IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 January 2025, accepted 5 February 2025, date of publication 10 February 2025, date of current version 13 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3540384

RESEARCH ARTICLE

Selecting Images With Entropy for Frugal Knowledge Distillation

MICHAIL KINNAS[1], JOHN VIOLOS[2], NIKOLAOS IOANNIS KARAPIPERIS[1], AND IOANNIS KOMPATSIARIS[2], (Senior Member, IEEE)

[1]Department of Informatics and Telematics, Harokopio University of Athens, 177 78 Athens, Greece
[2]Information Technologies Institute, Centre for Research and Technology Hellas, 151 25 Athens, Greece

Corresponding author: John Violos (violos@iti.gr)

This work was supported by European Union's Horizon Europe Research and Innovation Program [European Lighthouse of AI for Sustainability (ELIAS)] under Grant 101120237.

ABSTRACT Frugal knowledge distillation is becoming increasingly important as it enables the distillation process to function effectively in resource-constrained environments. A key aspect of achieving this efficiency lies in minimizing the amount of training data required. To address this, we propose an entropy-based data selection method that identifies smaller subsets from the original dataset, focusing on images that retain the highest informational content. We explore the effectiveness of entropy-based method in combination with five different image representations to determine the subsets most effective for transferring knowledge to the student model. Our experimental evaluation on benchmark datasets, including CIFAR-10, MNIST, and FashionMNIST, shows that our approach outperforms other state-of-the-art image selection methods in most scenarios. It achieves over 3% higher accuracy compared to random selection methods while maintaining similar knowledge distillation time and energy efficiency.

INDEX TERMS Entropy, frugal machine learning, image selection, input frugality, knowledge distillation.

I. INTRODUCTION

The rapid advancement of deep learning has led to the development of highly accurate and complex visual models, which often require substantial computational resources for both training and inference. Knowledge distillation has emerged as a crucial technique to reduce the needed resources by transferring knowledge from a large, pre-trained model (teacher) to a smaller, more efficient model (student). This process allows the student model to approximate the performance of the teacher model while being more suitable for resource-constrained environments [1].

Knowledge distillation is a mechanism that belongs to the compression of deep learning models, a technique aimed at reducing the size and complexity of these models without significantly sacrificing performance [2]. This process is a crucial aspect of frugal machine learning, a domain focused on optimizing machine learning systems to be more efficient in terms of computational resources, memory usage, and energy consumption [3]. Our work focuses on two aspects of frugal knowledge distillation: creating an accurate and compact student model, and ensuring the distillation process is efficient, requiring minimal time, resources, and energy. This efficiency is crucial for distillation in edge computing environments rather than data centers [4].

A key aspect of effective knowledge distillation is the appropriate selection of training data that maximizes the transfer of knowledge. Naive approaches typically rely on large randomly sampled datasets, which can be computationally expensive and time-consuming [5]. In our work, we hypothesize that not all images contribute equally to the learning process. In particular, we search for a systematic methodology that can rank images based on their contribution to the accuracy of the student model in order to form appropriate subsets to be used for knowledge distillation.

To this end, we propose a methodology that leverages entropy-based [6] image selection to identify and prioritize informative samples for knowledge distillation. Our approach entails a comprehensive exploration of different image representations, including histograms, averaged adjusted histograms, logits vectors, compressed feature vectors obtained through autoencoders, and patch entropy vectors.

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

By quantifying the information content of images through entropy analysis, we aim to select a subset optimized for efficient knowledge transfer in model distillation.

The experimental evaluation of our proposed methodology spans three popular datasets: CIFAR-10, MNIST, and FashionMNIST. We assess the performance and efficiency of our approach by varying the number of samples used in the knowledge distillation process presenting the accuracy, distillation time, and energy consumption of our experiments. Our experimental analysis aims at testing the hypothesis of whether images with higher entropy, which indicate greater information content, are also more valuable for transferring knowledge to the student model. We have also conducted experiments with other image selection methods based on variance and manifold-learning. Inspired by the paper that proposes the manifold-learning method [7], we conducted experiments that integrate the entropy-based approach with an additional clustering step, where similar images are grouped, and selections are made within these groups using the entropy-based process. The results demonstrate the superiority of the entropy-based method without clustering in selecting informative samples for knowledge distillation, leading to improved model performance and reduced computational overhead.

The major contributions of our work can be summarized in the following:

- We posit the idea of selecting an effective subset of a dataset instead of the original dataset in the knowledge distillation process.
- We propose an entropy-based approach to quantify the information content and consequently the importance of each image to be included in the subset.
- We compare different variations of the proposed method using five distinct image representations.
- Experiments conducted on three different datasets using a power meter device confirm the effectiveness of our method, demonstrating its applicability through efficiency and performance evaluation metrics.

The rest of the paper is organized as follows. Section II presents the background and related work regarding the topics of frugal machine learning, compression methods, knowledge distillation, data and image selection. Section III presents the main objectives of frugal knowledge distillation. Section IV proposes our methodology to achieve frugal knowledge distillation prioritizing images based on the entropy formula and selecting an effective subset of images. Section V presents the main methods to represent images. Section VI illustrates the experimental comparison and the results of our method with other state of the art methods. Finally, section VII concludes the paper giving some future directions for further exploration of our proposed method.

II. BACKGROUND AND RELATED WORK

Our research belongs to the domain of frugal machine learning, which is discussed in subsection II-A. We present compression and knowledge distillation techniques in subsection II-B. We then discuss data condensation in subsection II-C and data sampling methods, in subsection II-D, for constructing subsets used in training deep learning models. Following this, we dive into image selection techniques in subsection II-E, including an alternative image selection method based on variance instead of entropy, and review two key papers that are closely related to our proposed method. Last but not least, the subsection II-F presents how our work fills the research gap that exist in the literature and goes beyond the related work.

A. FRUGAL MACHINE LEARNING

Frugal machine learning is a specialized paradigm within the field of machine learning that focuses on developing efficient and resource-conscious methods to deliver accurate predictions with limited computational and energy resources in computing environments such as wearable devices, IoT devices, and edge computing systems [8]. The primary goal of frugal machine learning is to strike a balance between predictive performance and resource consumption, aiming to achieve high accuracy while minimizing the use of costly resources such as computational power, memory, and energy [9].

To achieve frugality, three main approaches exist: Input frugality, learning process frugality, and model frugality [3]. Input frugality emphasizes minimizing the cost associated with data acquisition and feature utilization by using fewer training data or fewer features, driven by data availability, resource, or privacy constraints. Learning process frugality emphasizes minimizing the computational and memory resources required for training a model, often resulting in a less accurate but more resource-efficient model, driven by constraints such as limited computational power and training time. Model frugality emphasizes minimizing the memory, computational resources, and energy required to store and use a machine learning model, often at the expense of optimal prediction quality.

B. COMPRESSION & KNOWLEDGE DISTILLATION

Compression techniques have been proposed as integral strategies in the field of model frugality [10]. Compression methods mostly include parameter pruning, quantization, low-rank factorization, and model distillation. Parameter pruning identifies and removes redundant or less impactful neurons, layers, or channels in the model, resulting in a more streamlined architecture. Quantization, on the other hand, reduces precision by representing weights with fewer bits, significantly reducing the model size. Low-rank factorization model compression works by approximating a large matrix (such as weight matrices in neural networks) with the product of two smaller matrices. Knowledge distillation focuses on transferring the knowledge encapsulated in a large ''teacher'' model to a more compact ''student'' model, ensuring computational efficiency without compromising predictive accuracy.

Knowledge distillation offers several advantages over other compression techniques for deep neural networks. First, knowledge distillation allows for the transfer of rich knowledge and insights from a large teacher model to a smaller student model, enabling the student model to benefit from the teacher's expertise and achieve comparable performance with reduced computational resources [2]. This knowledge transfer process helps in improving the generalization capabilities of the student model. Second, knowledge distillation can effectively compress deep models without the need for re-training or fine-tuning, making it a more efficient and practical approach for model compression and acceleration. In addition, knowledge distillation is particularly beneficial in scenarios where labeled data is scarce. By transferring knowledge from a well-trained teacher model, the student model can leverage the insights gained from the larger model, even with limited training data [11].

C. DATASET CONDENSATION

To achieve data frugality, recent research has proposed dataset condensation. This process involves synthesizing a compact and representative subset from a larger dataset. The goal is to create a smaller dataset that retains the essential information and characteristics of the original, while reducing storage and computational requirements [12]. By condensing the dataset, machine learning models can be trained more efficiently without sacrificing performance.

Various dataset condensation approaches have been proposed. One such approach is Synthetic-Data Parameterization [13], which synthesizes compact training datasets through an optimized parameterization that captures the regularity of data. Another approach formulates the condensation process as a gradient matching problem, which aligns the gradients of deep neural network weights trained on the original dataset with those trained on the synthetic dataset [14]. Furthermore, there is an approach that matches the feature distributions of synthetic and original training images across many sampled embedding spaces [15].

Dataset condensation has specific characteristics that make it infeasible for the frugal knowledge distillation process for two important reasons. First, the condensed dataset may be highly specific to the particular model or task used during the condensation process. Thus, it might not generalize well to other models or tasks, limiting its broader applicability. Second, dataset condensation is a computationally intensive process that cannot be performed with images stored in resource-limited computational environments. These reasons led us to investigate a data sampling approach using a criterion that can quantify the information each image contributes to the knowledge distillation process.

