

# **Analysis of Cumulative GPU Energy Consumption Across CNN Architecture Models**

*Savannah Maytom  
Independent Research Project  
2025*

*This project was conducted independently to explore the relationship between CNN model scale, numerical precision, and GPU energy consumption. While IEEE-style citations are used for clarity and attribution, all experimentation, analysis, and conclusions are original.*

## ***Abstract***

This paper analyses the effect of varying convolutional neural networks (CNN) configurations on GPU energy consumption during model training. These configurations vary with respect to the number of feature channels in each convolution layer and differing numerical precision modes, specifically involving standard FP32 and mixed-precision FP16 training. Energy expenditure is compared against performance metrics derived from training and validation losses to eventually assess the effectiveness of aforementioned precision modes as feature channels (widths) are increased. This project utilises a standard CNN deep-learning model accelerated by an NVIDIA T4 GPU, rated at a maximum power draw of 70W. Time is tracked and energy expenditure is approximated across each epoch, which is cumulatively stored and graphically visualised alongside loss and accuracy metrics. The experiment finds that increasing feature channel width significantly improves model accuracy and loss convergence, while mixed-precision training yields meaningful energy efficiency gains only for the largest model configurations, highlighting that the effectiveness of reduced numerical precision is strongly dependent on model scale and hardware utilisation.

## **Code Repository:**

<https://github.com/smeytom/gpu-energy-cnn-training>

## **1. Introduction**

Convolutional Neural Networks remain among the most widely used architectures for digital visual recognition tasks [1],[2]. However, the pursuit of higher accuracy and minimised losses has led to increasingly complex designs, placing greater strain on computational energy requirements. While model performance is often benchmarked on accuracy, far less attention is provided to energy consumption during training, underscoring the importance of energy consideration and analysis during CNN training stages. Although this report pertains to a small-scale neural network with a lightweight benchmark dataset, its outcomes reflect both the effect of precision modes on smaller models (associated with smaller channel widths) and larger CNNs where precision hardware such as tensor cores may be utilised to a greater extent. Hence, this small-scale network simulates marginal adjustments in width/precision, maintaining relevancy to the more complex and time-consuming models that are emerging in modern CNN designs. The goal of this analysis is to assess whether improvements in accuracy justify the increased energy requirements as network width and precision capability is scaled.

## 2. Related Theory

Convolutional Neural Networks are a type of deep learning model designed to recognise patterns in structured data, often used for computational image recognition [2]. CNNs utilise specialised layers; convolution layers, that scan small regions of the image and act as feature extractors. Each convolution layer slides multiple small kernels across the image, which acts as a learnable matrix that trains the network to recognise visual patterns. These kernels produce feature maps which highlight important elements such as edges, colour gradients, corners, textures, etc. Feature maps become the foundation to the construction of the network, as the number of maps used; represented by the feature map channels (*width* in this report), allows for the network to stack pattern learning procedures during the stage of processing in correspondence with the given width chosen. In this experiment, widths are represented as scalar integers; with a width of 1 pertaining to the smallest neural network, and a width of 3 to the largest. Thus, as channel width is increased, the task of processing becomes more complicated. Theoretically, larger widths improve model accuracy, despite requiring a higher computational energy per image.

As images are classified into dataset classes (which oversee the categorised objects within the datasets), many numerical operations are simultaneously conducted to transform kernels into a feature map. The amount of simultaneous operations depends on the numerical precision, used to represent the model's weight and activations. The standard precision for deep learning is a 32 bit floating point (FP32) representation [3], allocating a total of 23 bits for fractional characters, which makes it highly precise when carrying out matrix operations – despite requiring more memory and arithmetic work. Alternatively, mixed precision uses predominantly FP16, whilst still allocating FP32 where required to retain stability. This can accelerate the speed of calculations due to a lesser tax on memory. PyTorch, specifically, provides an Automatic Mixed Precision (AMP) Package which will be used to test energy expenditure across numerical precisions [3].

