Expand-and-Quantize: Unsupervised Semantic Segmentation Using High-Dimensional Space and Product Quantization

Jiyoung Kim, Kyuhong Shim, Insu Lee, Byonghyo Shim

Department of Electrical and Computer Engineering, Seoul National University, Korea
{jykim, khshim, islee, bshim}@islab.snu.ac.kr

Abstract

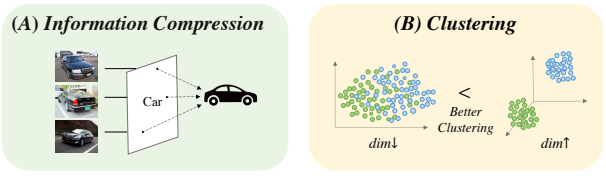
Unsupervised semantic segmentation (USS) aims to discover and recognize meaningful categories without any labels. For a successful USS, two key abilities are required: 1) information compression and 2) clustering capability. Previous methods have relied on feature dimension reduction for information compression, however, this approach may hinder the process of clustering. In this paper, we propose a novel USS framework called Expand-and-Quantize Unsupervised Semantic Segmentation (EQUSS), which combines the benefits of high-dimensional spaces for better clustering and product quantization for effective information compression. Our extensive experiments demonstrate that EQUSS achieves state-of-the-art results on three standard benchmarks. In addition, we analyze the entropy of USS features, which is the first step towards understanding USS from the perspective of information theory.

1 Introduction

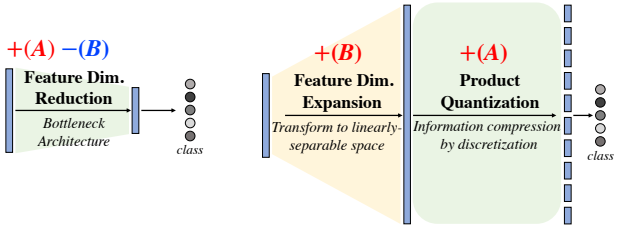
Semantic segmentation, a task to classify every pixel to the proper category, has numerous applications including autonomous driving (Feng et al. 2020; Siam et al. 2018), scene understanding (Meletis 2022; Kim, Park, and Shim 2023), and medical diagnostics (Asgari Taghanaki et al. 2021; Hatamizadeh et al. 2022). Despite its significance, dissemination of supervised semantic segmentation is a bit slow, in particular for real-world applications due to the difficulty and cost of the pixel-wise annotations. As a surrogate, unsupervised semantic segmentation (USS) has received much attention recently since it can save the time and effort to label the dataset (Seong et al. 2023; Hamilton et al. 2022; Cho et al. 2021).

In most USS models, a pre-trained 'backbone' module is employed to extract rich features from the input image. The output of the backbone is a high-dimensional feature vector containing both class-relevant and class-irrelevant information (Ji, Henriques, and Vedaldi 2019; Kim, Shim, and Shim 2022). In the absence of meaningful guidance, a USS model cannot efficiently judge which pixels are semantically similar. To handle the issue, a USS 'head' module should refine the feature such that the class-relevant information is preserved while class-irrelevant information is discarded. In the information-theoretic perspective, this process can be considered as a form of *lossy information compression* (Dameau 1964). For instance, if there are 27 classes (Caesar, Uijlings, and Ferrari 2018), a perfect classification can be achieved with 5 bits, which is far smaller than the amount of information contained in the 384-dim feature vector (384 elements × 32-bit floating point = 12,288 bits). After the information compression, USS model clusters the outputs using k-means such that pixels in the same group are aggregated to the same class.

So far, the primary goal of USS frameworks is to compress information while preserving as much class-relevant information as possible. Popularly used method to achieve the goal is *feature dimension reduction* (e.g., 384-dim to 70-dim) for information compression, followed by training to retain class-relevant information in the compressed feature (Ouali, Hudelot, and Tami 2020; Ji, Henriques, and



(a) Key factors for successful USS



(b) Conventional USS (c) Proposed USS

Figure 1: To achieve success in USS, two key factors should be considered: (*A*) Information Compression and (*B*) Clustering. The conventional model typically employs feature dimension reduction for (*A*), but it can be counterproductive for (*B*). In contrast, our proposed model takes advantage of both by employing feature dimension expansion for (*B*) and product quantization for (*A*).

Vedaldi 2019; Cho et al. 2021; Yin et al. 2022; Hamilton et al. 2022) (see Fig. 1(b)). Despite its design simplicity and low-complexity, conventional USS framework has a serious shortcoming; loss of class-relevant information and introduction of quantization noise. Thus, without proper guidance, features cannot be *easily clustered* in a metric space. The moral of the story is that the features should achieve a higher level of 'clusterability'. In the sequel, we use this term as the capability to achieve accurate decision boundaries between classes.

To facilitate the clustering of features (i.e., better clusterability), we cast the feature vector into a high-dimensional space. This approach is motivated by the well-known fact that a nonlinear transformation to a high-dimensional latent space can enhance the model's ability to find hyperplanes that properly separate classes (Hofmann, Schölkopf, and Smola 2008). Essentially, as the dimension increases, the volume of the space where the feature vector exists also increases exponentially. This makes the class-relevant information more sparsely distributed, facilitating the model easier to identify linear decision boundaries between groups of data (classes). As a result, training methods for enhancing the class-relevance of features can be more effective in high-dimensional space than in low-dimensional space.