D. DATA SAMPLING

In the field of deep learning, discussions on data sampling or selection have primarily focused on the need for diverse data sampling [16]. This means, data engineers should collect as many diverse data samples as possible, ensuring they cover all potential instances and variations of observations. In addition, when dealing with class imbalance in data, researchers have proposed random over/under-sampling, and hybrid methods to mitigate the cardinality difference between large and small classes [17]. However, these sampling approaches do not consider whether the selected instances provide more information than those left out.

Paul et al. [18] proposed a method for identifying important data samples in the training of ANNs. The researchers found that in vision datasets, simple scores averaged over several neural weights can be used to identify crucial data samples early in training. They introduced two such scores: the Gradient Norm and the Error L2-Norm. Despite the excellent evaluation outcomes of these scores, they require that all data will be used for several epochs to quantify the importance and prioritize the data samples based on their information content. This is a significant limitation, particularly when transferring knowledge between a large teacher model and a smaller student model with large datasets on resource-constrained devices.

E. IMAGE SELECTION

Image selection methods rely on a ranking criterion to quantify and prioritize data samples before they enter the knowledge distillation process. Such approaches include methods based on the Highest Variance Criterion, Manifold Learning methods, and methods based on entropy criteria. As baseline, we also use Random Image Subset Selection. In the experimental comparison of our article we evaluate all these approaches. The Random Image Subset Selection, the Highest Variance Criterion, and the Manifold Learning-based are described in this subsection. The Entropy criterion is discussed in the subsection IV-B.

Random Image Subset Selection [17] is a method used in data processing and machine learning tasks where a subset of images is randomly chosen from a larger dataset. This approach involves randomly selecting images from a given dataset without any specific criteria or bias, aiming at a diverse representation of the original dataset. Random Image Subset Selection is commonly used in scenarios where processing or training on the entire dataset is computationally expensive or impractical, allowing researchers or practitioners to work with a smaller, manageable subset while still maintaining the diversity of the original dataset.

The Highest Variance Criterion offers an alternative approach for selecting a subset of images based on the variance of their histogram representations. In previous studies, this criterion has been applied as an attribute selection technique for binary classification in imbalanced datasets [19] and for assessing image quality by quantifying visual features [20]. However, it has not yet been used for image selection. In our experiments, each image in the dataset is initially represented by its color intensity histogram. The variance of these histograms is then computed across

the entire dataset. Images with histograms exhibiting higher variance are considered to contain more diverse pixel color intensity distributions, indicating a wider range of visual content. By selecting images with the highest variance in their histogram representations, this method aims to assemble a subset that encompasses a broad spectrum of visual characteristics present in the original dataset.

Manifold Learning-based Data Sampling [7] involves converting high-dimensional data into a lower-dimensional representation using manifold learning techniques, such as Locally Linear Embedding and k-Means clustering. By projecting the data onto an n-dimensional space, the dataset represents a volumetric image, allowing for a more efficient analysis of the data distribution. This helps in addressing the challenges of data selection by ensuring that the subset maintains similar topology as the original. By resampling the data to the same image spacing before applying manifold learning, biases due to data selection can be minimized, leading to improved model training and more accurate results in visual tasks such as image classification.

F. BEYOND THE RELATED WORK

Although knowledge distillation produces lightweight models, the process itself is computationally demanding, making it infeasible for devices with limited resources. Dataset condensation approaches can be used to reduce computational load and memory requirements, but these transform the data in a way that the initial pre-trained models cannot work with the new data representations.

To address these limitations, we propose an entropy-based data sampling methodology to select the most informative images. This ensures that the selected subsets maintain high information content, thereby improving the efficiency of the distillation process and preserving the performance of the student model. This aligns with the overarching goals of frugal machine learning.

III. OBJECTIVES IN FRUGAL KNOWLEDGE DISTILLATION

Frugal knowledge distillation addresses the need for efficient model compression in resource-constrained environments by focusing on minimizing model size, computational complexity, memory usage, and energy consumption, while maintaining performance. The challenge on the one hand lies in distilling the essential knowledge from a large teacher model into a compact student model without sacrificing performance or accuracy. On the other hand, the process of knowledge distillation should be as lightweight as possible requiring a small amount of computation and energy consumption. In this context, frugality encompasses both the student model and the process of knowledge distillation that produces the student model. In the following, we briefly outline the five key objectives that we will also use in the experimental evaluation section:

- **Number of student parameters:** By reducing the model's parameter count, we aim to enhance its computational efficiency, memory footprint, and inference speed, making it suitable for deployment on resource-constrained devices such as mobile phones, edge devices, and IoT devices.
- **Predictive Performance**: Maintaining high predictive performance (in terms of metrics like accuracy, precision, recall, and F1-score) and generalization despite reductions in model size and complexity is crucial. The challenge lies in compressing models without significant performance degradation, ensuring the student model retains essential knowledge from the teacher model. This requires a careful balance between model compression and predictive performance.
- **Distillation time:** The efficient utilization of computational resources, the optimization techniques and the best use of available training data play a crucial role in accelerating the distillation process. Shorter distillation times enable rapid model development and bigger re-usability of the infrastructure, facilitating agile experimentation and iterative refinement.
- **Energy consumption:** Energy efficiency is a key consideration particularly in battery-powered devices and energy-constrained environments. Minimizing energy consumption during both distillation and student inference phases is essential to prolonging device battery life and reducing environmental impact.
- **Less data:** Knowledge distillation with less data focuses on achieving effective knowledge transfer using smaller datasets, which offer a minimization in training time and resource utilization. The approach assumes that not all data samples equally contribute to the distillation process and aims to systematically select a subset that maximizes the student model's predictive performance.

IV. PROPOSED METHODOLOGY

The main proposed idea involves utilizing entropy on image representations to identify image samples that make a greater contribution to the knowledge distillation process, resulting in a high performing student model. The utilization of entropy in the image representations is included as the most important step in frugal knowledge distillation workflow as depicted in Figure 1. In Section V we comprehensively discuss five approaches to represent images. In this section we outline the key steps of the entire workflow, beginning from the original full-size dataset and ending in the creation of the compressed student model. We will particularly underscore the significance of image selection, which is pivotal for the subsequent frugal knowledge distillation process.

A. IMAGE REPRESENTATIONS

With a full-size dataset, the first step in the image selection with entropy workflow is the image representation as shown in Figure 1 steps a and b respectively. In the Image Representation step, we aim to generate a vector representation that encapsulates the unique insights and the

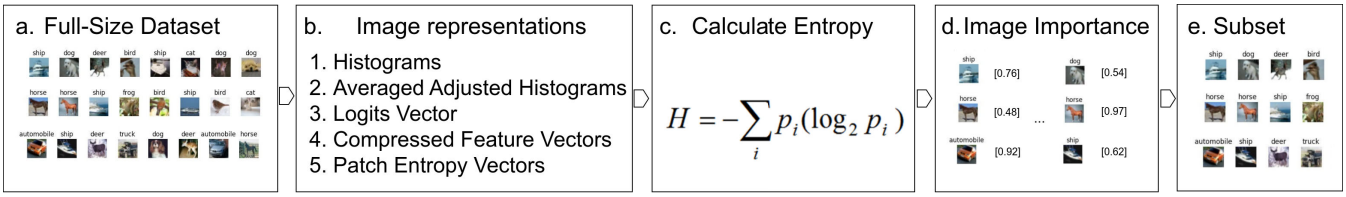

FIGURE 1. Subset of dataset: selecting images with entropy.

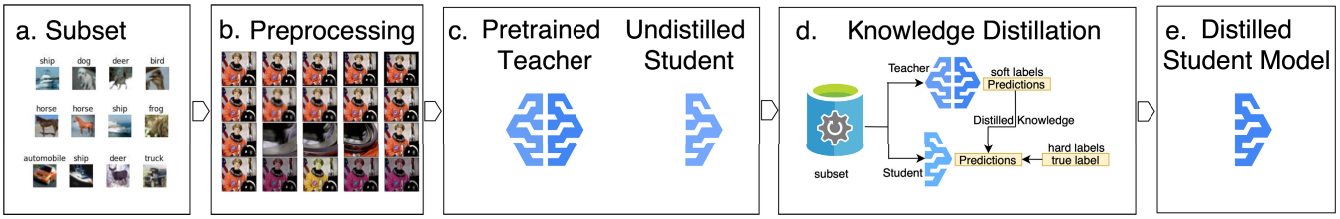

FIGURE 2. Knowledge distillation with image selection.

intrinsic characteristics of the images. As we can see in the Figure 1, step c, in these image vector representations we will apply the cross entropy formula to provide a quantitative measure of the information content within the images. We explored various approaches for image representation, each offering distinct characteristics for capturing and encoding visual information from images. These approaches are: 1. histograms, 2. averaged adjusted histograms, 3. logits vectors, 4. compressed feature vectors, and 5. patch entropy vectors. In section V we will investigate these different representations.