Feature channel width and numerical precision are to be compared against a variety of performance metrics, tested through the following width-precision combinations; FP32 width 3, AMP width 3, FP32 width 2, AMP width 2, FP32 width 1, AMP width 1. One of the primary outcomes of this report concerns the total and cumulative energy expended. Importantly, during the training process, an entire iteration over the dataset (epoch) gradually improves model accuracy and associated metrics. Thus, cumulative energy can be calculated after each epoch to observe the variation over time. Total and cumulative energies are to be calculated by equations (1) and (2) respectively [6].

$$(1) E_{total} = P_{GPU} \cdot t_{total} \quad (2) E_{cumulative} = P_{GPU} \cdot t_{epoch} + E_{previous}$$

The experiment utilises PyTorch's inbuilt NVIDIA T4 GPU, which is rated at 70W power draw. This means that the  $P_{GPU}$  variable is constant, and the respective time periods remain the only independent variables; implicating that energy is directly proportional to training time. Hence, it is expected that more complex CNN training stages should correlate with higher computational energy requirements [6].

A variety of other metrics are required to assess the performance of each model, which pertain to a more standard analysis of CNN configurations. During training, the model is shown a variety of labelled examples from each class, so that it can undergo the process of analysis to learn the relationship between an image and its category. To measure this, the dataset is divided into two segments; training and validation. The training dataset is passed through the network so that the model can update its weights to

reduce errors. This dataset is iterated over an amount of times defined by the epoch parameter. The validation dataset, however, is not used for model refinement, instead utilised to test the model's ability to generalise new images into classifications. To do this, PyTorch's CIFAR-10 dataset is used, with a split of 50,000 images for training and 10,000 for validation, each of resolution 32x32 RGB [4]. The dataset contains 10 classes, intending to be a lightweight benchmark dataset to enable efficient experimentation upon a small-scale neural network. Training loss curves illustrate how effectively the models reduce their error on the training dataset across each epoch. Loss itself is a measure of how 'wrong' the model's predictions are, as the model compares its probability distribution guesses to the true category of the training image, producing a single numerical value to reflect the error. Across a dataset of  $N$  images and  $C$  classes, the cross entropy loss function is given by equation (3).

$$(3) \quad L = -\frac{1}{N} \sum_{i=1}^N \log \left( \frac{\exp(z_{i,y_i})}{\sum_{j=1}^C \exp(z_{i,j})} \right)$$

The symbol  $z_{i,j}$  represents the logit (unnormalised score) assigned by the model to class  $j$  for sample  $i$ , and  $z_{i,y_i}$  denotes the logit corresponding to the correct class for sample  $i$ . This calculation is repeated across each class and image, then summed. Thus, lower values of  $L$  indicate that the model is producing more confident and accurate predictions, as the probability distribution becomes more dense upon the correct classifications. Note that in the neural network itself, this is carried out by PyTorch's inbuilt `CrossEntropyLoss` function.

Similarly, validation loss curves assess the model's reduction of error, instead centering around the validation dataset, with completely unmapped images that are fed into the network. Validation loss encompasses the model's ability to generalise beyond data it was trained on, and is extremely important to verify its capabilities following training. The validation loss is also given by equation (3). Though the equation is identical to that of the training loss function, it is expected to be more variable across batch iterations, despite still following the overall pattern of a reduction in the rate of decrease over time.

Accuracy curves track the percentage of images that the model has classified correctly after each epoch. Likewise to loss, accuracy is tracked across training and validation datasets, generally increasing rapidly across initial epochs, then plateauing in later ones. For  $N$  samples in the dataset, accuracy is given by equation (4).

$$(4) \quad A = \frac{1}{N} \sum_{i=1}^N 1(\hat{y}_i = y_i) \cdot 100\%$$

The indicator function  $1(\hat{y}_i = y_i)$  produces 1 if the model's prediction is correct, and 0 otherwise. It should be noted that if the validation does not mirror training (in both loss and accuracy), then the model may have overfitted to the dataset, meaning it has learned patterns specific to the training images as opposed to generalised patterns associated with the classifications. This is usually attributed to excessive model complexity, too many or too few epochs, or insufficient training data. Considering both loss and accuracy allows performance to be assessed not just in terms of correctness, but also how confidently and reliably the model reaches predictions.

For the purpose of comparison, the results of accuracy during training over overall energy expenditure during training are normalised through equation (5), to produce an artificial *performance* metric that weighs accuracy against energy.

$$(5) \quad performance = \frac{A_{training}}{E_{total}}$$

### 3. Related Literature

Although prior studies demonstrate that mixed precision reduces energy consumption on large CNNs, there is limited analysis regarding the impact of lightweight precision variation, where tensor cores may be underutilised. This gap motivates the present study, which compares precision modes across differing channel widths on CIFAR-10.

