

Evaluating the Use of Interpretable Quantized Convolutional Neural Networks for Resource-Constrained Deployment

Harry Rogers¹^a, Beatriz De La Iglesia¹^b, Tahmina Zebin²^c

¹*School of Computing Science, University of East Anglia, UK*

²*School of Computer Science, Brunel University London, UK*

{Harry.Rogers, b.iglesia}@uea.ac.uk, tahmina.zebin@brunel.ac.uk

Keywords: Class Activation Maps, Deep Learning, Quantization, XAI.

Abstract: The deployment of Neural Networks on resource-constrained devices for object classification and detection has led to the adoption of network compression methods, such as Quantization. However, the interpretation and comparison of Quantized Neural Networks with their Non-Quantized counterparts remains inadequately explored. To bridge this gap, we propose a novel Quantization Aware eXplainable Artificial Intelligence (XAI) pipeline to effectively compare Quantized and Non-Quantized Convolutional Neural Networks (CNNs). Our pipeline leverages Class Activation Maps (CAMs) to identify differences in activation patterns between Quantized and Non-Quantized. Through the application of Root Mean Squared Error, a subset from the top 5% scoring Quantized and Non-Quantized CAMs is generated, highlighting regions of dissimilarity for further analysis. We conduct a comprehensive comparison of activations from both Quantized and Non-Quantized CNNs, using Entropy, Standard Deviation, Sparsity metrics, and activation histograms. The ImageNet dataset is utilized for network evaluation, with CAM effectiveness assessed through Deletion, Insertion, and Weakly Supervised Object Localization (WSOL). Our findings demonstrate that Quantized CNNs exhibit higher performance in WSOL and show promising potential for real-time deployment on resource-constrained devices.

1 INTRODUCTION


As technology progresses, the field of Artificial Intelligence (AI) continues to advance enabling more sophisticated solutions. Typically, for automated image-processing tasks, Vision Transformers (ViTs) and variants of deep Convolutional Neural Networks (CNNs) are commonly employed for object classification and detection. However, as researchers propose increasingly complex architectures, these architectures are becoming larger and more computationally intensive, posing challenges for training and inference time. Moreover, the hardware limitations of resource-constrained devices further complicate the deployment of these larger networks. As a result, researchers have turned to compression methodologies to address these challenges.


To tackle the need for efficient model deployment and inference on devices with limited resources, quantization has emerged as a promising technique.


By reducing the memory footprint and computational requirements of Neural Networks, Quantized networks offer a viable solution that balances model size and inference speed while maintaining acceptable levels of accuracy.

Quantization involves converting Neural Network weights from 32-bit floating-point to 8-bit integer representation, reducing memory usage and computation requirements without sacrificing accuracy. By leveraging the benefits of quantization, larger and more complex models can be deployed on resource-constrained devices. The reduced memory footprint and faster inference time make Quantized CNNs particularly suitable for real-time applications

In this paper, we explore the application of quantization on CNNs, aiming to leverage the benefits of reduced memory usage and faster inference times using publicly available architectures from PyTorch (Paszke et al., 2019). We investigate the impact of quantization on CNN activations, CNN accuracy, CNN size, and inference speed. Our goal is to provide insights into the trade-offs and advantages of Quantized networks for efficient deployment on resource-constrained devices in a Weakly Supervised Object

^a <https://orcid.org/0000-0003-3227-5677>

^b <https://orcid.org/0000-0003-2675-5826>

^c <https://orcid.org/0000-0003-0437-0570>

Localization (WSOL) task.

Within literature, there has not been a comparison drawn between Quantized and Non-Quantized CNNs to identify regions of interest. Proposed in this paper is a Quantization Aware eXplainable AI (XAI) pipeline to visualize and compare regions of interest between Quantized and Non-Quantized CNNs using Class Activation Maps (CAMs). CAMs are computed using the last convolutional layer in a specified CNN. In this paper we use EigenCAM (Bany Muhammad and Yeasin, 2021) as the method for CAM computation is gradient-free, allowing for activation only inference. Using CAMs from Quantized and Non-Quantized CNNs, Root Mean Squared Error (RMSE) is applied to identify differences between CAMs. After taking the top 5% where the CAMs are different, we create a subset of images to test. Using this subset, we apply several metrics to activations to identify differences between Quantized and Non-Quantized CNNs. To evaluate CAM effectiveness XAI metrics Deletion and Insertion are applied with a WSOL task. To facilitate the contrast between networks we use the ImageNet dataset (Russakovsky et al., 2015). Our usage of WSOL is intended to allow for resource-constrained devices to have a lightweight object detector on board without the need for training a more complex model that cannot be deployed due to its size.

The core contributions of this paper are as follows:

1. Comparison of Quantized CNNs and Non-Quantized CNNs using XAI to identify different regions and features utilized for image classification.
2. Comparison of Quantized CNNs and Non-Quantized CNNs in a Weakly Supervised Object Localization task, considering the feasibility in real-time usage.
3. Comparison of Quantized and Non-Quantized activation blocks using several statistical metrics, to identify similarities and differences.

The remainder of the paper is organized as follows. Section 2 presents related literature on quantization methods, with a focus on the application of XAI methods. It also provides a review of XAI CAM methods and their evaluations. Section 3 presents the details of the Quantization Aware XAI pipeline, including CAM evaluation metrics, activation metrics, classification metrics, inference speed test information, and ImageNet baselines with the applied quantization methodologies. The performance of CAM explanations is reported and evaluated in Section 4. Finally, Section 5 presents the conclusions of the paper and outlines future work.

2 RELATED LITERATURE

Quantized Neural Networks have been combined with XAI with differing methodologies to enable more efficient deployment and use of Neural Networks. The combination of CAMs and Quantized Neural Networks has been identified in literature but not fully explored as we do in this paper.