Since previous USS approaches adopt feature dimension reduction architecture, they could not achieve both the benefits of high-dimensional space and information compression at the same time (see Fig. 1(b)). In contrast, we aggressively exploit *feature dimension expansion* to improve clusterability for USS. To make the most of the benefits of high-dimensional space while achieving information compression, we exploit the *product quantization* (see Fig. 1(c)). Intuitively, this quantization process can be interpreted as a filtering mechanism that preserves only the common information represented by cluster centroids and discards unwanted class-irrelevant information and the quantization noise. We expect that the centroids represent the class-relevant information, as long as the features are properly clustered.

In this paper, we propose an entirely different USS framework referred to as Expand-and-Quantize Unsupervised Semantic Segmentation (**EQUSS**). In order to take advantage of both high-dimensional space for improved clustering and PQ for class-aware information compression, we employ two-stage processing, viz., 'expand' and 'quantize'. To the best of our best knowledge, we are the first to employ either high-dimensional features or PQ in previous USS literature.

By measuring the number of bits to represent a class, we show that EQUSS requires significantly smaller number of bits than the previous state-of-the-art (SOTA) method (Hamilton et al. 2022) (246 bits vs. 475 bits). We also show that the number of bits to the class tends to be proportional to the representation difficulty of the class. For example, EQUSS allocates fewer bits for classes with small variations (i.e., sky, water) while relatively large number of bits for classes with divergent appearances (i.e., sports, indoor). We find that this behavior is a unique characteristic of EQUSS and is not observed in other USS models. From our empirical experiments on CocoStuff-27 (Caesar, Uijlings, and Ferrari 2018), Cityscapes (Cordts et al. 2016), and Potsdam-3 USS benchmarks, we show that the proposed EQUSS outperforms the recent SOTA (Hamilton et al. 2022) by a substantial margin.

The contributions of our work are as follows:

- We propose a novel USS framework referred to as EQUSS, that utilizes high-dimensional spaces to improve clusterability. We are the first to recognize the critical role of clusterability in USS architecture design.
- Instead of the feature dimension reduction, we use the product quantization for the information compression. From our experiments, we show that EQUSS achieves SOTA performance on three USS benchmarks in all metrics.
- We quantify the information capacity of USS features in terms of *bits*. Our analyses establish a relationship between the entropy of features and segmentation accuracy, providing valuable insights guideline for USS design.

2 Related Work

2.1 Unsupervised Semantic Segmentation

Over the years, various training methods for USS have been proposed to retain class-relevant information while removing irrelevant ones. Due to the absence of labels, previous USS approaches focused on developing reliable guidance that is presumably related to the class. One line of research is the mutual information-based learning (Ji, Henriques, and Vedaldi 2019; Ouali, Hudelot, and Tami 2020). The main idea behind this approach is that each pixel's class remains the same after image augmentations. For example, IIC (Ji, Henriques, and Vedaldi 2019) maximizes the mutual information between different views of the inputs generated by the augmentation. Similarly, PiCIE (Cho et al. 2021) exploits multiple photometric and geometric transformations to learn consistent class representations. Another line of research is the k-means clustering-based methods (Caron et al. 2018; Cho et al. 2021; Yin et al. 2022) that generate pixel-level pseudo-labels to train the model with a classification loss. TransFGU (Yin et al. 2022) obtains high-level semantic information from feature clustering and generates pseudo-labels from Grad-CAM (Selvaraju et al. 2017). These pseudo-labels can be refined iteratively by altering between clustering and training steps (Cho et al. 2021). Recently, STEGO (Hamilton et al. 2022) suggests the feature correspondence distillation from backbone features to head features. A common ingredient in the aforementioned studies is to use the feature dimension reduction as a means of information compression and then apply the training techniques to the compressed features. In contrast, in our EQUSS, we expand the feature dimension and train these expanded features to contain class-relevant information.

2.2 Information Bottleneck

Feature dimension reduction has been widely used in many fields of designing information bottleneck architecture (Goldfeld and Polyanskiy 2020; Tishby, Pereira, and Bialek 2000). This architecture has proven to be useful in

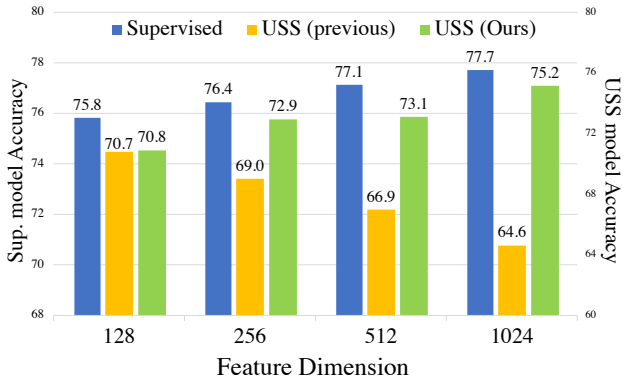


Figure 2: The relationship between feature dimension size and segmentation accuracy on the supervised model, previous USS model (Hamilton et al. 2022), and our proposed USS model.

a wide range of applications, including image classification (He et al. 2015), latent representation learning (Kingma and Welling 2013), model compression (Yu et al. 2017), and lightweight model adaptation (Hu et al. 2022). These applications, except for USS, typically have clear guidance for compression. For example, in the image classification, the model learns to identify classes using the supervised learning. In the latent representation learning, a model aims to find a compact representation of an input by minimizing the reconstruction error. As discussed, USS lacks such clear guidance so the feature dimension reduction might not be a proper choice. Instead of relying on the feature dimension reduction, our EQUSS exploits the product quantization for information bottleneck design.