As a foundation to this study, there remain various notable scientific works and areas of research, which should be considered prior to its commencement.

Firstly, it's important to review the current position of large CNNs under the wider scope of neural networks and machine learning. The roots of convolutions neural networks originate from early works of LeCun, et al. [1], which effectuates the combination of convolution, pooling and learning filters to recognise hand-written digits. Following this development, a study released constituting the practicality of CNNs trained on larger datasets [2], marking among the first implementations of GPUs for such matters, dramatically improving image recognition, and introducing the prospect of deep learning in combination with computer vision.

As technology has developed, the increasing computational demands of deep CNNs have motivated extensive research into hardware acceleration and numerical optimisation. Modern GPUs enable highly parallelised matrix operations, allowing large-scale CNNs to be trained efficiently, but with substantial energy consumption. As model size and depth increases, training time and power usage become vital constraints, motivating the exploration of reduced numerical precision as a means of improving computational efficiency. Mixed-precision training has been proposed as an effective solution, most notably by Micikevicius et al. [3], demonstrating that a combined FP16 with selective FP32 can significantly accelerate training while maintaining numerical stability. By reducing memory bandwidth requirements and enabling the use of specialised tensor cores, mixed precision has been shown to reduce energy consumption for large-scale CNNs. These benefits are most evident when models exhibit sufficient arithmetic intensity to fully utilise GPU units and subsequent tensor cores [5].

Although much of existing literature focuses on large and computationally demanding architectures, there remains comparatively less analysis of how mixed-precision training behaves within lightweight CNNs, where reduced model complexity may limit hardware utilisation and introduce precision-related overheads. As a result, it is unclear whether the energy efficiency benefits reported for large models translate to smaller architectures. Thus, this study evaluates the interaction between feature channel width, numerical precision, and GPU energy consumption in a controlled and lightweight CNN environment.

## 4. Method

This section outlines the experimental methodology used to investigate the relationship between model capacity (feature channel widths), numerical precision (precision modes), and energy consumption during neural network training. In this experiment, a convolutional neural network is used for image classification, trained and validated on the CIFAR-10 dataset. Model success is assessed using training/validation loss and accuracy, while computational efficiency is evaluated through runtime and subsequent estimated energy consumption; relationship outlined in equation (5). To achieve this, the network structure can be defined by the model architecture, data preprocessing/loading, training procedure, validation/evaluation process, and metric calculation/processing.

The model architecture consists of three convolutional blocks with increasing feature channel depths and decreasing spatial resolutions, followed by a fully connected classifier. The model capacity is controlled by a *width* multiplier which scales the number of channels accordingly across layers. The final layer outputs tensors of unnormalised logits (class scores assigned by the model) for the ten CIFAR-10 classes.

The CIFAR-10 dataset is used for all models, consisting of 60,000 coloured images. Prior to training, input images are transformed into tensor representations, and normalised using the typical CIFAR-10 dataset mean and standard deviation to stabilise optimisation. Random cropping and horizontal flipping is also implemented to reduce overfitting. Validation undergoes an identical tensor conversion and normalisation, but excludes overfitting reduction measures to ensure unbiased performance evaluation.

The model training is performed using mini-batch stochastic gradient descent with momentum, optimising the categorical cross-entropy loss between the predicted class logits, and the true ground class labels. Following each training epoch, the model parameters and input weights are updated through backpropagation to minimise loss in future iterations across the training dataset. During this stage, training time for a singular epoch and total runtime of the model is tracked and stored. From this data, energy is estimated. Validation is conducted following this, within which the model operates in validation mode and no parameter updates are performed. Across the process of training and validation, each function calculates loss and accuracy respectively, which are used in later function calls to gain lists of data that is in turn graphically visualised.

The full experimental code and visualisation pipeline used in this study are publicly available at:

<https://github.com/smaytom/gpu-energy-cnn-training>

During the experiment, specific variables maintained constant between models; utilised dataset, optimisation settings, initial dataset transformations, convolutional layers, number of epochs, and GPU. This enables a stable experimental method where independent variables that influence performance metrics are isolated, and can be directly analysed.

