HuggingGreen - A Probabilistic Attention-Based Method for Energy Efficiency

Mattia Limone
Dipartimento di Informatica
Università degli Studi di Salerno

Carmine Iannotti
Dipartimento di Informatica
Università degli Studi di Salerno

***Abstract*—Vision Transformers (ViTs) have achieved state-of-the-art performance in many computer vision tasks. However, their high computational cost and energy consumption during training remain a major obstacle for their widespread adoption in resource-constrained scenarios. In this work, we propose a modified ViT architecture with probabilistic attention mechanisms that reduce the energy consumption during training without sacrificing the model's performance. Our approach utilize a probabilistic interpretation of the attention score in order to reduce the computational complexity required for self-attention, as well as a sparsity-inducing regularization term that encourages the attention weights to be more sparse. Additionally, we introduce a novel technique that selectively activates the attention mechanism only for important feature maps, further reducing energy consumption. We evaluate our proposed method on image classification and object detection task, and show that our method achieves comparable or better accuracy than the state-of-the-art ViT models while reducing energy consumption. The results aims to demonstrate the effectiveness of our modified ViT architecture in achieving energy-efficient training, which can benefit a wide range of applications, especially in resource-constrained environments.**

A PyTorch implementation can be found at: https://github.com/MattiaLimone/HuggingGreen

***Index Terms*—ViT, Image Classification, Object Detection, Energy Consumption**

I. Introduction

For decades, image classification has been a prominent research area in computer vision. With the advancements in deep learning, models can now identify objects, patterns, and features in images with high accuracy, revolutionizing image recognition. Convolutional neural networks (CNNs) have been the dominant approach for image recognition tasks, including object detection, but recent research has shown that transformers can also achieve state-of-the-art performance in this domain.

Transformers were initially developed for natural language processing tasks but have gained attention in the computer vision community due to their ability to model long-range dependencies and capture global context. The Vision Transformer (ViT) introduced by Dosovitskiy et al. [1] has shown impressive performance in various image recognition tasks, surpassing the performance of CNNs on several benchmarks.

In this paper, we attempt to go further and study the energy impact of these architectures and propose a version of the same model that makes use of a probabilistic attention mechanism in order to reduce electricity consumption.

The paper is organized as follows. In Section II, we provide an overview of related work on transformers and their applications in computer vision. In Section III, we describe the work goal and the research questions this work is structured around. In Section IV, we will present the datasets that we will use as benchmarks to compare our model with others in literature. In Section V, we are going to describe the pipeline that we are going to follow. In Section VI, we are going to specify the evaluation metrics to compare the model with others. In Section VII, finally we will describe our architecture. In Section VIII, we will present the results of the experiment.

Symbol	Description
ViT	Vision Transformer
CNN	Convolutional Neural Network
RQs	Research Question
PDFs	Probability Distribution Functions
MLE	Maximum Likelihood Estimation
KWh	KiloWatt-hour

Tabella I: Symbols used throughout the document

II. Related works

Convolutional neural networks (CNNs) have been widely used for image recognition tasks, but recent studies have shown that vision transformers (ViTs) can outperform CNNs on several benchmark datasets [2]. However, ViTs have high computational and memory requirements, making them challenging to deploy on resource-constrained devices.

Transformers are a type of deep learning model that has revolutionized the field of natural language processing (NLP) and has also been used in computer vision and other fields.

Transformers were first introduced in 2017 by Vaswani et al.[3]. The key innovation of transformers is the use of self-attention mechanisms instead of recurrent neural networks (RNNs) or convolutional neural networks (CNNs), which were the predominant models for NLP at the time.

Self-attention is a mechanism that allows the model to weigh the importance of different parts of the input sequence when making predictions. This mechanism allows the model to handle long-range dependencies much better than RNNs or CNNs, which makes them ideal for NLP tasks such as language modeling, machine translation, and sentiment analysis.

Transformers consist of an encoder and a decoder. The encoder takes in the input sequence and produces a sequence of hidden representations, while the decoder takes in the output of the encoder and generates the output sequence. The key to the success of transformers is the use of multi-head attention, which allows the model to focus on different parts of the input sequence simultaneously.