2.3 Product Quantization

Product Quantization (PQ) was initially proposed for the nearest neighbor search (Jegou, Douze, and Schmid 2010). Due to its ability to compress high-dimensional vectors with extremely fast searching, PQ has been popularly used in large-scale retrieval tasks (Jégou et al. 2010; Jang and Cho 2021; Klein and Wolf 2019). For example, SPQ (Jang and Cho 2021) has employed PQ for the unsupervised image retrieval by jointly learning codewords and feature extractors in a self-supervised manner. Recently, PQ has been applied to other domains such as neural network compression (Huang et al. 2022; Yu et al. 2020; Stock et al. 2021), text-to-speech synthesis (Chen, Watanabe, and Rudnicky 2023; Guo et al. 2022), and zero-shot learning (Li et al. 2019). While these studies mainly utilize PQ's ability for the fast indexing and data size reduction, we exploit PQ in the information compression. To the best of our knowledge, we are the first to utilize PQ for semantic segmentation.

3 Expand-and-Quantize Unsupervised Semantic Segmentation (EQUSS)

In this section, we first present motivating experiments describing our central idea and then discuss two key components of EQUSS: feature dimension expansion and product quantization. Fig. 3 illustrates the overview of the proposed framework, EQUSS.

3.1 Motivation

In Fig. 2, we plot the segmentation accuracy of the supervised and unsupervised semantic segmentation models for different feature dimensions. We show that an increase in the feature dimension is beneficial for the supervised model but not for the previous USS model (see blue and yellow bars in Fig. 2).

We believe that feature dimension expansion can help the supervised model accurately delineate the decision boundary between different classes. This can be supported by the celebrated Cover's theorem (Cover 1965), saying that patterns (classes in our context) that are difficult to separate in low-dimensional space might be better separable when projected into high-dimensional space by the nonlinear transformation. In other words, feature dimension expansion can contribute to better clusterability. We note that such behavior cannot be contained by the conventional USS model since the dimension reduction significantly degrades the clusterability. We claim that naively increasing the feature dimension may result in the loss of the ability to compress information. Despite the potential benefits of high-dimensional features for clustering, the absence of information compression causes a detrimental performance loss.

Motivated by these findings, we raise a question: Can we achieve the best of both, better clusterability and information compression? In this paper, we propose a novel architecture that leverages the expansion of feature dimensions without sacrificing the capability to compress information.

3.2 Feature Extraction

Given an unlabeled dataset $D = \{I_1, I_2, \ldots, I_N\}$ of N training images, the goal of USS is to learn a model that maps $I(i,j) \mapsto c \in C$, where (i,j) represents a spatial location in an image I and C indicates a set of classes. To extract rich visual features, the backbone model $\mathcal{F}$ takes an image I and returns the features $F \in \mathbb{R}^{d_F \times h \times w}$ where d_F and (h,w) indicate the feature dimension and spatial size, respectively. We denote a feature vector at (i,j) as $f_{(i,j)} = F(i,j) \in \mathbb{R}^{d_F}$; for simplicity, we omit the spatial indices (i,j) for f.

3.3 Feature Dimension Expansion and Product Quantization

Expansion For the information refinement process discussed in Sec. 1, the extracted feature f, obtained from the backbone, passes through an expansion head $\mathcal{E}$. Essentially, the role of this head is to perform the nonlinear transform on f to generate the expanded feature vector $x \in \mathbb{R}^{d_E}$ where d_E is the dimension of the expanded feature. Note that $d_E \ll d_F$ for the conventional USS framework and $d_E > d_F$ for our EQUSS framework.

Quantization In order to compress the information, the quantization head $\mathcal{Q}$ takes an expanded feature vector x as an input and generates a quantized output $q \in \mathbb{R}^{d_E}$. Note that the quantization process does not change the feature

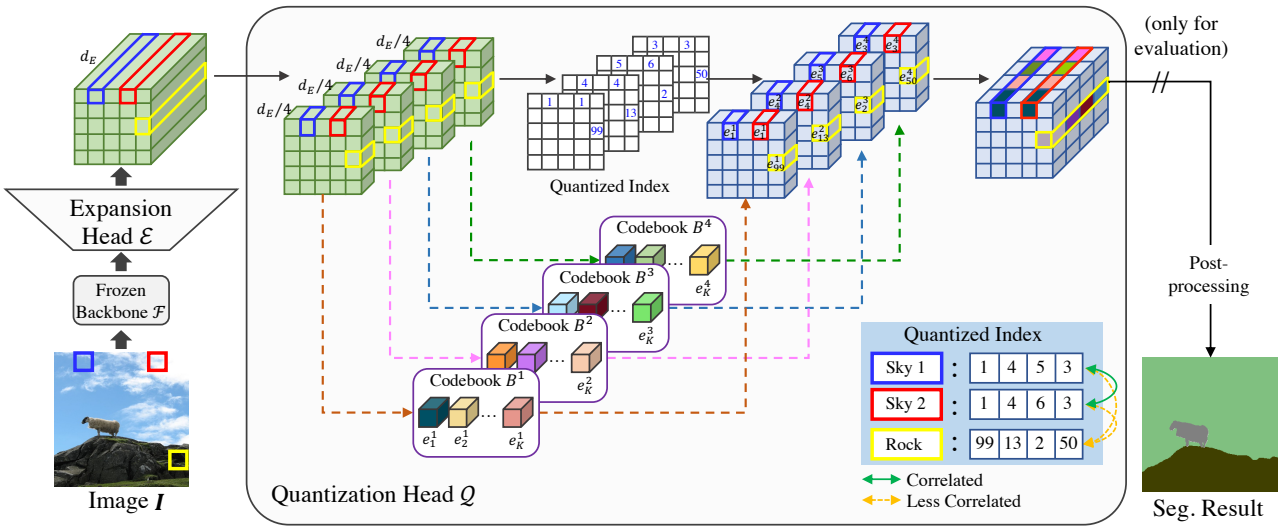


Figure 3: Overview of the proposed EQUSS. The expansion head $\mathcal{E}$ first expands the feature extracted from the backbone $\mathcal{F}$ into high-dimensional spaces. Then, the quantization head $\mathcal{Q}$ applies product quantization to generate the quantized output. Finally, this output is used for clustering and linear probing during the evaluation. For a better understanding, we set $M = 4$.

dimension. In our study, we employ PQ instead of vector quantization (VQ). The essence of PQ is to decompose a high-dimensional space into a Cartesian product of low-dimensional subspaces and then individually apply VQ for each partition. In doing so, PQ can exponentially increase the pool of possible candidates without greatly increasing the number of codewords for each codebook.

Specifically, the quantization head $\mathcal{Q}$ consists of M codebooks $\{B^1, \ldots, B^M\}$, each of which contains K codewords as $B^m = \{e_1^m, \ldots, e_K^m\}$ where $e_j^i \in \mathbb{R}^{d_E/M}$ is the j-th codeword in the i-th codebook B^i. In the first stage of PQ, x is divided into multiple subvectors $\{x^1, \ldots, x^M\}$ as

$$x = \texttt{concatenate}[x^1, \ldots, x^M], \quad x^i \in \mathbb{R}^{d_E/M} \tag{1}$$

For the second stage, each subvector x^m is mapped to the closest codeword in the m-th codebook as:

$$q^m = Q^m(x^m) = e_k^m, \quad \text{where } k = \arg\min_j \|\hat{x}^m - \hat{e}_j^m\|_2^2. \tag{2}$$

Q^m is a quantization operator for m-th subvector. The distance function is a squared Euclidean distance between $\hat{x}^i$ (= $x^i/\|x^i\|_2$) and $\hat{e}_j^i$ (= $e_j^i/\|e_j^i\|_2$). We empirically discover that this normalization is essential for stabilizing the training as we initialize codewords from a random distribution. Finally, by concatenating M quantized outputs $\{q^1, \ldots, q^M\}$, we obtain the final output $q = \texttt{concatenate}[q^1, \ldots, q^M]$.

3.4 Training Objective

The loss for PQ consists of two components: codebook loss and commitment loss. The codebook loss minimizes the distance between each subvector and its selected codeword. Then, the commitment loss encourages the subvector to stay close to the chosen codeword. The codebook loss and the commitment loss are formally defined as:

$$L_{codebook} = \frac{1}{M}\sum_{m=1}^{M} \|sg[x^m] - e_k^m\|_2^2 \tag{3}$$

$$L_{commit} = \frac{1}{M}\sum_{m=1}^{M} \|x^m - sg[e_k^m]\|_2^2 \tag{4}$$

where sg denotes the stop-gradient operation. Since the operation $\arg\min$ in Eq. (2) is non-differentiable, we approximate the gradient of x^i by copying gradients from a selected codeword e_k^i. Note that this process is known as the straight-through gradient estimation technique (Bengio, Léonard, and Courville 2013).

The overall objective function is the sum of the training loss for the expansion head L_{head} (Hamilton et al. 2022), codebook loss $L_{codebook}$, and commitment loss L_{commit}:

$$L = L_{head} + \lambda_1 L_{codebook} + \lambda_2 L_{commit} \tag{5}$$

where $\lambda_1, \lambda_2 > 0$ are weighting coefficients. Please refer to previous work (Hamilton et al. 2022) for more details about L_{head}.

3.5 Other Details

Evaluation Protocol For measuring the performance of the USS model, we need additional steps that map the learned feature q to the ground truth label c. The USS performance is evaluated with two measurement processes, namely, linear probing and unsupervised clustering. Note that these measuring processes do not update the model parameters as they are completely separated from USS training (see Fig. 3).

CRF Using Conditional Random Field (CRF) (Krähenbühl and Koltun 2011), a widely used post-processing in semantic segmentation, one can optionally refine the final segmentation mask. CRF is a probabilistic graphical model that utilizes the relationship between neighboring pixels, such as their distances or RGB values. CRF improves the result by sharpening the edges and reducing the noises based on statistical relationships between pixels.

3.6 Measuring the Information Capacity

To quantify the ability for preserving class-relevant information after the information compression, we measure the Shannon entropy (Shannon 1948). Intuitively, an increase in class-irrelevant information will cause an increase in the number of bits to represent the feature.

Since the mechanism of EQUSS is based on PQ, we can easily obtain an empirical probability mass function (PMF) over codewords in each codebook based on the frequency. For the given set of features X, we first calculate p_i^m, a probability that a subvector x^m is mapped to the e_i^m as

$$\begin{aligned} p_i^m &= \mathbb{E}_X[P(q^m = e_i^m)] \\ &\approx \frac{1}{|X|} \sum_{x \in X} \mathbb{I}(Q^m(x^m) = e_i^m) \end{aligned} \tag{6}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition is true and 0 otherwise. In practice, we approximate the expectation using a sufficient number of randomly selected features. To quantify the amount of information on features, we define the sum of entropy $J(X)$ as

$$J(X) = \sum_{m=1}^{M} \left(-\sum_{i=1}^{K} p_i^m \log_2(p_i^m) \right). \tag{7}$$

Note that $J(X)$ can be interpreted as the number of bits to represent the entire feature. In Sec. 5.1, we compare the per-class entropy of EQUSS and STEGO (Hamilton et al. 2022) for information capacity in representing each class.

4 Experiments

We evaluate EQUSS on three standard semantic segmentation datasets and compare with the recent SOTA methods.

4.1 USS Performance

Table 1 compares the performance of EQUSS with the recent USS models on CocoStuff-27. EQUSS outperforms previous works by a large margin in all metrics. In particular, EQUSS outperforms STEGO (Hamilton et al. 2022) by **+5.5** in the unsupervised accuracy and **+2.9** in the linear mIoU, respectively. Table 2a and 2b report the USS performance on Cityscapes and Potsdam-3, respectively. While STEGO and TransFGU (Yin et al. 2022) suffer from the trade-off between accuracy and mIoU on Cityscapes, EQUSS shows outstanding performances in both accuracy and mIoU. In Potsdam-3, EQUSS outperforms STEGO by **+5.0** in the unsupervised accuracy. We guess the performance gain is mainly due to 1) better clusterability of high-dimensional space and 2) effective compression of the class-irrelevant information.

Model	Unsupervised		Linear Probe	
	Acc.	mIoU	Acc.	mIoU
ResNet50 (He et al. 2015)	24.6	8.9	41.3	10.2
MoCoV2 (Chen et al. 2020)	25.2	10.4	44.4	13.2
DINO (Caron et al. 2021)	30.5	9.6	66.8	29.4
Deep Cluster	19.9	-	-	-
SIFT (Lowe 1999)	20.2	-	-	-
AC	30.8	-	-	-
IIC	21.8	6.7	44.5	8.4
MDC (Cho et al. 2021)	32.2	9.8	48.6	13.3
PiCIE	48.1	13.8	54.2	13.9
PiCIE+H (Cho et al. 2021)	50.0	14.4	54.8	14.8
ACSeg (Li et al. 2023)	-	16.4	-	-
TransFGU	52.7	17.5	-	-
STEGO	48.3	24.5	74.4	38.3
EQUSS	**53.8**	**25.8**	**75.2**	**41.2**

Table 1: USS performance comparison on CocoStuff-27.

Model	Unsupervised	
	Acc.	mIoU
IIC	47.9	6.4
MDC	40.7	7.1
PiCIE	65.5	12.3
TransFGU	77.9	16.8
STEGO	73.2	21.0
EQUSS	**79.9**	**22.0**

(a) Cityscapes

Model	U.Acc.
Deep Cluster	41.7
SIFT	38.2
Isola et al.	63.9
IIC	65.1
STEGO	77.0
EQUSS	**82.0**

(b) Potsdam-3

Table 2: Performance comparisons on (a) Cityscapes and (b) Potsdam-3.

4.2 Qualitative Result

In Fig. 5, we present the segmentation results of EQUSS and STEGO on CocoStuff-27. We observe that STEGO is often biased to certain visual information such as colors or edges, which causes the mask of an object to be split into multiple parts. For example, in Fig. 5 (a) and (h), the written texts on the wall and rock behind a giraffe are inaccurately predicted. In contrast, EQUSS accurately groups the pixels with diverse characteristics into the same class. This is because EQUSS can effectively compress class-irrelevant information and output features that are highly correlated with classes.

5 Analysis

5.1 Relationship between Entropy and Accuracy

As discussed in Sec. 3.6, we quantify the information capacity of features in terms of bits. Specifically, we compare the per-class entropy of EQUSS and STEGO on the CocoStuff-27 (see Fig. 4(a)). Unlike EQUSS, STEGO outputs a continuous variable, so we cannot directly measure the entropy using Eq. (7). To deal with the problem, we discretize the continuous distribution by histogram and calculate entropy for each feature dimension based on the frequency.