2.1 Quantization and Explainable Quantization

Quantization was originally proposed to be used for faster inference times for mobile devices with real-time deployment. Jacob et al. (2017) was the first to propose Quantization Aware Training (QAT) which involves training Neural Networks directly with low-bit Quantized weights and activations. QAT maintains the performance of higher bit Non-Quantized weights and activations whilst achieving inference speed-ups as well as a lower memory usage. Quantization is achieved through the usage of straight-through estimation. The method approximates gradients during backpropagation to account for the reduction of information from quantization, converting 32-bit floating point values into lower bit values.

Following that, there have been many methodologies to achieve a more efficient inference using quantization with the aim of minimising the error from quantization (Gholami et al., 2021; Ghimire et al., 2022; Liang et al., 2021). Various optimization methods have been explored to enhance the performance of Neural Networks on hardware with limited computational power. These methods include hardware-aware training, which aims to improve network efficiency on specific hardware platforms. Additionally, zero-shot quantization techniques have been employed to convert weights and activations without the need for retraining or fine-tuning, similar to post static quantization. These approaches collectively contribute to improving Neural Network efficiency and adaptability on resource-constrained hardware. However, each method may be good on a specific test case, but may have downsides when compared to each other; there seems too not be a universal best fit.

Since the adoption of Quantized networks, developments have been made to make networks more efficient with other compression methods. For example, pruning by removing weights has been used in conjunction with quantization (Xu et al., 2020; Liu et al., 2020). Pruning methods can be structured or unstructured, where structured refers to removing entire nodes from a network and unstructured refers to setting specific parameters to zero, making the net-

work sparse. Both methods make networks lighter and can have speed-ups whilst retaining accuracy. Pruning has been used with Quantized networks and XAI with the usage of Deep Learning Important Features (DeepLIFT) (Shrikumar et al., 2019) to make models lighter and more efficient (Sabih et al., 2020). DeepLIFT operates by assigning importance scores to individual input features based on the difference they make in the networks activations compared to a reference baseline. The usage of XAI is an excellent way to identify weights in a network that could be pruned as they serve no purpose for inference. However, Sabih et al. (2020) could have a comparison to the Non-Quantized counterpart to show key differences to highlight improvements. Furthermore, usage of other XAI methods following this should be explored.

Methods such as guided backpropagation with Quantized networks have been explored in the literature. For example, Zee et al. (2022) compares a Quantized CNN to a Non-Quantized baseline CNN, pruned CNN, ablated CNN, and a pruned and Quantized CNN. The usage of guided backpropagation results in a CAM for each CNN, however, there are no metrics used to evaluate the CAMs with regards to how effective each CAM is at representing what the CNN actually uses for prediction. Furthermore, there is only one Quantized CNN tested with QAT.

Similar research has been completed with image retrieval tasks with explainable Quantized CNNs (Ma et al., 2023). A novel quantization methodology, Deep Progressive Asymmetric Quantization, is proposed and CNNs are visualised with CAMs. The CAM method utilised is not evaluated with metrics like Zee et al. (2022). There also is no comparison to a Non-Quantized CNN counterpart to identify what baselines could be achieved.

From our review of quantization, it can be stated there are many methodologies for quantization but not all are openly available online. A key part of research is being able to replicate results and data, therefore for this paper we will use pretrained weights that are publicly available from PyTorch. Comparing quantization methods to the Non-Quantized counterpart has also not been fully explored, with little research on publicly available datasets. Therefore, comparisons between QAT and Post Static quantization of CNNs using the ImageNet dataset will be explored in this paper.

2.2 Class Activation Maps and Evaluations

Within XAI there are methods to identify regions of interest from CNN predictions. These methods

visualise gradients or activations from CNN predictions by taking the last layer of the architecture and computing regions of interest called Class Activation Maps (CAMs). Previously, gradient based methods have been used with GradCAM, GradCAM++, Full-Grad, with success (Selvaraju et al., 2019; Chattopadhyay et al., 2018; Srinivas and Fleuret, 2019). More recently, gradient-free methodologies have been used to account for networks that can have negative or non-differentiable values within gradients.

AblationCAM was one of the first gradient-free methods proposed (Ramaswamy et al., 2020). Ablation-CAM measures activations by measuring the dropout. If the output drops by a large margin, then that activation is important and receives a higher weight. Ablation-CAM is reported to perform more effectively for CNNs with the ImageNet dataset when compared to using GradCAM. Methods like EigenCAM (Bany Muhammad and Yeasin, 2021) have also been proposed for gradient-free CAMs. EigenCAM returns the first principal component of the activations in the network. These correspond with the dominant object in the image. Similarly, to Ablation-CAM, EigenCAM reports new baselines for the ImageNet dataset. However, EigenCAM is much more lightweight and efficient than AblationCAM. Finally, we look at ScoreCAM (Wang et al., 2020). ScoreCAM uses a two-phase system, in phase one, activation maps are collected from the CNN using up-sampling. These activations then work as a mask on the original image to obtain the forward pass for each target. In phase two there is a point wise manipulation of these masks using a loop that is the same size as the number of activation maps. The result is then a linearly generated combination of the outputs from phase one and two. This method is slower than EigenCAM (Bany Muhammad and Yeasin, 2021) but is much faster than Ablation-CAM (Ramaswamy et al., 2020).

Several metrics can be employed to assess the effectiveness of CAMs in explaining the regions used by a CNN. In this context, we focus on three metrics: Deletion, Insertion, and WSOL (Petsiuk et al., 2018). Deletion and Insertion are two complementary metrics commonly used to evaluate the effectiveness of explanations. Deletion quantifies the change in classification confidence when different regions of an image are removed. Insertion measures the confidence change resulting from adding regions, either with surrounding noise or in isolation from any surrounding context. WSOL is a simple approach that involves calculating the Intersection over Union (IoU) of regions of high interest with labeled objects.

From the gradient-free methods we have re-

viewed, EigenCAM will be used as this method stands out as the most suitable choice for our specific application due to its lightweight and efficient nature. As we are using Quantized CNNs, the inference time will be considered. Therefore, the computation time for the CAM method should also be recorded as this has not been explored to be used as a real-time application. We will evaluate our CAMs for both Quantized and Non-Quantized CNNs with Deletion, Insertion and WSOL, further details are in section 3.