Vision Transformer (ViT), introduced by Dosovitskiy et al. [1], is a deep learning model that uses the transformer architecture for image classification.

Traditionally, convolutional neural networks (CNNs) have been the dominant model architecture for image classification tasks. However, ViT uses the transformer architecture to process image data without using any convolutional layers.

In ViT, the image is divided into a set of fixed-size patches, and each patch is treated as a token. These tokens are then processed by a transformer encoder, which allows the model to capture global information about the image. The transformer encoder consists of multiple layers, each containing a multi-head self-attention mechanism and a position-wise feedforward network.

ViT achieves state-of-the-art results on several benchmark image classification datasets, including ImageNet and CIFAR-100. One of the benefits of ViT is that it allows for better generalization to out-of-distribution data since it relies on global features instead of local patterns.

To address the issues of energy efficiency, several techniques have been proposed to reduce the energy consumption of ViTs. One approach is to use algorithms or heuristics that delete the large amount of redundancy present in self-attention operations.

For example Yangfan Li et al.[4] propose a delta patch encoding which expresses information in a compressed, more space-efficient and communication-efficient manner and a novel algorithm design of differential attention that leverage this patch locality to avoid these redundancies without loss of accuracy.

Instead Jing Liu et al. [5] propose a new binarization paradigm customized to high-dimensional softmax attention via kernelized hashing, called EcoFormer, to map the original queries and keys into low-dimensional binary codes in Hamming space. In this study based on PVTv2-B0 and ImageNet-1K datasets EcoFormer achieves a 73% reduction in on-chip energy footprint with only a slight performance drop of 0.33% compared to the standard attention.

In addition, some studies instead have approached the problem of energy consumption in a different way, for example, Ibrahim et al. [6] propose ImageSign, a methodology in which images are processed as signatures and processed through one-dimensional convolution (conv1d). Through this study, the authors were able to show that on some datasets they were able to drastically reduce both the number of parameters and the size of the model compared to a ViT, from 4,915,401 to 37,112 parameters and from 59.5 MB to 0.6 MB, respectively. In addition, they also managed to reduce the number of FLOPs from 4.65 to 1.69 without having a loss of accuracy; in fact, in one of their studies, permonances improved compared to ViTs, from 75.23% to 95.02%.

In 2021 Gabbur et al. [7] proposed a probabilistic interpretation of attention and suggested the use of Expectation Maximization algorithms for online adaptation of key and value model parameters, which can improve transformer model performance in tasks that require adaptation to new information during inference. Based on this work we have implemented their solution in a ViT architecture to address if the solution can be used in offline learning and Nguyen et al. [8] proposed a novel transformer architecture called Transformer-MGK, which replaces redundant attention heads in transformers with a mixture of Gaussian keys. Transformer-MGK accelerates training and inference, has fewer parameters, and achieves comparable or better accuracy across tasks than its conventional transformer counterpart. This work highlights the potential of using mixture models to improve transformer performance and reduce computational complexity.

Overall, these studies demonstrate the importance of addressing the energy efficiency of ViTs, and highlight several techniques that can be used to reduce their computational and memory requirements. Our work builds on these prior studies by proposing a new technique that combines pruning and knowledge distillation to improve the energy efficiency of ViTs while maintaining their accuracy.

III. Goals and Research Questions

The goal of this study is to assess whether a probabilistic approach as well as different training techniques results in an energy-efficient model with less impact on the environment. Based on this, after an analysis on the topic, we have structured our research around three research questions (**RQs**).

RQ$_1$. *Could a probabilistic approach lead to more energy efficient Transformer-based model?*

RQ$_2$. *Does removing parts of the information based on a heuristic allow the model to converge sooner?*

RQ$_3$. *Can different training techniques lead to lower energy consumption while preserving performance?*

With the first research question (**RQ**$_1$), based on work of Movellan et al. [7], we want to show that it is possible to use the attention mechanism in a mixture model treating the keys and values as the data points to cluster, and the queries as the test points for which the probability density function is estimated. The attention weights can be used as the mixture weights, and the values associated with each key can be used as the component PDFs. The mixture model can be trained using the MLE method, with the attention mechanism serving as the mixture density estimator.