B. ENTROPY FORMULA

We use the entropy formula on the image representations as we can see in Figure 1 step c. Entropy in the context of image analysis, measures the uncertainty or disorder within the distribution of pixel intensities, latent representations, or other image features. A higher entropy value indicates greater complexity or information content within the image, while lower entropy suggests more predictability or uniformity. By computing entropy for each image representation in an image dataset, we can gauge its relative importance based on the diversity and richness of information it contains.

The entropy function $H(X)$ is given in Equation in Equation 1.

$$H(X) = -\sum_{i=1}^{n} P(i) \log P(i) \tag{1}$$

Here, $H(X)$ denotes the entropy of the image representation X, where $P(i)$ denotes the probability of occurrence of each image feature i. The natural logarithm log amplifies the importance of rare features, while the summation process aggregates contributions from all possible outcomes within the image.

C. IMAGE IMPORTANCE

We apply an image importance criterion to prioritize which images should be included in the subset, as illustrated in Figure 1, step d. Image importance is a measure of the significance of individual images within a dataset. In our approach we quantify image importance through the calculation of entropy. To ensure balanced representation across categories, since the original datasets are also balanced, we select an equal number of images per class (Algorithm 1). In this way we capture a diverse range of visual characteristics present in the dataset while maintaining equal focus on each class.

Algorithm 1 Select Top N Entropy Score Samples Per Category

SELECT_SAMPLES (e, y, N)
// Select the top N samples per category based on entropy scores. e are the entropy scores of the images in the dataset y are the corresponding class labels and N is the number of total samples to select.

initialize $selIdxs = [\,]$
for $i \leftarrow 0$ **to** $max(y) - 1$ **do**
  initialize $labelIdxs = [\,]$
  for $j \leftarrow 0$ **to** $size(y) - 1$ **do**
    if $i = y[j]$ **then**
      append $labelIdxs \leftarrow j$
    end if
  end for
  initialize $labelEntropies = [\,]$
  for $i \leftarrow 0$ **to** $size(labelIdxs) - 1$ **do**
    append $labelEntropies \leftarrow e[labelIdxs[i]]$
  end for
  set $sortIdxs \leftarrow argsort(labelEntropies)$
  for $i \leftarrow size(sortIdxs) - N$ **to** $size(sortIdxs) - 1$ **do**
    append $selIdxs \leftarrow labelIdxs[sortIdxs[i]]$
  end for
end for
return: $selIdxs$

D. OUTPUT: SUBSET

After quantifying image importance, we retain the highest-entropy images while ensuring class balance (Figure 1, step e). This curated subset is optimized for efficient knowledge transfer, serves as the input for the knowledge distillation

process, and it is the first step in the pipeline depicted in Figure 2.

E. PRE-PROCESSING

Pre-processing prepares the images of the subset for distillation, as shown in Figure 2, steps a and b. This involves resizing the images to a uniform dimension, applying data augmentation (like random horizontal flipping) to increase diversity, and converting the images into tensor representations. The data is then normalized to ensure consistent pixel intensity, reducing variations in illumination and improving model stability and performance. This pre-processing step is crucial for optimizing the data, ensuring that the models are effectively trained and that knowledge distillation is robust and accurate.

F. PRETRAINED TEACHER

The Knowledge distillation takes place in a pretrained teacher and an undistilled student as we can see in the Figure 2 step c. The pretrained teacher is a deep learning model that facilitates the transfer of knowledge to the student model using the images included in the subset. Throughout the distillation process, the teacher model provides supervisory signals as soft targets to the student model, enabling effective knowledge transfer, robust performance and enhance the generalization capabilities in image classification tasks.

G. UNDISTILLED STUDENT

The undistilled student refers to the initial, untrained state of the student model prior to undergoing the distillation process. The student model's architecture should strike a balance trade-off between model complexity and computational efficiency while still being capable of learning the distilled knowledge from the teacher model effectively. Typically, the student model is designed to be simpler and more compact than the teacher model, reducing inference time as well as computational and energy requirements. However, at this undistilled stage, the student model has not yet been trained, meaning it has no learned parameters or knowledge.

H. KNOWLEDGE DISTILLATION

Using Knowledge distillation the compressed, more efficient model (the student) is trained to emulate the performance of the larger, more complex model (the teacher). The goal of this article is not to provide an in-depth description of knowledge distillation; rather, it aims to offer a brief overview of the process as it pertains to our work. For readers seeking a comprehensive understanding of knowledge distillation, we recommend the article knowledge distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks by Lin Wang and Kuk-Jin Yoon [11].

Knowledge distillation involves the student model learning from the softened output probabilities (soft labels) of the teacher model using the images included in the subset as depicted in Figure 2 step c. Soft labels, which capture the teacher's prediction distribution over all classes, provide richer information about the relative importance of different classes and inter-class similarities. By leveraging the most informative images through these soft labels, the student model can achieve high accuracy while consuming significant less computational resources.

It is important to mention that the knowledge distillation process can be very computationally intensive, particularly when working with large amounts of data. To address this challenge is the main research contribution of our work. We propose a methodology to select a subset of the available images, aiming to minimize the processing resources and energy requirements. This strategic selection mitigates the heavy computational load of the distillation process, ensuring that it remains efficient and manageable without compromising the quality of the distilled knowledge.

I. DISTILLED STUDENT

When the knowledge distillation process is complete, the outcome is the distilled student as depicted in Figure 2 step e. The student model is a compressed, efficient version of the larger, more complex teacher model. The student model is designed to replicate the performance of the teacher while significantly reducing computational demands. It consumes considerably less CPU and memory, resulting in faster response times and lower energy consumption. These attributes make the student model highly suitable for deployment in resource-constrained environments, such as mobile devices and edge computing scenarios, where efficiency and quick processing are paramount.

J. CLUSTERING SELECTION

We explored the use of clustering on image representations as part of a comparison to our main methodology, drawing inspiration from the Manifold-Learning approach [7], which uses clustering to select representative subsets. The aim was to create a diverse subset that closely mirrors the characteristics of the original dataset. In this process, images were first grouped by their classes, and K-Means clustering was applied to each group. From each cluster, we selected samples with the highest entropy scores, proportional to the cluster size, to avoid selecting overly similar images, which would yield nearly identical entropy scores (Algorithm 2). This step appears between steps 1(c) and 1(d) in Figure 1. However, as shown in Section VI Experimental Evaluation, using clustering did not improve performance over using entropy alone. Therefore, while we discuss this approach, we do not recommend it as part of our primary methodology, nor do we include it in the workflow presented in figures.

V. EXPLORING DIFFERENT APPROACHES TO IMAGE REPRESENTATION

Image representations play a pivotal role in analyzing, and modeling visual data, serving as the foundation for various computer vision tasks. We make a comprehensive investigation of five distinct methodologies for representing images, each offering unique insights into their visual

Algorithm 2 Select Images by Clustering on Each Label

SELECT_SAMPLES (X, y, N)
// Perform clustering on image representations for each label category. X are the image representations on which the clustering will be performed. y are the corresponding labels and N is the number of total samples to select.
for $i \leftarrow 0$ **to** $max(y) - 1$ **do**
  initialize $label\_indices = [\,]$
  for $j \leftarrow 0$ **to** $size(y) - 1$ **do**
    if $i = y[j]$ **then**
      append $labelIdxs \leftarrow j$
    end if
  end for
  initialize $selIdxs = [\,]$
  set $clusters \leftarrow clustering(X[labelIdxs])$
  for $j \leftarrow 0$ **to** $size(clusters) - 1$ **do**
    initialize $clusterEntropies = [\,]$
    for $k \leftarrow 0$ **to** $size(clusters[j]) - 1$ **do**
// calculate entropy on the representations from each cluster
      set $repr \leftarrow clusters[j][k]$
      set $imageEntropy \leftarrow calcEntropy(repr)$
      append $clusterEntropies \leftarrow imageEntropy$
    end for
    set $sortIdxs \leftarrow argsort(clusterEntropies)$
    set $sel \leftarrow round(size(clusters[j])/size(X)) \cdot (N/(max(y)+1)$
    for $k \leftarrow size(sortIdxs) - sel$ **to** $size(sortIdxs) - 1$ **do**
      append $selIdxs \leftarrow clusters[j][sortIdxs[k]]$
    end for
    append $selIdxs \leftarrow size(clusters[j])$
  end for
end for
return $selIdxs$

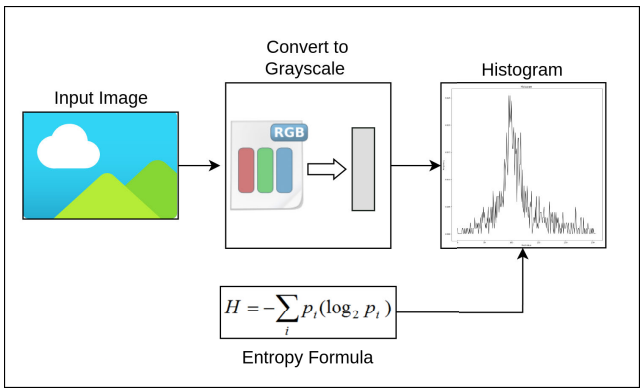

FIGURE 3. Histogram representation.

characteristics and content. These methodologies encompass histogram analysis, averaged adjusted histograms, logits vector extraction, compressed feature vectors obtained through autoencoder-based encoding, and entropy vectors based on image patches as depicted in Figures 3, 4, 5, 6 and 7. By representing the images as vectors and applying the entropy formula to them, we assess the significance of each image for inclusion in the subdataset.