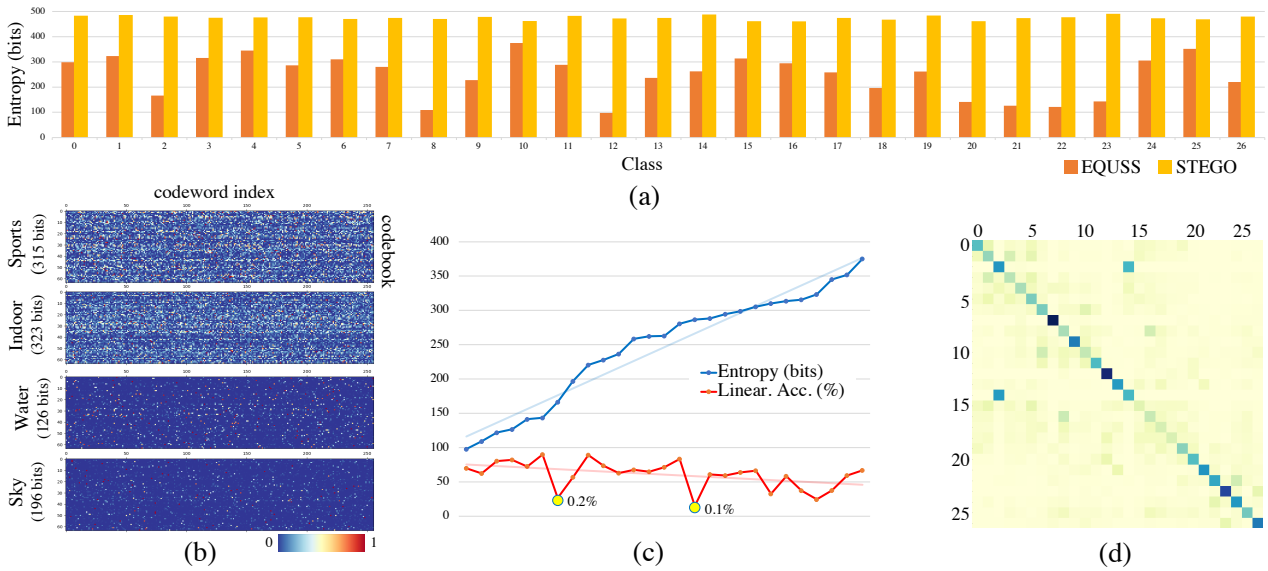


Figure 4: Analyses on EQUSS. (a) Per-class entropy of EQUSS and STEGO (Hamilton et al. 2022). (b) Frequency of each codeword to be selected, along with its corresponding codebook. (c) Relationship between entropy and accuracy. Yellow points with particularly low accuracy correspond to the most infrequent minority classes (0.1% and 0.2%). (d) Inter-class distance between the combination of codewords. Each row and column corresponds to the class index. The darker the color, the closer the distance.

The most noticeable observation is that the entropy of EQUSS is far smaller than that of STEGO, 246 bits vs. 475 bits, on average. Intuitively, the number of bits corresponds to the number of distinct representations for each class. If there are few combinations of codewords for representing a certain class, we can assume that such candidates contain common core information shared among the samples in that class. Consequently, when the class is represented with fewer candidates, the output features are also more likely to contain class-relevant information, empirically supported by our superior USS performance.

Interestingly, unlike STEGO, EQUSS shows different entropy values for different classes. To better understand this relationship, we plot the frequency of each codeword (x-axis) being selected for each codebook (y-axis) in Fig. 4(b). We observe that classes containing various objects tend to have high entropy, while those with similar appearances have low entropy. For example, in classes consisting of various objects such as 'Sports' (e.g., surfboard, kite, skis, ...) and 'Indoor' (e.g., toothbrush, vase, clock, ...), a variety of codewords are activated. In contrast, classes that consistently appear in similar forms such as 'Sky' and 'Water' tend to concentrate on specific codewords that are assigned more frequently than others. Our analysis verifies that EQUSS indeed reflects the diversity of classes, which aligns with our common intuition.

We conjecture that the class with diverse appearance may confuse the model to learn consistent representation, thus leading to poor performance. To investigate this, we plot per-class accuracy and its corresponding entropy in Fig. 4(c). Note that the classes are sorted in ascending order according to their entropy values. As expected, the accuracy tends to decrease as the number of bits needed to represent a class increases. From these results, we establish the quantitative relationship between entropy and accuracy for the first time.

5.2 Distances between Classes After PQ

As illustrated in the blue box of Fig. 3, the combination of selected codewords within the same class would exhibit a strong correlation, whereas those between different classes would have a weak correlation. To investigate such a correlation, we calculate the distance between codeword combinations of two pixels. Specifically, we define a codeword combination of q as a sequence of selected codeword indices:

$$\texttt{combination}(q) = [k^1, k^2, \ldots, k^M] \tag{8}$$

where k^i is the $\arg\min$ index of i-th subvector (see Eq. (2)). Then, we define the distance between two combinations as the sum of the number of different elements as below:

$$\texttt{distance}(\texttt{c}(q_1), \texttt{c}(q_2)) = \sum_{i=1}^{M} \mathbb{I}(k_1^i \neq k_2^i) \tag{9}$$

where $\texttt{c}(q)$ is an abbreviation of $\texttt{combination}(q)$. Fig. 4(d) shows the average distance of codeword combinations between classes. Note that we randomly select 10,000 pixels from every class for distance computation. We observe that the diagonal of the matrix, corresponding to the distance within the same class, shows much smaller values than other elements of the matrix. This implies that our quantization head outputs similar results for the same class, which enables clustering to become easier.

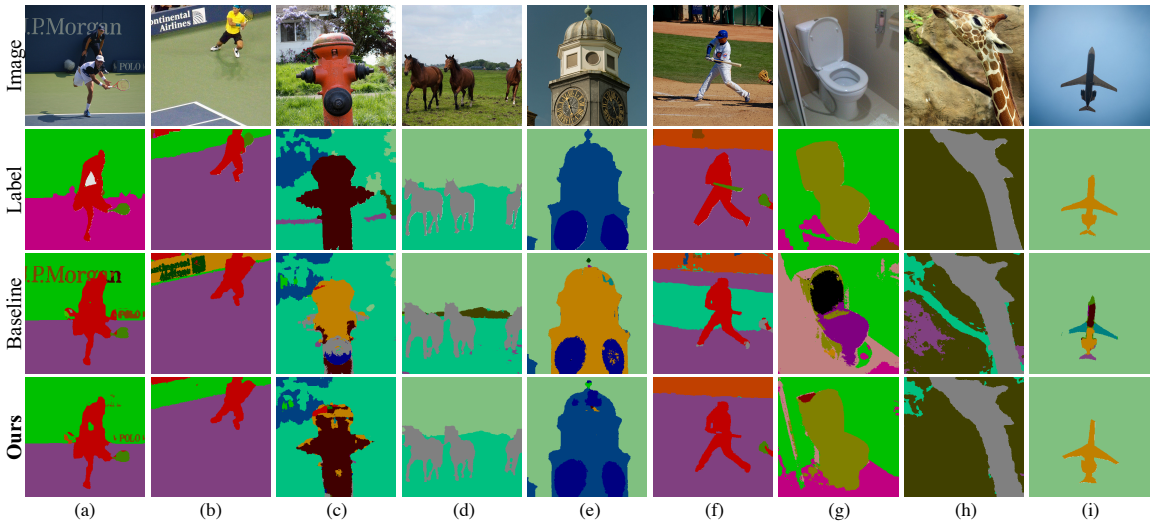


Figure 5: Qualitative comparison between the baseline (STEGO) and ours (EQUSS) on CocoStuff-27.

Model	Dim.	Quant.	Unsupervised Acc	Unsupervised mIoU	Linear Probe Acc	Linear Probe mIoU
M1	70		48.3	24.5	74.4	38.3
M2	384		48.6	24.2	67.9	31.6
M3	1024		48.1	21.9	64.6	31.5
Q2	384	✓	49.7	25.1	72.4	36.7
Q3 (**EQUSS**)	1024	✓	**53.8**	**25.8**	**75.2**	**41.2**

Table 3: Ablation study on CocoStuff-27.

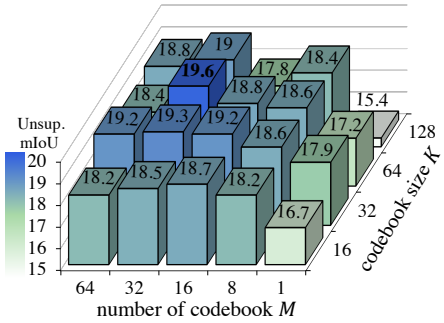


Figure 6: Unsupervised mIoU performance regarding the number and size of the codebook on the Cityscapes. Note that the dimension of the feature is fixed to 512.

5.3 Ablation Study

Component Ablation Table 3 shows the effect of feature dimension expansion and product quantization. As discussed in Sec. 3.1, naively increasing the feature dimension hurts the performance, specifically resulting in a consistent decrease in linear probing accuracy and mIoU (see M1, M2, and M3). When PQ is applied for compression, the model achieves higher performance in all cases compared to not using PQ (see M2 *vs.* Q2, M3 *vs.* Q3). Furthermore, the effectiveness of PQ increases as the feature dimension size increases (see Q2 $<$ Q3). By combining both feature dimension expansion and product quantization, we can achieve the best result (see Q3).

PQ Ablation To study the effect of the number of codebooks (M) and the size of the codebook (K), we conduct experiments while varying M and K with fixed feature dimension. In Fig. 6, we plot unsupervised mIoU performance on Cityscapes. The results show that the performance generally improves with M. Specifically, $M > 1$ outperforms M=1 in all cases. Note that M=1 is identical to VQ. Furthermore, performance gain is higher for a large codebook size (K) since it expands the number of cases that a single codebook can represent.

However, excessively large values of M or K slightly degrade the performance. When the fixed input dimension is divided into too many small subspaces (i.e., M=64), such small dimensions of each subvector would restrict their ability to capture necessary representations. Additionally, for overly large K (i.e., K=128), each codeword may not receive sufficient updates due to the limited number of training samples, so that we observe unstable performance. Nonetheless, in most cases where $M \in \{8, 16, 32\}$ and $K \in \{32, 64\}$, the performance is consistently higher than the baseline.

6 Conclusion

In this paper, we proposed EQUSS, a novel USS framework that takes advantage of high-dimensional space and product quantization. From the numerical experiments, we show that EQUSS achieves the state-of-the-art performance on three USS benchmark datasets. We also quantified the information capacity of USS features and discovered meaningful analyses such as class-specific tendency and the relationship between entropy and accuracy.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022M3C1A3099336) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00208985).

References

Asgari Taghanaki, S.; Abhishek, K.; Cohen, J. P.; Cohen-Adad, J.; and Hamarneh, G. 2021. Deep semantic segmentation of natural and medical images: a review. *Artificial Intelligence Review*, 54: 137–178.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Caesar, H.; Uijlings, J.; and Ferrari, V. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1209–1218.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 132–149.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Chen, L.-W.; Watanabe, S.; and Rudnicky, A. 2023. A Vector Quantized Approach for Text to Speech Synthesis on Real-World Spontaneous Speech. *arXiv preprint arXiv:2302.04215*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Cho, J. H.; Mall, U.; Bala, K.; and Hariharan, B. 2021. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16794–16804.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Cover, T. M. 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3): 326–334.

Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3): 171–176.

Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; and Dietmayer, K. 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3): 1341–1360.

Goldfeld, Z.; and Polyanskiy, Y. 2020. The information bottleneck problem and its applications in machine learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 19–38.

Guo, H.; Xie, F.; Wu, X.; Lu, H.; and Meng, H. 2022. Towards High-Quality Neural TTS for Low-Resource Languages by Learning Compact Speech Representations. *arXiv preprint arXiv:2210.15131*.

Hamilton, M.; Zhang, Z.; Hariharan, B.; Snavely, N.; and Freeman, W. T. 2022. Unsupervised Semantic Segmentation by Distilling Feature Correspondences. In *International Conference on Learning Representations*.

Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H. R.; and Xu, D. 2022. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 574–584.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *CoRR, abs/1512*, 3385: 2.

Hofmann, T.; Schölkopf, B.; and Smola, A. J. 2008. Kernel methods in machine learning.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, L.; Zhang, Z.; Du, Z.; Li, S.; Zheng, H.; Xie, Y.; and Tan, N. 2022. EPQuant: A Graph Neural Network compression approach based on product quantization. *Neurocomputing*, 503: 49–61.

Jang, Y. K.; and Cho, N. I. 2021. Self-supervised product quantization for deep unsupervised image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12085–12094.

Jegou, H.; Douze, M.; and Schmid, C. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1): 117–128.

Jégou, H.; Douze, M.; Schmid, C.; and Pérez, P. 2010. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 3304–3311. IEEE.

Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9865–9874.

Kim, J.; Shim, K.; and Shim, B. 2022. Semantic feature extraction for generalized zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1166–1173.

Kim, S.; Park, D.; and Shim, B. 2023. Semantic-aware superpixel for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1142–1150.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klein, B.; and Wolf, L. 2019. End-to-end supervised product quantization for image search and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5041–5050.

Krähenbühl, P.; and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24.

Li, J.; Lan, X.; Liu, Y.; Wang, L.; and Zheng, N. 2019. Compressing unknown images with product quantizer for efficient zero-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5463–5472.

Li, K.; Wang, Z.; Cheng, Z.; Yu, R.; Zhao, Y.; Song, G.; Liu, C.; Yuan, L.; and Chen, J. 2023. ACSeg: Adaptive Conceptualization for Unsupervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7162–7172.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, 1150–1157. Ieee.

Meletis, P. 2022. Towards holistic scene understanding: Semantic segmentation and beyond. *arXiv preprint arXiv:2201.07734*.

Ouali, Y.; Hudelot, C.; and Tami, M. 2020. Autoregressive unsupervised image segmentation. In *European Conference on Computer Vision*, 142–158. Springer.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Seong, H. S.; Moon, W.; Lee, S.; and Heo, J.-P. 2023. Leveraging Hidden Positives for Unsupervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19540–19549.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423.

Siam, M.; Gamal, M.; Abdel-Razek, M.; Yogamani, S.; Jagersand, M.; and Zhang, H. 2018. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 587–597.

Stock, P.; Fan, A.; Graham, B.; Grave, E.; Gribonval, R.; Jegou, H.; and Joulin, A. 2021. Training with Quantization Noise for Extreme Model Compression. In *International Conference on Learning Representations*.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Yin, Z.; Wang, P.; Wang, F.; Xu, X.; Zhang, H.; Li, H.; and Jin, R. 2022. TransFGU: a top-down approach to fine-grained unsupervised semantic segmentation. In *European Conference on Computer Vision*, 73–89. Springer.

Yu, K.; Ma, R.; Shi, K.; and Liu, Q. 2020. Neural network language model compression with product quantization and soft binarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 2438–2449.

Yu, X.; Liu, T.; Wang, X.; and Tao, D. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7370–7379.