The results on the first research question directly lead to the experimentation of the **RQ**$_2$, in which we are going to implement a feature selection layer that use an heuristic to select which part of the image pass to the Vision Transformer model. In a Vision Transformer architecture, each image is divided into n patches. Given a threshold $k < n$ we are going to select the a k sample of patches according to the probability distribution described by the contrast, variance or entropy value of the patches.

With the last research question (**RQ**$_3$), we want to test if different approaches of training in the Image Recognition task can lead to a faster training and thus lead to lower electricity consumption. In common practice, a transformer-based model first goes through a pre-training phase on a large amount of data and then is fine-tuned for the specific task. We are going to train the model in a different way hoping that it will need less time to achieve the same performance. Firstly a pre-training phase will be done but for much less epochs compared to the common practice, then the model will be trained using the hard-negative contrastive learning where the model is trained to distinguish between a positive pair (consisting of two samples that are similar or belong to the same class) and a hard-negative pair (consisting of two samples that are dissimilar or belong to different classes) and only then will a specific fine-tuning phase be carried out.

IV. Dataset

The selection of a suitable dataset is a critical aspect of training machine learning models. The primary reason for choosing CIFAR-10/100 and TinynImageNet is the computational cost that larger datasets such as ImageNet and JFT-300M requires. These datasets contain a significantly larger number of images, and training on them requires a large amount of computational power and storage capacity. Given our research aim, which is to prove the energy efficiency of our modified Vit architecture, it is not feasible to use these datasets.

The datasets we have used are commonly used to train and evaluate machine learning models, particularly in the field of deep learning. Because the images are relatively small and low-resolution, the datasets can be trained on standard hardware without requiring a large amount of memory or computational resources. Additionally, because the images contain a wide variety of objects and backgrounds, they are a good benchmark for evaluating a model's ability to generalize to new, unseen data.

Over the years, many state-of-the-art models have been trained and evaluated on these datasets, including various convolutional neural network (CNN) and Vision Transformer (ViT) architectures. Achieving high accuracy on these datasets has become a standard benchmark for evaluating the performance of image classification models.

This decision allows us to focus on proving the energy efficiency of our architecture, which is the primary objective of our research. CIFAR-10, CIFAR-100 are benchmark datasets commonly used for image classification tasks in computer vision research while Tiny ImageNet can give us a good understanding on the generalization capacity of the model.

CIFAR-10 consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The 10 classes are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

CIFAR-100 also consists of 60,000 32x32 color images but divided in 100 classes, with 600 images per class. The 100 classes are grouped into 20 superclasses, each containing 5 subclasses. For example, the ”aquatic mammals” superclass contains the ”beaver”, ”dolphin”, ”otter”, ”seal”, and ”whale” subclasses.

Tiny ImageNet is a reduced version of the ImageNet dataset, which is a large-scale image dataset used for training deep learning models for computer vision tasks. The Tiny ImageNet dataset consists of 200 classes, each with 500 training images and 50 validation and test images. The images in this dataset are of size 64x64 pixels, significantly smaller than the images in the original ImageNet dataset.

V. Execution Plan

The general execution pipeline, in Figure 1, provides an overview of the steps that will be followed. Hereafter, we will analyse for each research question how the experiment was conducted.

A. *RQ*$_1$ - *Comparing a standard ViT with Our Proposal*

After implementing in PyTorch our architecture described in Section VII we are going to train from scratch a ViT and our model and comparing them using the metrics explained in Section VI.

For the energy efficiency analysis we will use CodeCarbon, which provides an accurate estimation of the energy producted by RAM, GPUs and CPUs during the executing of the code (we will use Colab to perform the operations in as isolated an environment as possible *”Intel Xeon @ 2.20GHz and Tesla T4 GPU”*).

At this stage, the hyper-parameters will be fixed and common between the ViT architecture and our model, as our intention is to demonstrate that with the same configuration using a probabilistic approach, performance does not degrade and energy consumption is reduced.

Here follows the configurations for this phase of research, parameters marked by the * symbol are used in the ViT standard architecture, this configuration has been taken from the ViT-H/14, the current state-of-the-art ViT for Image Classification and Object Detection:

- in_channels*: 3
- patch_size*: 16
- emb_size*: 3072
- img_size*: 224
- depth*: 16
- top_k: 196 (100% of the patches)
- heuristic: none
- probabilistic: false
- prob: 1 (means that the heuristic function will not be applied)
- prob_decay_rate: 0
- batch_size: 512
- learning_rate: 1e-3

Further explaination of the additional hyperparameters can be found in the **RQ**$_2$ execution plan.