A. HISTOGRAMS

Histograms can be used for representing the distribution of pixel intensities within an image, offering a succinct overview of its visual characteristics. Each bin in the histogram corresponds to a specific range of pixel intensity values, with the height of each bin indicating the frequency of pixels falling within that range. In our approach (Algorithm 3), firstly we make a grayscale conversion of the input images and then we compute the histogram, capturing the frequency of occurrence for each pixel intensity value across the entire image. Subsequently, we normalize the histogram values by dividing the count of each bin by the total number of pixels in the image. This normalization step ensures that the histogram represents a probability distribution, enabling seamless comparison across images of varying sizes. After these steps, we can compute the entropy of the normalized histogram to quantify the level of uncertainty in the distribution of pixel intensities, Figure 3 illustrates the histogram representation of an image.

When evaluating entropy using histograms, $P(i)$ represents the normalized histogram bin values, with $\log P(i)$ being the natural logarithm of these probabilities.

Algorithm 3 Grayscale Histogram Entropy

CALCULATE_ENTROPY (X)
// Compute the Grayscale Histogram Entropy representation of each image from the dataset and compute the entropy score on these representations. X are the images from the dataset, and $entropies$ are the calculated entropy scores.
initialize $entropies = [\,]$
for $i \leftarrow 0$ **to** $len(X) - 1$ **do**
  set $hist \leftarrow calcHistogram(X[labelIdxs[i]])$
  set $hist \leftarrow hist/sum(hist)$
  set $imgEntropy \leftarrow calcEntropy(hist)$
  append $entropies \leftarrow imgEntropy$
end for
set $sortIdxs \leftarrow argsort(labelEntropies)$
return: $entropies$

B. AVERAGED ADJUSTED HISTOGRAMS

In our quest to better estimate the image importance, we explore the concept of averaged adjusted histograms. This approach (Algorithm 4) involves aggregating histograms from a collection of images to derive a representative distribution, thereby enabling a comparative analysis of individual image characteristics against these collective aggregated characteristics. In this approach we iterate over each image, converting it to grayscale to facilitate histogram computation. Then we generate the histogram for each grayscale image, as described in the previous subsection. These individual histograms are aggregated to form a cumulative representation, from which the average histogram is computed.

With the average histogram computed, we proceed to assess individual images' entropy relative to this baseline. This image representation method quantifies the degree of divergence in pixel intensity distributions between each image and the average histogram. By computing the difference between the normalized histogram of each image and the average histogram, we effectively filter out noise or irrelevant details, allowing us to focus on the significant

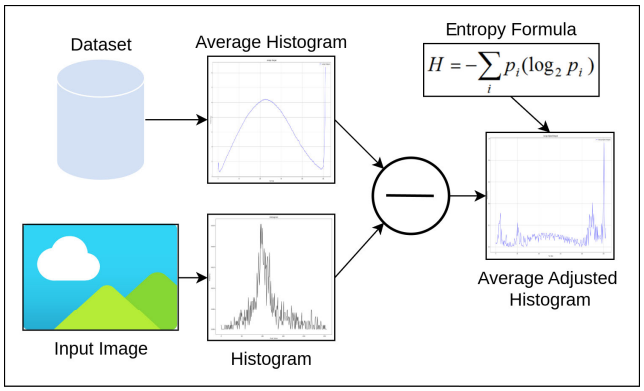

FIGURE 4. Averaged adjusted histogram representation.

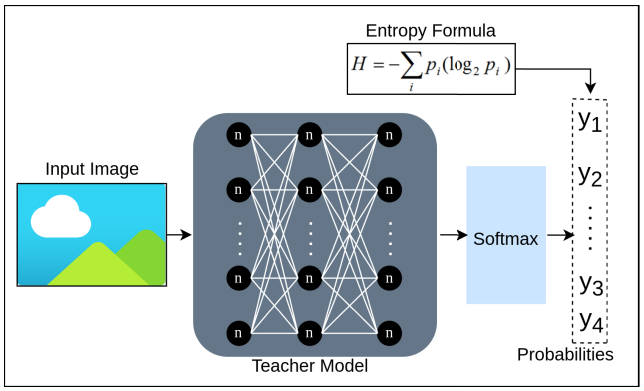

FIGURE 5. Logits vector representation.

visual characteristics of each image. Figure 4 illustrates the averaged adjusted histogram representation of an image.

Algorithm 4 Average Adjusted Histogram Entropy

CALCULATE_ENTROPY (X)
// Compute the Average Adjusted Histogram representation of each image from the dataset and compute the entropy score on these representations. X are the images from the dataset, and *entropies* are the calculated entropy scores.
set *histogram* $\leftarrow 0$
for $i \leftarrow 0$ **to** $size(X) - 1$ **do**
  set *histogram* $\leftarrow$ *histogram* + *calcHistogram*$(X[i])$
end for
set *avgHist* $\leftarrow$ *histogram*$/size(X)$
initialize *imgHistograms* $\leftarrow [\,]$
for $i \leftarrow 0$ **to** $size(X) - 1$ **do**
  set *avgHistN* $\leftarrow$ *normalize*(*avgHist*, *range* $\leftarrow (0, 1)$)
  set *imgHist* $\leftarrow$ *calcHistogram*$(X[i])$
  set *imgHistN* $\leftarrow$ *normalize*(*imgHist*, *range* $\leftarrow (0, 1)$)
  set *adjHist* $\leftarrow$ *abs*(*avgHistN* $-$ *imgHistN*)
  append *imgHistograms* $\leftarrow$ *adjHist*
end for
initialize *entropies* $= [\,]$
for $i \leftarrow 0$ **to** $size(imgHistograms) - 1$ **do**
  set *entr* $\leftarrow$ *calcEntropy*$(imgHistograms[i])$
  append *entropies* $\leftarrow$ *entr*
end for
return *entropies*

C. LOGITS VECTOR

In this approach, we extract the logits vector from the output of the last layer of our teacher model. Logits represent the unnormalized scores assigned to each class by the model before applying the softmax function. By capturing these raw prediction scores, we gain insights into the model's confidence levels and decision-making process.

Next, we apply the softmax function to the logits, transforming them into probabilities representing the model's predicted probability distribution over the classes. Then, we compute the entropy where p_i represents the probability of the i^{th} according to the softmax output, and N is the total number of classes. In the case of using logits from the last layer of the teacher model, $P(i)$ corresponds to the softmax probabilities generated by the teacher model, while $\log P(i)$ represents the natural logarithm of these probabilities. Figure 5 represents the logits vector representation and Algorithm 5 the image selection process. By calculating entropy on the teacher's output probability vector, we assign each image a score based on the teacher's prediction confidence, with higher entropy indicating lower confidence. By prioritizing high-entropy (low-confidence) samples during the student's training, we encourage the student to focus on more challenging examples, ultimately improving its performance across the entire dataset.

Algorithm 5 Logits Vector Entropy

CALCULATE_ENTROPY (Z)
// Calculate the entropy score on the logits vectors of the teacher. Z are the output logits of the teacher model corresponding to each sample from the dataset, and *entropies* are the calculated entropy scores.
initialize *entropies* $= [\,]$
for $i \leftarrow 0$ **to** $size(Z) - 1$ **do**
  set $p \leftarrow$ *softmax*$(Z[i])$
  append *entropies* $\leftarrow$ *calcEntropy*(p)
end for
return *entropies*

D. COMPRESSED FEATURE VECTORS

In this approach, we employ an autoencoder to capture and encode visual information from images into compressed feature vectors (Algorithm 6). The autoencoder consists of an encoder and a decoder. The encoder compresses the input images into latent vectors (latent representations), while the decoder reconstructs the input images from these latent vectors. The architecture of the autoencoder includes convolutional layers for encoding and transposed convolutional layers for decoding. During training, the autoencoder learns to minimize the reconstruction error between the input and the reconstructed output images. When assessing entropy based on latent vectors obtained from an autoencoder, $P(i)$ signifies the probability distribution of features in the latent space, with $\log P(i)$ being the natural logarithm of these probabilities.

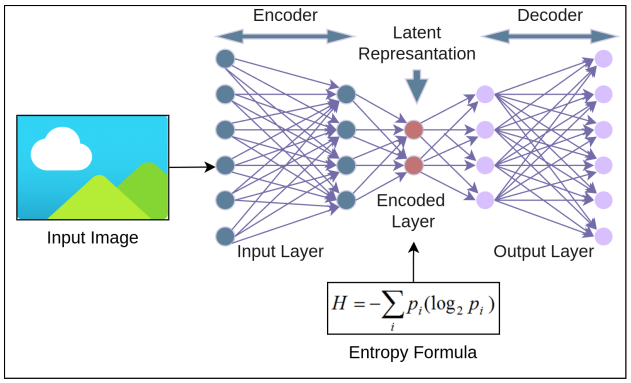

FIGURE 6. Compressed feature vector representation.