3 QUANTIZATION AWARE XAI PIPELINE

Our quantization aware pipeline has multiple steps to rigorously compare Quantized CNNs against their Non-Quantized counterparts. This comprehensive evaluation process ensures that we thoroughly assess the performance, efficiency, and trade-offs associated with quantization techniques in the context of deep learning models.

Figure 1 provides a visual representation of the various stages involved in our quantization aware pipeline. We begin with collecting weights from publicly available libraries (PyTorch) to generate CAMs for the ImageNet dataset.

3.1 CAM Generation

To determine if there are differences between the regions of interest used between Quantized and Non-Quantized CNNs we utilize CAMs. Specifically, EigenCAM which computes the CAM for both Quantized and Non-Quantized networks. As previously mentioned, EigenCAM is computationally lighter and faster than other gradient-free methods therefore this method is applied.

EigenCAM has been adapted in our work to be able to work with Quantized or Non-Quantized (8-bit or 32-bit) weights by checking the type of activation passed. Once this check is complete the activations are computed with Singular Value Decomposition (SVD). SVD is defined in Equation (1), where M is the activation matrix, U represents the orthogonal matrix of left singular vectors, Σ represents the diagonal matrix of singular values, and V^t represents the transpose of the orthogonal matrix of right singular vectors.

$$M = U\Sigma V^t \quad (1)$$

3.2 CAM Evaluation Metrics

To create our subset from the validation dataset we first have to generate all 50,000 CAMs from each validation image. Once these are computed we compare Quantized CAMs to Non-Quantized CAMs using Root Mean Squared Error (RMSE). RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - f_i)^2} \quad (2)$$

where n is the total number of pixels in each image, d_i is the grayscale value of the i -th pixel in a Quantized CAM, and f_i is the grayscale value of the corresponding i -th pixel in a Non-Quantized CAM. RMSE is used to select CAM pairs (Quantized and Non-Quantized) that are different to each other. The highest scoring 5% are selected as these have the largest difference. We are using the top 5% as this equates to 2,500 images creating a subset from the original 50,000 images. We report in section 4 the average RMSE across the entire validation set and the average RMSE across the top 5% selected by this method.

After creating the RMSE subset, Deletion and Insertion are calculated from each CAM. The confidence of the CNNs inference will be recorded with Deletion and Insertion increasing by 1% until the entire image is deleted or inserted. After plotting the confidence values against the amount of image deleted or inserted, the Area Under the Curve (AUC) is calculated using the Trapezoidal Rule:

$$AUC = \frac{h}{2} [y_0 + 2(y_1 + y_2 + y_3 + \dots + y_{n-1}) + y_n] \quad (3)$$

where y is the prediction confidence, n is equal to the number of plotted points, and h is equal to the increase in Deletion or Insertion change. Deletion scores that are lower are better and Insertion scores that are higher are better.

After Deletion and Insertion, WSOL is computed with each CAM. Intersection over Union (IoU) is used within WSOL and can be calculated as the area of overlap divided by the area of union using the ground truth boxes and prediction boxes from the CAM. Mathematically it is defined as:

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (4)$$

where A and B are the prediction and ground truth bounding boxes, respectively.

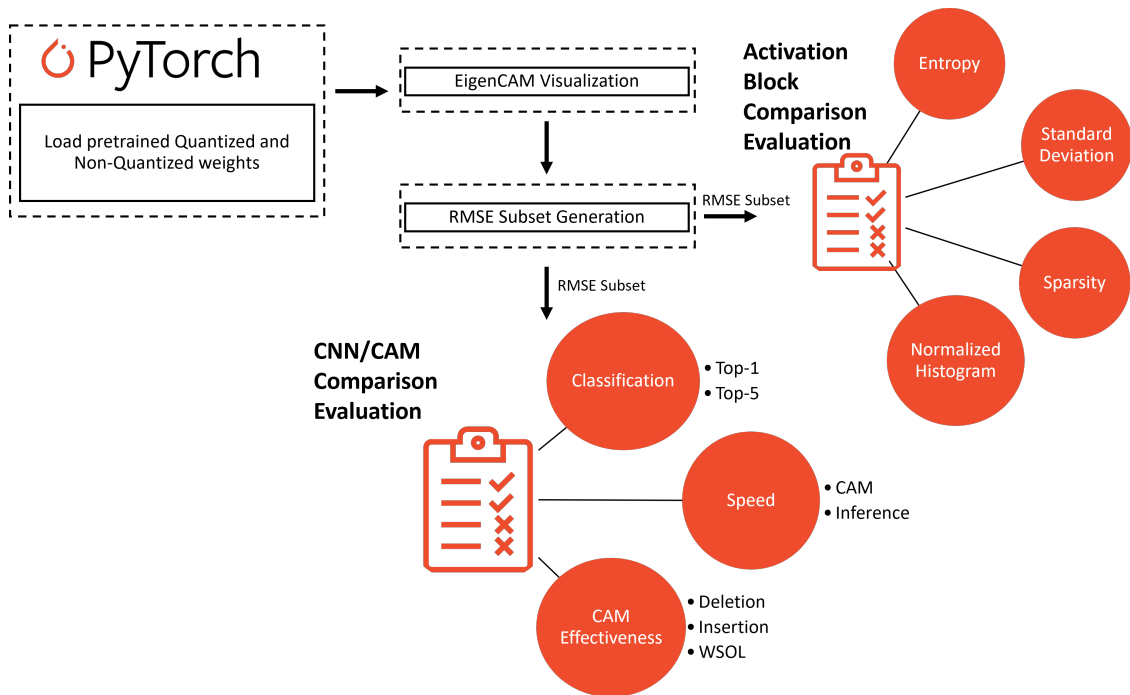


Figure 1: Pipeline for comparison of Quantized and Non-Quantized CNNs.

