fridrich, “Content-adaptive steganography by minimizing statistical detectability,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 221–234, 2016.
[7] L. Guo, J. Ni, and Y. Q. Shi, “Uniform embedding for efficient JPEG steganography,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 5, pp. 814–825, 2014.
[8] L. Guo, J. Ni, W. Su, C. Tang, and Y. Q. Shi, “Using statistical image model for JPEG steganography: Uniform embedding revisited,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2669–2680, 2015.
[9] V. Holub, J. Fridrich, and T. Denemark, “Universal distortion function for steganography in an arbitrary domain,” EURASIP Journal on Information Security, vol. 2014, no. 1, pp. 1–13, 2014.
[10] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, 2015.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. 2012 in neural information processing systems, (NIPS’2012), 2012, pp. 1097–1105.
[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[13] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2015), 2015, pp. 1–9.
TABLE III
Detection performance of CALPA-SRNet and original SRNet trained on one target and then tested on another target. The payload of the targets were fixed to 0.4 bpnzAC/bpp. The detection performance was measured with total error probability $P_E$.

|                       | JPEG domain | Spatial domain |
|-----------------------|-------------|---------------|
|                       | CALPA-SRNet |               |
|                       |             |               |
|                       | Trained on  | Tested on     |
| J-UNIRARD             | 0.0761      | 0.1063        |
| UERD                  | 0.242       | 0.2441        |
|                       | S-UNIRARD   |               |
|                       | 0.2604      | 0.2687        |
|                       | HILL        | 0.156         |
|                       |             |               |
|                       | SRNet       |               |
|                       |             |               |
|                       | Trained on  | Tested on     |
| J-UNIRARD             | 0.0795      | 0.1275        |
| UERD                  | 0.2604      | 0.2699        |
|                       | S-UNIRARD   |               |
|                       | 0.2604      | 0.2699        |
|                       | HILL        | 0.1697        |

Fig. 10. Comparison of the training accuracies, validation accuracies and testing accuracies vs. training iterations for CALPA-SRNet and the corresponding original SRNet on a subset of CLS-LOC dataset. The trained models are aiming at detecting J-UNIRARD/UERD stego images with 0.4 bpnzAC payload and QF 75. The blue dashed line in every sub-figure marks the highest testing accuracy achieved by the trained model. (a) CALPA-SRNet, aiming at J-UNIRARD; (b) Original SRNet, aiming at J-UNIRARD; (c) CALPA-SRNet, aiming at UERD; (d) Original SRNet, aiming at UERD.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’ 2016), 2016, pp. 770–778.
[15] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. International Conference on Machine Learning (ICML’ 2015), 2015, pp. 448–456.
[16] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868–882, 2012.
[17] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, “Selection-channel-aware rich model for steganalysis of digital images,” in Proc. 6th IEEE International Workshop on Information Forensic and Security (WIFS’2014), 2014, pp. 48–53.
[18] W. Tang, H. Li, W. Luo, and J. Huang, “Adaptive steganalysis based on embedding probabilities of pixels,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 734–745, 2016.
[19] S. Tan, H. Zhang, B. Li, and J. Huang, “Pixel-decimation-assisted steganalysis of synchronize-embedding-changes steganography,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 7, pp. 1658–1670, 2017.

[20] B. Li, Z. Li, S. Zhou, S. Tan, and X. Zhang, “New steganalytic features for spatial image steganography based on derivative filters and threshold LBP operator,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1242–1257, 2018.

[21] J. Kodovský and J. Fridrich, “Ensemble classifiers for steganalysis of digital media,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 432–442, 2012.

[22] S. Tan and B. Li, “Stacked convolutional auto-encoders for steganalysis of digital images,” in Proc. Asia-Paciﬁc Signal and Information Processing Association Annual Summit and Conference (APSIPA 2014), 2014, pp. 1–4.

[23] Y. Qian, J. Dong, W. Wang, and T. Tan, “Deep learning for steganalysis via convolutional neural networks,” in Proc. IS&T/SPIE Electronic Imaging 2015 (Media Watermarking, Security, and Forensics), 2015, pp. 94.090J–1–94.090J–10.

[24] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont, “Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch,” in Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2016), 2016, pp. 1–11.

[25] G. Xu, H. Z. Wu, and Y. Q. Shi, “Structural design of convolutional neural networks for steganalysis,” IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708–712, 2016.

[26] J. Ye, J. Ni, and Y. Yi, “Deep learning hierarchical representations for image steganalysis,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2545–2557, 2017.

[27] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, “JPEG-phase-aware convolutional neural network for steganalysis of JPEG images,” in Proc. 5th ACM Information Hiding and Multimedia Security Workshop (HI&MMSec'2017), 2017, pp. 75–84.

[28] G. Xu, “Deep convolutional neural network to detect J-UNITWARD,” in Proc. 5th ACM Information Hiding and Multimedia Security Workshop (HI&MMSec’2017), 2017, pp. 67–73.

[29] J. Zeng, S. Tan, B. Li, and J. Huang, “Large-scale JPEG steganalysis using hybrid deep-learning framework,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1242–1257, 2018.

[30] M. Boroumand, M. Chen, and J. Fridrich, “Deep residual network for steganalysis of digital images,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1181–1193, 2019.

[31] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, “WISERNet: Wider separate-then-reunion network for steganalysis of color images,” IEEE Transactions on Information Forensics and Security, 2019, doi: 10.1109/TIFS.2019.2904413.

[32] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in Proc. 2014 Advances in neural information processing systems, (NIPS’2014), 2014, pp. 2654–2662.

[33] A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.0486l, 2017.

[34] Y. LeCun, J. S. Denker, and S. A. Solla, “Optimal brain damage,” in Proc. 1990 Advances in neural information processing systems, (NIPS’1990), 1990, pp. 598–605.

[35] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in Proc. 2015 Advances in neural information processing systems, (NIPS’2015), 2015, pp. 1135–1143.

[36] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, “Network trimming: A data-driven neuron pruning approach towards efﬁcient deep architectures,” arXiv preprint arXiv:1607.03250, 2016.

[37] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, “Pruning ﬁlters for efﬁcient ConvNets,” in Proc. 2017 International Conference on Learning Representations, (ICLR’2017), 2017.

[38] J.-H. Luo, J. Wu, and W. Lin, “ThiNet: A ﬁlter level pruning method for deep neural network compression,” in Proc. 2017 IEEE international conference on computer vision, (ICCV’2017), 2017, pp. 5058–5066.

[39] Z. Huang and N. Wang, “Data-driven sparse structure selection for deep neural networks,” in Proc. 2018 European Conference on Computer Vision (ECCV’2018), 2018, pp. 304–320.

[40] A. Veit, M. Wilber, and S. Belongie, “Residual networks behave like ensembles of relatively shallow networks,” in Proc. 2016 Advances in neural information processing systems, (NIPS’2016), 2016, pp. 550–558.