Once trained, we use the encoder part of the autoencoder to extract the latent vectors for each image in the training dataset. These latent vectors capture the essential features of the input images in a compressed form. Next, we calculate the entropy of each latent vector. By computing the entropy of the latent vectors, we gain insights into the diversity and complexity of the visual information encoded in the compressed feature space. Images with higher entropy values in their latent vectors may contain more diverse and informative features. Figure 6 represents the compressed feature vector representation.

Algorithm 6 Compressed Feature Vector Entropy

CALCULATE_ENTROPY (X)
// Compute the Compressed Feature Vector Entropy representation of each image from the dataset and compute the entropy score on these representations. X are the images from the dataset, and *entropies* are the calculated entropy scores.
set $latentVecs \leftarrow autoencoder.encoder(X)$
set $flatLatentVecs \leftarrow flatten(latentVecs)$
set $logP \leftarrow logSoftmax(flatLatentVecs)$
set $entropies \leftarrow calcEntropy(logP)$
return *entropies*

E. PATCH ENTROPY VECTORS

We divide each image into patches of size $M \times N$ and flatten these patches into vectors of pixel intensities. For each flattened patch vector, we calculate the probability of each pixel intensity value by dividing its occurrence by the size of the vector. Using these probabilities, we compute the entropy for each patch. The resulting entropy values are aggregated into a feature vector, referred to as the *patch entropy vector*. This vector represents the entire image, where each entropy value corresponds to specific regions, such as sky, ground, horizon etc. The final entropy evaluation during image selection is obtained by summing all the entropy values in the vector to obtain the image importance score (Algorithm 7).

VI. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation of our proposed methodology across three popular datasets: CIFAR-10, MNIST, and FashionMNIST. We assess the performance and efficiency of our proposed method by varying the sizes of the dataset subsets used in the knowledge distillation process and comparing the results with other approaches. In subsection VI-A, we present the performance and efficiency evaluation metrics employed. Subsection VI-B outlines the experimental setup. In subsection VI-C, we summarize the experimental outcomes, followed by a discussion of the evaluation results.

Algorithm 7 Patch Entropy Vector

CALCULATE_ENTROPY (X)
// Compute the Patch Entropy Vector representation of each image from the dataset and compute the entropy score on these representations. X are the images from the dataset, and *entropies* are the calculated entropy scores.
initialize $entropies \leftarrow [\,]$
for $i \leftarrow 0$ **to** $size(X) - 1$ **do**
// Split image into smaller square patches
  set $patches = splitImg(X[i], patchSize = M \times M)$
  for $j \leftarrow 0$ **to** $size(patches) - 1$ **do**
    set $flattenedPatch \leftarrow flatten(imagePatches[j])$
// Calculate entropy on each flattened patch of the image
    set $patchEntropy \leftarrow calcEntropy(flattenedPatch)$
// Calculate the overall entropy of the image by summing the entropies of the patches
    append $entropies \leftarrow sum(patchEntropy)$
  end for
end for
return *entropies*

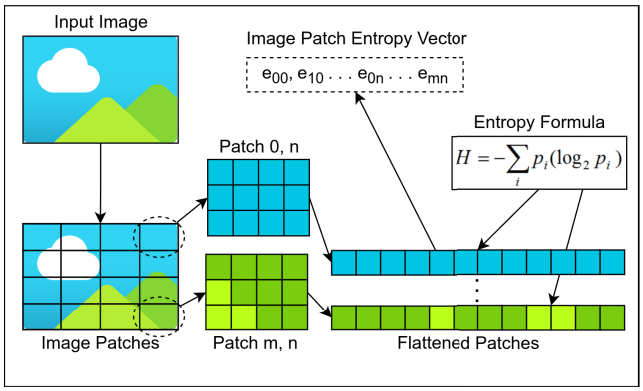

FIGURE 7. Patch entropy vectors representation.

A. EVALUATION METRICS

We evaluate the performance of the student model using the accuracy metric. In addition, we consider time and the energy consumption during the knowledge distillation process as key indicators of the efficiency and computational cost. We measured the elapsed time in hours (h) and minutes (m), noting that this includes both the image selection of the subset and the knowledge distillation phases. For brevity, we will refer to this simply as "distillation time" or "time". The energy consumption calculated with units of measurements the kilowatt-hour (KWh).

B. EXPERIMENTAL SETUP

The proposed model has been implemented in Python 3, using the libraries OpenCV, argparse, NumPy, pandas, Scikit-learn, and PyTorch. The environment used for the evaluation was an Ubuntu Linux computer with an intel i5-4670K CPU, 16 GBs of RAM at 2400 MHz and an ASUS RTX 2060 GPU. To monitor power consumption accurately, we utilized a power meter capable of measuring watt-hours (Wh) to the third decimal digit. The experiments' source code is available for any kind of reproduction and reexamination in the first author's GitHub repository.[1] We selected this configuration to closely resemble a realistic scenario involving a single, medium-range contemporary computer system. As such we evaluate the time and energy required directly on this machine under real-world conditions.

In our experiments, we employed a cross-architecture knowledge distillation approach [21], using a Visual Transformer as the teacher and a Visual Geometry Group (VGG) CNN as the student for image classification. This approach was selected to enable the student CNN to leverage the transformer's ability to capture long-range dependencies and complex patterns while retaining its computational efficiency. Specifically, we employed the Data-efficient Image Transformer (DeiT) [22], which uses self-attention mechanisms to simultaneously analyze relationships across all image regions, thereby understanding intricate patterns and structures. The distilled knowledge is then transferred to the simpler VGG CNN [23], which excels at capturing spatial hierarchies with minimal computational cost. This approach effectively balances model accuracy with resource efficiency, making it suitable for frugal machine learning applications.

We adopted pre-trained models from the torch hub repository.[2] For the teacher we used deit-tiny-patch16-224 and for the student we used VGG 19-layer model with batch normalization. We fine-tune the teacher model using the dataset employed in each experiment to enhance its feature representations and domain-specific knowledge. The VGG 19-layer model was chosen as the student model for its superior accuracy following a comparative analysis of the popular VGG-based architectures [23]. The VGG19 model consists of 19 convolutional layers organized into five blocks, each containing convolutional layers followed by batch normalization and rectified linear unit (ReLU) activation functions. Additionally, max-pooling layers are interspersed between the convolutional blocks to downsample the spatial dimensions of the feature maps, facilitating the extraction of hierarchical features from input images.

C. RESULTS & DISCUSSION

Figures 8 through 19 provide an overview of the experimental results for CIFAR-10, MNIST, and FashionMNIST, respectively. Specifically, Figures 8, 10, 12, 14, 16, and 18

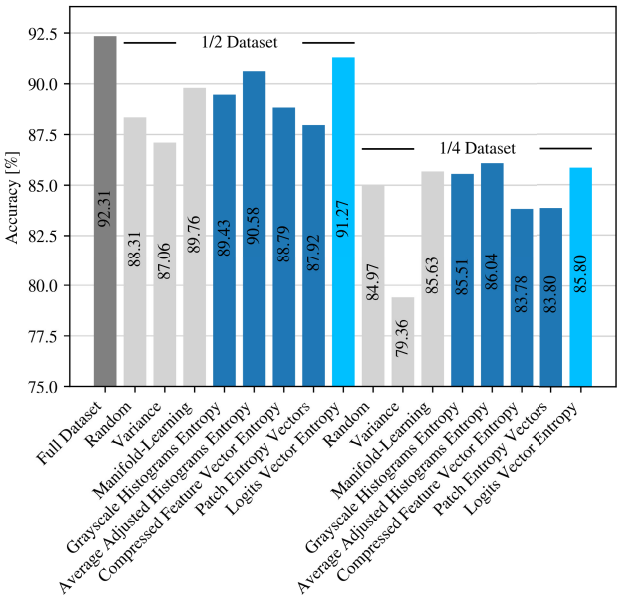

FIGURE 8. Student accuracy for CIFAR-10.

display the accuracy of the student model across various image representations, as detailed in Section V, two dataset ratios, with and without clustering selection. Our suggested representation of Highest Logits Vector Entropy is highlighted in each figure for clarity. Figures 9, 11, 13, 15, 17, and 19 illustrate the efficiency metrics for the distillation process, specifically time and energy. In each figure, energy is represented on the left y-axis in red, while time is shown on the right y-axis in green. The color of each bar corresponds to the respective axis it represents. The subsets ratios are indicated inside each figure, and are 1/2th and 1/4th the size of the original dataset, that being 25000 and 12500 images respectively for the CIFAR-10 dataset and 30000 and 15000 images respectively for the MNIST and the FashionMNIST datasets. We performed our experiments twice using two selection methods, first selecting the top N entropy samples for each class and secondly selecting top N entropy samples per cluster for each image class.

All experiments were conducted over 20 epochs with the same setup of teacher, student and knowledge distillation. To minimize computational resources, the goal is to perform knowledge distillation using the fewest possible epochs without compromising the student model's performance. Based on our observations of training and validation loss, we found that 20 epochs provide an optimal balance, preventing both underfitting and overfitting for our datasets. For better generalizability in different datasets, we recommend the readers reproducing our methodology with an early stopping criterion to avoid unnecessary training beyond the point of optimal performance.