## 5. Results

A summary of the experiment’s results are tabulated in *Table 1*, representing key metrics; the highest accuracy percentage achieved, the average time taken to iterate through one epoch, the total training time, and the total energy consumed across training. This is taken across each modeled configuration. The total time is seen to be proportional to the total energy expended. Notably, this renders model speed as the sole independent variable influencing energy expenditure. For this reason, time-specific metrics were omitted from the data visualisation segment (*Figures 5.1-5.6*). For the purpose of comparison, accuracy and energy have been normalised into a metric denoted ‘Efficiency’ to assess overall model performance; methodology having been covered in the *Related Theory*. However, this should be taken as a rough estimate of performance, as the efficiency metric’s validity breaks down in specific low accuracy/low energy combinations.

**Table 1: Summary of Configuration Metrics and Processes**

<i>Models</i>	<i>Highest Accuracy (%)</i>	<i>Avg Time/Epoch (s)</i>	<i>Total Time (s)</i>	<i>Total Energy (kWh)</i>	<i>Efficiency</i>
fp32_width3	81.87	22.20	665.85	0.01295	6.322
amp_width3	81.71	21.50	644.94	0.01254	6.516
fp32_width2	77.59	20.93	627.82	0.01221	6.354
amp_width2	77.28	20.98	629.31	0.01224	6.318
fp32_width1	69.96	20.60	617.92	0.01202	5.822
amp_width1	69.98	21.04	631.30	0.01228	5.700

The highest (convergent) accuracy achieved across models is observed to increase in correspondence with width, reinforcing that the addition of extra feature channels improves representational capacity. It is also seen that widths 3 consume the highest training time, and the substitution of mixed precision (AMP) reduces it significantly. On the contrary, at widths 1 and 2, the use of AMP increases energy slightly, directly impacting total energy expenditure through their directly proportional relationship.

These observations are mirrored in the efficiency metric, with 3 models producing the highest efficiency score (or lowest accuracy-energy ratio). Within this width group, applying mixed precision improves the efficiency rating from 6.322 to 6.516 due to an apparent energy expenditure drop while maintaining fairly constant accuracy. For width 2, the substitution of mixed precision has the opposite effect; lowering the efficiency from 6.354 to 6.318. Width 1 displays a similar trend, with a more significant difference where mixed precision reduces efficiency from 5.822 to 5.700.

In general, the width 3 configurations deliver the strongest balance between performance and energy use, and mixed precision provides a clear benefit by reducing the energy cost without harming accuracy. As model width is reduced, the advantages of mixed precision steadily diminish.

Figures 5.1 and 5.2 portray standard training and validation loss curves, displaying the expected loss trends; as loss improves rapidly in early epochs, then slows in later ones (after  $\sim 10$  epochs). Both training and validation loss plots mirror a similar trend, indicating that the dataset has not been overfitted by the model, which is important to validate performance.

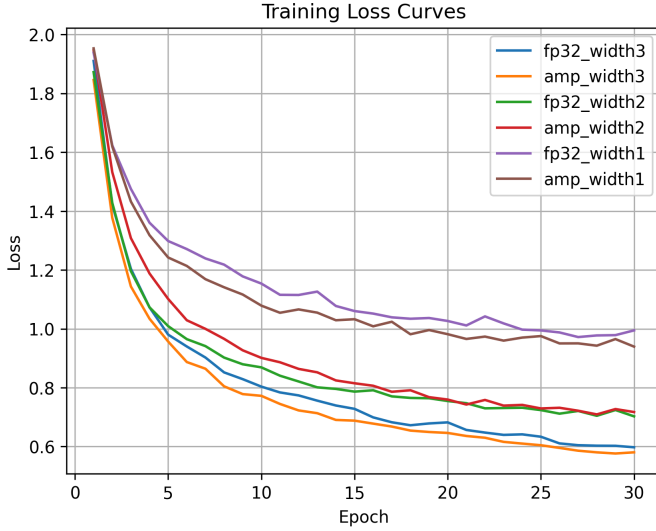


Figure 5.1: Training Loss per Epoch for Models Configurations

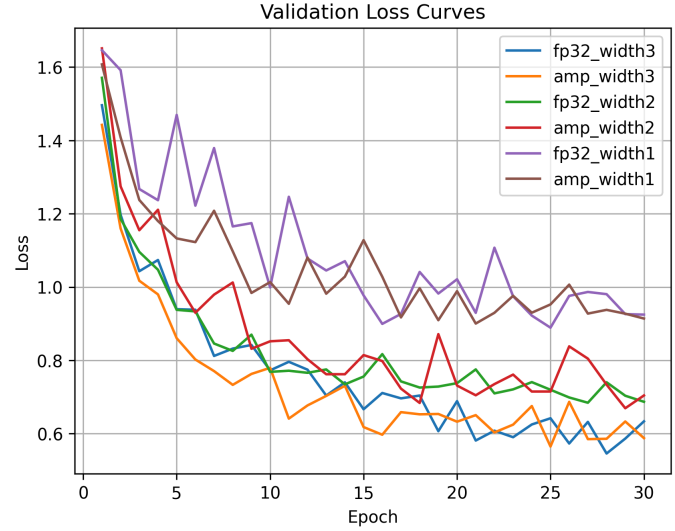


Figure 5.2: Validation Loss per Epoch for Models Configurations

The majority of the model's effective optimisation occurs during these first 10 epochs, eventually slowing to a convergence value; where the loss decreases at such a minimal rate that the model's improvement can be essentially viewed as stagnant. In the Figures, channel *widths* 2 and 3 achieve a noticeably lower loss convergence than *width* 1, indicating more sound probability predictions, especially as epochs increase. *Widths* 2 and 3 reach said convergence slightly faster than 1, which signifies that the additional feature maps assist the network in extracting fundamental patterns more efficiently. This outcome highlights that an expanded number of feature channels benefits representation learning on CIFAR10, and remains important when observing model capability.

As most prominent in Figure 5.1, the mixed (amp) and 32 bit (fp32) precision modes closely overlap throughout the curve period, especially as loss values approach convergence. This implies that precision variation has minimal impact on loss. This can also be observed in Figure 5.2, though at greater variability (due to the inherent instability of validation curves on account of such dataset images being previously unseen). It should be noted that width separation only begins within the first few epochs, as the plots begin at an effectively identical y-value, then gradually increase in separation as epoch magnitude approaches 30.

Ultimately, it is demonstrated that the loss curves are predominantly impacted by feature channel width, as global loss convergence decreases inversely proportional to width. Such curves also imply that precision modes have limited impact on both convergence, and overall loss across each epoch.

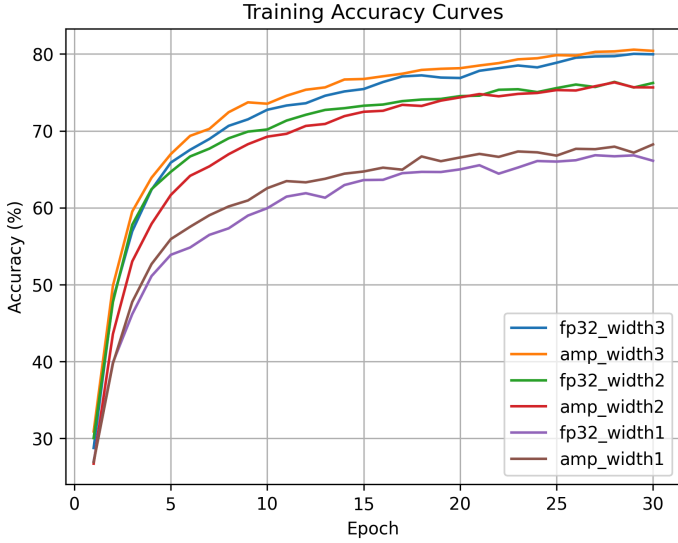


Figure 5.3: Training Accuracy per Epoch for Models Configurations

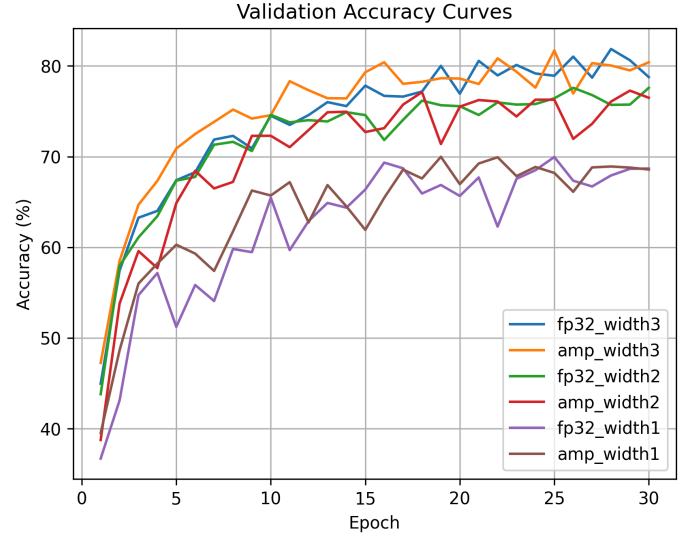


Figure 5.4: Validation Accuracy per Epoch for Models Configurations

Likewise to the loss curves, the accuracy curves in *Figure 5.3* and *5.4* demonstrate that a majority of the model's accuracy is acquired during the beginning of the training/validation process, as the accuracy slope is great across the initial 10 epochs. The accuracy curves, to the most extent, present as the inverse of the loss plots, indicating that an increased confidence in predictions directly translates to a higher proportion of correct classifications. Additionally, close alignment between training and validation accuracy signifies that the model has not been overfitted, with channel width maintaining a significant impact on accuracy, and precision modes almost none. This is to be expected due to preestablished observations from the loss curves in *Figure 5.1* and *5.2*.

Overall, the higher final accuracy achieved by wider configurations highlights their stronger representational capacity on CIFAR-10, while the minimal separation between precision settings reinforces that numerical format has negligible influence on classification performance for this network.

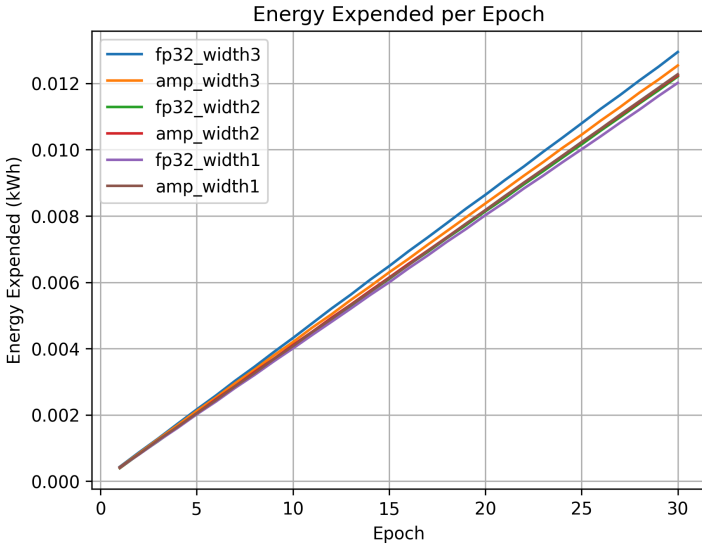


Figure 5.5: Energy Expended per Epoch for Models Configurations

otherwise be visually compressed in Figure 5.5. Thus, the numerical gradient of each model enables a more robust results analysis, which is displayed in Figure 5.6.

Observable in Figure 5.6, the highest per-epoch energy consumption rate occurs for channel widths 3, matching their greater computational work load. Across the width 3 variants, the mixed-precision mode appears at a significantly smaller rate than the FP32, supporting the notion that a greater precision type significantly increases energy expenditure for higher channel widths. For the channel width 2, however, both precision types appear to consume energy at a rate of effective equality, suggesting that tensor cores are only beginning to see usable parallelism at this scale and any benefit remains minimal. Furthermore, at width 1, the mixed-precision configuration is marginally less efficient, which reinforces that smaller models cannot utilise the specialised hardware to offset precision-related overheads.

It is important to note the scaling of the graphical energy consumption rate, as the values are extremely narrowly spread, at roughly 0.39-0.43 Wh per epoch. The narrow spread indicates that precision modes operate within a similar energy envelope, and thus per-epoch differences are modest within such lightweight model structures. This aligns with the prior observations that time, and therefore computational complexity, is the primary driver of total expenditure.

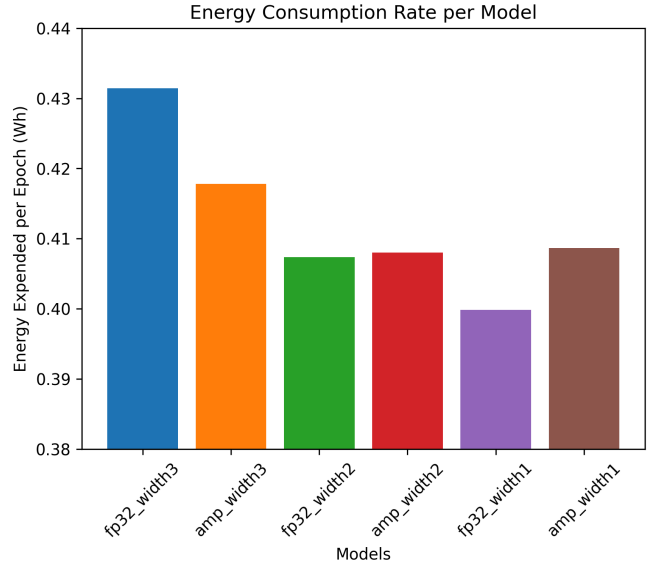


Figure 5.6: Energy Consumption Rate (Slope) for Models Configurations

Overall, as feature channel widths decrease, the effectiveness of mixed-precision modes on energy efficiency is observed to lessen, concluding that mixed-precision is most appropriate as applied upon higher model complexities.

## 5. Discussion

This study pertains to examining the impacts of feature channel widths and numerical precisions upon convolutional neural network GPU energy output during training, whilst also analysing model performance. Through the findings of the experiment, technical explanations must be investigated, alongside limiting factors and improvements to the experimental process.

The results demonstrate that channel width is the predominant factor in determining classification accuracy and loss, whilst precision modes have been observed to have limited impact. As channel width increases from 1 to 3, the models consistently achieve higher convergent accuracy values, and lower loss, confirming that increased representational capacity improves learning effectiveness on CIFAR-10. On the contrary, precision modes vary in effectiveness on total energy expenditure, with mixed precision yielding a higher energy output on lower widths, resulting in a looser (albeit minor) correlation between higher usage and an increasing channel width. When comparing the two metrics, overall performance of the training process was found to be most effective with a model using scalar width 3 and AMP mixed-precision, which provided the most effective balance at a maximum training accuracy of 81.71% and an energy expenditure of 0.01254 kWh. Although this was not the highest accuracy nor expenditure of the data, it yielded the strongest ratio between the two, hence being determined as most effective. It should be noted that this finding is subject to change as the training constraints are adjusted, this instance being specific to extremely lightweight neural networks.

There are two primary findings of this report; loss/accuracy acting proportionally to width increase and mixed precision achieving a measurable energy reduction only for the largest model width. These behaviours can be explained by the interaction between model arithmetic intensity, memory bandwidth utilisation, and GPU hardware specialisation. As channel width increases, the number of concurrent matrix multiplications grows, increasing arithmetic density and enabling more effective scheduling of parallel operations. This correlates with the observed increase in model accuracy. At sufficient scale, this also allows mixed-precision execution to leverage tensor core acceleration, reducing memory traffic and shortening kernel execution time, which in turn lowers total training duration and cumulative energy consumption. Conversely, at smaller channel widths, the computational workload is insufficient to saturate these specialised units, causing precision casting overheads, kernel launch latency, and reduced parallel efficiency to dominate runtime. As energy expenditure in this study is directly proportional to training time, these inefficiencies manifest as equal or increased energy usage under mixed precision. Therefore, the observed divergence in energy efficiency across widths reflects a hardware utilisation threshold, beyond which reduced numerical precision transitions from a computational liability to a tangible efficiency advantage. This suggests that mixed precision is most beneficial for large CNNs, while FP32 may remain more energy-efficient for lightweight architectures.

Several factors within the experimental design limit the generalisability and precision of the findings. Firstly, energy consumption was estimated indirectly using a fixed GPU power rating as opposed to real-time measured power telemetry, assuming constant power draw across all training phases and disregarding transient fluctuations during kernel execution, memory access, idle periods, etc. This may disguise energy discrepancies between precision modes; being predominantly relevant to lightweight models where per-epoch variations may be less significant. Subsequently, the use of fixed GPU architecture restricts robust hardware conclusions, as the effectiveness of mixed-precision is strongly

dependent on tensor core availability, memory hierarchy, and driver-level optimisations [5]. The relatively shallow network depth and narrow range of channel widths further limits insight into scaling behaviour, as the tested configurations may not fully capture the instances where mixed precision consistently outperforms FP32. Furthermore, several experimental factors reduce the effectiveness of direct model-to-model comparison, introducing unforeseen and unaccounted independent variables. Stochastic elements associated with CNN training; random weight initialisation, mini-batch ordering, and data augmentation, introduce variance that cannot be recognised in single-run evaluations [2]. Additionally, the use of identical training hyperparameters across models, though necessary for controlled cross-model analysis, may disadvantage specific model width/precision combinations that may have benefitted from alternative or tuned configurations. In turn, the narrow spread of energy consumption across models reduces observable differences, becoming more sensitive to minor fluctuations in the network, and other randomly occurring variables that have been discussed prior. Finally, a limiting factor arises from the creation and interpretation of the artificial performance metric that compares accuracy against energy expenditure. While this normalised quantity provides a convenient scalar representation of the trade-off between model performance and computational cost, it in turn simplifies a multi-dimensional optimisation problem. The metric assumes a linear and equally weighted relationship between accuracy and energy, which often does not reflect practical priorities where marginal accuracy gains can outweigh energy costs, or vice versa. This simplification becomes problematic for low-accuracy configurations, where small differences in energy consumption can produce disproportional swings in the performance score, reducing its robustness as a comparative measure [2].

Following the discussion of limitations, improvements to the experimental method should be considered. A primary addition involves the incorporation of direct power measurement tools such as the NVIDIA Management Library logging [7], enabling finer energy profiling at a kernel or batch level. This permits the inclusion of deeper networks, larger width scalars, and alternative layer composition which would enable better analysis of the threshold at which mixed-precision assists model efficiency. Likewise, the experiment would become more refined upon an expansion to alternative datasets and datatypes, avoiding overgeneralisation of the results to the specific CIFAR-10 dataset, and enabling mixed-precision threshold analysis across alternative parameters. In cases such as this, result analysis can be achieved more effectively, with fine-tuned interpretations; such as that of the performance metric, to attain more adequate conclusions and a more experimentally sound process.

Overall, the study highlights that while energy efficiency can be governed by feature channel width and numerical precision, it also extends to the interaction between model depth, hardware utilisation, and the foundational optimisation context of the model itself. This discussion finds that various limiting factors hinder the experiment, impacting said optimisation context, and potentially impacting key findings through randomly occurring neural network inconsistencies across model-to-model comparisons. As model energy consumption becomes an increasingly critical consideration in both research and deployment, particularly under constrained budgets, this underscores how architectural design choices and precision modes should be jointly considered to achieve meaningful and context-appropriate efficiency gains, as well as how model scope is a foundational concept in achieving model optimisation.

## 6. Conclusion

This study investigated the relationship between convolutional neural network feature channel width, numerical precision, and GPU energy consumption during training, alongside their impact on model performance. Through controlled experimentation on a lightweight CNN trained on CIFAR-10 using an NVIDIA T4 GPU, the results demonstrate that feature channel width is the dominant factor influencing both classification accuracy and cumulative energy expenditure. Increasing model width consistently improved loss convergence and final accuracy, albeit at the cost of longer training times and higher overall energy usage. In contrast, numerical precision was shown to have minimal influence on accuracy, while its effect on energy consumption was strongly dependent on model scale.

Mixed-precision training provided a measurable energy efficiency benefit only for the largest model configuration, where sufficient computational intensity enabled effective utilisation of specialised GPU hardware. For smaller and more lightweight models, mixed precision failed to reduce energy consumption and in some cases marginally increased it, indicating that reduced numerical precision does not inherently guarantee improved efficiency. These findings highlight the presence of a hardware and model-dependent utilisation threshold, beyond which mixed precision transitions from an overhead-dominated regime to an advantageous optimisation strategy.

Overall, this report demonstrates that energy-efficient deep learning cannot be achieved through isolated optimisation choices. Instead, it requires a holistic consideration of model architecture, numerical precision, hardware capabilities, and evaluation methodology. By highlighting the limitations of mixed precision in lightweight CNNs and emphasising the central role of model scale, this study contributes to a more in-depth understanding of energy-aware neural network design. As energy consumption becomes an increasingly critical constraint in modern machine learning systems, particularly in large-scale training and deployment environments, these insights provide a foundation for more informed and context-appropriate optimisation decisions.

## References

- [1] Y. LeCun et al., “Handwritten Digit Recognition with a Back-Propagation Network.” Available: <https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2012.
- [3] S. Narang et al., “Published as a conference paper at ICLR 2018 MIXED PRECISION TRAINING.” Available: <https://arxiv.org/pdf/1710.03740>
- [4] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” Apr. 2009. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [5] “NVIDIA A100 Tensor Core GPU Architecture” Available: <https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf>
- [6] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in NLP,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019, doi: <https://doi.org/10.18653/v1/p19-1355>.
- [7] “NVIDIA Management Library (NVML),” NVIDIA Developer. <https://developer.nvidia.com/management-library-nvml>