Deep Neural Networks' Compression Techniques Study

Appling Compressing Methods for Investigating their Effects on Siren based Deep Neural Networks

Chiarlo Francesco Maria

University Polytechnic of Turin

March 3, 2021

POLITECNICO
DI TORINO

Siren DNNs

$$F(x, \Phi, \nabla_x \Phi, \nabla_x^2 \Phi, \dots) = 0, \Phi : x \to \Phi(x)$$

Short Presentation of Siren Architecture

Siren Deep Neural Network Architectures (2)

Main characteristics of Siren based Deep Models employed for as major purpose of implicitly representing input processed image:

- It belongs to **MLP family** of Deep Learning Networks, this means that It resembles a Fully Connected Architecture;
- It employes as non-linear activation function a **trigonometric sine function**.
- It is feeded by means of a set of **input coordinates** throught which **output pixel value**, in other words pixel magnitude, in a given scale and so range of values is predicted;
- It does not employes a well-known and widespread weigths initialization technique as **Xavier Initialization** but instead decided to adopt a **Custom Uniform Initialization**, as described within the same Siren paper.

Siren Based Network Topolgy

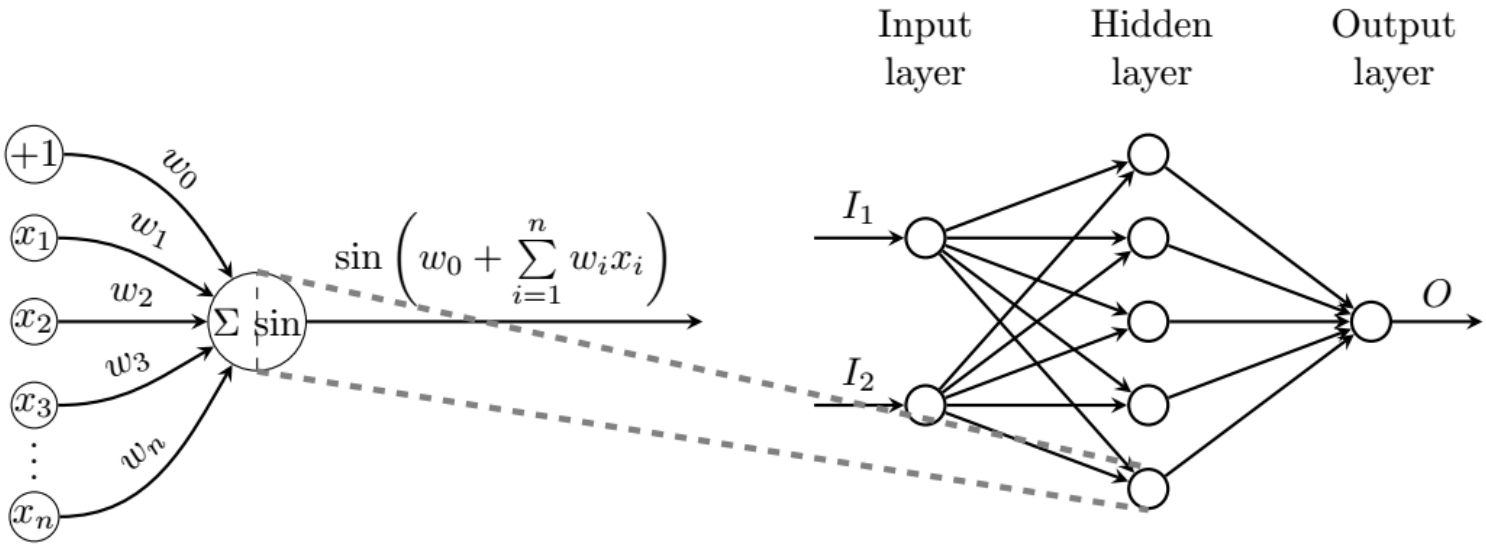


Figure: Siren Architecture Example

Example of tiny Siren like architecture, in order to show the minimal configuration that can be employed to describe network's major behavior.

Compression Techniques

Short Introduction about Deep Nets Compression Techniques

Compression Techniques

Deep Neural Network Models Compression Techniques have been studied from many years, and are now a day in ever high demand due to several reasons which can be summerized in the following:

- Pushing state-of-the-art models on salient tasks within domanins as **Computer Vision, NLP** corresponds to models becoming larger, increasing **Memory and Storage Requirements** at both computation and storing time.
- Bigger models lead to **Larger Carbon Footprint**, where this issue follows also from observation just done above.
- Other issues araise when we are attempting to deploy state-of-the-art models within those contexts where we find ourselves constrained by computing limited resources, as well as, reduced amount of battery capability as for **Mobile Applications and IoT Devices**.

Compression Techniques (2)

Deep Neural Network Models Compression Techniques can be devided into a given number of different classes or categories depending on the approaches they follow. In particular we can devide such methods, without the claim of being for sure exhaustive but at least quite accurate, as follows:

Weight Sharing

- Cluster-based Weight Sharing
- Learning Weight Sharing
- Weight Sharing in Large Archs
- ...

Network Pruning

- Pruning via Weights Regularization
- Pruning via Loss Sensitivity
- Structured Pruning
- ...

Quantization

- Adaptive Range and Clipping
- Linear Range Quant
- DoReFa Net Quant
- WRPN Net Quant
- ...

Knowledge Distillation

- Recurrent (Autoregressive NNs
- Transformer-based (Non-Autoregressive) NNs
- Data Free KD

Selected Compression Techs.