First, we examine knowledge distillation using the original dataset without any selection (Full dataset). Next we evaluate the proposed methodology using two subset ratios with the

[1] https://github.com/michaelkinnas/Selecting-Images-with-Entropy-for-Frugal-Knowledge-Distillation

[2] https://pytorch.org/hub/

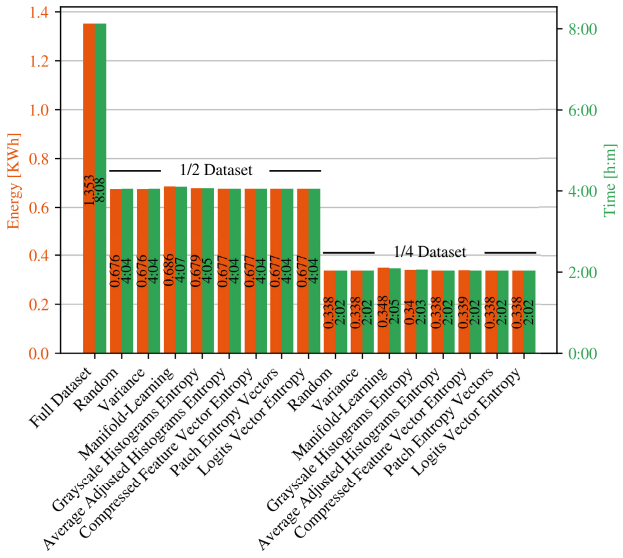

FIGURE 9. Distillation energy and time for CIFAR-10.

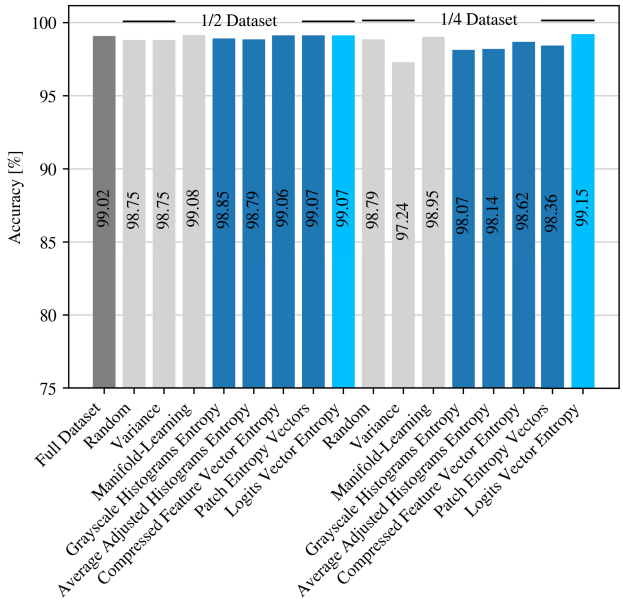

FIGURE 10. Student accuracy for MNIST.

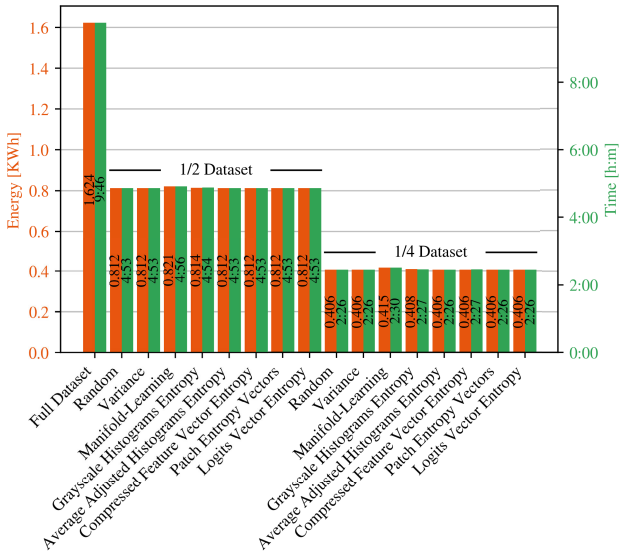

FIGURE 11. Distillation energy and time for MNIST.

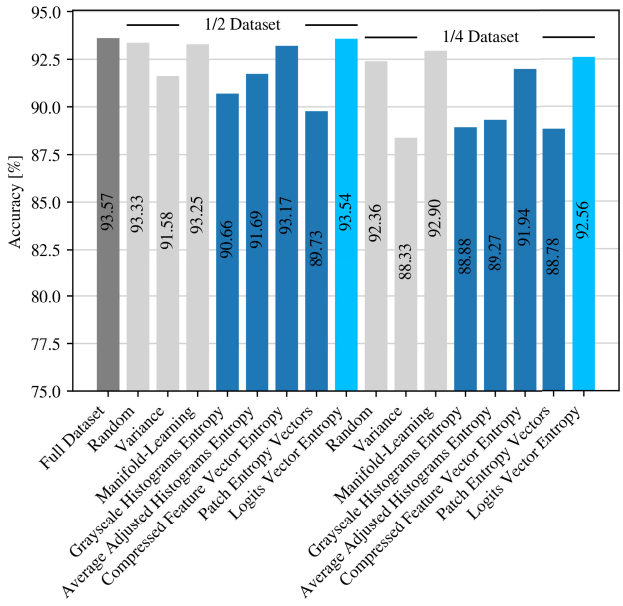

FIGURE 12. Student accuracy for fashionMNIST.

following representations as described in section V: Highest Grayscale Histogram Entropy, Highest Average Adjusted Histogram Entropy, Highest Compressed Feature Vector Entropy, Highest Patch Entropy Vectors, and Highest Logits Vector Entropy. Furthermore we examine a random selection without applying any specific representation or selection method (Random) and we also evaluate the highest variance criterion and the state-of-the-art manifold learning-based method as described in subsection II-E.

Considering energy consumption and distillation time (Figures 9, 11, 13, 15, 17, and 19), we observe consistent trends across all datasets, representation and selection methods. Specifically, there is a clear correlation between dataset size and the energy and time required for knowledge distillation. Halving the dataset size effectively reduces both the distillation time and energy consumption by half. Similarly, reducing the dataset size to one-quarter results in a proportional decrease in energy and time requirements. The preprocessing overhead introduced by most representation methods is minimal, typically under one minute of additional computation time and an insignificant amount of energy, below the third decimal place of our energy measurement device. For instance, the Highest Logits Vector Entropy method increased computation time by approximately 1.5 seconds compared to the random

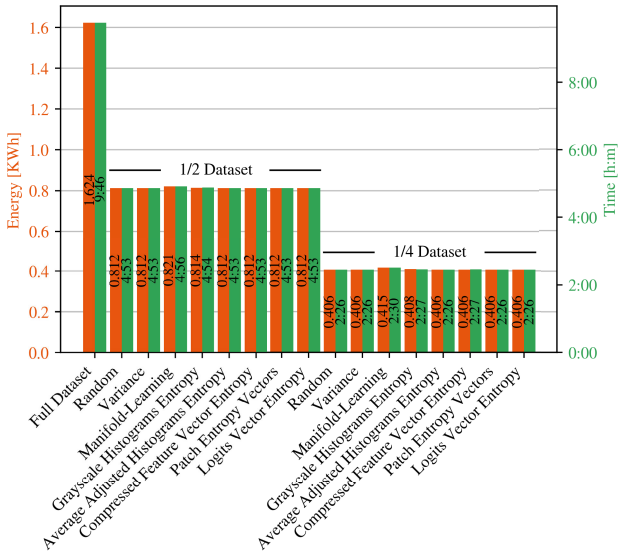

FIGURE 13. Distillation energy and time for fashionMNIST.

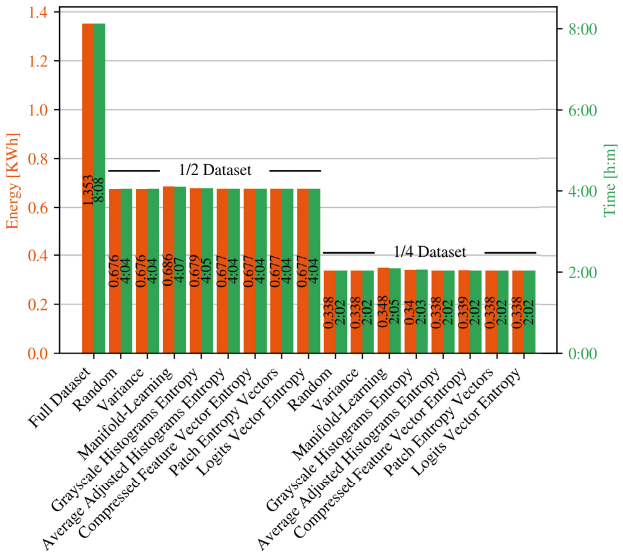

FIGURE 15. Distillation energy and time for CIFAR-10 with clustering.

