Deep Attribute-Preserving Hashing for Content Based Image Retrieval

Wonkyung Lee
leewk92@estsoft.com
ESTsoft

Uijung Chung
dict@estsoft.com
ESTsoft

Hyeongjin Byeon
kessar@estsoft.com
ESTsoft

ABSTRACT

The techniques of extracting hash codes of images through deep learning have recently been studied due to its high efficiency in large scale image retrieval systems. In the past, we have supervised the label to train the hash function in the same manner to the classification task. Although labels have semantic information, they couldn't contain all of attribute information such as shape, color, and texture, so it is difficult to find similarity of attribute in ranking results even though Mean Average Precision score (mAP) is high.

In this paper, we propose a novel hashing method, Deep Attribute-Preserving Hashing (DAPH), to include not only semantic information but also attribute information. Comprehensive experiments show that it is possible to rank images which have similar attributes to queried image without dropping the mAP using the proposed method. In addition, we compare the existing studies with the mAP to see how well our hash function learned semantic information.

CCS CONCEPTS

• **Information systems → Information retrieval**; • **Computing methodologies → Visual content-based indexing and retrieval**.

KEYWORDS

convolutional neural network, image retrieval, deep hashing, center loss

ACM Reference Format:
Wonkyung Lee, Uijung Chung, and Hyeongjin Byeon. 2019. Deep Attribute-Preserving Hashing for Content Based Image Retrieval. In *Ottawa '19: ACM International Conference on Multimedia Retrieval, June 10–13, 2019, Ottawa, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Amount of image data has grown tremendously in recent years, and the need for efficient image retrieval method has emerged. One of the significant research streams on the image retrieval is that aims to increase the efficiency of large-capacity retrieval systems by creating hashes to retrieve a large number of images quickly.

Traditionally, a hashing method such as LSH [7], which expresses local similarity as a distance between hash codes, has been devised, but after the evolution of deep learning, parameters of the hash function are automatically learned [2, 6, 13, 14, 20].

The key to learning a hash function is to minimize the hamming distance between hash codes of similar images. The quality of trained hash function is dependent on how we define the similarity of images. A most common way is to supervise it with data labels for classification. Therefore, image retrieval researches mainly focus on problems regarding classification tasks such as object recognition. Deep features from global descriptors extracted from neural networks show better performance in classification tasks than traditional shallow features, and it is easy to learn and quantitative evaluation is possible with labeled datasets [11].

Although, there is a limitation of learning similarity of images with the labels because they only could represent specific properties of images. Which means, the classification label is the semantic information of an object expressed as a finite number of predefined categories. For example, if the label is given as 'dog' or 'cat', all images of 'dog' have same categorized semantic information, so it matches even if an image of 'white hairy dog' is searched for an image of 'brown skinny dog'.

As in the example above, uncategorized attributes are also important in image retrieval. Categories of objects as well as similarity of visual appearance, that is, the attributes of objects such as color, shape, and texture also have an essential effect on user satisfaction of image retrieval. These attributes of objects have been traditionally engineered to shallow (relatively less deep) features from local descriptors to calculate similarity and use them for retrieval ranking. In other words, attributes, as well as semantic information, should be considered as a criterion for judging whether images are similar or not. And the two pieces of information can be extracted from the trained CNNs (Convolutional Neural Networks).

CNNs are trained as a not only global descriptor to distinguish semantic information but also local descriptor covering various representability and region from low level to high level for each layer [1]. If CNN's local descriptor is used to represent details and attributes of images, it means that a single CNN model can be expanded to create an embedded vector that represents both semantic and attribute information.

We propose a hybrid hash function that allows CNN-based semantic hash to preserve attributes. By using the hamming distance of the hash code, not only the category of objects can be matched, but also the attributes of objects can arrange the search results in a similar order. To preserve attribute information during hashing, we developed several methods such as binarization loss, center loss, local feature extraction. Then we show attribute-preserved querying results with maintaining qualitative performances on image retrieval task for CIFAR-10 dataset.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Ottawa '19, June 10–13, 2019, Ottawa, Canada