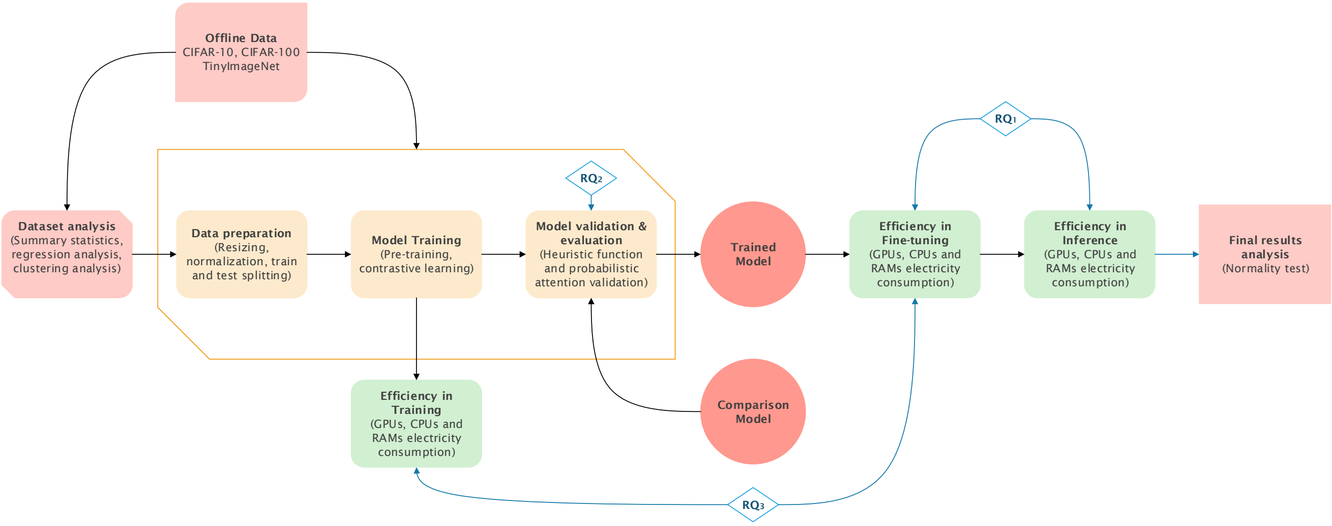
Offline Data
CIFAR-10, CIFAR-100
TinyImageNet
Dataset analysis
(Summary statistics, regression analysis, clustering analysis)
Data preparation
(Resizing, normalization, train and test splitting)
Model Training
(Pre-training, contrastive learning)
RQ2
Model validation & evaluation
(Heuristic function and probabilistic attention validation)
Trained Model
RQ1
Efficiency in Fine-tuning
(GPUs, CPUs and RAMs electricity consumption)
Efficiency in Inference
(GPUs, CPUs and RAMs electricity consumption)
Final results analysis
(Normality test)
Efficiency in Training
(GPUs, CPUs and RAMs electricity consumption)
Comparison Model
RQ3

Fig. 1: Overview of the experiment pipeline

We will perform a statistical validity test with null hypothesis stating that the sample (electric energy consumed repeating the same experiment) follows a normal PDF and alternative hypothesis stating that it does not come from that distribution:

- We will perform the Shapiro-Wilk test with significance level 0.05 in the hope of validating the null hypothesis
- If the tests confirm the null hypothesis, then the energy values we will report will be the average of the calculated values.
- Finally, we will calculate the Carbon Intensity, which is the grams of CO_2 produced by running the code as the product of the Net Carbon Intensity (A weighted average of the emissions from the different energy sources that are used to generate electricity consumed by the Cloud provider or Country) and the KWh consumed.

*B. **RQ**$_2$ - In Search of Heuristic that lead to a faster convergence*

When a human being is asked to understand an object represented on an image, a logical process is generally carried out that leads to discarding all information that does not concern the object itself. Based on this theory, we will implement a feature selection layer based on three different heuristics. Given that every patch in which the image is divided is an $n \times n$ matrix the value of each patch x is calculated as follows:

- Contrast (Figure 2):

$$F(x) = \frac{max(x) - min(x) + 10^{-8}}{max(x) + min(x)}$$

- Variance (Figure 3):

$$F(x) = \frac{\sum_{i=1}^{n} (x_i - \overline{\mathbf{x}})^2}{n-1}$$

- Entropy (Figure 4):

$$F(x) = -\sum_{i=1}^{n} (p_i * log2(p_i))$$

where

$$p_i = \frac{count(i)}{N}$$

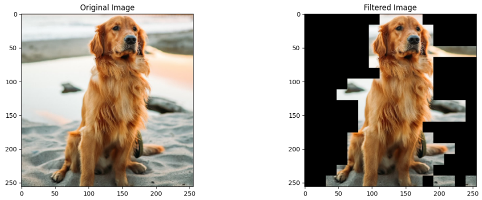

Fig. 2: Image filtered based on contrast values distribution preserving 50% of image

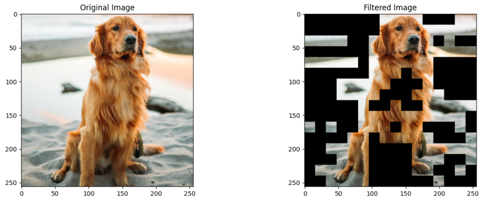

Fig. 3: Image filtered based on variance values distribution preserving 50% of image

We are going to perform the experiments on different hyperparameters, keeping fixed the learning rate fixed at $1e{-}3$, showed in Table II, parameters marked by the * symbol are used in the ViT standard architecture counterpart.

	4 heads			8 heads			12 heads		
in_channels*	3	3	3	3	3	3	3	3	3
patch_size*	16	16	16	16	16	16	16	16	16
emb_size*	64	64	64	64	64	64	64	64	64
img_size*	224	224	224	224	224	224	224	224	224
depth*	4	4	4	8	8	8	12	12	12
top_k	138	138	138	138	138	138	138	138	138
heuristic	*contrast*	*variance*	*entropy*	*contrast*	*variance*	*entropy*	*contrast*	*variance*	*entropy*
probabilistic	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
prob	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
prob_decay_rate	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3
batch_size	512	512	512	512	512	512	512	512	512

Tabella II: RQ$_2$ table of hyperparameters

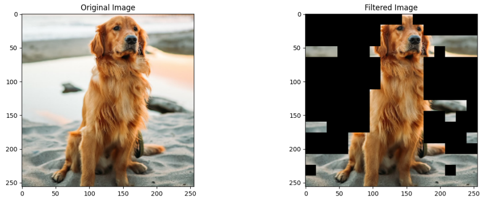

Fig. 4: Image filtered based on entropy values distribution preserving 50% of image

- in_channels: refers to the number of channel of the image, 1 if gray scale, 3 if rgb
- patch_size: is the size of each $n \times n$ patch into which the image will be divided
- emb_size: the number of dimensions in the feature representation used to encode each patch or region of the image
- img_size: is the size in which the image will be resized
- depth: is the number of head of the multi-head attention layer
- top_k: is the number of patches to preserve (e.g 138 for 224×224 is equivalent to preserve the 70% of the 196 patches in which the image will be divided)
- heuristic: is the heuristic method that will be used to select the patches
- probabilistic: if true the values of each patches will be interpreted as a PDF, otherwise they will be sorted descending
- prob: is the probability to chose to use the patches selected or the full image
- prob_decay_rate: is the decay rate over epoch for the prob hyperparameter in order to use the full image with the succession of epochs
- batch_size: refers to the number of samples or data points processed at a time

C. ***RQ$_3$*** *- Assessing the Performance of Our Model on different training technique*

In this phase we will train our model using two different techniques

1) Pre-training + Fine-tuning

2) Pre-training + Contrastive Learning + Fine-tuning

Each phase will be executed as follows:

- Pre-training: the model will be trained on a large amount of unlabeled data to learn general features. In our pre-training, the model is trained on Tiny ImageNet using a self-supervised task called 'masked language modeling.' This involves randomly masking 15% of patches of the input image and then training the model to predict the masked patches. This way, the model learns to represent the image in terms of meaningful patches of varying sizes and positions.
- Contrastive Learning: after pre-training, the model will be further fine-tuned using contrastive learning, which is a form of self-supervised learning. In contrastive learning, the model learns to distinguish between similar and dissimilar image by projecting them into a high-dimensional space and comparing their distances. This way, the model learns to identify patterns and features that are relevant for the downstream task, such as image recognition. Positive pair will be similar images. Negative pari will be an image randomly selected from a different class.
- Fine-tuning : once the model has been pre-trained and fine-tuned with contrastive learning, it will be further fine-tuned on image recognition task on CIFAR-10/CIFAR-100 and object detection on Tiny ImageNet. During fine-tuning, the model is trained on labeled data with a supervised learning approach. In this phase, the last few layers of the model are replaced with new ones, and only the newly added layers are trained on the specific task, while the earlier layers are frozen to retain the learned features. This way, the model can adapt to the specific dataset and learn to recognize the relevant features for the task at hand.

VI. Evaluation Metrics

Since the goal of our work is to obtain a more energy-efficient model, the energy consumption due to the use of RAMs, GPUs and CPUs will be monitored throughout the pipeline in order to identify the most affected hardware parts.

Furthermore, since we aimed to build a model that has comparable performance to other proposals in the literature, the trade-off between Energy Consumption/micro-average F1 Score (3) will also be considered.

The micro-average F1 score (3) is a commonly used performance metric in machine learning classification tasks. It is a type of F1 score that is calculated by taking the overall precision and recall across all classes, and then using these values to compute a single F1 score.

In order to calculate the micro-average F1 score, we first need to calculate the precision and recall for each class separately. Precision (1) measures the fraction of true positives out of all predicted positives for a given class, while Recall (2) measures the fraction of true positives out of all actual positives for that class.

Once we have calculated the precision and recall for each class, we can compute the micro-average F1 score as the harmonic mean of the overall precision and recall:

$$Precision_{\mu\text{-}avg} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{1}$$

$$Recall_{\mu\text{-}avg} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{2}$$

$$F1\ Score_{\mu\text{-}avg} = 2 \cdot \frac{Precision_{\mu\text{-}avg} \cdot Recall_{\mu\text{-}avg}}{Precision_{\mu\text{-}avg} + Recall_{\mu\text{-}avg}} \tag{3}$$

VII. Architecture

Our ViT Architecture in Figure 5 follows a three-stage process to process an input image:

- Patch Selection: The input image is divided into a grid of fixed-size non-overlapping patches, typically 16x16 pixels in size. For each patch is calculated a score based on an heuristic (contrast/variance/entropy). The calculated values are interpreted as a probability distribution and a k sample of patches is selected and passed to next step. The number of patches is determined by the size of the input image and the patch size.
- Patch Embeddings: Each patch is then linearly projected to a low-dimensional space to obtain a patch embedding vector. The patch embeddings are arranged in a sequence to form the input to the transformer encoder.
- Transformer Encoder: The patch embeddings are fed into a modified transformer encoder, which consists of multiple identical layers of self-attention and feed-forward layers. The self-attention mechanism allows the model to attend to different parts of the image and model the spatial relationships between patches. The feed-forward layers process the information from the previous layers and output a sequence of feature vectors.

In this particular implementation, in Figure 6, of the Transformer Encoder block, the Multi-Head Attention layer has been modified to be probabilistic in nature. Specifically, the attention weights computed by the layer are used as mixture weights and the values associated with each key are used as the component of the PDFs.

This modification allows for a more flexible and expressive representation of the attention mechanism. By using a probabilistic approach, the model can capture a wider range of dependencies between different positions in the sequence.