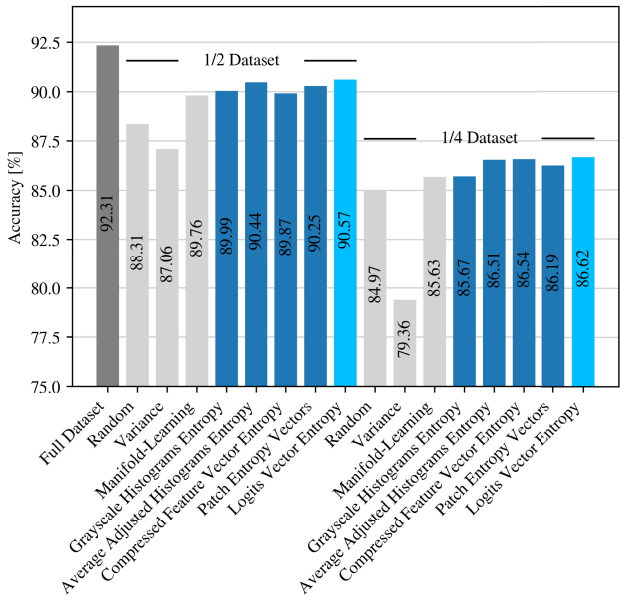

FIGURE 14. Student accuracy for CIFAR-10 with clustering.

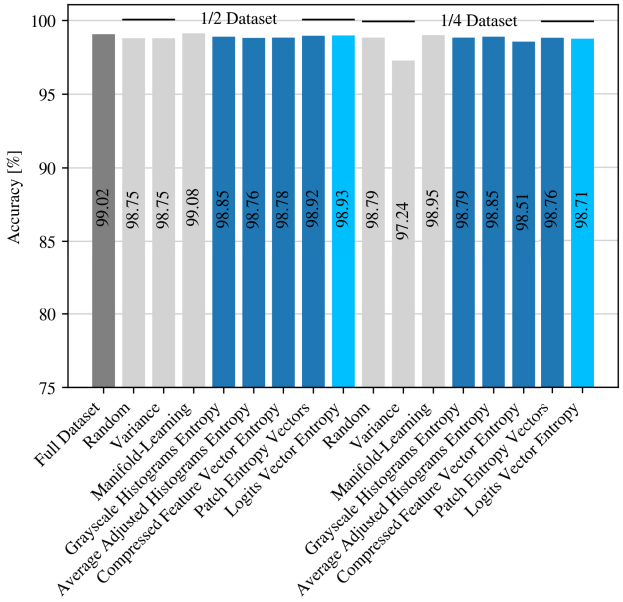

FIGURE 16. Student accuracy for MNIST with clustering.

selection method. An exception is the Manifold-Learning method, which incurred a more significant overhead, adding approximately 3 minutes of computation time and 0.010 kWh of energy across all datasets. When combined with the clustering selection method, an additional 1.5 seconds of computation time was required.

In terms of performance, the accuracy results without clustering (Figures 8, 10, and 12) reveal variability depending on the representation and selection method and dataset size. For the CIFAR-10 dataset, a random selection with a 1/2-sized subset results in a performance degradation of 4% compared to the full dataset. In contrast, our recommended method, Highest Logits Vector Entropy, exhibits a smaller degradation of 1.04%, outperforming all other methods for this subset size. With a 1/4-sized subset, the best-performing method is Manifold-Learning, which shows a 6.27% reduction in performance relative to the full dataset, followed closely by Highest Logits Vector Entropy at 6.51%. By comparison, the random selection method yields a larger degradation of 7.34%.

For the MNIST dataset, the 1/2-sized subset demonstrates a slight performance improvement over the full dataset with the Highest Logits Vector Entropy method, achieving an increase of 0.05%, while the best-performing Manifold-Learning

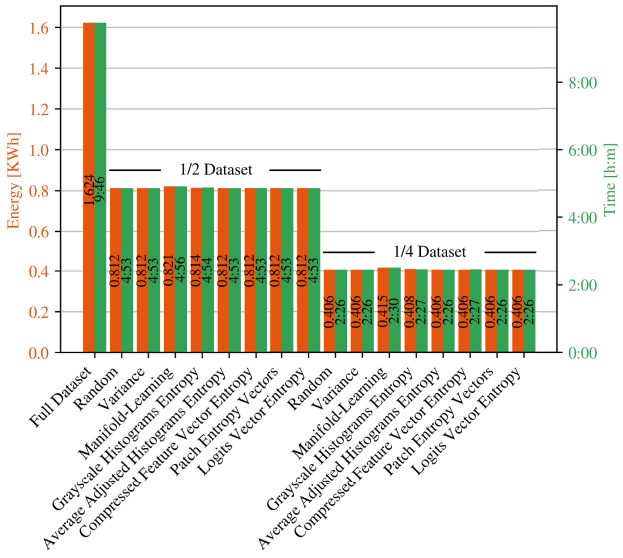

FIGURE 17. Distillation energy and time for MNIST with clustering.

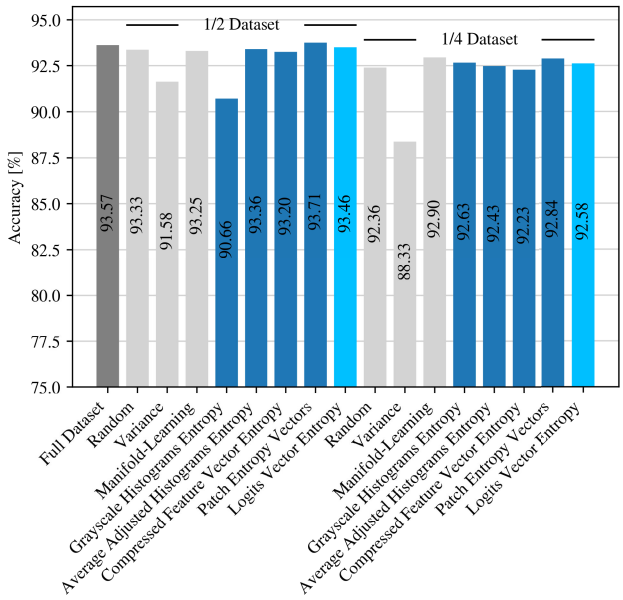

FIGURE 18. Student accuracy for fashionMNIST with clustering.

method achieves a slightly higher improvement of 0.06%. At the 1/4-sized subset, however, our recommended Highest Logits Vector Entropy representation method surpasses all others, even exceeding the performance of the full dataset by 0.13%. The reduced performance observed when using the full dataset compared to smaller subsets can be attributed to overfitting, given the simplicity of the MNIST dataset. Our primary goal is to select fewer samples based on entropy; however, a welcome bonus is that doing so also helps mitigate overfitting, and as such we considered it a success of our work.

For the FashionMNIST dataset, our recommended representation method Highest Logits Vector Entropy outperforms all others in the 1/2th dataset size, showing a 0.03%

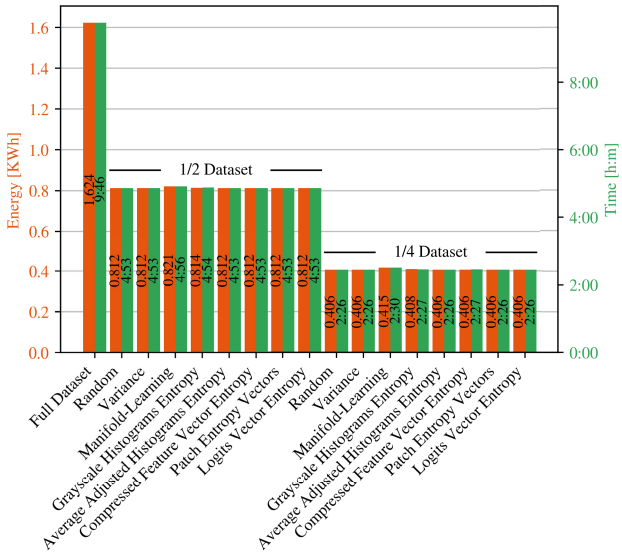

FIGURE 19. Distillation energy and time for fashionMNIST with clustering.

performance degradation vs the random selection of 0.25% over the full dataset. With the 1/4th size dataset the best performing method is the Manifold-Learning showing 0.64% performance degradation with the Highest Logits Vector Entropy method coming second at 0.98% performance degradation, while a random selection of images resulted in 1.18% performance degradation over the full dataset.

Analyzing the accuracy results when using clustering selection (Figures 14, 16, and 18), we observe mixed outcomes, with some representation methods performing better and others worse compared to non-clustering selection.

For the CIFAR-10 dataset, using the 1/2-sized subset, the Highest Logits Vector Entropy method outperforms all others, showing a 1.75% performance degradation relative to the full dataset and a 0.7% degradation compared to the non-clustering selection. At the 1/4-sized subset, our recommended method also demonstrates superior performance, with a 5.69% degradation versus 7.34% for random selection. Compared to the Highest Logits Vector Entropy method without clustering, clustering provides a 0.82% performance improvement.

For the MNIST dataset, at the 1/2-sized subset, the best-performing method is Manifold-Learning, with a slight performance increase of 0.06% over the full dataset, while our recommended method ranks close to it with a 0.09% degradation. Compared to our method without clustering, a 0.14% degradation is observed when clustering is used. At the 1/4-sized subset, Manifold-Learning performs best, with a marginal 0.02% increase over the full dataset. But performs worst compared to Highest Logits Vector Entropy method without clustering.