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

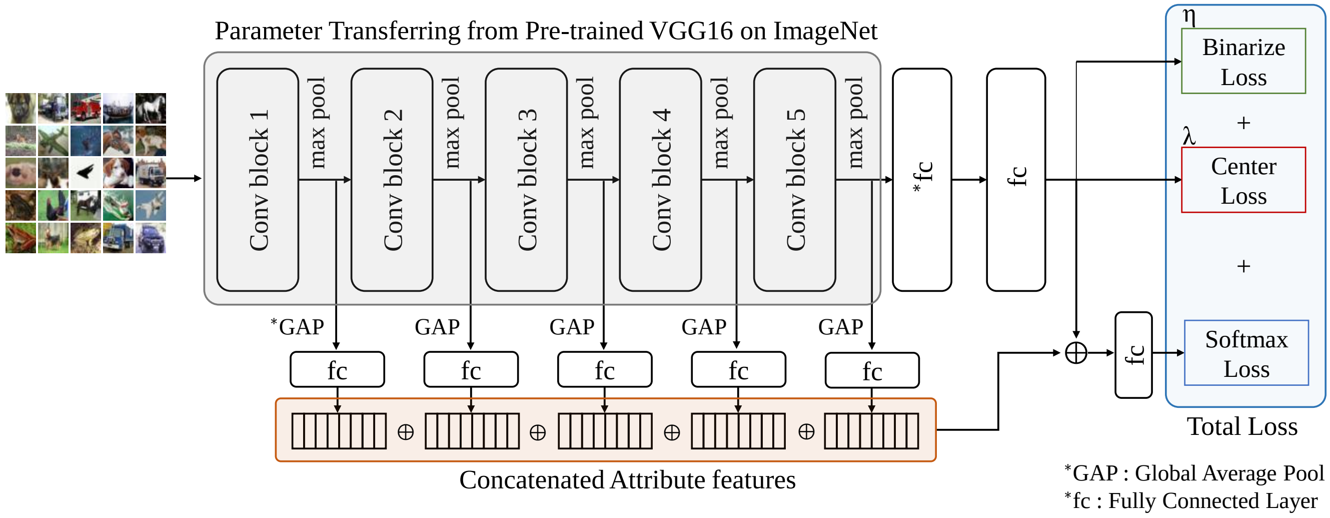

Figure 1: Proposed network structure to obtain a semantic hash code and an attribute hash code from the image. (a) Binarize loss makes the output values of connected fully connected layer two modes on -0.5 and 0.5. (b) Center loss and softmax cross entropy loss are used for extracting semantic hash code. Center loss centralizes output vectors from last fully connected layer, and softmax cross entropy makes them possible to linearly separable from those of the other labels. (c) For extracting an attribute hash code from convolutional blocks, 2-dim Global Average Pooling (GAP) is applied to each block_pool layers. Fully connected layers are used for a dimension reduction with binarize-activation function. Then concatenate five output vectors to get attribute hash code.

2 RELATED WORKS

Image Retrieval with Pre-trained Model

There are several attempts to build global descriptors by using deep learning that can contain semantic properties [8, 10, 11]. Furthermore, Zheng et al. [21] shows that when using the output of the layer which closes to the Conv layer as a feature is better to generalization to a different domain rather than use the feature of the Fully connected layer close to the logit layer. Also, RMAC [18] have been introduced that can combine the output of the CNN layer to provide features for objects of various sizes without additional operations such as image resizing and cropping.

Image Retrieval with Fine Tuning

There are also researches to obtain discriminative features that reflect the characteristics of the other image domains by fine-tuning neural networks with the huge image datasets. For example, Contrastive loss of Siamese network[4], Triplet loss[9], center loss[19] and arc loss[5] were designed to achieve the goal of discriminative feature learning in large-scale face image domain. They perform metric learning that narrows the feature distance of semantically similar samples and broadens the distance between features of different samples. They also use the last layer of fully connected as an image representation without additional pooling or encoding steps.

Deep Hashing Methods

One of the major research streams in image retrieval is to create a compact feature to increase efficiency in large-volume retrieval systems. Lin et al. [13] show that a feature containing image properties can be expressed and retrieved by binary hash code through deep learning using sigmoid activation in the feature layer. To improve the performance by using the learning method of siamese and triplet loss have been further developed [12, 14]. They reduce the Hamming distance of semantically similar images and increases the Hamming distance of semantically distant images. Deep Cauchy Hashing (DCH) [2] generates compact and concentrated binary hash codes to enable efficient and effective Hamming space retrieval. DCH intended to concentrate generated hash codes of images whose labels are same.

3 OUR APPROACHES

Our goal is to extract and merge the semantic information and the attribute information from the image with a hash code, respectively. The neural network is a powerful nonlinear model that performs well in various tasks of image classification and retrieval. Therefore, we extracted each semantic information and attribute information using a neural network. First, we binarize the output feature vector by adding binarization loss to the neural network. Second, we extracted the semantic information using center loss function. Third, get the Attribute hash code from the pooling outputs of the convolutional blocks. Finally, considering the perplexity and relative distances of the semantic hash codes, the semantic and the attribute hash codes are merged so as not to affect the mAP score.

3.1 Binarization Loss

Binarization loss which is a kind of activity regularization loss supports the output of the last fully connected layer to be binarized.

When there are N data and with feature vector $\mathbf{b}$ with length as M, the equation of binarization loss function is as follows.

$$L_b = \eta \sum_{i}^{N} \sum_{j}^{M} ||\mathbf{b}_{ij}|_1 - 0.5|_1 \tag{1}$$

We use the feature values quantized to 1s or 0s depending on whether those are positive or negative. Therefore, if there is a feature value around the threshold value of 0, the quantized value changes due to some noise. To prevent this, we introduced the binarization loss function above, so that modes appear equally around 0.5 and -0.5, with very few values around 0.

3.2 Center Loss

Figure .1 shows the structure of our neural network. The center loss was applied to the last fully connected output vector of the network. Using center loss, the embedding features of the images with the same label are brought close to the center vector. This because the meaning of the center loss is to solve the regression problem of targeting the center vector of each label from the image's embedding vector. Therefore, the variance and bias of the estimated embedding vector can be reduced, which decreases the entropy of the embedding vector. Since the target center vector contains only the semantic information, the center loss can perform the role of removing attribute information. Let's assume that $\mathbf{c}_{\mathrm{y_i}}$ as a center vector of ith image's label and $\mathbf{e}$ as an embedding vector of ith image. Then the equation of center loss is as follows.

$$L_c = \lambda \frac{1}{N} \sum_{i} (\mathbf{c}_{y_i} - \mathbf{e}_i)^2 \tag{2}$$

When creating a semantic hash code, extract the fully connected layer output from the input image and convert the positive number to 1 and the negative number to 0.

3.3 Extract Attribute Information

The convolutional blocks of the pretrained model can act as a local feature extractor of the image. The feature maps of the convolutional layer close to the input layer learn to represent the local and simple features of the image and learn the complex features corresponding to the broad region as they close to the output layer. These trained features can be used as attribute information of images.

We trained our model by fine-tuning the VGG16 [17] model pretrained with ImageNet [11]. The attribute features are extracted from each pooling layers from the VGG16 model. Then, we add Global Average Pooling to each max pooling layer to extract layers that represent all the attributes in the image patches. For dimension reduction, we use non-trainable orthogonally initialized fully connected layer. Then we use binarize-activation for the layer, which sets a positive number to 1 and a negative number to zero. And concatenate them into a feature vector that represents attributes of various layers. This process is demonstrated in Figure. 1.

3.4 Merge Semantic and Attribute Hash code

From the above methods, we could obtain the semantic hash code and the attribute hash code. The final goal is to keep the retrieval evaluation result (mAP) when both codes are merged.

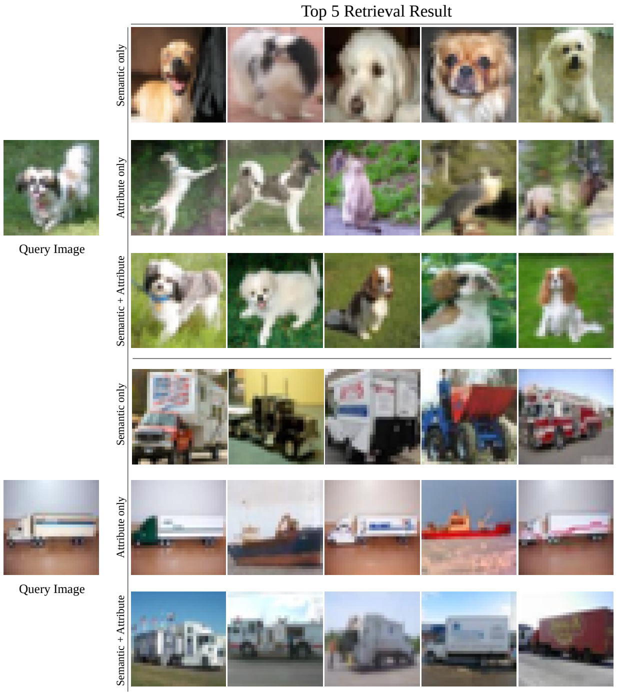

Figure 2: Examples of top 5 retrieval results on CIFAR10. Most left images are queried images of two retrieval tasks. The images at the first column have nearest hash code from the queried image, rank-1 images. For each example, images on the first row are ranked by those semantic hash code. Likewise, images on the second row are ranked by attribute hash code, and the third one are by merged codes, we called it hybrid hash codes. The Precision @ top5 scores are both 1, but the result images that are retrieved with the semantic hash code are less attribute related to the queried image than the images obtained by the hybrid hash code.

Concatenating the semantic hash code and attribute hash code by 1:1 results a drop of the mAP score due to the hamming distance variance of the attribute in the same label. To prevent this problem, the following conditions must be satisfied:

For images with the same label, let the maximum hamming distance of attribute hash code as $DA(L_i)$ and the difference of the semantic hash code of different labels as $DS(L_i, L_j)$. If the condition of $DA(L_i) + DA(L_j) < DS(L_i, L_j)$ is satisfied, there is no decrease of mAP.

For all i, $DA(L_i)$ can not exceed the number of bits assigned to the attribute hash code which is called as B_a. Therefore, $DS(L_i, L_j) > B_a$ must be satisfied. To do this, we tiling the Semantic hash as much as B_a. This is because $min(DS(L_i, L_j)) > B_a$ is satisfied.

By concatenating the semantic hash code which created by tiling and attribute hash, we could merge the two hash without decreasing the mAP score.

Method	CIFAR-10			
	16 bits	32 bits	48 bits	64 bits
KSH [15]	0.4368	0.4585	0.4012	0.3819
SDH [16]	0.562	0.6428	0.6069	0.5012
CNNH [20]	0.5512	0.5468	0.5454	0.5364
DNNH [12]	0.5703	0.5985	0.6421	0.6118
DHN [22]	0.6929	0.6445	0.5835	0.5883
HashNet [3]	0.7476	0.7776	0.6399	0.6259
DCH [2]	0.7901	0.7979	0.8071	0.7936
DAPH-S	0.7188	**0.7531**	0.6143	0.5697
DAPH-S + CM	**0.7356**	0.7501	**0.7688**	**0.7691**

Table 1: $mAP@H \leq 2$ **comparison to the other studies.**

Methods	mAP	bits	mAP	bits	mAP	bits
DAPH-S	0.6610	32	0.7019	48	0.6928	64
DAPH-A	0.2090	48	0.2175	48	0.2179	48
DAPH-C	0.4946	80	0.6137	96	0.6362	112
DAPH-H	**0.6659**	1584	**0.7049**	2352	**0.6952**	3120

Table 2: mAP comparison of four hash codes extracted from our models. We could verify that hash code was well merged with the semantic and attribute hash code without dropping the mAP.

4 EXPERIMENTS

In this section, we show through experiments that the attributes of an image can be ranked in similar order without decrease of the mAP. We also compare the mAP with previous researches to see how well learned semantic information.

4.1 Dataset

CIFAR10

Cifar10 is a dataset with a larger class variance than MNIST, which means it has various image attributes in one label. According to the protocol of [22], we used 500 per class as a train set, 100 per class as a test set, and the rest as database set.

4.2 Experiment Settings

We use pre-trained VGG16 on ImageNet as a base model. Since the height and weight of CIFAR10 are 32, we limited the pooling layer to be used in VGG16. In detail, attribute hash is extracted from all three pooling layers of block1_pool, block2_pool, and block3_pool, and block3_pool is set as the last layer of VGG16. In the attribute pooling block, 16bit attribute code is extracted for each block_pool layer. The size of the semantics hash was tested separately for 16 bits, 32 bits, 48 bits, and 64 bits for comparison with other researches.