Compression Techiques adopted to compress Siren Net Instances

Selected Compression Techniques

The Compression Techniques we decided to adopt and investigate among the wide varieaty of possible choices are the two following:

- **Automated Gradual Pruning** by *Michael Zhu et al., 2017* - an instance of a possible pruning like compressing method:
  - Model pruning is the art of discarding the weights that **do not improve a model's performance**, in other words **non-significant or non-salient parameters**;
  - Careful pruning enables us to compress and deploy our **state-of-the-art neural networks** onto **mobile phones and other resource-constrained devices**.
- **Linear Range Quantization** by *Benoit et al., 2018* - an instance of a possible quantizing aware training method;

Automated Gradual Pruner

Deep Neural Network Pruning Based Compression Techinque

Automated Gradual Pruner

Automated Gradual Pruner (AGP), discussed in **To prune, or not to prune: exploring the efficacy of pruning for model compression**, the authors Michael Zhu and Suyog Gupta propose an intuitive, fresh new pruning approach briefly described as:

- **new automated gradual pruning algorithm** in which the sparsity is increased from an initial sparsity value s_i (usually 0) to a final sparsity value s_f over a span of n pruning steps;
- The intuition behind this sparsity function in equation below is to **prune the network rapidly in the initial phase** when:
  - the redundant connections are abundant, and;
  - gradually reduce the number of weights being pruned each time as there are fewer and fewer weights remaining in the network.

Automated Gradual Pruner (2)

Other interesting properties related to **Automated Gradual Pruner (AGP)**, and discussedwithin Michael Zhu and Suyog Gupta's paper, for better appreciating and understanding the usefulness of such a technique are:

- It Requires a lower number of trials, and attempts for identifying meaningful set of hyper-params for leading the pruning approach, compared to other techniques such Magnitude Level Pruning and other similars;
- It does not made particular assumptions on weight values density distribution;
- It is agnostic with respect to the particular Deep Neural Network Architecture chosen;

Input Dataset

Describing selected Input Target Image for our trials

Figure: Camera 512x512 cameramen image

Target Image for Training Siren Models

Differently from Siren paper's Camera Image, we decide to resize it, by cropping the full image down to 256x256 image about its center, leading to the following update image:

Figure: Camera 256x256 target image

Image Feature	Value
name	Camera
shape	(256, 256)
size_byte	65536
image_band	(L,)

Table: Cropped Camera Image Characteristics

Data Distribution for Target Image for Training Siren Models

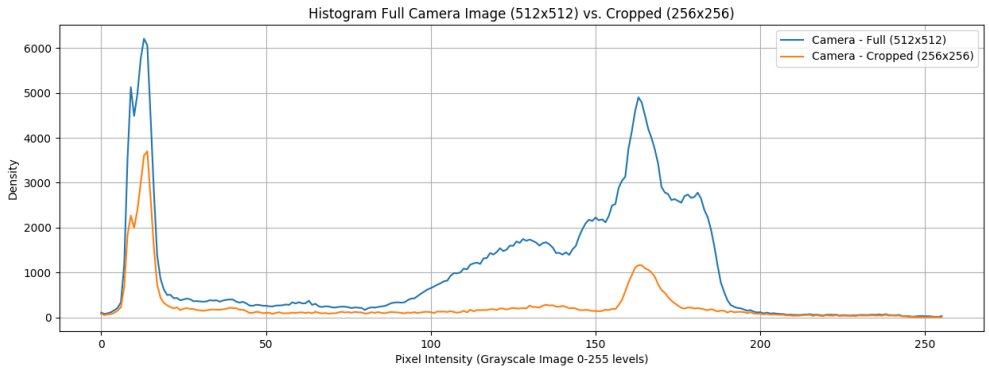
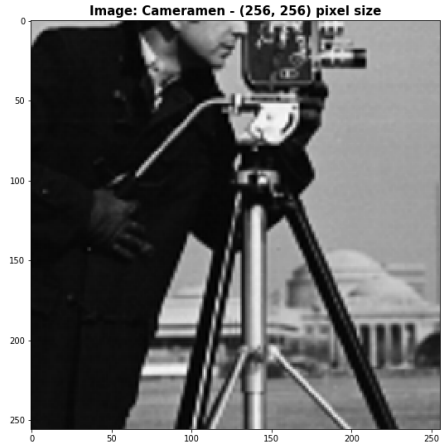
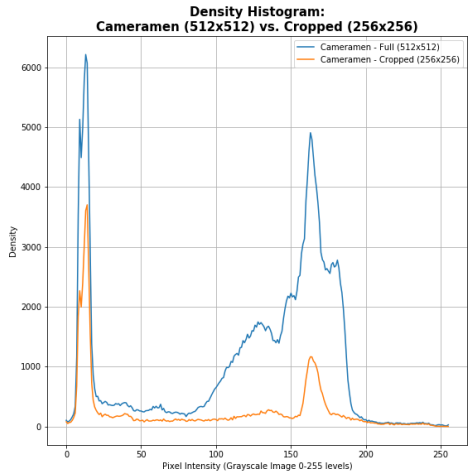


Figure: Camera target image, full and cropped data distribution

General Summarizing Image Overview

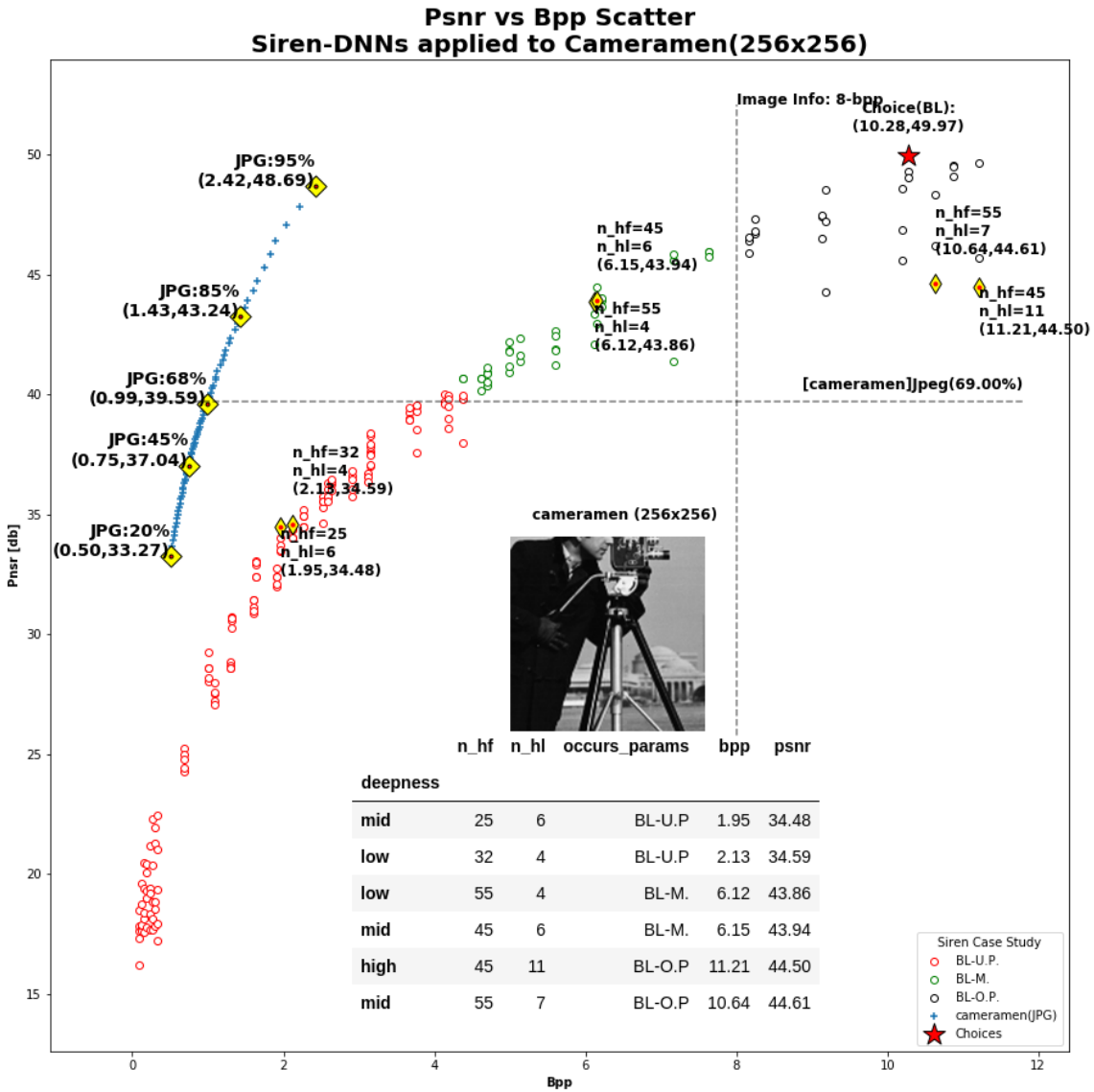




Summary Table Image Cameramen: full size and cropped versions

	name	shape	size_byte	image_band	entropy
full	Cameramen	(512, 512)	262144	(L,)	7.047955
cropped	Cameramen	(256, 256)	65536	(L,)	6.780318

Base Pnsr [db] vs. Bpp Jpeg Scatterplot



deepness	n_hf	n_hl	occurs_params	bpp	psnr
mid	25	6	BL-U.P	1.95	34.48
low	32	4	BL-U.P	2.13	34.59
low	55	4	BL-M.	6.12	43.86
mid	45	6	BL-M.	6.15	43.94
high	45	11	BL-O.P	11.21	44.50
mid	55	7	BL-O.P	10.64	44.61

Training Strategies and Workflow

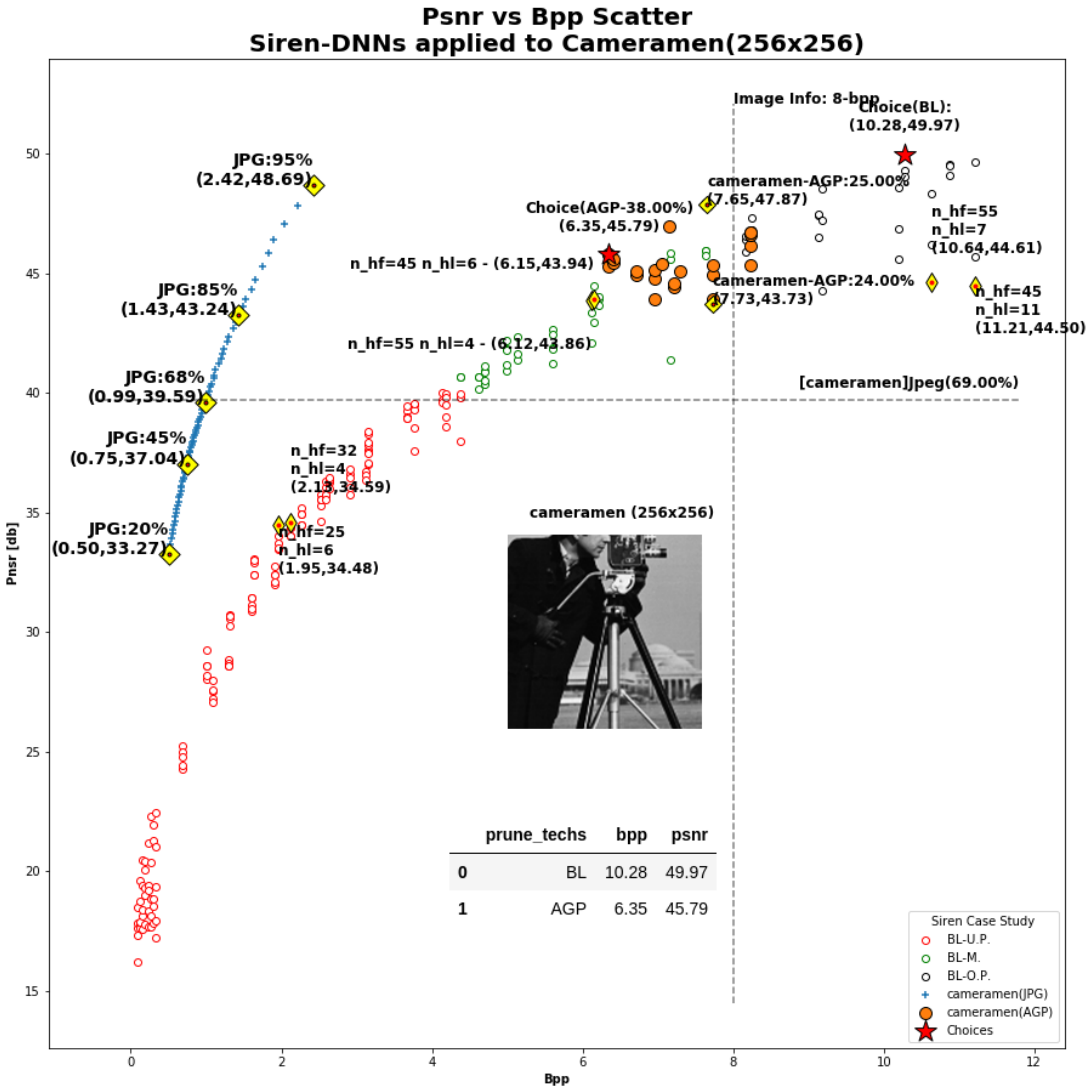
Identified and Applied for learning Pruned Deep Net Models

Pruning Technique Workflow

For pruning Siren based Deep Models via AGP pruning method, and after having performed some random trials, via **Random Search Approach** to indetify most suitable hyper-params, we follow the subsequent training strategy:

- We determine the degree of pruning for each layer depending on the **Sensitivity Level Analysis** done ahead of pruning time.
- We let the **Frequency** Hyper-parameter to be picked up from: $\{50, 100, 200\}$ possible choices, essential to let net model
- We follow a generically **Suggested Heuristic in literature**, where we avoid pruning first and last layers in order to let performance to not degradate too much;

AGP Pruning Technique: Psnr [db] vs. Bpp Scatterplot



Range Linear Quantization

Quantization Aware Training Technique for Compressing Deep Nets

Linear Range Quant

In order to satisfy quantizing requirements we made use of Linear Range Quantization technique, which is a Quant Aware Training method developed by Benoit et al, (2018), whose main features are:

- **quantization scheme** that relies only on integer arithmetic to approximate the floating-point computations in a neural network;
- **Training Procedure** that simulates the effect of quantization helps to restore model accuracy to near-identical levels as the original.
- The Algorithm uses exponential moving averages to track activation ranges.

Linear Range Quant (2)

In order to satisfy quantizing requirements we made use of Linear Range Quantization technique, which is a Quant Aware Training method developed by Benoit et al, (2018), whose main features are:

- They requiring that thequantization scheme be anaffine mapping of integers q toreal numbers r, i.e. of the form:

$$r = S(q - Z) \tag{1}$$

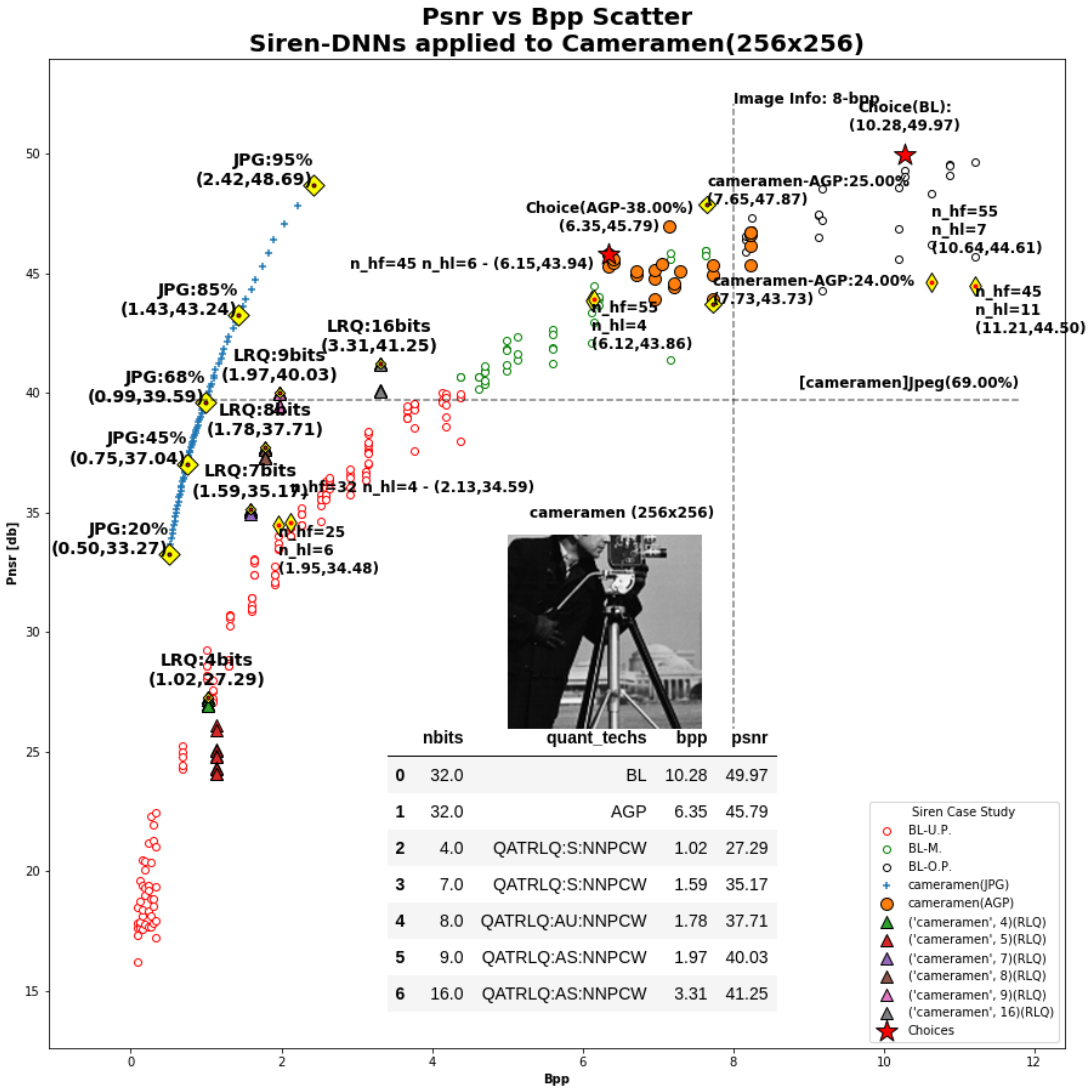
  for some constants S and Z quant-parameters.
- where, The constant S (for "scale") is an arbitrary positive realnumber. It is typically represented in software as a floating-point quantity, like the real values r.
- The constant Z (for "zero-point") is of the same type as quantized values q, and is in fact the quantized value q corresponding to the real value 0.
- Finally, The motivationfor this requirement is that efficient implementation of neural network operators often requires zero-padding of arraysaround boundaries.

Linear Range Quant Workflow

For Quantizing Siren based already pruned Deep Models via **LR QAT Compressing Method**, and after having performed some random trials, via **Random Search Approach** to indetify most suitable hyper-params, we follow the subsequent training strategy:

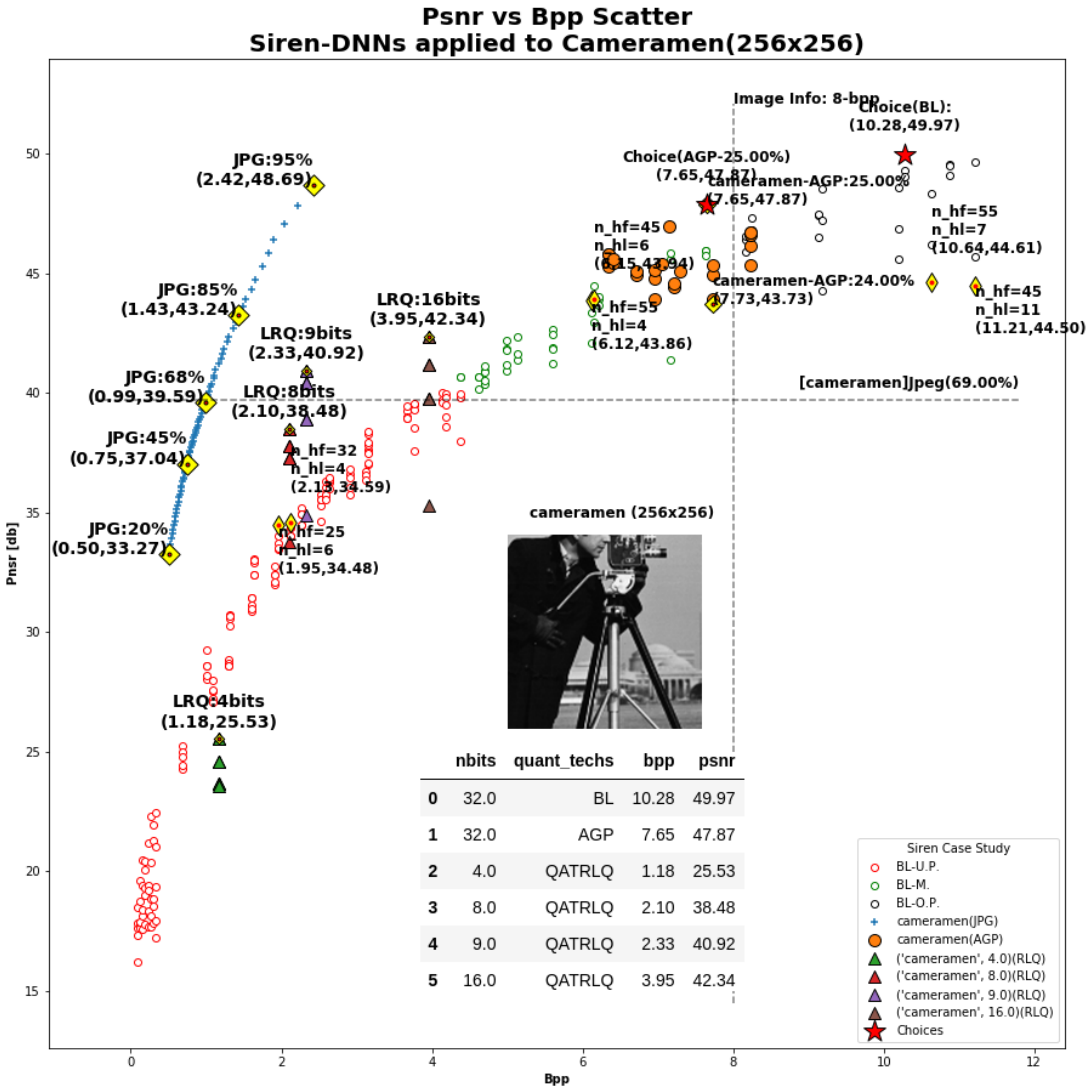
- We let **quantization** levels to be in the interval represented from the following values 4, 8, 9, 16;
- We let the **frequency of updates** to modify quantizing parameters to be 2, 5, 10, 25 from trials to trails;
- We let the Learning Rate to be reduced from $1e-4$ to $7.5e-5$ and even a lower value as $5.5e-5$;

Linear Range Quantization: Cameramen Pnsr [db] vs Bpp Scatter Plot



	nbits	quant_techs	bpp	psnr
0	32.0	BL	10.28	49.97
1	32.0	AGP	6.35	45.79
2	4.0	QATRLQ:S:NNPCW	1.02	27.29
3	7.0	QATRLQ:S:NNPCW	1.59	35.17
4	8.0	QATRLQ:AU:NNPCW	1.78	37.71
5	9.0	QATRLQ:AS:NNPCW	1.97	40.03
6	16.0	QATRLQ:AS:NNPCW	3.31	41.25

Linear Range Quantization: Cameramen Pnsr [db] vs Bpp Scatter Plot (2)

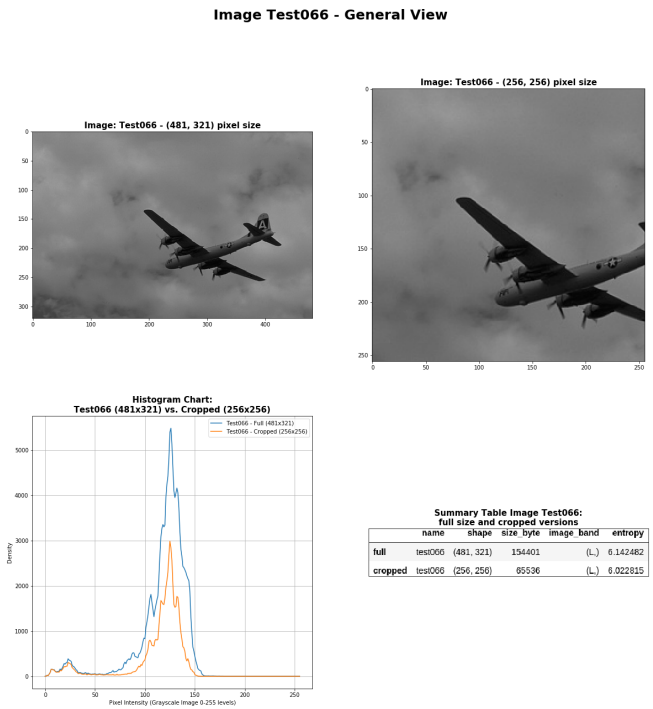


	nbits	quant_techs	bpp	psnr
0	32.0	BL	10.28	49.97
1	32.0	AGP	7.65	47.87
2	4.0	QATRLQ	1.18	25.53
3	8.0	QATRLQ	2.10	38.48
4	9.0	QATRLQ	2.33	40.92
5	16.0	QATRLQ	3.95	42.34

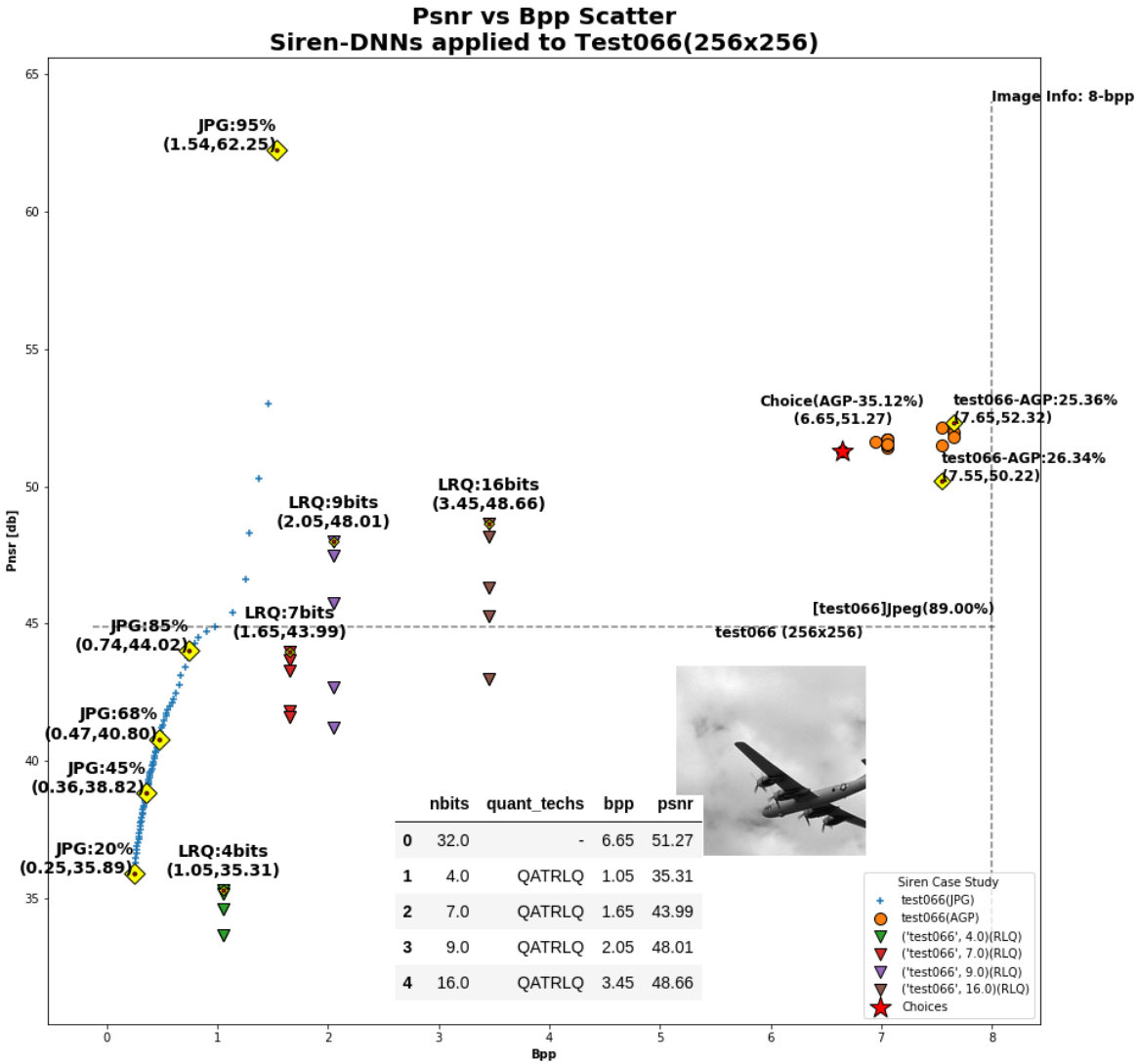
Airplane Test066 BSD Target Image

Other Trials onto A new Image

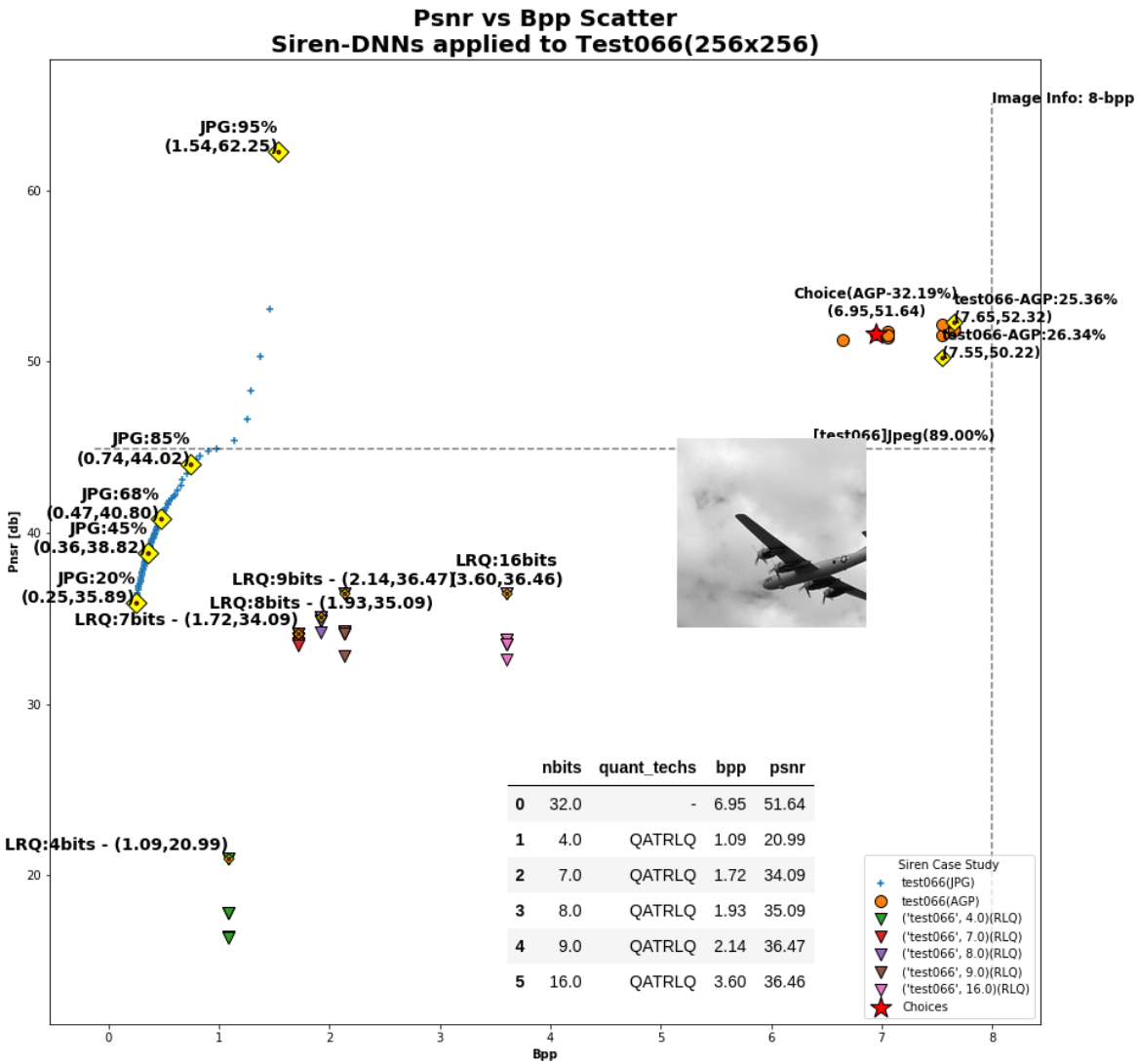
Shortly, presenting results when another target image is employed



Linear Range Quantization: Test066 Pnsr [db] vs Bpp Scatter Plot



Linear Range Quantization: Test066 Pnsr [db] vs Bpp Scatter Plot (2)



	nbits	quant_techs	bpp	psnr
0	32.0	-	6.95	51.64
1	4.0	QATRLQ	1.09	20.99
2	7.0	QATRLQ	1.72	34.09
3	8.0	QATRLQ	1.93	35.09
4	9.0	QATRLQ	2.14	36.47
5	16.0	QATRLQ	3.60	36.46

Conclusions

Last Remarks learnt from compressing Siren Based Nets

Conclusions and Remarks

The major remarks we have found while appling Deep Networks Compression Techiniques such as pruning-based approaches and quantizing-based approaches are the following. In particular, speaking about adopted Pruning Algorithm, which is Automated Gradual Pruning (AGP) we notice that:

- Models seem to need a reasonable extensive number of epochs to improve over brain damage, i.e. weights prunining
- It result a better choice to let models to be pruned when an adequate large number of steps last between two consecutive wheight pruning
- item We were able to identify Hyper-params combinations for AGP pruning method that
  - outperform baseline architectures, i.e. plain Siren-based Nets.
  - satisty constraints of being at least lower in size w.r.t Learned Image as well as still performing better than some comparing in size plain Siren-nets.
- However, when looking at performance that should be compared to Jpeg compression results we still notice that the existing gap between values related to Bpp score are too large to justify just such a

Conclusions and Remarks (2)

The major remarks we have found while appling Deep Networks Compression Techiniques such as pruning-based approaches and quantizing-based approaches are the following. Where, talking about adopted Quant Procedure, which is Linear Range quant-aware training we claim that:

- We have quantized starting from already trained and pruned models, obtaining in best cases that:
  - I wide reduction in Bpp score corresponded a drammaticaly reduction in Pnsr values
  - Still we have retrieved quantized model instances that show an acceptable Psnr score value w.r.t both Jpeg compressed counterpart, and even an indisputable results wr.t plain Siren-based models.
- However, Learning Rate Hyper-param was crucial for identifying promising Linear Range Quant Configurations, and lower Learning Rate values seems to outperform larger ones.
- Differently from AGP pruning technique, we observed that for training by means of quant awere procedures as LRQ:
  - it was better to increase frequency with which updating learnable parameters to let quantization to take place

Future Works

Suggestions for widening current intial analyses carried out

Future Works: Other Pruning Techniques

As we already mentioned, NN Pruning Techniques can reduce the parameter counts of training networks by over 90%, decreasing **storage requirements** and improving **computational performance of inference**, without compromising accuracy. So, Other availabe techniques we can adopt are:

- The **Lottery Ticket Hypothesis** by Jonathn Frankle et al. (March 2019), where the main reason for using it are:
  - They found out standard **pruning techniques uncover trainable subnetworks**, i.e. winning tickets, from FCs and CNNs.
  - The **winning tickets**, i.e. best initial params intialization, lead to connections having initial weigths that make training particularly effective;
  - They show in their paper that there consistently exist **smaller subnetworks** that when trained from the start are able to learn at least as fast as their larger counterpart while reaching similar test accuracy.
  - It supports both **one-shot** and **iterative-pruning** behaviour, making it suitable in a variety of context when hardware and time constraints arise. Furthermore, LTH follows **unstructured-pruning heuristics**, focusing on reaching sparisiy.