For the FashionMNIST dataset, at the 1/2-sized subset with clustering, we see that the entropy approach surpasses the manifold learning with the logit vector representation

showing a 0.11% degradation compared to the full dataset. However it shows a 0.09% degradation compared to non-clustering selection. It is also worth noting the good performance achieved with the patch entropy vector representation, which resulted in a 0.14% improvement over the full dataset. On the 1/4-sized subset, the Manifold-Learning method and the entropy-based approach utilizing the patch entropy vector representation deliver the best performance, with only a slight difference of 0.06% between them.

In summary, reducing the dataset size to 1/2 or 1/4 results in a performance degradation of approximately 2% and 6% at most, respectively. However, this reduction also leads to a decrease in energy consumption by around 50% and 75% directly correlating to dataset size. Furthermore, employing entropy as a sample selection criterion ensures that the selected subsets yield favorable results, outperforming random selection in most cases and matching it in a few, while never performing worse.

Furthermore, we have considered the case of blurry and noisy images. Based on our empirical experience, blurry images tend to have lower entropy scores due to their more uniform pixel distributions. Our method naturally selects less blurry images, leading to improved performance. In contrast, noisy images typically exhibit higher entropy scores, which can negatively impact performance by favoring their selection. To address this, a preprocessing step to filter out noisy samples could be incorporated. This is not included in the article, as it will be covered in a future work.

VII. CONCLUSION & FUTURE WORK

The conclusion of this work is that reducing the number of images used in knowledge distillation decreases both the time and energy consumption, but also diminishes accuracy. To mitigate this performance degradation, we select images with the highest information content for the knowledge distillation process, using the entropy formula along with the Logits vectors image representation. This approach ensures that the distilled student model remains both accurate and efficient.

As future work, we plan to propose a criterion that selects images based on a combination of high entropy and diversity. This approach will ensure that the selected dataset includes the most informative images while avoiding redundancy. By incorporating diversity, we aim to prevent the inclusion of high-entropy images that are similar based on a criterion we will investigate. Therefore, we will enhance the representativeness and overall quality of the condensed dataset. This dual-criterion selection method is expected to further improve the efficiency and performance of the knowledge distillation process. In this direction we also also aim to investigate the combination of entropy with other measures of informativeness, such as mutual information, gradient-based metrics or meta-heuristics. Exploring hybrid approaches that integrate multiple metrics may lead to even more effective subset selection.

REFERENCES

[1] H.-I. Liu, M. Galindo, H. Xie, L.-K. Wong, H.-H. Shuai, Y.-H. Li, and W.-H. Cheng, "Lightweight deep learning for resource-constrained environments: A survey," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–42, Oct. 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3657282

[2] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021, doi: 10.1007/s11263-021-01453-z.

[3] M. Evchenko, J. Vanschoren, H. H. Hoos, M. Schoenauer, and M. Sebag, "Frugal machine learning," 2021, *arXiv:2111.03731*.

[4] J. Park, S. Wang, A. Elgabli, S. Oh, E. Jeong, H. Cha, H. Kim, S.-L. Kim, and M. Bennis, "Distilling on-device intelligence at the network edge," 2019, *arXiv:1908.05895*.

[5] S. E. Whang and J.-G. Lee, "Data collection and quality challenges for deep learning," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3429–3432, Aug. 2020, doi: 10.14778/3415478.3415562.

[6] K. Lee and S. Lee, "A new framework for measuring 2D and 3D visual information in terms of entropy," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2015–2027, Nov. 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7254154

[7] S. Chen, S. Dorn, M. Lell, M. Kachelrieß, and A. Maier, "Manifold learning-based data sampling for model training," in *Bildverarbeitung Für Die Medizin 2018*. Berlin, Germany: Springer, 2018, pp. 269–274. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-56537-7_70

[8] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1595–1623, Apr. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821003335

[9] X. Tu, A. Mallik, D. Chen, K. Han, O. Altintas, H. Wang, and J. Xie, "Unveiling energy efficiency in deep learning: Measurement, prediction, and scoring across edge devices," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Dec. 2023, pp. 80–93. [Online]. Available: https://ieeexplore.ieee.org/document/10419248

[10] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*.

[11] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, Jun. 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9340578

[12] Y. He, L. Xiao, J. T. Zhou, and I. Tsang, "Multisize dataset condensation," 2024, *arXiv:2403.06075*.

[13] J.-H. Kim, J. Kim, S. Oh, S. Yun, H. Song, J. Jeong, J.-W. Ha, and H. O. Song, "Dataset condensation via efficient synthetic-data parameterization," in *Proc. 39th Int. Conf. Mach. Learn.*, Jun. 2022, pp. 11102–11118. [Online]. Available: https://proceedings.mlr.press/v162/kim22c.html

[14] B. Zhao, K. R. Mopuri, and H. Bilen, "Dataset condensation with gradient matching," 2020, *arXiv:2006.05929*.

[15] B. Zhao and H. Bilen, "Dataset condensation with distribution matching," 2021, *arXiv:2110.04181*.

[16] A. Munappy, J. Bosch, H. H. Olsson, A. Arpteg, and B. Brinne, "Data management challenges for deep learning," in *Proc. 45th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2019, pp. 140–147. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8906736

[17] J. M. Johnson and T. M. Khoshgoftaar, "The effects of data sampling with deep learning and highly imbalanced big data," *Inf. Syst. Frontiers*, vol. 22, no. 5, pp. 1113–1131, Oct. 2020, doi: 10.1007/s10796-020-10022-7.

[18] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," 2021, *arXiv:2107.07075*.

[19] S. H. Ebenuwa, M. S. Sharif, M. Alazab, and A. Al-Nemrat, "Variance ranking attributes selection techniques for binary classification problem in imbalance data," *IEEE Access*, vol. 7, pp. 24649–24666, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8651567

[20] L. C. Tan, H. Yazid, and Y. F. Chong, "Image quality assessment (IQA) using high-frequency and image variance (HFIV) for colour image," *J. Phys., Conf. Ser.*, vol. 1372, no. 1, Nov. 2019, Art. no. 012034, doi: 10.1088/1742-6596/1372/1/012034.

[21] Y. Liu, J. Cao, B. Li, W. Hu, J. Ding, and L. Li, "Cross-architecture knowledge distillation," 2022, *arXiv:2207.05273*.

[22] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 10347–10357. [Online]. Available: https://proceedings.mlr.press/v139/touvron21a.html
[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

MICHAIL KINNAS received the B.Sc. degree (Hons.) from the Department of Informatics and Telematics, Harokopio University of Athens, Greece, in 2024. He is currently working as a Researcher Intern with Information Technologies Institute, Centre for Research and Technology Hellas. His research interests include machine learning, artificial neural networks, and computer vision.

JOHN VIOLOS is currently working with the Center for Research and Technology Hellas. He is also an Adjunct Professor with the Harokopio University of Athens, and the National and Kapodistrian University of Athens. Previously, he was a Senior Researcher at the Department of Software Engineering and Information Technology, École de Technologie Supérieure, Université du Québec, Montreal, Canada; and a Senior Researcher at the National Technical University of Athens, Greece. He has more than 50 publications in top-tier conferences and journals. He was a member of European Commission's Digital Single Market working group on the code of conduct for switching and porting data between cloud service providers. He has participated as Work Package Leader, a Task Leader, and a Researcher in more than 14 research projects funded by European Union, South Korea, and Canada. He received the Best Paper Award of the IEEE CISOSE 2023 Conference, the Best Paper Award of the IEEE iThings 2021 Conference, and the Best Course Teaching by an Adjunct Professor Award of the Harokopio University of Athens, in 2021.

NIKOLAOS IOANNIS KARAPIPERIS received the B.Sc. degree from the Department of Informatics and Telematics, Harokopio University of Athens, Greece, in 2024. He is currently working as a Software Engineer with Accenture. He began his professional career as a Junior Software Engineer at European Dynamics, where he worked for 1.5 years during his third year of studies. His professional research interests include computer science, data science, software development, and back-end engineering.

IOANNIS KOMPATSIARIS (Senior Member, IEEE) is currently working as a Research Director with ITI, CERTH, the Head of Multimedia Knowledge and Social Media Analytics Laboratory, and the Deputy Director of Institute. He is the co-author of 171 papers in refereed journals, 63 book chapters, eight patents, and more than 500 papers in international conferences. Since 2001, he has been participated in 89 National and European research programs, 18 of which he has been the Project Coordinator. He has also been the PI in 14 research collaborations with industry, including Motorola U.S. and U.K. His research interests include image and video analysis, big data and social media analytics, semantics, human–computer interfaces (AR and BCI), multimedia, eHealth, security, and culture applications. He is a member of the Scientific Advisory Board of the CHIST-ERA funding program, an Elected Member of the IEEE Image, Video, and Multidimensional Signal Processing—Technical Committee (IVMSP—TC), and a member of ACM. He has been the Co-Chair of various international conferences and workshops, including the 13th IEEE Image, Video, and Multidimensional Signal Processing (IVMSP 2018) Workshop, and he has served as a regular reviewer, an associate editor, and the guest editor for a number of journals and conferences. He is currently an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING.