



# Deep CNN Model Compression via Filter Pruning for Efficient ConvNets

**Project Supervisor**  
Dr. Viswanath Pulabaigari

## Group Members

- Dupally Nikhil (S20170010045)
- Farazuddin Mohammad (S20170010097)
- V Madhukar Reddy (S20170010183)



# Outline

- Introduction to the **Problem Statement**
- **Related** works
- About our **BTP** Work
- **Results** on experiments conducted
- **Future** Plan of Action



## Problem Statement

- **Bigger** and **Deeper** CNN architectures come at a cost of High **computational power** for training and High **Storage capacity**.
- As a result of which the **deployment** of these models on some low grade devices like microcontrollers and embedded devices becomes extremely **difficult**.
- To address this **issue** we attempt at reducing the size of CNN using an approach called **Filter** pruning aka **Channel** pruning.



# Motivation

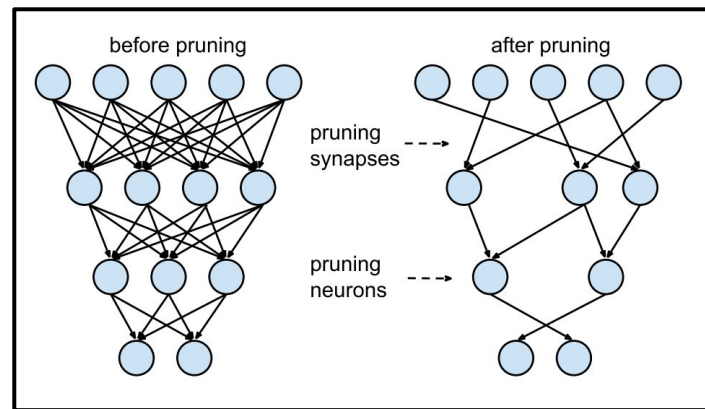
- It was **empirically** found that in a NN there are **redundant** and **unimportant** Weights present which if **removed** would not affect the **performance** of the model.
- This was observed when **some** of the filters were **removed** from the CNN model at **random** i.e there was **no criteria** for discarding the filters and there was almost **no drop** in the **performance** [1].

[1]

Mittal, Deepak, et al. "Recovering from random pruning: On the plasticity of deep convolutional neural networks." *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.

# Pruning

- It is a model **optimization** technique that involves eliminating unnecessary values in the weight tensor.
- It aids in the development of smaller and more efficient **neural networks**.



## Image Source [2]

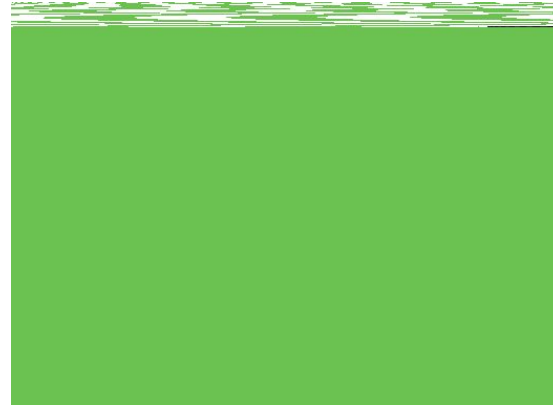
Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems*. 2015.



# Floating Point Operations

- The number **Floating Point Operations** of a model is a measure of the **computational power** that it will demand.
- When there are **constraints** on **computational resources** we use the number of Floating Point Operations (**FLOPs**) as a metric to compare which model is more **efficient**.

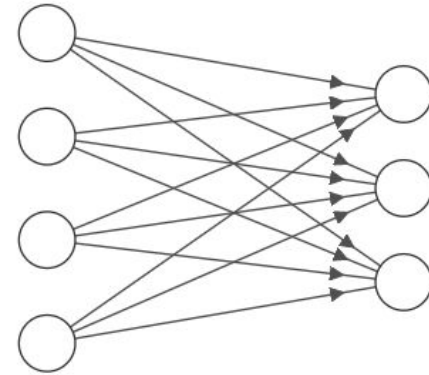
## Calculating Flops <sup>(1)</sup>



$$FLOP_{conv}(L_i) = F * F * C_{in} * H_{out} * W_{out} * C_{out}$$

## Calculating Flops <sup>(2)</sup>

$$FLOP_{fcC}(L_i) = C_{in} * C_{out}$$



Input Layer  $\in \mathbb{R}^4$

Output Layer  $\in \mathbb{R}^3$





## Calculating Flops <sup>(3)</sup>

$$FLOP_{total} = \sum_{i=1}^K FLOP_{conv}(L_i) + \sum_{j=1}^N FLOP_{fc}(L_j)$$

- **VGG-16** has  $3.137 \cdot 10^8$ ,
- **ResNet-56** has  $1.26 \cdot 10^8$ ,
- **LeNet-5** has  $4.40 \cdot 10^6$ .

# About Filter Pruning

- Filter pruning is one of the **model compression** techniques.
- We discard the filter using a **particular criteria** and the corresponding **feature map** gets **dropped** with it.

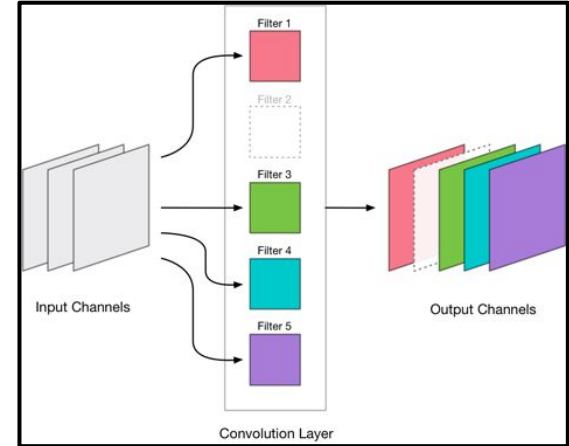
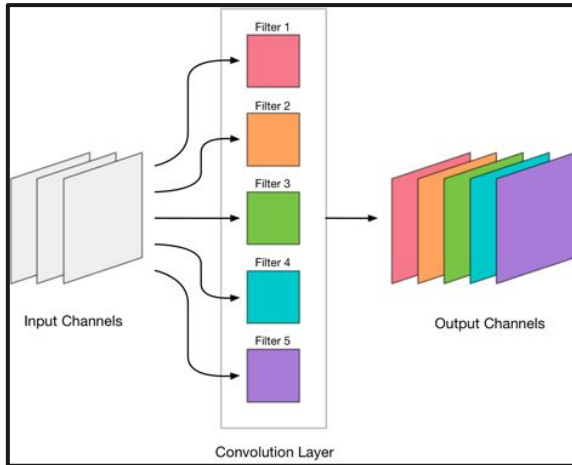


Image Source

<http://machinethink.net/blog/compressing-deep-neural-nets/>



## BTP Work

- To **Extend** the recent works on Filter pruning from two relevant papers.

Paper I - [1]. Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).

Paper II - [2]. Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

- To Leverage the model's training **History** to identify the redundant filters in CNN. A **Novel** approach suggested by our mentor.



# Paper 1

## Ranking Filters based on L1 Norm

- Use L1-norm to select **unimportant** filters and **discard** them.
- **Fine-tuning** process is the same as the conventional training procedure.
- It is a stage wise process,
  - Select some parameter “**m**”, prune the last ‘m’ ranking filters from each layer in one stage.
  - Fine tune the model until you reach the previous accuracy.
  - Repeat until the desired compression is reached.

$$L_1(f_k^i) = \|f_k^i\|_1 = \sum_{p=1}^9 |w_{k,p}^i|$$

## Related Works (continued)

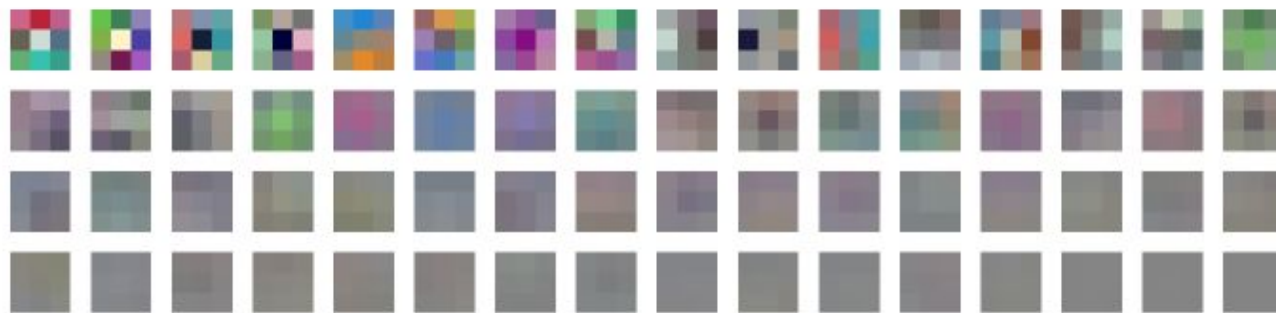


Figure 5: Visualization of filters in the first convolutional layer of VGG-16 trained on CIFAR-10. Filters are ranked by  $\ell_1$ -norm.

Source: Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).

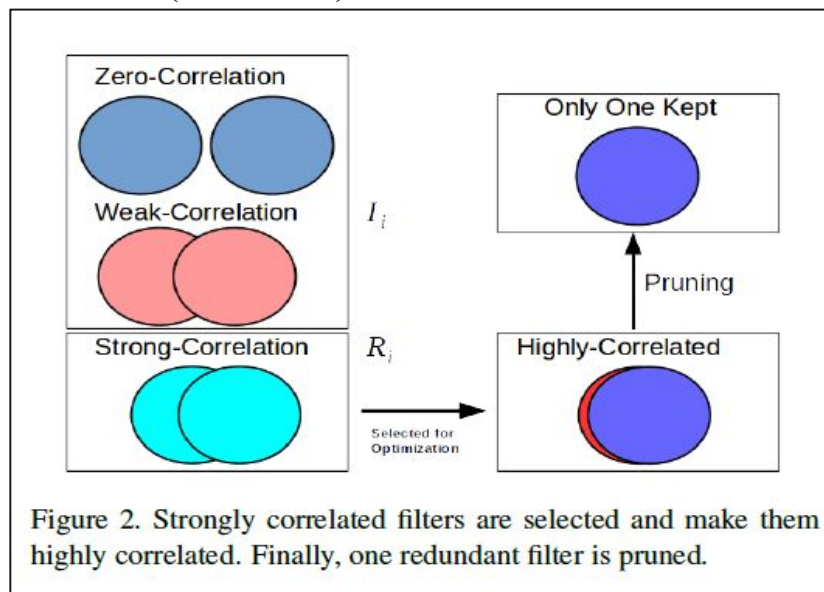


## Related Works (continued) paper 2

### Approach

- For a given Conv Layer we generate all pairs of filters i.e  $\text{NC}_2$
- Iteratively identify pairs of filters with the largest **pairwise correlations**.
- The model is **re-optimized** to make the filters in these pairs **maximally correlated**.
- After **discarding** the filters in each round, we further **fine tune** the model to recover from the potential small loss incurred by the compression.

## Related Works (continued)





## Objectives(Continued)

- We will Test our approach for the following CNN's

S.No	CNN Model	Data Set
1.	LeNet-5 [1]	MNIST
2.	VGG-16 [2]	CIFAR-10
3.	ResNet-56 [3]	CIFAR-10

[1]Yann LeCun, L'eon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11

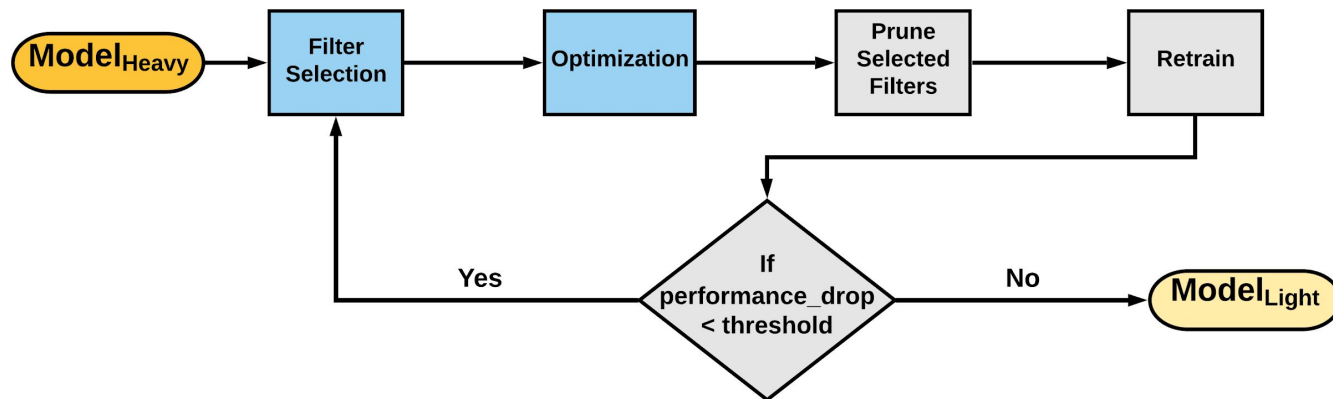
[2]Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition

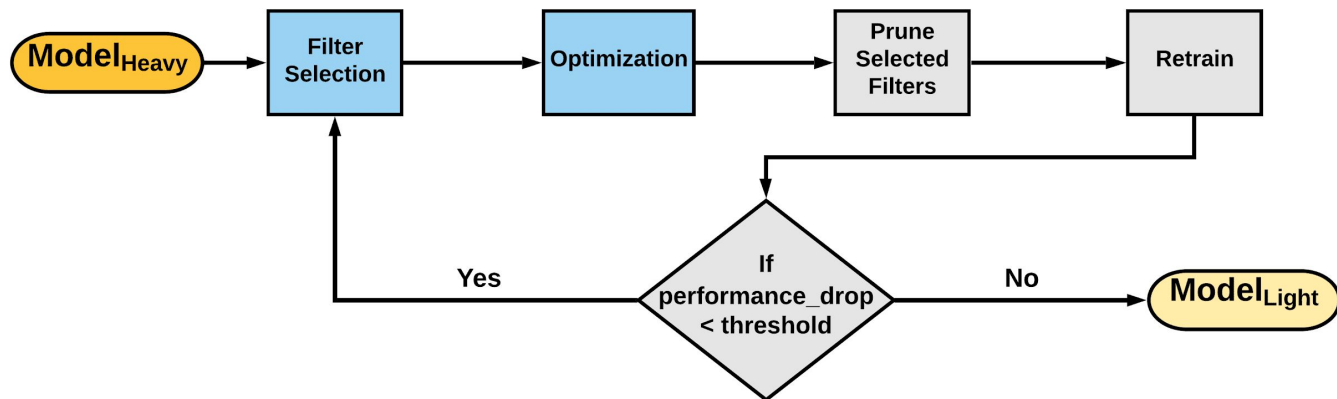
[3]Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.



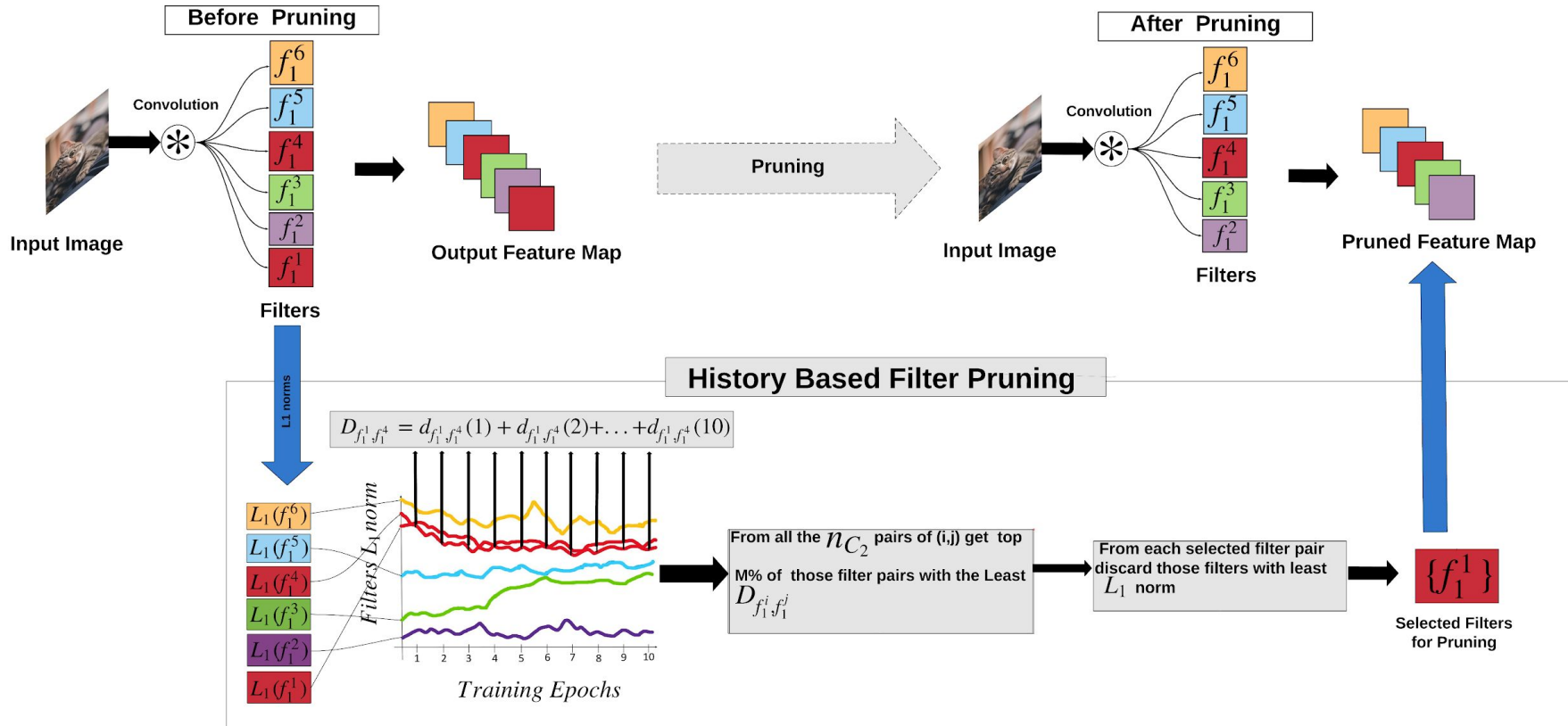
# History Based Filter Pruning

- We identify the **redundant filters** by observing the similar patterns in the weights (parameters) of filters during the network training, which we refer as the **model's training history**.





```
1 while validation_accuracy - max_val_acc >= -0.01:
2     print("ITERATION {} ".format(count+1))
3     all_models.append(model)
4     if max_val_acc < validation_accuracy:
5         max_val_acc = validation_accuracy
6
7     optimize(model,weight_list_per_epoch,10,30,False)
8     model = my_delete_filters(model,weight_list_per_epoch,30,False)
9     model,history,weight_list_per_epoch = train(model,20,False)
10
```





## Results of LeNet-5 on MNIST

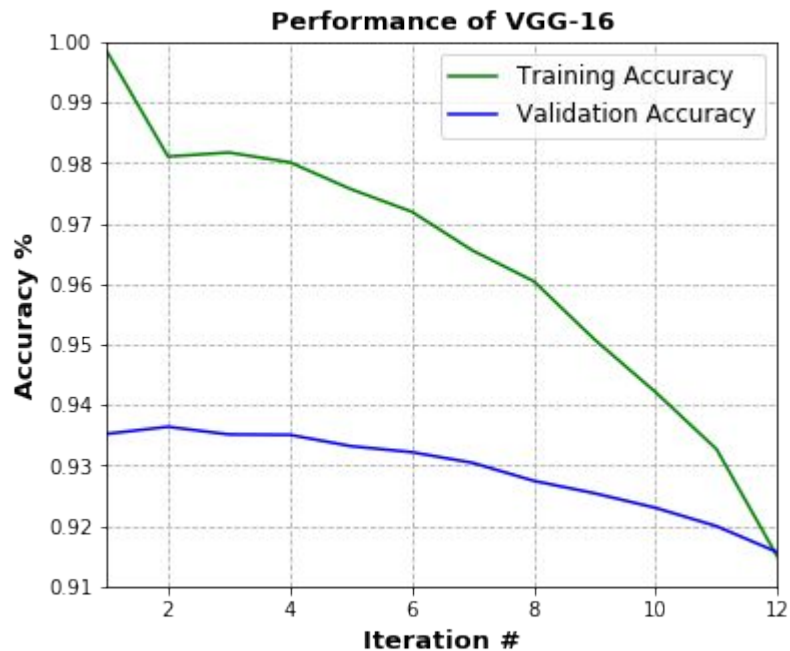
Method	R1, R2	Error%	FLOPs ( $10^5$ )	Pruned%
Base Line	20,50	0.83 +/-	44	0
SSL [1]	3,12	1.00	2.89	93.42
ABFP [2]	3,5	0.83	1.58	96.41
CFP[3]	2,3	1.77 +/- 0.08	0.89	97.98
HBFP (Ours)	2,3	1.40 +/- 0.10	0.89	97.98