3.3 Activation Metrics

As we are using a gradient-free CAM method, we need to compare the activations used not only with CAM metrics but with statistical tests. Therefore we have decided to use: Entropy (\mathcal{S}), Standard Deviation (σ), Sparsity (λ), and normalized activation histograms. These metrics are designed to provide insights into the characteristics of the activations and their contribution to the CNNs predictions. The average score for each metric across the subset is reported in Section 4.

- **Entropy:** Entropy is a measure of the uncertainty in the distribution of activations. A higher entropy will create a more uncertain prediction meaning the CNN is less confident in its prediction. A lower entropy will show a smaller distribution meaning a CNN is more confident in its prediction. Entropy will not directly help with the WSOL task, however, it will help with explaining the effectiveness of CAMs as a higher entropy will cause less certain CAMs to be generated where Deletion and Insertion may be not as effective. Whereas a lower entropy will generate more certain CAMs that will have regions that are more important drawn causing Deletion and Insertion to be effective. Entropy can be defined as follows:

$$\mathcal{S} = -\sum_i P(i) \log_2(P(i) + \epsilon) \quad (5)$$

Where $P(i)$ represents the probability of the i th element in the probability distribution P , ϵ is a small constant of $1e^{-10}$ to prevent taking the logarithm of zero.

- **Standard Deviation:** The standard deviation shows the range of neuron values used in the activation. A high standard deviation would generate a wide range of activation neurons, this means that more activation neurons respond to different input patterns meaning a more used activation block. Whereas, a lower standard deviation would do the opposite with lower usage of the activation block. When considering the WSOL task, a higher standard deviation could lead to larger areas that cluster around objects of interest as a higher distribution would likely cause a wider activation pattern. Standard deviation can be defined as follows:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}(x))^2} \quad (6)$$

where n is the number of activation neurons, x_i represents the i th neuron in the activation block.

- **Sparsity:** Sparsity is used to identify neurons that are not activated that have the value of 0. These neurons could be pruned to make the CNN lighter. Sparse activations result in faster execution time and reduced memory requirements. Sparsity can

be defined as follows:

$$\lambda = \frac{\text{No. of Zero Activation Neurons}}{\text{Total No. of Activation Neurons}} \times 100 \quad (7)$$

- **Histograms:** Histograms provide a visual representation of the distribution of activation values. This will show the frequency of values and the overall spread of activations. The average activation will be computed, normalized, and plotted as a histogram across 256 bins. Using the Sørensen–Dice Coefficient of the two histograms the similarity can be computed. The Sørensen–Dice Coefficient is defined as follows:

$$\text{Dice} = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (8)$$

Where A and B are the activations.

3.4 Classification Metrics

Top-1 and Top-5 accuracy are used within this paper to describe how accurate CNNs are when using the ImageNet dataset.

Accuracy can be defined as the measure of how correctly a CNN predicts the class, shown in Equation (9). Accuracy refers to the percentage of images in the dataset that are correctly classified by the CNN model.

$$\text{Accuracy} = \frac{\text{No. of correctly classified images}}{\text{Total No. of images}} \times 100 \quad (9)$$

When considering Top-1 accuracy, the CNN predicts the correct class label for the given image with the highest confidence among all the possible classes predicted. On the other hand, Top-5 accuracy takes into account a more lenient criterion. It means that the CNN predicts the correct class label for the given image within the top five most confident predictions.

3.5 Inference Speed Test

The average inference time for each CNN, Quantized and Non-Quantized, against the entire ImageNet validation dataset is recorded. Images are passed through a CNN running on a PC with an Intel Core i7-1800H CPU and the inference time is recorded. EigenCAM computation time will also be recorded to identify any potential speed ups from Quantized weights. The times for CNN inference and EigenCAM are recorded in seconds and are reported in Section 4.

3.6 ImageNet Baselines

In this section, we have reported the Top-1 and Top-5 accuracy on ImageNet as baselines for easy reference. These will be compared too in Section 4.

The models we have used in this work include MobileNetV2 (Sandler et al., 2019), MobileNetV3 (Howard et al., 2019), ResNet18 (He et al., 2015), and ShuffleNetV2 (Ma et al., 2018). To ensure the reproducibility of our work, we utilized publicly available weights for the models from PyTorch. PyTorch provides both Quantized alternatives and original weights. MobileNetV2 and MobileNetV3 use QAT, while ResNet18 and ShuffleNetV2 are converted using Post Static quantization. We have chosen lightweight versions of each architecture as this is for a resource-constrained device.

Table 1 presents the Non-Quantized networks' Top-1 accuracy, Top-5 accuracy, and file size in megabytes (MB) on the ImageNet dataset. MobileNetV3 achieves the highest Top-1 and Top-5 accuracy with 74.042% and 91.340% respectively. MobileNetV2 shows slightly lower scores, with a Top-1 accuracy of 71.878% and a Top-5 accuracy of 90.286%. ResNet18 and ShuffleNetV2 perform comparatively lower, with Top-1 accuracies of 69.758% and 60.552%, and Top-5 accuracies of 89.078% and 81.746% respectively. The file sizes are 13.598 MB for MobileNetV2, 21.114 MB for MobileNetV3, 44.661 MB for ResNet18, and 5.282 MB for ShuffleNetV2.

In Table 2, we present the Quantized networks' Top-1 accuracy, Top-5 accuracy, and file size on the ImageNet dataset. MobileNetV3 maintains its position as the top-performing model with a Top-1 accuracy of 73.004% and a Top-5 accuracy of 90.858%. MobileNetV2 closely follows with a Top-1 accuracy of 71.658% and a Top-5 accuracy of 90.150%. Notably, MobileNetV2 demonstrates the best retention scores, with the smallest decrease in Top-1 accuracy of 0.220% and Top-5 accuracy of 0.136% compared to the Non-Quantized version. ResNet18 achieves a Top-1 accuracy of 69.494% and a Top-5 accuracy of 88.882%, while ShuffleNetV2 exhibits a Top-1 accuracy of 57.972% and a Top-5 accuracy of 79.780%. The file sizes of the Quantized models are significantly reduced, with MobileNetV2, ResNet18, and ShuffleNetV2 shrinking by approximately 3.9 times to 3.423 MB, 11.238 MB, and 1.501 MB respectively. However, the file size of MobileNetV3 increases to 21.554 MB.