Experiments were conducted using Keras, a deep learning framework. The hyper parameters settings are as follows. RMSProp was used as the optimizer, and the learning rate was 1e-4 and the decay factor was 1e-6. The center loss λ was 1e-4, and the binarization loss η was 1e-3.

4.3 Results

Qualitative Evaluation

The qualitative evaluation was carried out to check whether the retrieval took into consideration both the semantic information and the attribute information, and the result can be confirmed in Figure. 2. Five semantically relevant images were retrieved by both the semantic hash code and the hybrid hash code. Therefore, the Precision @ top5 scores are both 1, but the result images that are retrieved with the semantic hash code are less attribute related to the queried image than the images obtained by the hybrid hash code. The result images which retrieved with the attribute hash code have a high attribute relevance with the queried image. However, we can see that the Precision @ top5 scores are 2/5 and 3/5, respectively, and do not fully reflect the semantic information.

Quantitative Evaluation

We compared the performance of each hash codes (hash code with only semantic information and semantic information with center matching) with other papers. Center matching is a method of retrieving the center hash code that is closest to the semantic hash code extracted from the query image, resulting in an improvement of the mAP score by about 0.05 and 0.2, respectively, in the experimental results of 48 bits and 64 bits. Although the results of DCH outperform all other methods, our method using center matching (DAPH-S + CM) has only a slight performance difference from that of DCH.

The mAP scores of CIFAR10 were obtained for the four hash codes extracted from our models to verify that hash code was well merged with the semantic and Attribute hash code without dropping the mAP. DAPH-S, DAPH-A, and DAPH-H represent the semantic hash code, the attribute hash code, and the hybrid hash code, respectively. DAPH-C is a comparison group included to compare the merge method. It is a hash code that simply concatenates Semantic hash code and Attribute hash code. As can be seen from the results of Table. 2, we can see that the mAP score of the hybrid hash code is rather higher than that of the semantic hash code.

5 CONCLUSION

In this paper, a novel deep hashing method - DAPH - was proposed for content-based image retrieval, which learned a hashing function that preserves attribute information that couldn't be supervised from labels. DAPH could rank not only semantically but also attributively similar images and it was demonstrated in the qualitative results. Center loss was adopted to extract semantic information from images and Binarize loss was used to reduce the quantization noise of feature vectors binarization. To extract attribute hash codes from CNNs, we used global average pooling to all of block_pool layers, reduced fixed size dimension by non-trainable fully connected layers and concatenated them. Finally we merge the semantic hash code and attribute hash code safely.

However, it is hard to measure how our method works well on attribute-preserving hashing by only showing qualitative evaluation. Accordingly for the direction of future work, it is valuable to measure attributive similarity by quantitative method to evaluate our method correctly.

REFERENCES

[1] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. 2016. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2874–2883.

[2] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. 2018. Deep cauchy hashing for hamming space retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1229–1237.

[3] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. 2017. Hashnet: Deep learning to hash by continuation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*.

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1. IEEE, 539–546.

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2018. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698* (2018).

[6] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2475–2483.

[7] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *Vldb*, Vol. 99. 518–529.

[8] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. Deep image retrieval: Learning global representations for image search. In *European Conference on Computer Vision*. Springer, 241–257.

[9] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*. Springer, 84–92.

[10] Eva Hörster and Rainer Lienhart. 2008. Deep networks for image retrieval on large-scale databases. In *Proceedings of the 16th ACM International Conference on Multimedia*. ACM, 643–646.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[12] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3270–3278.

[13] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. 2015. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 27–35.

[14] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2016. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2064–2072.

[15] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. 2012. Supervised hashing with kernels. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2074–2081.

[16] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. 2015. Supervised discrete hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 37–45.

[17] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[18] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2015. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* (2015).

[19] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*. Springer, 499–515.

[20] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised hashing for image retrieval via image representation learning.. In *AAAI*, Vol. 1. 2.

[21] Liang Zheng, Yali Zhao, Shengjin Wang, Jingdong Wang, and Qi Tian. 2016. Good practice in CNN feature transfer. *arXiv preprint arXiv:1604.00133* (2016).

[22] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval.. In *AAAI*. 2415–2421.