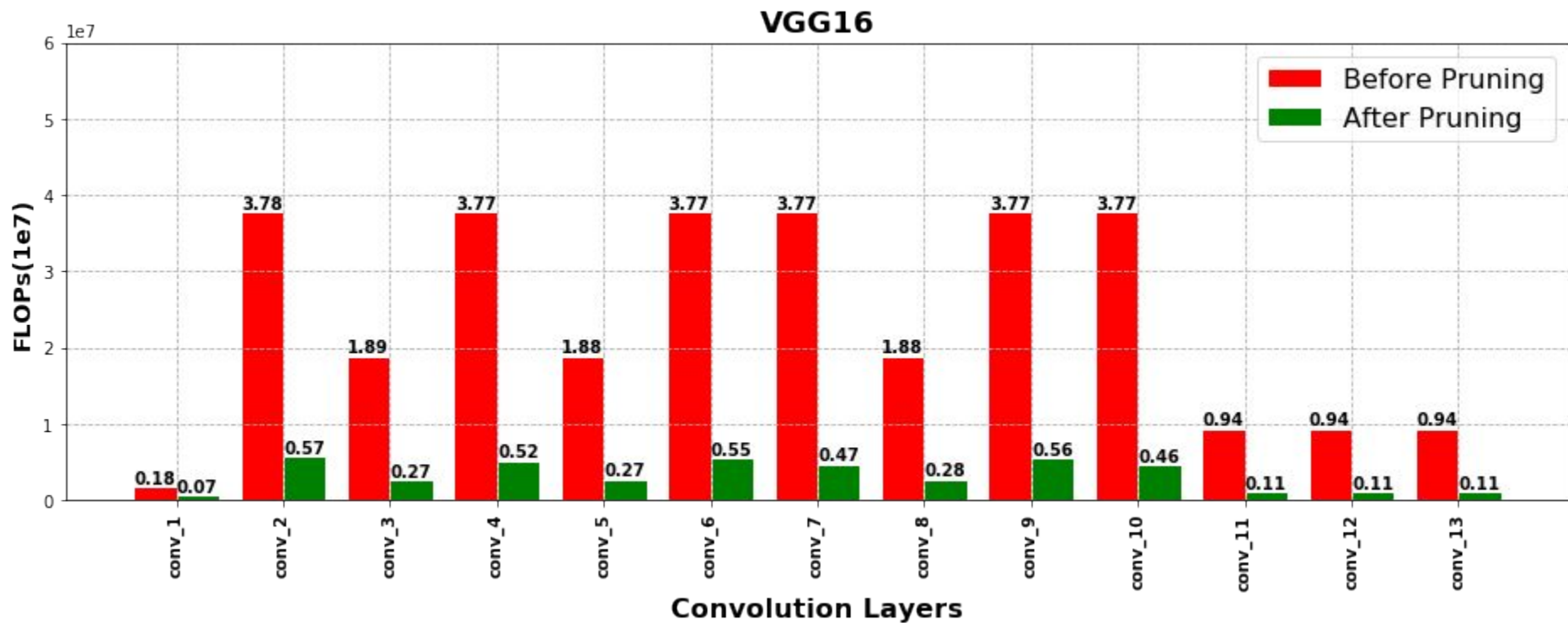
[1]Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems, pages 2074–2082, 2016.

[2]Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[3] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay Namboodiri. Leveraging filter correlations for deep model compression. In The IEEE Winter Conference on Applications of Computer Vision, pages 835–844, 2020.

# Results of VGG-16 on Cifar-10







## Results of VGG-16 on Cifar-10<sub>(continued)</sub>

Method	Base Error %	FLOPs ( $10^8$ )	Retrain Error %	Pruned %
L1 Norm [1]	6.75	2.06	6.60	34.20
ABFP [2]	7.08	0.58	7.13	81.39
CFP [3]	6.51	0.57	7.02 +/- 0.16	83.49
HBFP (Ours)	6.51	0.43	8.23 +/- 0.09	86.21

[1] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. arXivpreprint arXiv:1608.08710, 2016

[2] Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

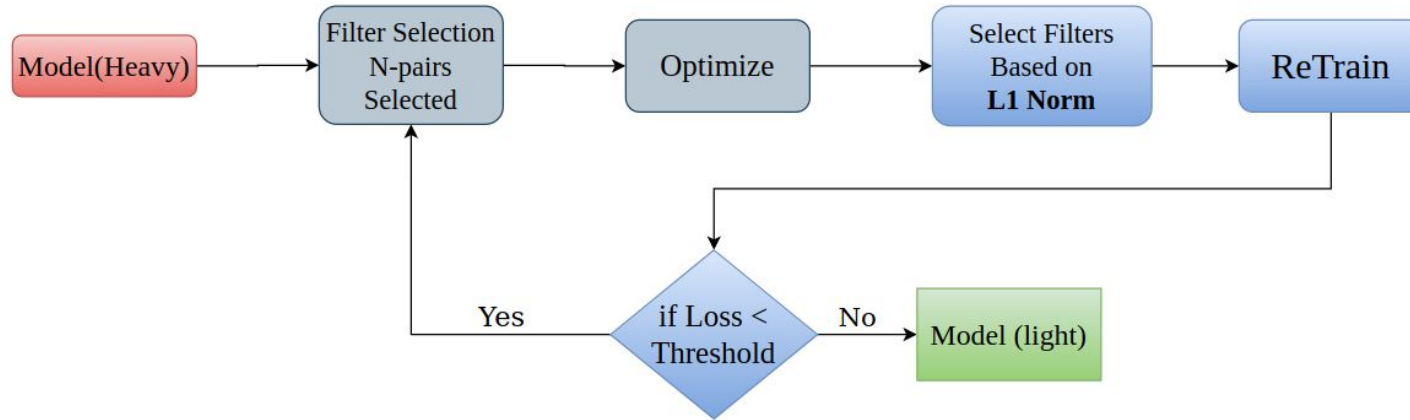
[3] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay Namboodiri. Leveraging filter correlations for deep model compression. In The IEEE Winter Conference on Applications of Computer Vision, pages 835–844, 2020.



## Extension to CFP

- Combining the approach of pruning filters based on **Importance** and **Redundancy**.
- **Redundancy** - Selecting the Filter pairs which are highly correlated.
- **Importance** - Selecting the filter based on L1 Norm between highly correlated filter pairs.





```
while validation_accuracy - max_val_acc >= -0.02 :  
    print("ITERATION {}".format(count+1))  
    all_models.append(model)  
    if max_val_acc < validation_accuracy:  
        max_val_acc = validation_accuracy  
    optimized_model = optimize(model)  
    pruned_model = delete_filters(optimized_model)  
    model , history = fine_tune(pruned_model ,35 )
```



## Results of LeNet-5 on MNIST

Method	R1, R2	Error%	FLOPs ( $10^5$ )	Pruned%
Base Line	20,30	0.83 +/-	44	0
SSL [1]	3,12	1.00	2.89	93.42
ABFP [2]	3,5	0.83	1.58	96.41
CFP [3]	2,3	1.77 +/- 0.08	0.89	97.98
CLFP (Ours)	2,3	1.47 +/- 0.10	0.89	97.98

[1]Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems, pages 2074–2082, 2016.

[2]Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[3] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay Namboodiri. Leveraging filter correlations for deep model compression. In The IEEE Winter Conference on Applications of Computer Vision, pages 835–844, 2020.



## Why did we choose pruning?

Method	R1, R2	Error%	FLOPs ( $10^5$ )	Pruned%
Training From Scratch	2,3	2.7%	0.89	-
HBFP (ours)	2,3	1.40 +/- 0.10	0.89	97.98
CLFP (ours)	2,3	1.47 +/- 0.10	0.89	97.98



## Plan of Action

- Experiment the above methods for Remaining Architectures i.e
  - HBFP - Resnet-56
  - CLFP - VGG-16 , Resnet-56
- Include Decorrelation Metric in CLFP.



## References

- [1].Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.
- [2]Mittal, Deepak, et al. "Recovering from random pruning: On the plasticity of deep convolutional neural networks." *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.
- [3].Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).



# Deep CNN Model Compression via Filter Pruning for Efficient ConvNets

Phase - II



# Outline

- Revisit the Problem Statement (Brief)
- Experiments Conducted in Phase 2 (Present)
- Reviewer Comments from **WACV 2021**
- Further Plan (For Concluding BTP)



# Motivation

- It was **empirically** found that in a NN there are **redundant** and **unimportant** Weights present which if **removed** would not affect the **performance** of the model.
- This was observed when **some** of the filters were **removed** from the CNN model at **random** i.e there was **no criteria** for discarding the filters and there was almost **no drop** in the **performance** [1].

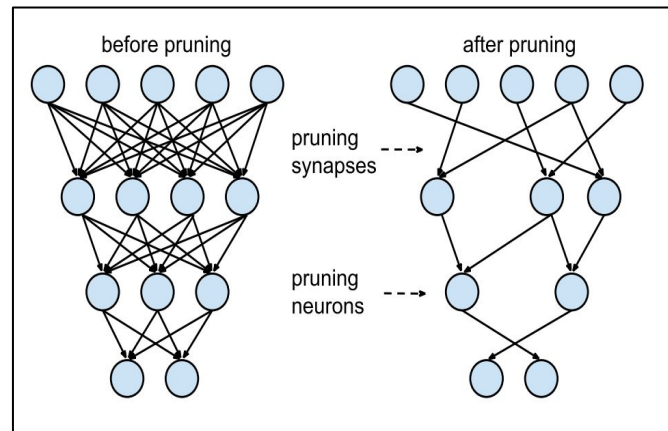
[1]

Mittal, Deepak, et al. "Recovering from random pruning: On the plasticity of deep convolutional neural networks." *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.



# Pruning

- It is a model **optimization** technique that involves eliminating unnecessary values in the weight tensor.
- It aids in the development of smaller and more efficient **neural networks**.

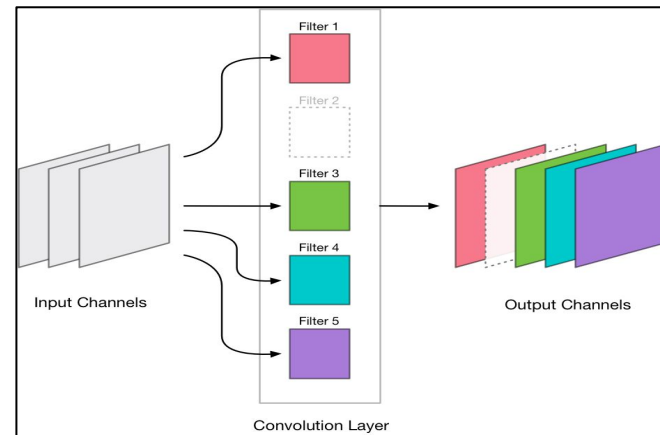
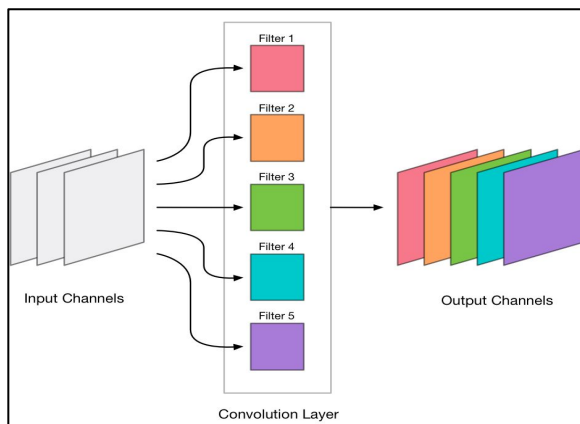


## Image Source [2]

Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems*. 2015.

# About Filter Pruning

- Filter pruning is one of the **model compression** techniques.
- We discard the filter using a **particular criteria** and the corresponding **feature map** gets **dropped** with it.





# Experiments conducted in Phase- I

## History Based Filter Pruning

- LeNet-5 on MNIST
- VGG-16 on Cifar-10

## Correlation Based Filter Pruning

- Lenet-5 on MNIST



## Why did we choose pruning?

Method	R1, R2	Error%	FLOPs ( $10^5$ )	Pruned%
Training From Scratch	2,3	2.7%	0.89	-
HBFP (ours)	2,3	1.40 +/- 0.10	0.89	97.98
CLFP (ours)	2,3	1.47 +/- 0.10	0.89	97.98



## Plan of Action (Phase - I)

- Experiment the above methods for Remaining Architectures i.e
  - HBFP - Resnet-56 (**Done**)
  - CLFP - VGG-16 (**Done**)
  - CLFP - Resnet-56 (**In Progress**)
- Include Decreasing correlation Metric in CLFP. (**In Progress**)

Note: By “Done” we mean that we have reasonable results which can be reported.



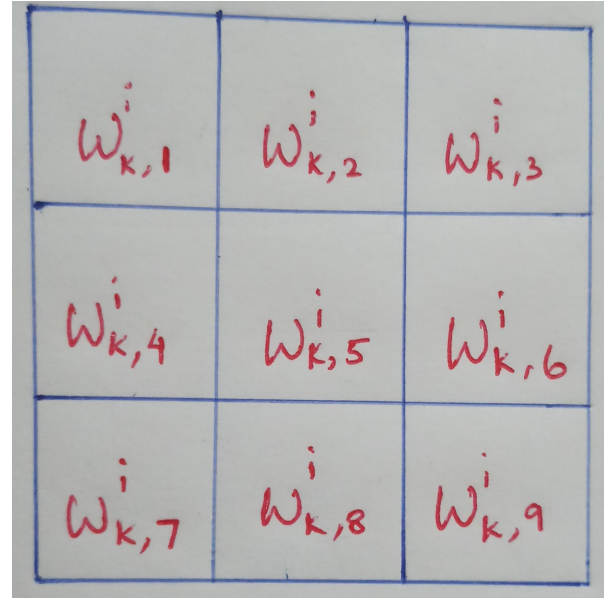
## Additional Work Done (Phase - II)

- Developed **API** for automatic FLOP calculation for tensorflow models.
- Effect of Custom Loss **with** and **without** it.
- Submitted Work to WACV 2021.

## Important Terminologies (For HBFP)

- **L1 Norm** of filter computed as follows

$$\ell_1(f_k^i) = \|f_k^i\|_1 = \sum_{p=1}^9 |w_{k,p}^i|$$



A 3x3 grid of handwritten labels representing filter weights. The labels are arranged in three rows and three columns, with each cell containing a weight symbol  $w_{k,p}^i$  where  $i$  is the row index and  $p$  is the column index.

$w_{k,1}^i$	$w_{k,2}^i$	$w_{k,3}^i$
$w_{k,4}^i$	$w_{k,5}^i$	$w_{k,6}^i$
$w_{k,7}^i$	$w_{k,8}^i$	$w_{k,9}^i$

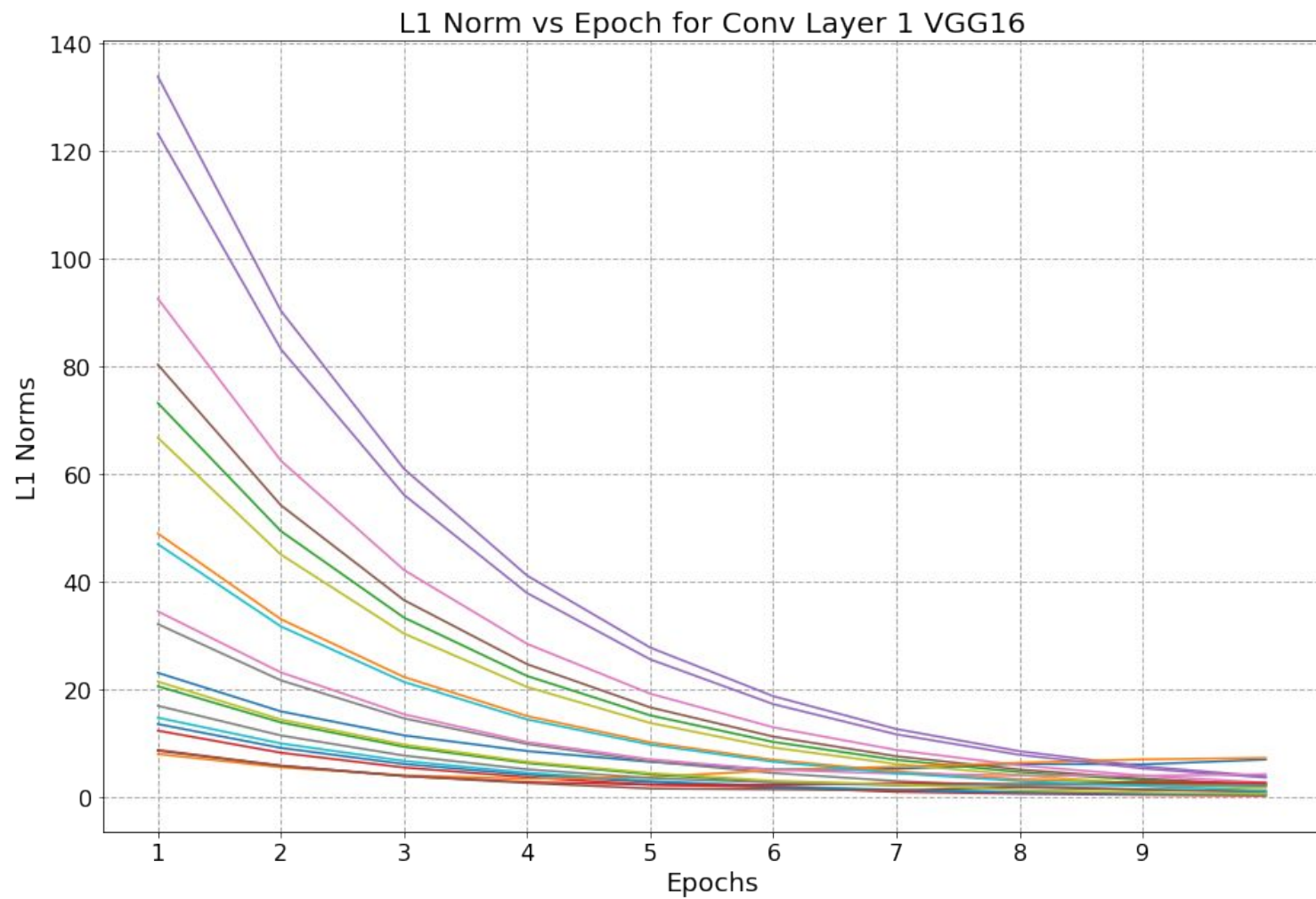


## Important Terminologies (For HBFP)

- **Absolute Difference** Between L1 norm.
- Here ' $t$ ' indicates the training epoch.

$$d_{f_k^i, f_k^j}(t) = \left| \ell_1(f_k^i) - \ell_1(f_k^j) \right|$$





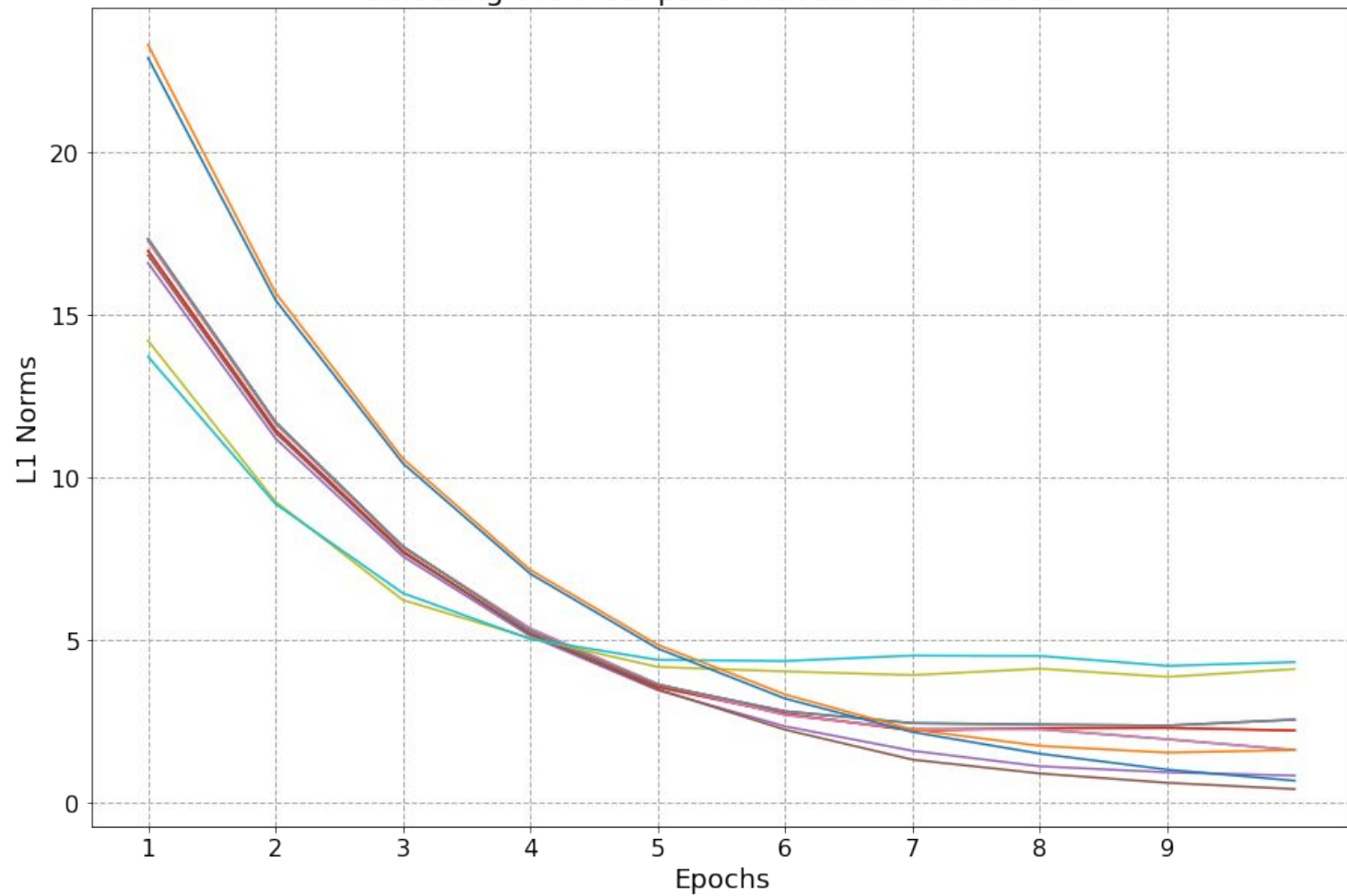


## Important Terminologies (For HBFP)

- Filter Selection **criteria**.
- Here ‘ $N$ ’ is the number of training epochs.

$$D_{f_k^i, f_k^j} = \sum_{t=1}^N d_{f_k^i, f_k^j}(t)$$

Selecting the Filter pairs with similar behaviour





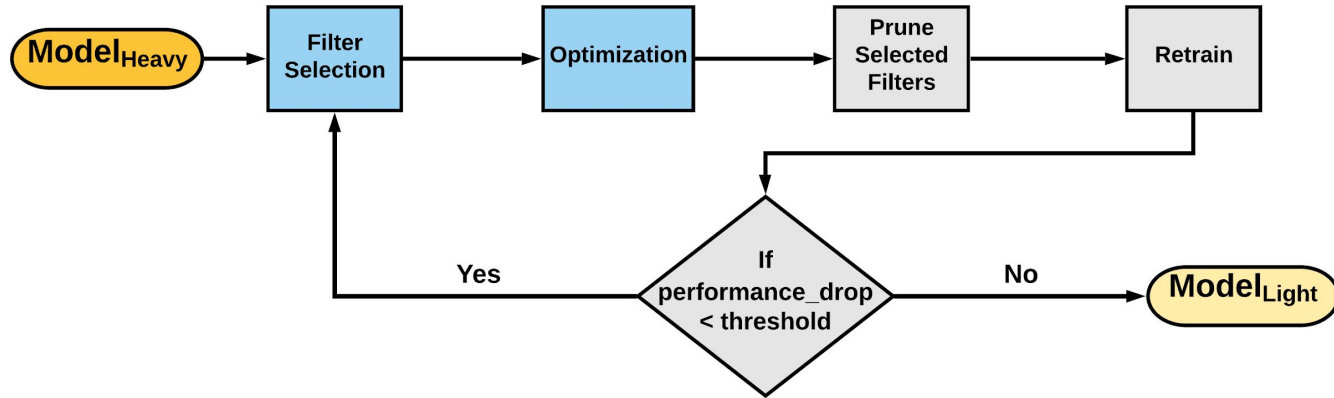
## Custom Loss (For HBFP)

- We add a **new** term to the loss function.
- We then, train our model with this **new loss function**.

$$C_1 = \exp \left( \sum_{f_k^i, f_k^j \in Q_i} d_{f_k^i, f_k^j}(t) \right)$$

$$\mathbb{L}_{HBFP} = \arg \min_W \left( C(W) + \lambda * C_1 \right)$$

# Algorithm

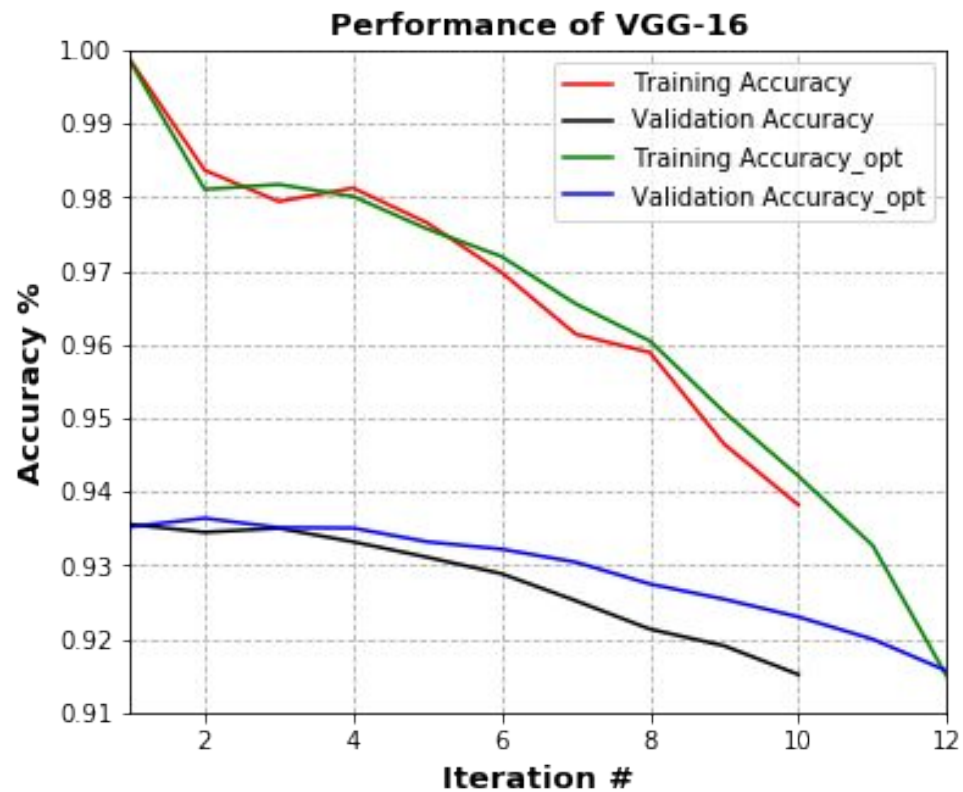




## Effect Of Custom Loss on vgg-16 (Result)

- We will use two plots demonstrate the effect of custom loss function.
  1. Overall Performance Plot.
  2. Plot of accuracy and epochs for every Iteration.

## Overall Plot for all Iterations

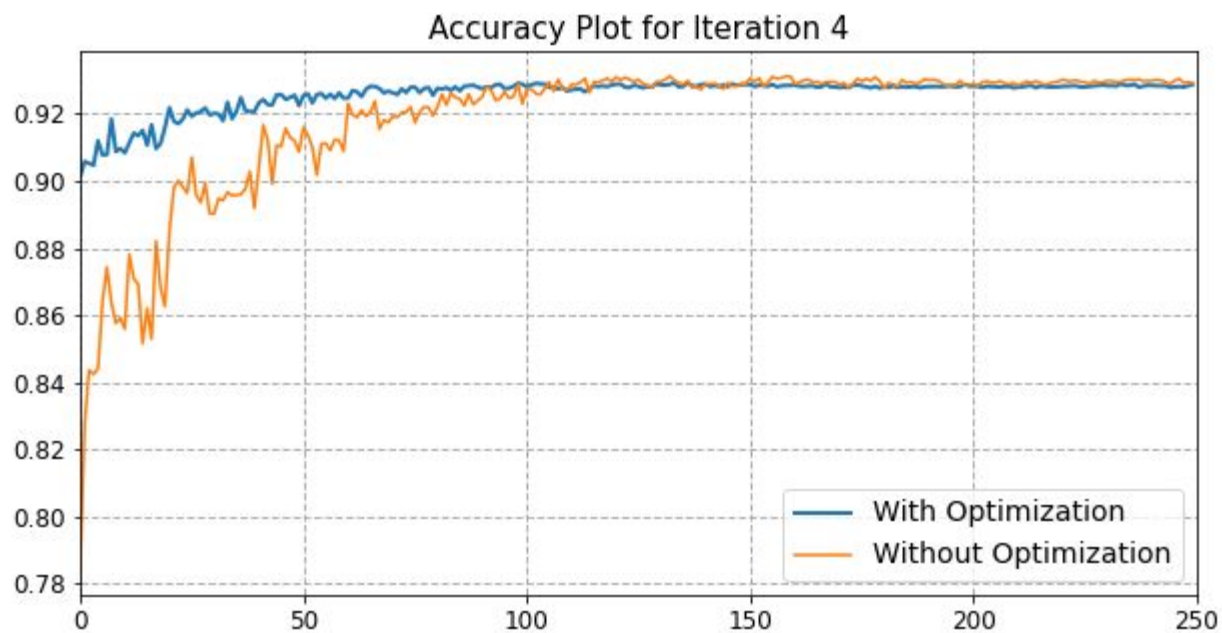


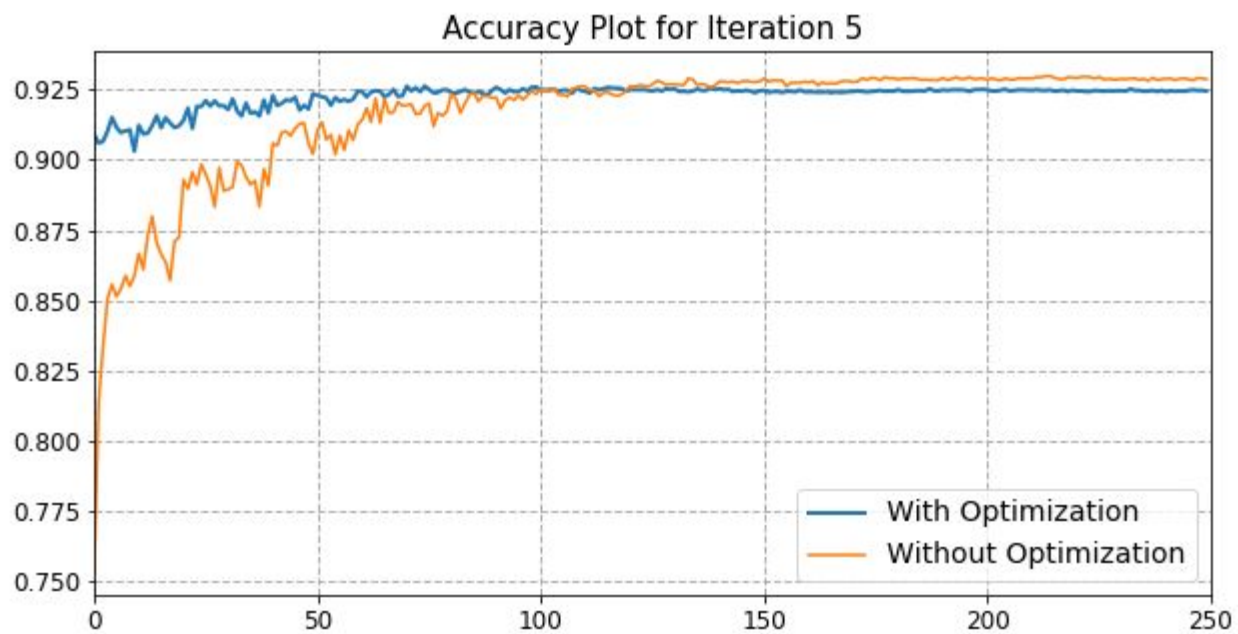


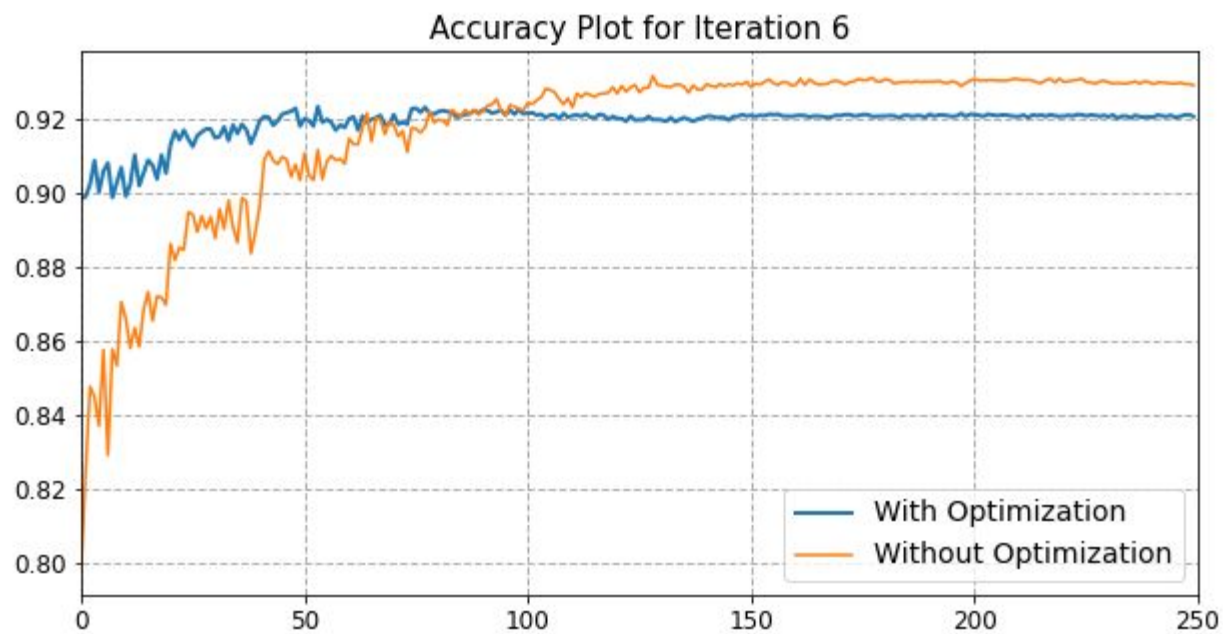


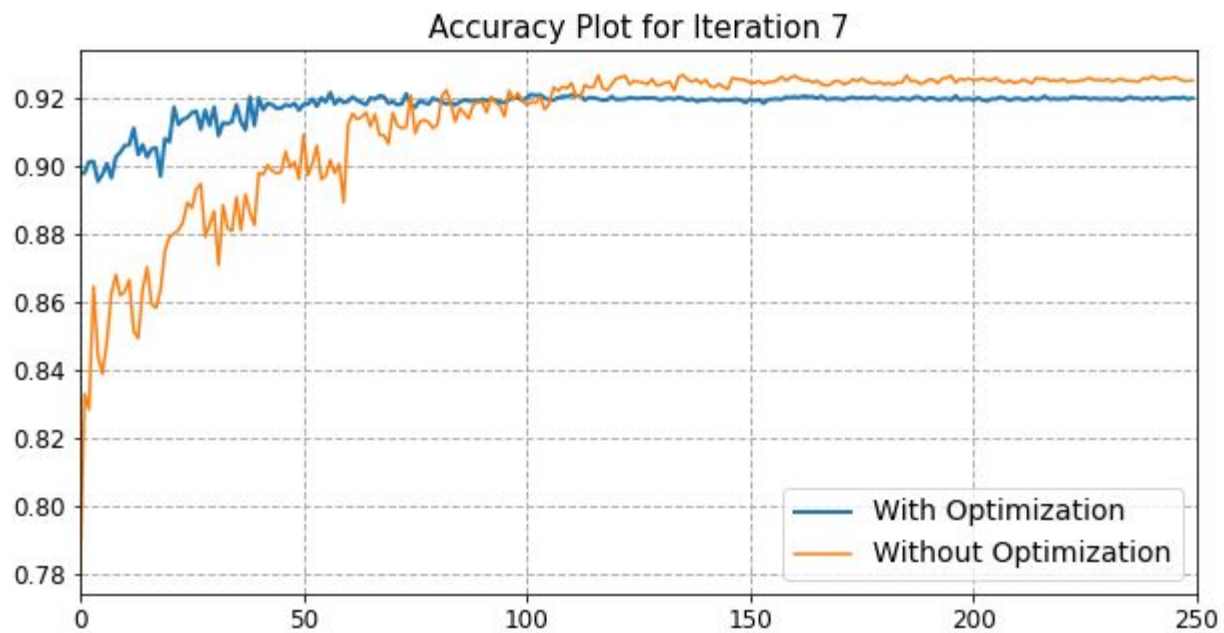


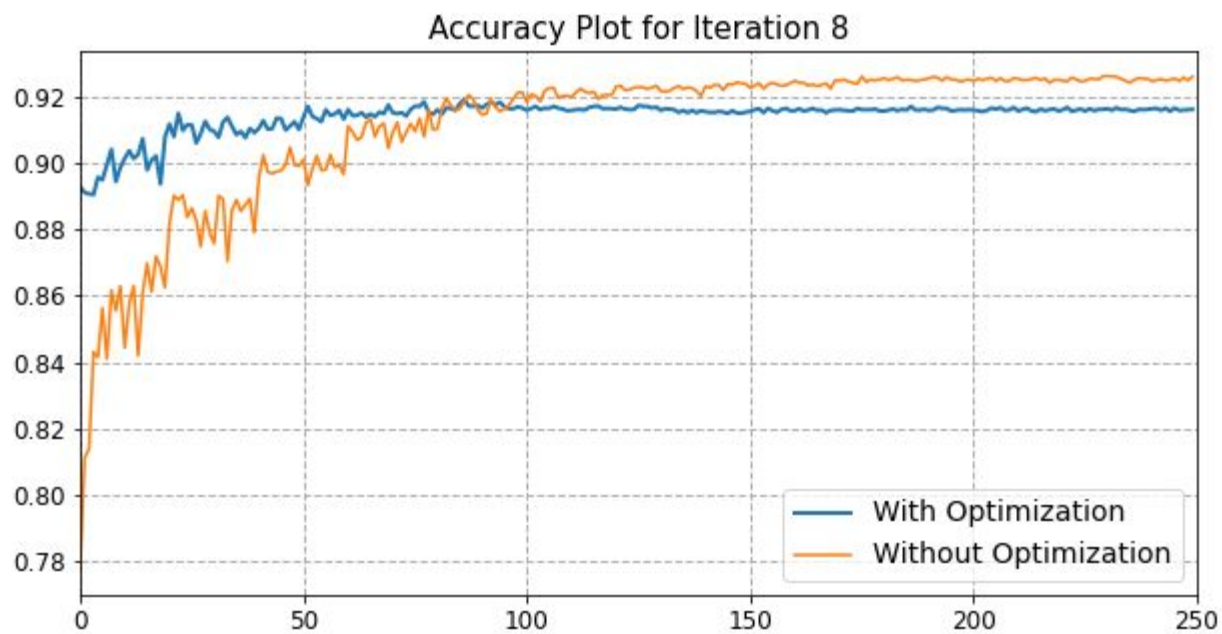




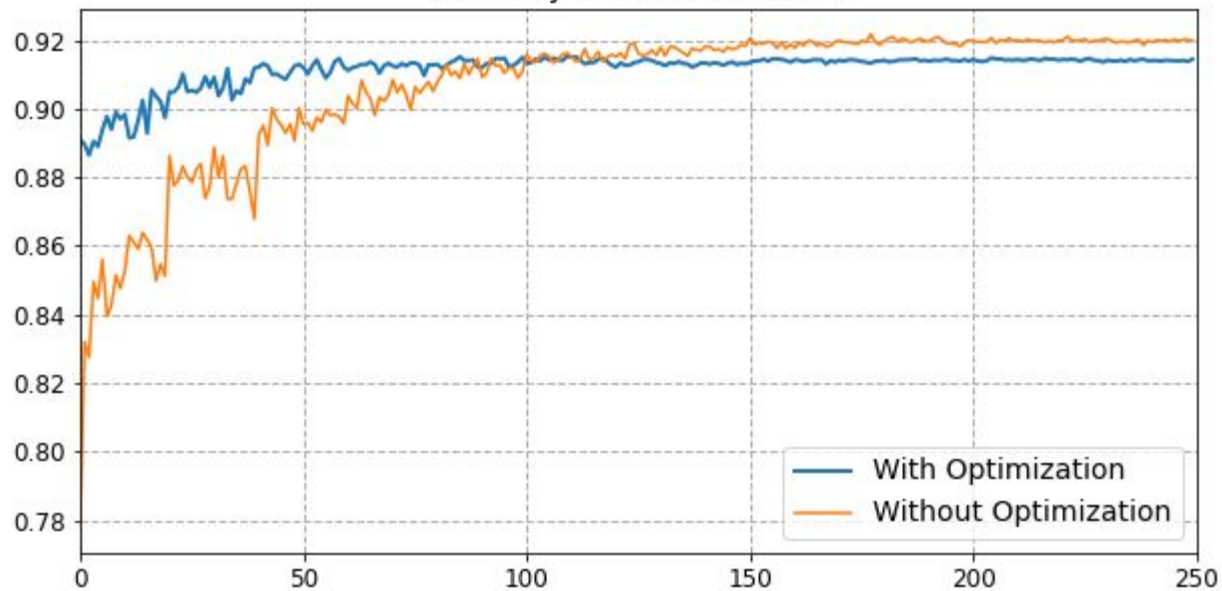






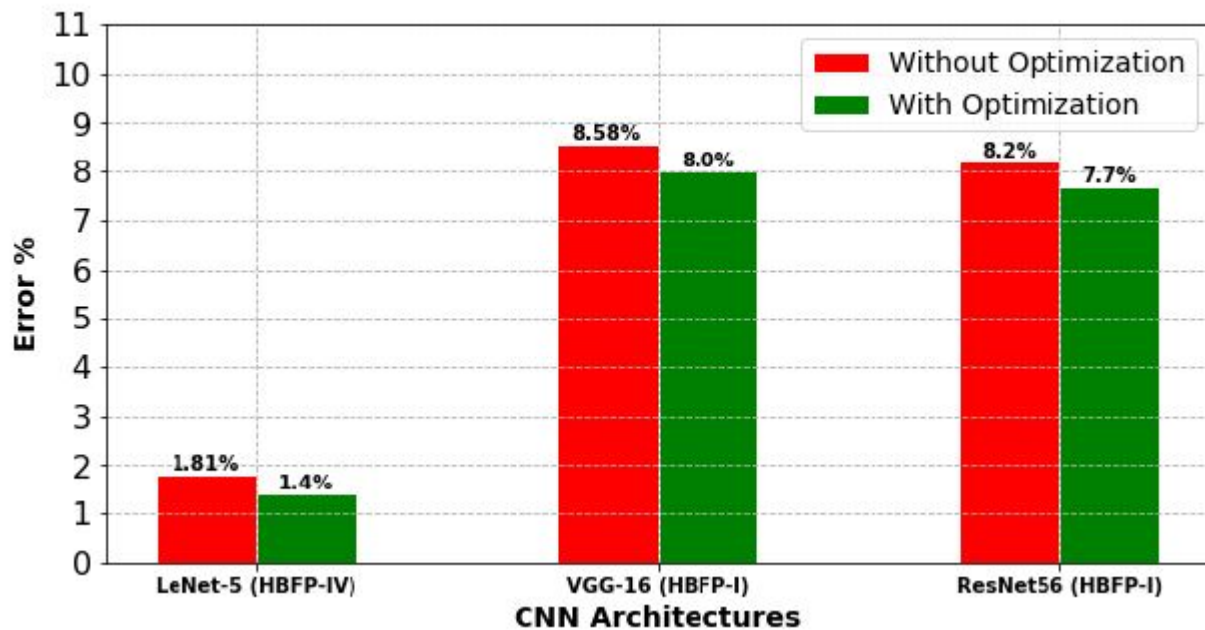


Accuracy Plot for Iteration 9





## Effect of CL on all 3 architectures (Result)





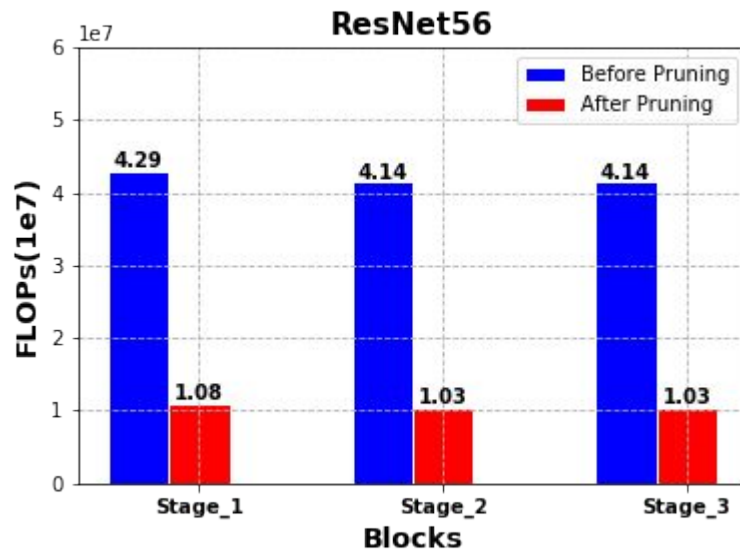
## Result of ResNet-56 on Cifar 10

Method	r1,r2,r3	Accuracy	Flops( $10^8$ )	Error	Pruned Flops
L1 Norm[1]		-0.02	0.91	6.94	27.60
CFP[2]	8,16,27	0.94	0.29	7.37+/-0.17	75.59
HBFP-I	8,16,32	0.6	0.31	8.2 +/- 0.11	74.91
HBFP-II	7,15,31	0.98	0.27	8.58+/- 0.04	78.43

[1]Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).

[2]Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

## Result of ResNet-56 on Cifar 10 (For HBFP)





## WACV Reviewer Comments (For HBFP)

### Reviewer - I (Weak Reject)

Summary: This submission aims to propose a new method to prune a heavy network.

### Reviewer - II (Weak Reject)

Summary: This paper presents a new channel pruning method by computing the criteria for pruning using historical values. This method is also incorporated with an optimization-based pruning method. Results on MNIST and CIFAR are reported..



## WACV Reviewer Comments (For HBFP)

Reviewer - III (Border Line)

**Summary:** This paper proposes to prune filters during the training of the convolutional neural networks. This is done by group the filters into pairs and prune one filter from the pairs that are most similar to each other. The prune stage is done iteratively until the accuracy of the network drops below a predefined threshold.



## **WACV Reviewer Comments (For HBFP)**

Reviewer - I

Strength: The writing and Idea is clear.

Weakness: The experimental results can not demonstrate the efficiency of the proposed method



## WACV Reviewer Comments (For HBFP)

Reviewer - II

Strength:

- The idea of utilizing historical information in training process for channel pruning is new and interesting.
- The proposed method can be incorporated with other model compression methods.

Weakness:

- The experiment part is very weak in terms of both experiment settings and results.



## WACV Reviewer Comments (For HBFP)

Reviewer - II

Strength:

- The proposed method is straightforward and easy to implement. The idea of pruning during training is interesting because there is no need to first fully train an uncompressed model and then prune it. Instead, the networks can be pruned on the fly.
- This paper is well written and easy to follow.

Weakness:

- It is suggested that the author conduct more experiments on those datasets on more network architectures





## Results of VGG16 on CIFAR 10 (For CBFP)

Method	Base Error %	FLOPs ( $10^8$ )	Retrain Error %	Pruned %
L1 Norm [1]	6.75	2.06	6.60	34.20
ABFP [2]	7.08	0.58	7.13	81.39
CFP [3]	6.51	0.57	7.02 +/- 0.16	83.49
CLFP(Ours)	6.51	0.43	8.15 +/- 0.09	86.21



## About API

- Input is the Tensorflow CNN model.
- Output is a dictionary with key as Layer name and value is the number of Flops.



## Plan for concluding BTP

- Conducting More experiments using different data sets using different architectures.
  - ResNet-110
  - DenseNet
  - ResNet-18
  - On datasets of cifar 100, tiny-imagenet.
- Reducing the Error Rate obtained on the done experiments using hyper parameter tuning.



# Deep CNN Model Compression via Filter Pruning for Efficient ConvNets

End Evaluation



## Problem Statement

- **Bigger** and **Deeper** CNN architectures come at a cost of High **computational power** for training and High **Storage capacity**.
- As a result of which the **deployment** of these models on some low grade devices like microcontrollers and embedded devices becomes extremely **difficult**.
- To address this **issue** we attempt at reducing the size of CNN using an approach called **Filter** pruning aka **Channel** pruning.



## Plan for concluding BTP (based on previous evaluation)

- Conducting More experiments using different architectures.
  - ResNet-110 on cifar-10
  - VGG-16 on datasets of cifar 100
- Reducing the Error Rate obtained on the done experiments using hyper parameter tuning.
- Extension To CFP with L1 norm metric for selecting important filters.



## Reducing Error Rate for HBFP

- We employed changes to the Custom Loss function, such that the Custom loss parameter is now bounded unlike the previous Loss function.

$$C_1 = \exp \left( \frac{-1}{1n(\sum_{f_k^i, f_k^j \in Q_i} d_{f_k^i, f_k^j}(t))} \right)$$

## Previous Loss Function ( PLF )

$$C_1 = \exp \left( \sum_{f_k^i, f_k^j \in Q_i} d_{f_k^i, f_k^j}(t) \right)$$

## Current Loss Function ( CLF )

$$C_1 = \exp \left( \frac{-1}{1n(\sum_{f_k^i, f_k^j \in Q_i} d_{f_k^i, f_k^j}(t))} \right)$$



S.No.	$\sum_{f_k^i, f_k^j \in Q_i} d_{f_k^i, f_k^j}(t)$	CLF	PLF
1.	1	0	2.71
2.	10	0.64	$22 \times 10^3$
3.	20	0.71	$48 \times 10^7$
4.	50	0.77	inf
5.	70	0.79	inf
6.	100	0.80	inf
7.	500	0.85	inf
8.	1000	0.86	inf
9.	10000	0.89	inf



## Experiments Conducted on HBFP

1. LeNet-5 on MNIST
2. VGG-16 on Cifar-10
3. VGG16 on Cifar-100
4. ResNet-56 on Cifar-10
5. ResNet-110 on Cifar10

# Results of LeNet-5 on MNIST

S.No	Set	Method	R1, R2	Error%	FLOPs (M)	Pruned%
1.	--	Base Line	20,50	0.83	4.4M	0
2.	1	CFP(I) [3]	4,5	0.91 +/- 0.07	0.19M	95.57
3.	1	<b>HBFP (III)</b>	4,5	0.98 +/- 0.05	0.19M	95.57
4.	2	ABFP [2]	3,5	2.21	0.15M	96.41
5.	2	<b>HBFP (II)</b>	3,5	1.08 +/- 0.10	0.15M	96.41
6.	3	CFP(II) [3]	2,3	1.77 +/- 0.08	0.08M	97.98
7.	3	<b>HBFP (I)</b>	2,3	1.36 +/- 0.10	0.08M	97.98

[1]Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems, pages 2074–2082, 2016.

[2]Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[3] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay Namboodiri. Leveraging filter correlations for deep model compression. In The IEEE Winter Conference on Applications of Computer Vision, pages 835–844, 2020.

# Results of VGG-16 on Cifar-10

S.No	Set	Method	Acc%	Flops (M)	Flops reduction (%)	Parameters (M)	Parameters reduction (%)
		VGG-16 [1]	93.96	313.73	0	14.98	0
1.	--	L1 Norm [2]	93.40	206	34.3	5.40	64
2.	1	GM [3]	93.58	201.1	35.9	--	--
3.	1	<b>HBFP (I)</b>	93.04	90.23	71.21	4.2	71.8
4.	2	Hrank [4]	91.23	73.7	76.5	1.78	92.0
5.	2	<b>HBFP (II)</b>	92.54	75.05	76.05	3.5	76.56
6.	3	CFP [5]	91.83	59.15	81.14	2.8	81.1
7.	3	<b>HBFP (III)</b>	92.3	62.3	80.09	2.9	80.47

## References for previous slide:

- [1] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).
- [3] He, Yang, et al. "Filter pruning via geometric median for deep convolutional neural networks acceleration." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [4] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [5] Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

## Results of VGG-16 on Cifar-100

S.No.	Method	Base Accuracy (%)	Retrain Accuracy (%)	FLOPs (M)	Pruned Flops
	VGG-16 [1]	72.21	--	320	0
1.	L1 Norm [2]	73.14	71.11	196	37.32
2.	Taylor [3]	73.14	72.02	196	37.32
3.	FOFP [4]	73.14	72.31	196	37.32
4.	<b>HBFP (ours)</b>	73.08	72.76	193.8	38.1

## References for previous slide:

[1] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).

[3] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *arXiv preprint arXiv:1611.06440*, 3, 2016.

[4] Qin, Zhuwei, et al. "Functionality-Oriented Convolutional Filter Pruning." *arXiv preprint arXiv:1810.07322* (2018).

[5] Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

# Result of ResNet-56 on Cifar-10

S.No.	Set	Method	Accuracy	Flops (M)	Flops reduction (%)	Parameters (M)	Parameter reduction (%)
		ResNet-56 [1]	93.26	125.49	0	0.85	0
1.	1	L1 Norm [2]	93.06	90.9	27.6	0.73	14.1
2.	1	VFP [3]	92.26	96.6	20.3	0.67	20.49
3.	1	<b>HBFP (I)</b>	92.42	70.81	43.68	0.48	46.38
4.	2	AMC [4]	91.9	62.74	50	--	--
5.	2	CP [5]	91.8	62.74	50	--	--
6.	2	<b>HBFP (III)</b>	91.79	31.54	74.91	0.21	74.9
7.	3	HRank [6]	90.72	32.52	74.1	0.27	68.1
8.	3	GAL [6]	90.36	49.99	60.2	0.29	65.9
9.	3	<b>HBFP (II)</b>	92.25	49.22	60.85	0.33	60.85



## References for previous slide:

- [1] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).
- [3] Zhao, Chenglong, et al. "Variational convolutional neural network pruning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [4] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the Euro-pean Conference on Computer Vision (ECCV)*, pages 784–800, 2018.
- .
- [5] He, Yihui, Xiangyu Zhang, and Jian Sun. "Channel pruning for accelerating very deep neural networks." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
- [6] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [7] Lin, Shaohui, et al. "Towards optimal structured cnn pruning via generative adversarial learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

# Result of ResNet-110 on Cifar 10

S.No.	Set	Method	Accura cy	Flops (M)	Flops reduction(%)	Parameter (M)	Parameter r reduction (%)
		ResNet-110	93.5	252.89	0	1.72	0
1.	1	VFP [1]	92.96	160.7	36.44	1.01	41.27
2.	1	L1-Norm [2]	93.3	155	38.7	1.16	32.6
3.	1	GAL [3]	92.55	130.2	48.5	0.95	44.8
4.	1	<b>HBFP (I)</b>	93.01	119.7	52.69	0.81	52.66
5.	2	Hrank [4]	92.65	79.3	68.6	0.53	68.7
6.	2	<b>HBFP (III)</b>	92.83	80.2	68.31	0.54	68.28
7.		<b>HBFP (II)</b>	92.91	98.9	60.89	0.67	41.27
8.		<b>HBFP (IV)</b>	91.96	63.3	74.95	0.43	74.92

## References for previous slide:

- [1] Zhao, Chenglong, et al. "Variational convolutional neural network pruning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).
- [3] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [4] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

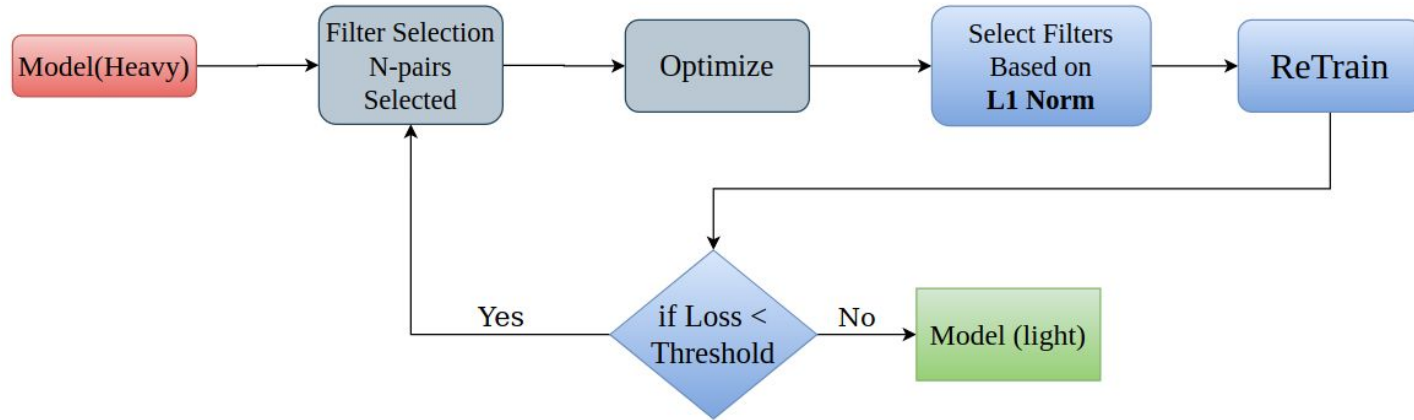


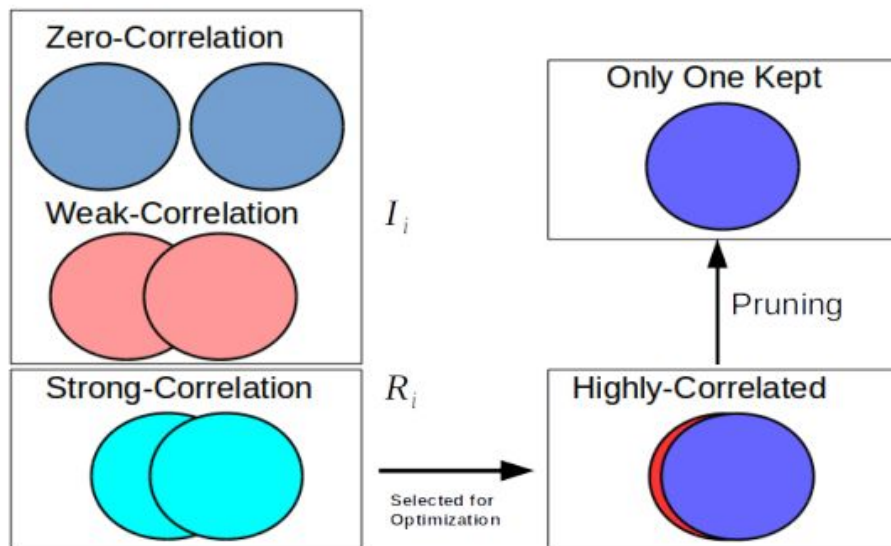
## Extension to CFP

- Combining the approach of pruning filters based on **Importance** and **Redundancy**.
- **Redundancy** - Selecting the Filter pairs which are highly correlated.
- **Importance** - Selecting the filter based on L1 Norm between highly correlated filter pairs.

Source: Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

# Algorithm







## Experiments Conducted on CFP-Extension

- LeNet-5 on MNIST
- VGG-16 on CIFAR-100
- ResNet-56 on CIFAR-100

# Results of LeNet-5 on MNIST

S.No	Set	Method	R1, R2	Error%	FLOPs (M)	Pruned%
1.	--	Base Line	20,50	0.83	4.4M	0
2.	1	CFP(I) [3]	4,5	0.91 +/- 0.07	0.19M	95.57
3.	1	<b>CBFP (III)</b>	4,5	0.96 +/- 0.05	0.19M	95.57
4.	2	ABFP [2]	3,5	2.21	0.15M	96.41
5.	2	<b>CBFP (II)</b>	3,5	1.12 +/- 0.10	0.15M	96.41
6.	3	CFP(II) [3]	2,3	1.77 +/- 0.08	0.08M	97.98
7.	3	<b>CBFP (I)</b>	2,3	1.58 +/- 0.10	0.08M	97.98

[1]Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Advances in neural information processing systems, pages 2074–2082, 2016.

[2]Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[3] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay Namboodiri. Leveraging filter correlations for deep model compression. In The IEEE Winter Conference on Applications of Computer Vision, pages 835–844, 2020.



# Results of VGG-16 on Cifar-10

S.No	Set	Method	Acc%	Flops (M)	Flops reduction (%)	Parameters (M)	Parameters reduction (%)
		VGG-16 [1]	93.96	313.73	0	14.98	0
1.	--	L1 Norm [2]	93.40	206	34.3	5.40	64
2.	1	GM [3]	93.58	201.1	35.9	--	--
3.	1	<b>CBFP (I)</b>	93.24	90.23	71.21	4.2	71.8
4.	2	Hrank [4]	91.23	73.7	76.5	1.78	92.0
5.	2	<b>CBFP (II)</b>	92.14	75.05	76.05	3.5	76.56
6.	3	CFP [5]	91.83	59.15	81.14	2.8	81.1
7.	3	<b>CBFP (III)</b>	92.21	62.3	80.09	2.9	80.47

## References for previous slide:

- [1] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).
- [3] He, Yang, et al. "Filter pruning via geometric median for deep convolutional neural networks acceleration." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [4] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [5] Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

# Result of ResNet-56 on Cifar-10

S.No.	Set	Method	Accuracy	Flops (M)	Flops reduction (%)	Parameters (M)	Parameter reduction (%)
		ResNet-56 [1]	93.26	125.49	0	0.85	0
1.	1	L1 Norm [2]	93.06	90.9	27.6	0.73	14.1
2.	1	VFP [3]	92.26	96.6	20.3	0.67	20.49
3.	1	<b>CBFP (I)</b>	92.42	70.81	43.68	0.48	46.38
4.	2	AMC [4]	91.9	62.74	50	--	--
5.	2	CP [5]	91.8	62.74	50	--	--
6.	2	<b>CBFP (III)</b>	91.44	31.54	74.91	0.21	74.9
7.	3	HRank [6]	90.72	32.52	74.1	0.27	68.1
8.	3	GAL [7]	90.36	49.99	60.2	0.29	65.9
9.	3	<b>CBFP (II)</b>	91.65	49.22	60.85	0.33	60.85

## References for previous slide:

- [1] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).
- [3] Zhao, Chenglong, et al. "Variational convolutional neural network pruning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
- [4] Zhao, Chenglong, et al. "Variational convolutional neural network pruning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019..
- [5] He, Yihui, Xiangyu Zhang, and Jian Sun. "Channel pruning for accelerating very deep neural networks." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
- [6] Lin, Mingbao, et al. "HRank: Filter Pruning using High-Rank Feature Map." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [7] Lin, Shaohui, et al. "Towards optimal structured cnn pruning via generative adversarial learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.



## Conclusion of BTP

- Implemented 2 existing techniques for CNN filter pruning.
  - “Pruning filters for efficient convnets”
  - “Leveraging filter correlation for deep model compression”
- Proposed and implemented a novel criteria for filter pruning that leverages the training History.
- Employed modification to loss function as an optimization for the pruning criteria.
- Extended the idea of correlation based filter pruning by adding the l1 norm metric to prune only the unimportant filter in a pair.



## Major References

[1].Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." *The IEEE Winter Conference on Applications of Computer Vision*. 2020.

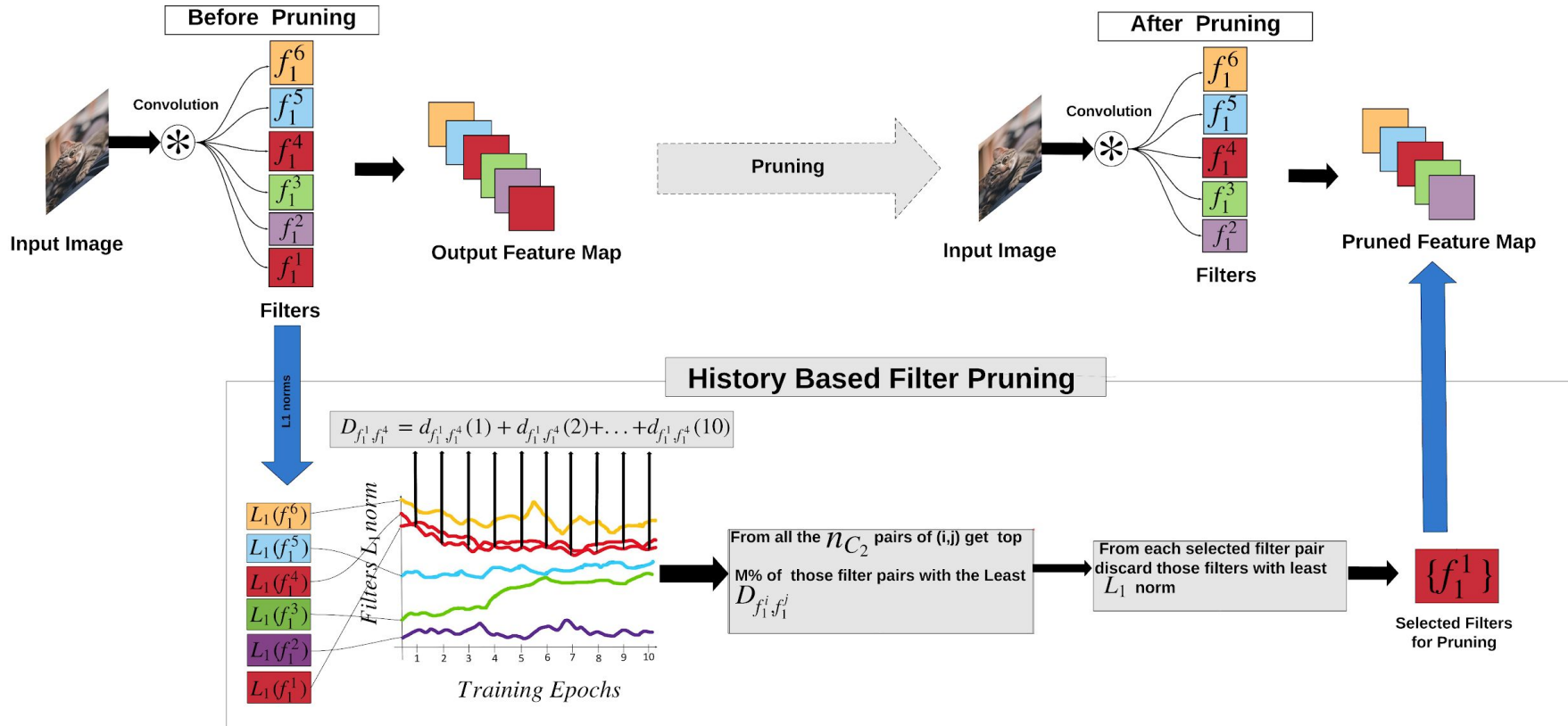
[2] Li, Hao, et al. "Pruning filters for efficient convnets." *arXiv preprint arXiv:1608.08710* (2016).



## Inspiration for HBFP

- We treat L1 Norm as a Time Series of the epoch number.
- Every filter in a particular layer has its own time series.
- We use the point-to-point distance metric as a measure for similarity.

[1] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata and Alfredo Pulvirenti (September 12th 2012). Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining, Advances in Data Mining Knowledge Discovery and Applications, Adem Karahoca, IntechOpen, DOI: 10.5772/49941. Available from: <https://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/similarity-measures-and-dimensionality-reduction-techniques-for-time-series-data-mining>





# Measuring Similarity between Two Time Series [1]