Table 1: Network Top-1 and Top-5 Accuracy (ImageNet).

Model	Top-1 (%)	Top-5 (%)	Size (MB)
MobileNetV2	71.878	90.286	13.598
MobileNetV3	74.042	91.340	21.114
ResNet18	69.758	89.078	44.661
ShuffleNetV2	60.552	81.746	5.282

Table 2: Quantized Network Top-1 and Top-5 Accuracy (ImageNet).

Model	Top-1 (%)	Top-5 (%)	Size (MB)
MobileNetV2	71.658	90.150	3.423
MobileNetV3	73.004	90.858	21.554
ResNet18	69.494	88.882	11.238
ShuffleNetV2	57.972	79.780	1.501

4 RESULTS

To create the RMSE subset, EigenCAM was used to compute each CAM for each validation image for all Quantized and Non-Quantized CNNs. In Table 3 the average RMSE for each CNN is reported, it can be stated that the average RMSE across the 50,000 images is the highest with the QAT MobileNetV3 with 96.926, followed by the Post Static ShuffleNetV2 with 90.742, then the QAT MobileNetV2 with 79.541, and finally the Post Static ResNet18 with 29.220. When taking the average from the top 5% RMSE values the scores increase to 144.449 with MobileNetV2, 167.484 with the MobileNetV3, 148.001 with the ResNet18, and 148.725 with the ShuffleNetV2.

From these results it could be concluded that quantization changes the regions of interest for networks whether they are retrained (QAT) or not (Post static). This means that Quantized CNNs are likely learning different features to classify objects within images across 50,000 validation images from ImageNet. This is further explored when considering the top 5% highest scoring RMSE values as these will be completely different regions of interest.

Table 3: EigenCAM RMSE.

Model	Average RMSE	Top 5% Average RMSE
MobileNetV2	79.541	144.449
MobileNetV3	96.926	167.484
ResNet18	29.220	148.001
ShuffleNetV2	90.742	148.725

The activation metrics: Entropy (S), Standard Deviation (σ), and Sparsity (λ) are reported in Table 4 and Table 5 for the Quantized, and Non-Quantized CNNs, respectively. These metrics serve as essential

indicators of the efficiency and effectiveness of the respective activation blocks within these networks.

When comparing Quantized and Non-Quantized CNNs there is promising signs that each Quantized CNN activation block is more efficient and effective. Each CNN tested has a significantly lower entropy score when using quantization. MobileNetV2 increases by 8.732, MobileNetV3 increases by 5.293, ResNet18 increases by 4.104, ShuffleNetV2 increases by 9.587 when going from Quantized to Non-Quantized. This means that Quantized CNNs are much more certain in predictions as the entropy is lower. The standard deviation scores decrease by 17.313, 10.838, 6.884, and 4.811 for the MobileNetV2, MobileNetV3, ResNet18, and ShuffleNetV2, respectively against the Non-Quantized CNNs. This means that Quantized activation blocks are more utilized, and are more certain with the subset we have generated. Finally, all Quantized CNNs, apart from MobileNetV3, have a larger sparsity meaning that inference on resource-limited devices will be faster as there are less computations to complete.

These results are likely due to the conversion of quantization as values have been essentially generalized. The quantized activation blocks will be more certain in predictions as the Quantized activations are approximating the Non-Quantized activations. As quantization has been explored, the approximation is also efficient as the sparsity within activation blocks is higher showing that the quantization methodology for both QAT and Post Static. Which can achieve further speedups from less computation being required at inference time.

The normalized average activation histograms are plotted in Figure 2, from the distributions plotted it can be seen that quantized and non-quantized activations are very similar. Each Sørensen–Dice Coefficient is shown on each plot, MobileNetV2 scored 98.126, MobileNetV3 recorded 95.689, ResNet18 achieved 99.521, and ShuffleNetV2 scored 98.466. This could be due to classification and localisation being similar for each CNN whether Quantized or Non-Quantized. However, these scores are very high and further testing is needed to identify a spatially aware normalization process to ensure activations spatial information from neurons can be preserved.

Table 4: Quantized Network Activation Scores.

Model	S^1	σ^2	λ^3 (%)
MobileNetV2	0.203	18.259	70.417
MobileNetV3	0.30	12.189	10.425
ResNet18	0.38	8.362	48.589
ShuffleNetV2	1.029	4.898	91.914