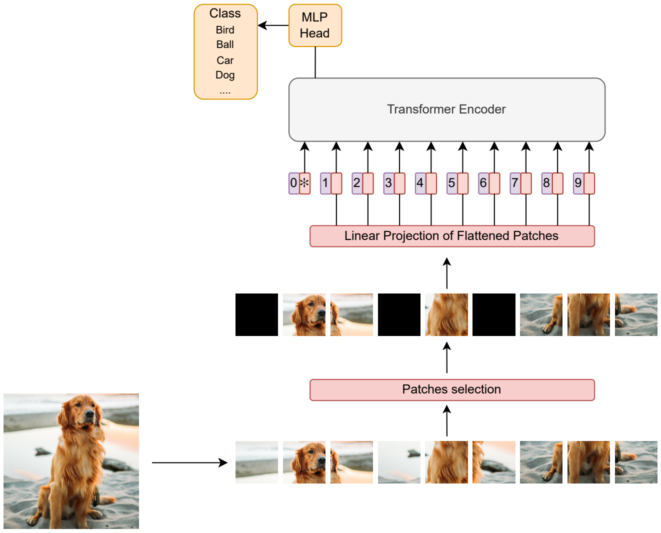

Fig. 5: Our ViT Architecture

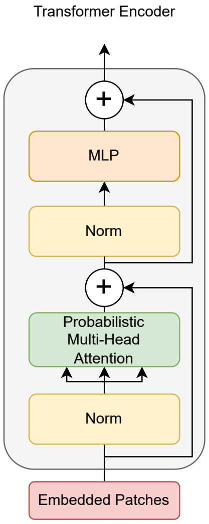

Fig. 6: Modified Transformer Encoder Block

VIII. Conclusion

Experiment not yet performed

IX. Future plans

In this paper, we presented HuggingGreen, a probabilistic attention-based method with the aim of reducing the environmental impact of transformer-based neural network models. However, there are several avenues for future research that could build upon our work and further advance the field of energy-efficient machine learning.

One potential direction for future research is to investigate the effectiveness of HuggingGreen on larger and more complex datasets. Probably it is possible that the performance of our method may degrade when faced with larger datasets or datasets with more different input features. Therefore, future research could focus on developing modifications to

HuggingGreen that enable it to scale to larger datasets without sacrificing performance.

Another area for future investigation is the potential for incorporating HuggingGreen into existing deep learning architectures (e.g. ResNet50, EfficientNetV2).

Finally, future research could explore the application of HuggingGreen to real-world energy-efficient machine learning problems because it is important to validate our method on real-world sceniario to ensure its practical effectiveness. In particular, future research could focus on the deploy phase of an HuggingGreen-based model.

In summary, we hope that our work on HuggingGreen provides a foundation for future research on energy-efficient machine learning. We believe that the potential for further improvements and applications of this method is vast, and we look forward to seeing how the field of energy-efficient machine learning continues to evolve in the coming years.

References

[1] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

[2] Stéphane Cuenat and Raphaël Couturier. "Convolutional Neural Network (CNN) vs Vision Transformer (ViT) for Digital Holography". In: *2022 2nd International Conference on Computer, Control and Robotics (ICCCR)*. 2022, pp. 235–240. DOI: 10.1109/ICCCR54399.2022.9790134.

[3] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[4] Yangfan Li et al. "DiVIT: Algorithm and architecture co-design of differential attention in vision transformer". In: *Journal of Systems Architecture* 128 (2022), p. 102520. URL: https://www.sciencedirect.com/science/article/abs/pii/S1383762122000868.

[5] Jing Liu et al. "Ecoformer: Energy-saving attention with linear complexity". In: *arXiv preprint arXiv:2209.09004* (2022). URL: https://arxiv.org/pdf/2209.09004.pdf.

[6] Mohamed R Ibrahim and Terry Lyons. "ImageSig: A signature transform for ultra-lightweight image recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3649–3659. URL: https://openaccess.thecvf.com/content/CVPR2022W/EVW/papers/Ibrahim_ImageSig_A_Signature_Transform_for_Ultra-Lightweight_Image_Recognition_CVPRW_2022_paper.pdf.

[7] Javier R. Movellan and Prasad Gabbur. "Probabilistic Transformers". In: *CoRR* abs/2010.15583 (2020). arXiv: 2010.15583. URL: https://arxiv.org/abs/2010.15583.

[8] Tam Nguyen et al. "Transformer with a Mixture of Gaussian Keys". In: *CoRR* abs/2110.08678 (2021). arXiv: 2110.08678. URL: https://arxiv.org/abs/2110.08678.