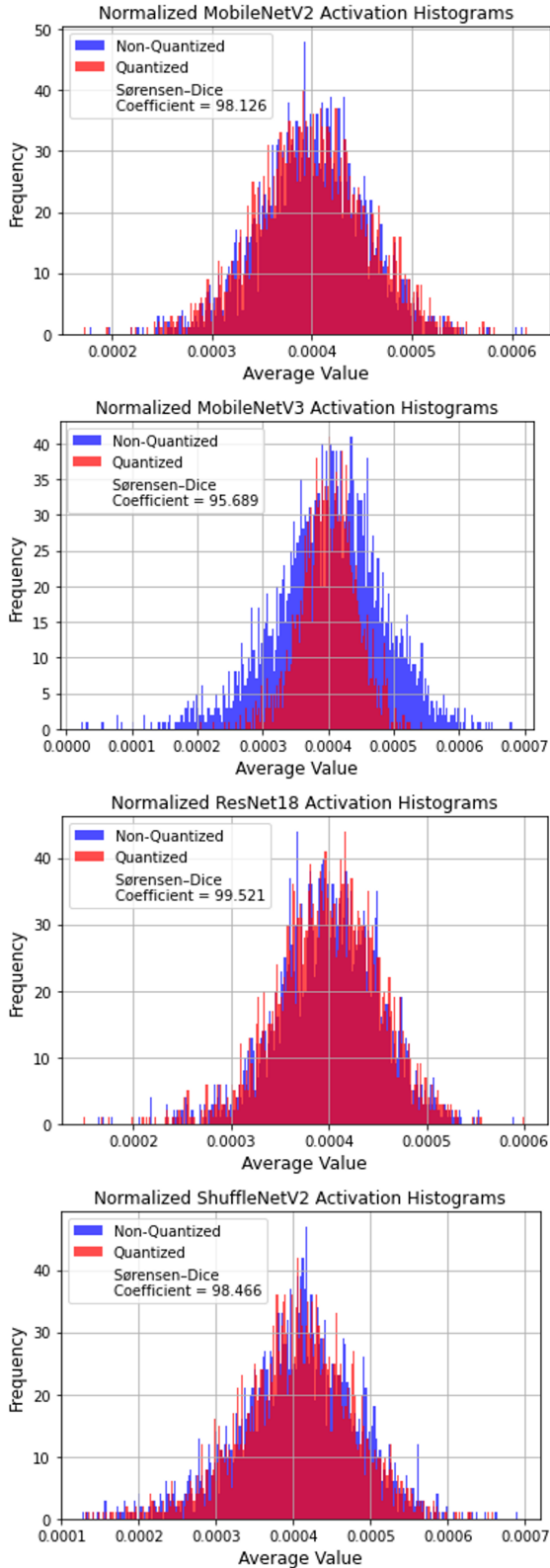


Figure 2: Normalized average Histogram distributions for activations in all CNNs.

Table 5: Non-Quantized Network Activation Scores.

Model	\mathcal{S}^1	σ^2	$\lambda^3(\%)$
MobileNetV2	8.931	0.946	55.791
MobileNetV3	5.596	1.351	16.686
ResNet18	4.484	1.478	45.339
ShuffleNetV2	9.991	0.087	91.877

To confirm that the Quantized and Non-Quantized CNNs are not inaccurate with the subset created, the Top-1 and Top-5 accuracies are recorded in Table 6, and Table 7, respectively. Analyzing Table 6, it can be observed that the Quantized CNNs, even with the created subset, retain high Top-1 and Top-5 accuracies when compared to their Non-Quantized counterparts. MobileNetV2 achieved a Top-1 accuracy of 68.932% and a Top-5 accuracy of 88.724%, MobileNetV3 achieved a Top-1 accuracy of 64.680% and a Top-5 accuracy of 86.840%, ResNet18 achieved a Top-1 accuracy of 63.640% and a Top-5 accuracy of 85.760%, and ShuffleNetV2 achieved a Top-1 accuracy of 53.263% and a Top-5 accuracy of 75.881%.

In contrast, Table 7 shows the Top-1 and Top-5 accuracies of the Non-Quantized networks. The QAT MobileNetV2 achieved a Top-1 accuracy of 69.212% and a Top-5 accuracy of 88.724%, the QAT MobileNetV3 achieved a Top-1 accuracy of 66.920% and a Top-5 accuracy of 87.840%, the post static ResNet18 achieved a Top-1 accuracy of 64.440% and a Top-5 accuracy of 86.120%, and the post static ShuffleNetV2 achieved a Top-1 accuracy of 55.938% and a Top-5 accuracy of 78.435%.

The differences when comparing Non-Quantized and Quantized CNNs for Top-1 and Top-5 accuracies for our subset are 0.28% Top-1, 0% Top-5 for MobileNetV2, whereas in Section 3 the baseline differences are Top-1 0.220% and 0.136% Top-5. For the MobileNetV3 our RMSE subset creates a 2.24% Top-1 difference, and 1% Top-5 change against 1.038% Top-1, 0.482% Top-5 in the baseline from Section 3. When using the ResNet18 our subset generates a difference of 0.8% Top-1, and 0.360% Top-5 whereas the baseline is 0.264% Top-1, and 0.196% Top-5. ShuffleNetV2 has a Top-1 difference in the subset of 2.675% and Top-5 of 2.554%, in the baseline is it 2.58% Top-1 and 1.966% Top-5. Therefore, our subset has a similar representation as the entire ImageNet validation dataset split. This shows that quantization is able to learn different features to approximate weights, whilst still being able to use different features for a similar performance. This is reinforced further when using the subset we have gener-

¹Entropy
²Standard Deviation
³Sparsity

ated showing the largest difference in features as the CAMs themselves are very different.

Table 6: Quantized Network Top-1 and Top-5 Accuracy (ImageNet Top 5% RMSE diff).

Model	Top-1 (%)	Top-5 (%)
MobileNetV2	68.932	88.724
MobileNetV3	64.680	86.840
ResNet18	63.640	85.760
ShuffleNetV2	53.263	75.881

Table 7: Network Top-1 and Top-5 Accuracy (ImageNet Top 5% RMSE diff).

Model	Top-1 (%)	Top-5 (%)
MobileNetV2	69.212	88.724
MobileNetV3	66.920	87.840
ResNet18	64.440	86.120
ShuffleNetV2	55.938	78.435

The inference speed and computation for Eigen-CAM with Quantized CNNs is reported in Table 8 and the Non-Quantized counterparts are in Table 9. Examining the Quantized CNNs, MobileNetV2 took 0.013 seconds on average for inference with CAM computation taking 0.010 seconds creating a total time of 0.023 seconds, MobileNetV3 was 0.065 seconds on average for inference and 0.007 seconds for CAM computation resulting in a total time of 0.072 seconds, ResNet18 classified in 0.023 seconds on average and CAMs were generated in 0.007 seconds equating to a 0.030 second total time, and ShuffleNetV2 took 0.073 seconds on average for inference but only 0.006 seconds for CAM computation meaning a total time of 0.079 seconds. Analyzing the Non-Quantized CNNs, the MobileNetV2 had an average inference time of 0.035 seconds with CAM computation taking 0.017 seconds therefore having a total time of 0.052 seconds. MobileNetV3 had an average classification time of 0.101 seconds and CAMs were generated in 0.016 seconds to then have a total time of 0.117 seconds, ResNet18 took 0.037 seconds for inference on average with a CAM computation time of 0.020 seconds resulting in a total time of 0.057 seconds, and the ShuffleNetV2 classified images in 0.088 seconds and CAMs were computed in 0.009 seconds totaling in 0.097 seconds.

When comparing Quantized CNNs to the Non-Quantized CNNs, MobileNetV2 demonstrated a remarkable improvement in inference speed, achieving a 2.69x faster performance compared to its Non-Quantized counterpart. MobileNetV3 followed with a 1.55x speedup, ResNet18 with a 1.61x speedup, and ShuffleNetV2 with a 1.21x speedup. These speedups

represent the improvements during inference only.

Moreover, when analysing the speed of the CAM generation, the Quantized networks also showcased notable improvements. MobileNetV2 achieved a 1.7x faster CAM generation, MobileNetV3 showed a 2.29x improvement, ResNet18 exhibited a 2.86x acceleration, and ShuffleNetV2 achieved a 1.5x boost.

Considering both inference speed and CAM generation, the Quantized networks demonstrate significant improvements. MobileNetV2 achieved a 2.26x overall speedup, MobileNetV3 achieved a 1.63x improvement, ResNet18 experienced a 1.9x acceleration, and ShuffleNetV2 saw a 1.23x boost.

These results show that using the Quantized CNNs for CAM computation and inference could be real-time with similar classification scores when compared to the Non-Quantized counterparts. This is for both QAT and Post Static methods of quantization.

Table 8: Quantised Network Speed Tests.

Model	Inference (s)	CAM (s)	Total (s)
MobileNetV2	0.013	0.010	0.023
MobileNetV3	0.065	0.007	0.072
ResNet18	0.023	0.007	0.030
ShuffleNetV2	0.073	0.006	0.079

Table 9: Non-Quantised Network Speed Tests.

Model	Inference (s)	CAM (s)	Total (s)
MobileNetV2	0.035	0.017	0.052
MobileNetV3	0.101	0.016	0.117
ResNet18	0.037	0.020	0.057
ShuffleNetV2	0.088	0.009	0.097

Table 10 displays the results of Quantized CAM evaluations and Table 11 showcases the results for the Non-Quantized CAM evaluations. The Quantized evaluations when considering Deletion for the MobileNetV3 and ResNet18 are lower than the Non-Quantized counterparts, showing that the regions highlighted are more important for predictions as when these are removed the confidence scores drop. The largest difference in Deletion is 0.534% when using the Quantized MobileNetV2, therefore it could be argued that quantized CNNs when using Deletion create more effective CAMs. Furthermore, the Quantized ResNet18 also has a higher Insertion score than the Non-Quantized counterpart showing the CAMs are more representative of the regions used for the ResNet18 architecture. Which in turn means that the Quantized ResNet18 is easier to visualize with Eigen-CAM.

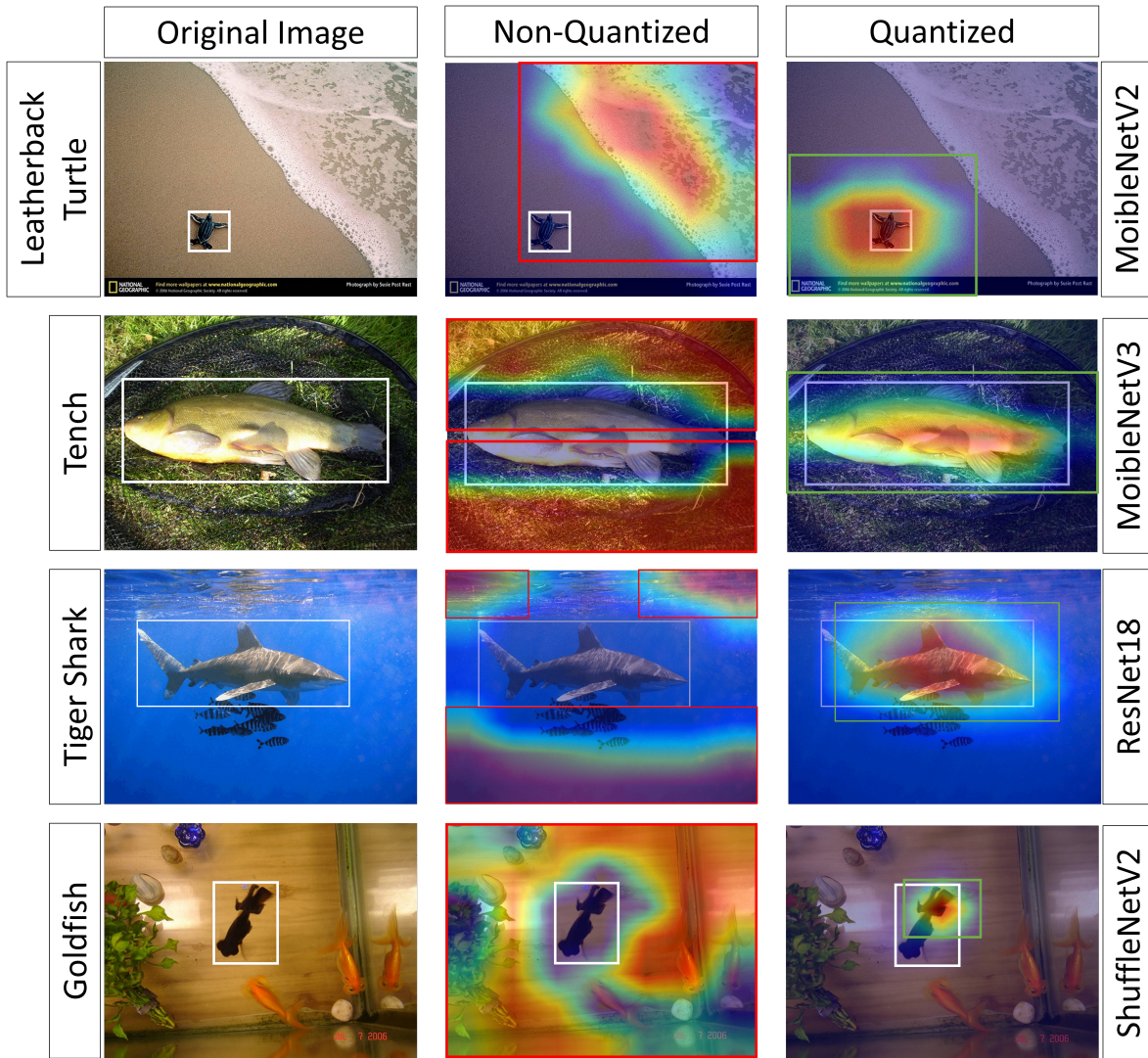


Figure 3: Comparison of Non-Quantized and Quantized CNNs, white boxes are the ground truth label, red boxes are the lower accuracy predictions, and green boxes are higher accuracy predictions.

Deletion and Insertion for all CNNs tested, Quantized and Non-Quantized, are causal where Deletion is smaller than Insertion showing that CAMs generated are effective explanations for each CNN.

Table 10: Quantized CAM Evaluation.

Model	Deletion (%)	Insertion (%)	WSOL (%)
MobileNetV2	0.984	10.589	30.717
MobileNetV3	1.529	32.032	26.896
ResNet18	0.496	9.244	27.942
ShuffleNetV2	0.289	2.613	28.845

The Quantized CNNs consistently outperformed their Non-Quantized counterparts in the WSOL

Table 11: Non-Quantized CAM Evaluation.

Model	Deletion (%)	Insertion (%)	WSOL (%)
MobileNetV2	0.450	12.658	30.701
MobileNetV3	7.560	44.659	26.790
ResNet18	0.531	10.464	27.840
ShuffleNetV2	0.117	0.127	28.740

task. Quantized CNNs scored 30.717%, 26.896%, 27.942%, and 28.845% for the MobileNetV2, MobileNetV3, ResNet18, and ShuffleNetV2, respectively. Whereas the Non-Quantized CNNs scored 30.701%, 26.790%, 27.840%, and 28.740% for the MobileNetV2, MobileNetV3, ResNet18, and ShuffleNetV2, respectively. The scores for the Quan-

tized CNNs were slightly higher, with increases ranging from 0.016% to 0.106% across different models with the MobileNetV2 increasing by 0.016%, MobileNetV3 increasing by 0.106%, ResNet18 increasing by 0.102%, and the ShufflenetV2 increasing by 0.105%. These results are not massive, however, they make sense. Quantization, as mentioned, is a method of generalization and therefore creates more generalizable CNNs.

For a visual comparison, Figure 3 illustrates the comparison between the Non-Quantized and Quantized CAMs. In the first column each image has a ground truth bounding box, in the second column is the Non-Quantized CAM, and in the third column is the Quantized CAM. The first row displays a prediction from MobileNetV2 on a Leatherback Turtle, the second row shows a prediction from MobileNetV3 on a Tench fish, the third row presents a prediction from ResNet18 on a Tiger Shark, and finally, the fourth row exhibits the ShuffleNetV2 with a prediction on a Goldfish. In each case, it is evident that the CAMs generated by the Quantized CNNs produce more concise bounding boxes for the respective images. This observation aligns with the reported WSOL IoU values.

5 CONCLUSION

To conclude, we have visualized and statistically compared activations from Quantized and Non-Quantized CNNs and identified differences within the activations themselves. Moreover, we have compared Quantized CNN activations in a WSOL task and compared the visualizations to their Non-Quantized CNN counterparts. Through this visualization, we have identified that Quantized CNNs utilize different features and regions for image classification using the ImageNet dataset. From this, we have demonstrated that Quantized CNNs exhibit higher performance in WSOL tasks and can be deployed in real-time using EigenCAM, and are statistically different. Thus, quantization should be considered in more academic papers, as it not only offers a more efficient network but also provides a more interpretable network in some cases.

For our future work, we will apply gradient-free methodologies to activations, incorporating ViTs, in order to explore the distinctions between Quantized and Non-Quantized ViT models.

Acknowledgements

This work is supported by the Engineering and Physical Sciences Research Council [EP/S023917/1].

REFERENCES

- Bany Muhammad, M. and Yeasin, M. (2021). Eigen-cam: Visual explanations for deep convolutional neural networks. *SN Computer Science*, 2:1–14.
- Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE.
- Ghimire, D., Kil, D., and Kim, S.-h. (2022). A survey on efficient convolutional neural networks and hardware acceleration. *Electronics*, 11(6).
- Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., and Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., and Adam, H. (2019). Searching for mobilenetv3.
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2017). Quantization and training of neural networks for efficient integer-arithmetic-only inference.
- Liang, T., Glossner, J., Wang, L., Shi, S., and Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403.
- Liu, J., Tripathi, S., Kurup, U., and Shah, M. (2020). Pruning algorithms to accelerate convolutional neural networks for edge applications: A survey. *arXiv preprint arXiv:2005.04275*.
- Ma, L., Hong, H., Meng, F., Wu, Q., and Wu, J. (2023). Deep progressive asymmetric quantization based on causal intervention for fine-grained image retrieval. *IEEE Transactions on Multimedia*, pages 1–13.
- Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In

- Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Petsiuk, V., Das, A., and Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models.
- Ramaswamy, H. G. et al. (2020). Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 983–991.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Sabih, M., Hannig, F., and Teich, J. (2020). Utilizing explainable ai for quantization and pruning of deep neural networks.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2019). Mobilenetv2: Inverted residuals and linear bottlenecks.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.
- Shrikumar, A., Greenside, P., and Kundaje, A. (2019). Learning important features through propagating activation differences.
- Srinivas, S. and Fleuret, F. (2019). Full-gradient representation for neural network visualization.
- Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. (2020). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25.
- Xu, S., Huang, A., Chen, L., and Zhang, B. (2020). Convolutional neural network pruning: A survey. In *2020 39th Chinese Control Conference (CCC)*, pages 7458–7463. IEEE.
- Zee, T., Lakshmana, M., and Nwogu, I. (2022). Towards understanding the behaviors of pretrained compressed convolutional models. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3450–3456. IEEE.