Efficient Fusion of Image Attributes: A New Approach to Visual Recognition

Dehao Yuan,[1], Minghui Liu, [1], Cornelia Fermüller[1], Yiannis Aloimonos[1]

[1]University of Maryland, College Park
dhyuan@umd.edu, minghui@umd.edu, fer@cfar.umd.edu, yiannis@cs.umd.edu

Abstract

Image attributes are often fused with other input data in many applications, where the existing fusing methods are usually data-driven. Such data-driven fusion can lead to overfitting or slow convergence. In this work, we propose a novel method of encoding image attributes. We also propose a novel fusion technique that combines the attribute encoding and the image encoding. Our novel technique requires very few or even no learnable parameters, but still achieves comparable performance as other data-driven fusion techniques. Besides, our fusion technique enables the network to learn faster when new attributes are added in on-the-fly. Hence, we claim that our fusion technique is efficient in terms of model size and training speed. With the novel fusion technique, we find that an image classifier can be enhanced by fusing it with a pre-trained attribute extractor. Since the fusion requires few or even no additional training parameters, the enhancement is for free. This opens a new direction of recognition: instead of training a larger image classifier to improve the accuracy, we can use a pre-trained attribute extractor to enhance the existing classifiers.

Introduction

In many vision applications, convolutional neural networks (CNN) for object recognition are fused with networks extracting features from other inputs (e.g. sounds, texts). Examples include the tasks of visual question answering (Antol et al. 2015; Zhang et al. 2016; Goyal et al. 2017) and cross-modal retrieval (Wei et al. 2016; Zhen et al. 2019). The fusion of the heterogeneous input data is usually modeled by deep neural networks (DNNs). Thanks to the strong approximation capability of DNNs, the data fusion usually performs well given enough training data and training epochs.

However, fusing heterogeneous input data via networks requires additional learning, which sometimes leads to overfitting or slow convergence. The training overhead is due to the data-driven approach. We found that the overhead may be reducible, especially when the additional information has a semantically meaningful structure. Specifically in this paper, we study the fusion of images with their attributes, where attributes have an inherent disentangled structure. That is, attributes can be decomposed into independent components (e.g. color, pattern), where each component contains several atomic attributes (e.g. blue, green). We ask the following question: if we can use high-dimensional vectors to represent atomic attributes and define operations to model their combination, can we get rid of learning the attribute encoders?

In this work, we answer this question by introducing a novel attribute fusing mechanism based on hyperdimensional computing (shortly HDC) (Kanerva 2009). HDC works by randomly sampling large-size vectors which are approximately orthogonal to each other. Additionally, HDC allows math operations like superposition and binding, through which hypervectors are combined in meaningful ways. This gives us a way to use a non-learning framework to approximate the learning of attribute encoders. The non-learning framework requires few or even no learnable parameters, and hence requires minimal training when new attributes are introduced.

We apply the fusion technique to enhance object recognition. We extract the visual attributes of an image and fuse the attributes with the image encoding. The fused output is fed to fully-connected layers for classification.

Our experiments show that when fusing images and their visual attributes, the accuracy and transferability improves significantly compared with a vanilla CNN classifier. In addition, our novel fusion technique achieves the largest performance gain compared with other strong baselines like feature concatenation and deep merge (Hu, Lu, and Tan 2017). This opens a new direction for visual recognition. Besides, if additional information (e.g. text descriptions) is also collected when the image is taken, our novel fusing technique can be applied to retrieve the joint embedding of both information for the latter tasks. Our contribution can be summarized as followed:

1. We propose an efficient and effective approach to encode attributes by adopting the mathematical operations in HDC. The encoding scheme requires very few (or even no) learnable parameters but still yields the highest accuracy when combining image embeddings of objects with attributes compared with other commonly-used techniques. It also takes fewer epochs for the network to converge.
2. We provide a new direction of visual recognition. Instead of training larger networks to improve the performance

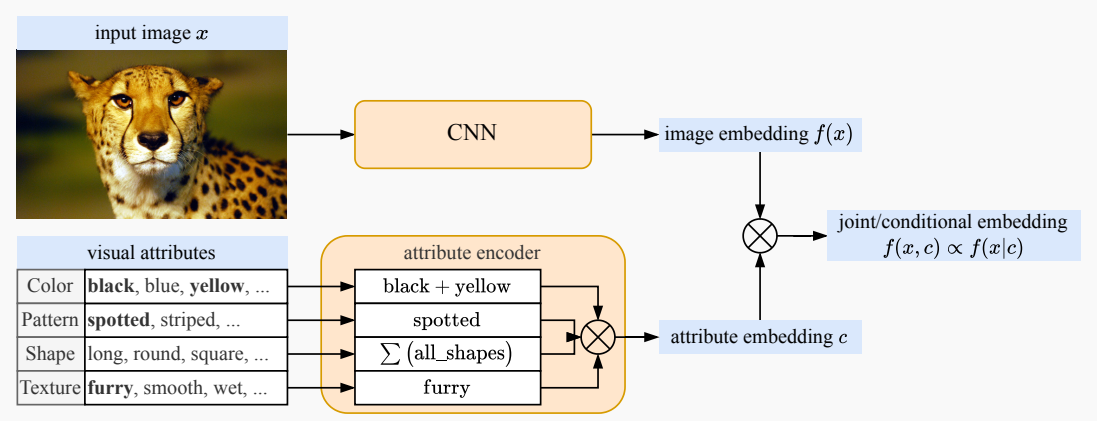

Figure 1: Fusion of images and their visual attributes. The visual attributes are disentangled into different components (i.e. color, pattern, shape, and texture). The embedding of each component is computed by summing up the positive attributes. The embedding of all components are then combined through the binding operation to obtain the final attribute embedding. Finally, the image embedding and the attribute embedding is fused through the binding operation to retrieve the joint embedding. $\otimes$ in the figure denotes the binding operation.

marginally, we can introduce an attribute predictor to enhance current image classifiers, in terms of both accuracy and transferability.

Background

In this section, we give a brief overview of Hyperdimensional Computing (HDC), also known as Vector Symbolic Architecture (VSA) (Kleyko et al. 2021). HDC uses very high dimensional vectors, called hypervectors, to perform symbolic computation. Two symbolic operations are supported for hypervectors. Using those basic elements and operations, we can design powerful encoders.

Hypervectors d-dimensional random vectors are called hypervectors if they are drawn from a distribution $\mathcal{H}$ such that any two vectors are very likely orthogonal to each other. That is, for any $\epsilon > 0$, $\text{Prob.}\big(|cos(\mathbf{x},\mathbf{y})| < \epsilon\big) \to 1$ when $d \to \infty$, where $\mathbf{x},\mathbf{y} \sim \mathcal{H}$ and cos is the cosine similarity between two vectors. For example, a hypervector $\mathbf{x}$ can be generated by independently choosing a random value for each element of the vector from the standard normal distribution, i.e. $\mathbf{x} \sim N(\mathbf{0}, I_d)$. Because of its simplicity, we will use this approach of generating hypervectors throughout the paper.

Hypervectors can be combined through two fundamental symbolic operations, superposition and binding:

Superposition A binary operation $+ : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a superposition if the following properties hold:

1. Commutative property: $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$.
2. Associative property: $\mathbf{x} + (\mathbf{y} + \mathbf{z}) = (\mathbf{x} + \mathbf{y}) + \mathbf{z}$.
3. Similarity: $cos(\mathbf{x} + \mathbf{y}, \mathbf{x}) >> 0$.

Superposition combines vectors into a single vector of the same dimension, and the superposed vector is similar to all of its components. For real vectors, superposition is done by summing or averaging vectors.

Binding A binary operation $\otimes : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a binding operation if the following properties hold:

1. Commutative property: $\mathbf{x} \otimes \mathbf{y} = \mathbf{y} \otimes \mathbf{x}$.
2. Distributive property: $\mathbf{x} \otimes (\mathbf{y} + \mathbf{z}) = \mathbf{x} \otimes \mathbf{y} + \mathbf{x} \otimes \mathbf{z}$.
3. Orthogonality: $cos(\mathbf{x} \otimes \mathbf{y}_1, \mathbf{x} \otimes \mathbf{y}_2) \approx 0$.

where $cos(\cdot,\cdot)$ is the cosine similarity between two hypervectors, and $\mathbf{y}_1$ and $\mathbf{y}_2$ are approximately orthogonal. In the context of real hypervectors, element-wise multiplication and circular convolution are two commonly-used binding operations. One can easily verify the three properties above. Element-wise multiplication is computationally more efficient but less robust in orthogonality than circular convolution. Nonetheless, both operations are differentiable, so they can be seamlessly incorporated into neural networks.

HDC Encoder Building upon the basic elements mentioned above, some works have designed HDC encoders for special data types, such as MNIST-style images (Kleyko et al. 2016; Manabat et al. 2019), trigrams (Alonso et al. 2021), and graphs (Nunes et al. 2022). However, the existing encoders have mainly been used for data compression and reconstruction. Few HDC encoders can be plugged into deep neural networks.

Methodology

In this section, we introduce the fusion of images and their visual attributes by adopting the binding operation from HDC. We elaborate: (a) the relation between symbolic operations and set operations; (b) how to encode attributes into embedding vectors using symbolic operations; (c) how to fuse the image encoding with the attribute encoding. Fig. 1 illustrates the fusion of images with their extracted/ground-truth visual attributes using our novel fusion technique.

Symbolic operations v.s. Set operations

Three symbolic operations can be applied to hypervectors: superposition $(\cdot + \cdot)$, binding $(\cdot \otimes \cdot)$, and subtraction $(\cdot - \cdot)$. Three set operations can be applied to sets: union $(\cdot \cup \cdot)$, intersection $(\cdot \cap \cdot)$, and set difference $(\cdot \backslash \cdot)$. We argue that there

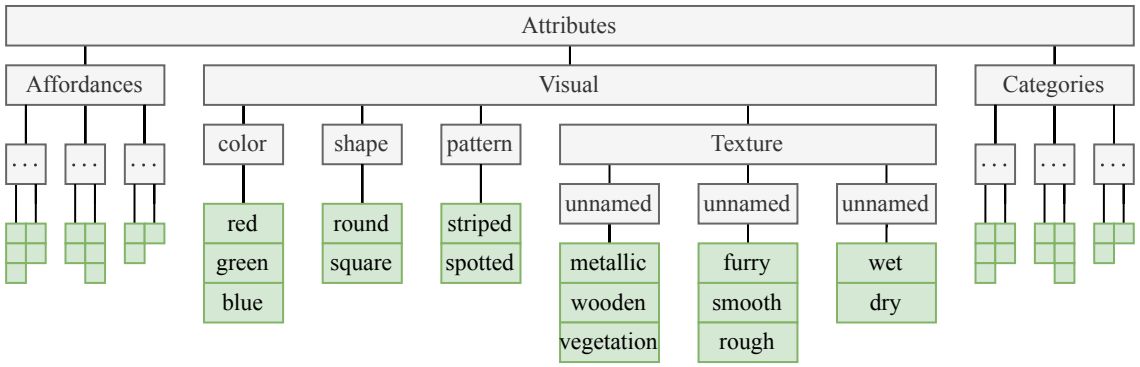

Figure 2: Disentangled structure of attributes. Thanks to the structure, the attributes of an object can be written as the intersection of union of some atomic properties. The corresponding attribute embedding can be obtained by directly translating the set operations to symbolic operations.

is a strong correspondence between symbolic operations and set operations. A similar argument can also be found at Furlong and Eliasmith (2022).

Before introducing the correspondence, we first state the premise: the set to be represented must be *the intersection of a disjoint union* of some *atomic sets*. In other words, the set should be disentangled into different independent components, where each component consists of finitely-many disjoint elements. Fig. 3 illustrates what sets satisfy the premise and what do not. Though it is not obvious why the premise is necessary at first glance, please allow us to introduce the correspondence, and then we will explain the premise in the last paragraph of this section.

First, hypervectors correspond to atomic sets. This is because hypervectors are almost always orthogonal to each other, the neighbor of one hypervector is not likely to be the neighbor of another hypervector. So it is appropriate to use hypervectors to represent atomic sets because atomicity requires each atom to be independent and orthogonal.

Second, superposition corresponds to disjoint union. The superposition of two hypervectors is close to both vectors, so the neighbor of the superposition is also the neighbor of the two hypervectors. Thanks to this property, we can represent the disjoint union of atomic sets within each independent component. Per the illustration in Fig. 3, we can represent $A_1 \cup A_2$, $B_1 \cup B_2$, etc.. But we cannot represent $A_1 \cup B_1$ because they belong to different components and they are not disjoint.

Third, binding corresponds to intersection. Since the binding of two hypervectors is orthogonal to both vectors: $cos(\mathbf{x}, \mathbf{x} \otimes \mathbf{y}), cos(\mathbf{y}, \mathbf{x} \otimes \mathbf{y}) \approx 0$, this means $\mathbf{x} \otimes \mathbf{y}$ points to a mysterious address in the huge hyper-dimensional space unknown to either $\mathbf{x}$ or $\mathbf{y}$. Both $\mathbf{x}$ and $\mathbf{y}$ must be known to reach the address pointed by $\mathbf{x} \otimes \mathbf{y}$. Therefore, $\mathbf{x} \otimes \mathbf{y}$ represents the intersection relation. Thanks to this property, we can encode the intersection of sets across different components. Per the illustration in Fig. 3, we can represent $A_1 \cap B_1$, $(A_1 \cup A_2) \cap B_1$, etc.. But we cannot represent $A_1 \cap A_2$ because the atomic set within each component is disjoint.

Fourth, subtraction corresponds to set difference. The subtraction of two hypervectors is close to the minuend and

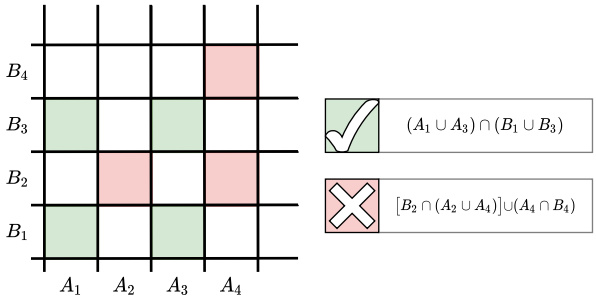

Figure 3: Illustration of the representable sets through HDC symbolic operations. The set to represent must be written as the intersection of disjoint union of some atomic sets. For example, the green set can be represented through the symbolic operations while the red set cannot.

far from the subtrahend: $cos(\mathbf{x}-\mathbf{y}, \mathbf{x}) > 0, cos(\mathbf{x}-\mathbf{y}, \mathbf{y}) < 0$. So the neighbor of $\mathbf{x}-\mathbf{y}$ is likely to be the neighbor of $\mathbf{x}$, but not likely to be the neighbor of $\mathbf{y}$.

Now we answer the necessity of the premise. Notice that only binding distributes over superposition (which coincides with the correspondence), but superposition does not distribute over binding (which does not coincide with the correspondence):

$$\begin{cases} X \cap (Y \cup Z) = (X \cup Z) \cap (Y \cup Z) \\ X \otimes (Y + Z) = (X + Z) \otimes (Y + Z) \end{cases}$$

$$\begin{cases} X \cup (Y \cap Z) = (X \cap Z) \cup (Y \cap Z) \\ X + (Y \otimes Z) \neq (X \otimes Z) + (Y \otimes Z) \end{cases}$$

Therefore, the correspondence does not hold when there exists a union of intersections. So we assume the atomic sets within each independent components must be disjoint to avoid the possibility of intersection within each component. On the other hand, we assume the set should be factorized into independent components. This is because the intersection-binding correspondence relies on the orthogonality property of binding, and the orthogonality property

works the best when the two input vectors are orthogonal. The assumption of independence is to ensure the effect of binding.

Attribute Encoder

First, we argue that the combination of attributes can be written as the intersection of the disjoint union of some atomic attributes. Attributes can be disentangled into different independent components, as illustrated in Fig. 2. The attributes within each component are disjoint. For example, a red object cannot be green and vice versa. Besides, intersections of attributes across the components are possible. For example, round and red, round and green, square and red, and square and green are all possible attributes. Therefore, combinations of attributes satisfy the premise imposed by the HDC encoding.

With the validation of the premise, the attribute encoding can be easily obtained by directly translating the set operations to symbolic operations. For example, $(blue$ OR $green)$ AND $round$ can be expressed as $(blue + green) \otimes round$.

Fuse Images and Attributes

After obtaining the image embedding $f(x)$ and attribute embedding c, their joint embedding can be obtained by binding the two embeddings. This can be understood in two ways: 1) The binding operation is modeling intersection, so $f(x) \otimes c$ is modeling the joint distribution of images and attributes $f(x \cap c)$. 2) $f(x) \otimes c$ can be viewed as moving the image embedding $f(x)$ to a new sub-space navigated by the attribute embedding c. So the fusion can be viewed as a conditional relation $f(x|c)$. Nonetheless, the two interpretations are unified because of the relation: $f(x|c) = \frac{f(x \cap c)}{f(c)} \propto f(x \cap c)$. Thanks to the distributed property (linearity) of binding, the disentangled structure of attributes is preserved in its embedding space.

In addition, to use binding to fuse the two embeddings, we can also encode the combined attributes as a longer vector, reshape it into a matrix and multiply it with the image embedding. Since the matrix multiplication is also linear like the binding operation, the structure of the attributes is also preserved. The benefit of doing this is to increase the robustness of attribute encoding. Since a matrix has more elements than a vector, its capacity is higher and can encode the combination of more attributes. As a trade-off, it also requires more computation.

Experiments

Comparison Settings

Throughout all the experiments, we consider the following fusing techniques, where binding (Hadamard) and binding (Matmul) are our novel fusing techniques. To make the comparison fair, all fusing architectures have roughly the same number of learnable parameters.

no attribute The image encoding is fed to a fully-connected layer for classification, without fusing with the attributes.

masking After the probability of each object class is computed, the probability of the impossible classes are masked as zero. Masking is only applicable for super-class attributes and not applicable for instance-level attributes.

score concat The score for each attribute is concatenated with the image encoding and then fed to a multi-layer perceptron.

feature concat The score for each attribute is first fed to a multi-layer perceptron to extract the attribute encoding. Then the attribute encoding is concatenated with the image encoding and fed to a fully-connected layer.

binding (Hadamard) Attributes are encoded into a hypervector by our novel attribute encoder. The attribute encoding is fused with the image encoding by the element-wise multiplication.

binding (Matmul) The attributes are encoded into a longer hypervector by our novel attribute encoder. The long hypervector is reshaped into a matrix and multiplied with the image encoding.

Experiment 1: Fuse with Super-Class Attributes

Dataset CIFAR100 (Krizhevsky, Hinton et al. 2009) contains 60k 32-by-32 images divided into 100 classes. It also has an official super-class annotation, where the 100 classes are divided into 20 super-classes. In the evaluation, we fuse the images with their super-class and classify the images. What we call a super-class refers to ontology (i.e. aquarium fish, flatfish, ray, shark, trout are fish), and is a special form of attributes because of its one-to-many structure, where all instances in a class are mapped to the same super-class.

Discussion Table 1 shows the result of the experiment. The masking pipeline is applicable in this context because of the one-to-many structure. Theoretically, masking gives the best performance and we compare other fusing techniques against the theoretical upper bound given by masking. Feature concatenation and binding all achieve the same performance as masking when the super-class is correctly provided. To see how robust each technique is against noisy attributes, we performed two additional comparisons: the *noisy* setting unions another random super-class with the correct one, the *uniform* setting unions all the super-classes. Experiments demonstrate that our fusing technique can achieve the optimal accuracy under all settings, while score concatenation and feature concatenation perform slightly worse in some settings.

Implementation We use ResNet-18 as our image encoder and a multi-layer perceptron as the classifier. We train the whole network from scratch. During training, the image is randomly cropped to 28-by-28 pixels. During testing, the image is centered and cropped to 28-by-28 pixels. We train the model for 1000 epochs using a learning rate of 0.1 and momentum of 0.9 and weight decay of 5e-5.

Experiment 2: Fuse with Instance-Level Attributes

Dataset Imagenet with Attributes (Russakovsky and Fei-Fei 2010) is a subset of the Imagenet dataset that contains 9600 images with average size of 256-by-256 pixel from 384 synsets. The dataset has 25 instance-level human annotated attributes, where the attributes span color, shape, texture and pattern. In the evaluation, we will fuse the images with their

	correct	noisy	uniform
no attribute		77.4%	
masking	86.4%	84.9%	77.4%
score concat	79.0%	77.6%	77.2%
feature concat	86.5%	84.6%	76.5%
binding (Hadamard)	86.6%	84.9%	77.1%
binding (Matmul)	86.4%	84.9%	77.4%

Table 1: Classification accuracy on CIFAR100 when fusing images with their super-classes. *correct* means the fused attribute is accurate. *noisy* means the fused super-class is the union of the correct super-class and another random one. *uniform* means the fused super-class is the union of all super-classes.

	correct	predicted	uniform
no attribute		68.3%	
score concat	72.8%	66.8%	**68.3%**
feature concat	83.0%	68.9%	67.6%
binding (Hadamard)	86.1%	**71.2%**	67.3%
binding (Matmul)	**86.3%**	**71.2%**	**68.3%**

Table 2: Classification accuracy on Imagenet-with-attributes dataset, where images are fused with the instance-level attributes. *correct* means the fused attributes are ground-truth attributes. *predicted* means the fused attributes are generated by an attribute predictor. *uniform* means that all the attributes are uncertain.

attributes and predict the synsets. In addition to using the human annotated attributes, we also evaluate the fusion with predicted attributes using the VAW network of Pham et al. (2021). The VAW network takes in an image and produces confidence scores for 620 attributes. Out of these, we only utilize the 25 attributes defined in Russakovsky and Fei-Fei (2010). During development 85% of the data is used as the training set and the remaining 15% is used as the test set.

Discussion Table 2 shows the accuracy comparison. It demonstrates that our novel fusing technique generally outperforms other techniques. In a more realistic setting (i.e. the *predicted* column), an attribute predictor can enhance the image classifier by 2.9% in this experiment, while the feature concatenation only provides extra 0.6%. Besides, the Hadamard fusion is slightly worse than the Matmul fusion. The reason may be that the Hadamard fusion uses fewer dimensions for attribute encoding, so the attribute encoding may be lost. Nevertheless, the capacity problem can be solved by using the Matmul fusion.

Implementation We adopt the official pretrained ResNet-152 network provided by PyTorch and freeze all the convolutional layers. We train the fused models for 100 epochs using a learning rate of 0.1, momentum of 0.9 and weight decay of 5e-5. To make the comparison fair, we make each fusing architecture equipped with roughly the same number of learnable parameters, and all the experiments are run with the same training hyper-parameters. Since the ResNet152 network is pretrained on 1000 commonly used synsets,

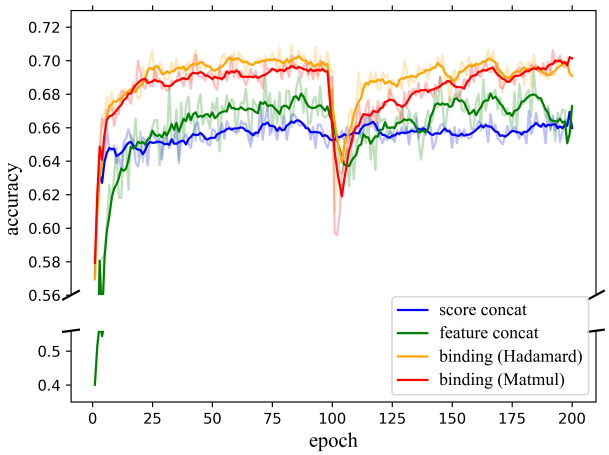

Figure 4: Training curve of the image classification using different fusion techniques. In the first 100 epochs, 25 attributes are fused with the images. Starting from the 100-th epochs, another 15 attributes are added in.

where only about 70 synsets overlap with the 384 synsets, this experiment is under the setting of transfer learning.

Experiment 3: Adding New Attributes On-the-Fly

In this experiment we test how the technique performs when new attributes are added. Using the same 9600 imagenet images, we first train the models for 100 epochs with the 25 predicted attributes used in Experiment 2. The setup and hyperparameters are the same. Then we add in 15 new attributes and see how quickly each architecture learns to fuse the new attributes. The network was then trained for another 100 epochs. We plot the training curve in Fig. 4.

Discussion Referring to Fig. 4, our novel fusing techniques train significantly faster than the other techniques and achieve higher accuracy after convergence. After adding in new attributes, all the fusion techniques experience a performance drop, but our novel techniques recover more quickly. Comparing the Hadamard fusion and the Matmul fusion, Hadamard fusion recovers faster because it uses less dimensions for attribute encoding. But the final accuracy achieved by both fusion techniques does not differ significantly.

Ablation Study 1: Learnable Embedding

We study whether learning the attribute embedding is necessary in the fusion pipeline. Table 3 shows that learning the attribute can bring 2.8% and 1.0% performance gain for Hadamard fusion and Matmul fusion, respectively. It demonstrates that our novel attribute encoder works better when the attribute embedding is learnable, but also works well even without any training.

Besides, the performance gap of the Matmul fusion is smaller than the Hadamard fusion. The reason may be that the Matmul fusion has higher dimension in the attribute encoding, so the attribute encoding has smaller noise and hence the fusion is more accurate.

	attribute embedding	
	Fixed	Learned
binding (Hadamard)	83.3%	86.1% (+2.8%)
binding (Matmul)	85.3%	86.3% (+1.0%)

Table 3: Comparison between fixing v.s. learning the attribute embedding during training.

	attribute structure	
	flat	disentangled
binding (Hadamard)	85.1%	86.1% (+1.0%)
binding (Matmul)	83.3%	86.3% (+3.0%)

Table 4: Effect of imposing the disentangled structure of attributes when encoding attributes. *flat* means all the attributes independent. *disentangled* means the attributes are disentangled into color, shape, texture and pattern.

Ablation Study 2: Effect of Disentangled Structure

We also explore the benefit of disentangling attributes into independent components. Table 4 shows that imposing the disentangled structure in the attribute encoding boosts the performance by 1.0% and 3.0%. On one hand, the Matmul binding with a flat attribute structure is worse for the flat attribute structure. The reason may be that the flat attribute structure makes the fusion harder to train for the Matmul fusion that has a higher dimension in the attribute encoding, and this leads to lower accuracy. On the other hand, the introduction of the attribute hierarchy resolves the problem and even makes the Matmul performance higher than that of Hadamard.

Ablation Study 3: Fusing with Noisy Attributes

We add artificial noise to the attributes and test how the performance changes as the noise level changes. During testing, the attributes are flipped with a certain probability (i.e. the error rate). Table 5 shows that our fusing techniques perform better than the others when the error rate is low. When the error rate is higher, the feature concatenation technique works better. It may be because the feature concatenation has more non-linear structure in the data fusion, so the attributes may interfere less with the image classification. For the binding fusion, the attribute encoding affects the image classification directly, so it is more sensitive to the attribute errors.

Related Work

Hyper-Dimensional Computing

The standalone HDC framework has been used to solve some simple learning problems with higher speed (Imani et al. 2019; Joshi, Halseth, and Kanerva 2016; Rahimi, Kanerva, and Rabaey 2016; Imani et al. 2018). But there has not been any success in making HDC achieve performance comparable to those of neural networks in more complicated problems, e.g. imagenet classification. Therefore, there has been several (but not many) works adopting HDC in modern deep learning (DL) technology to gain mutual benefit from HDC and DL. Here we enumerate some of them.

	attribute error rate				
	0%	5%	10%	15%	20%
no attribute			68.3%		
score concat	72.8%	72.7%	72.6%	72.3%	72.3%
feature concat	83.0%	82.2%	81.0%	**79.4%**	**77.5%**
binding (Hadamard)	86.1%	83.1%	81.0%	78.2%	75.3%
binding (Matmul)	**86.3%**	**84.1%**	**81.3%**	79.0%	76.1%

Table 5: Performance change of fusion techniques under different levels of attribute noise. The attributes are flipped with a certain probability, i.e. the attribute error rate.

First, neural networks can accept hypervectors (HVs) as inputs. Bandaragoda et al. (2019); Mirus et al. (2019); Mirus, Stewart, and Conradt (2020) encode inputs of variable size into HVs. Ma et al. (2018) encodes knowledge graphs into HVs. Karlgren and Kanerva (2019) encode sentences into HVs. The HVs have fixed length and can be fed to a deep neural network, but the performance is not guaranteed. Second, neural networks can be trained to produce HVs. Yilmaz (2015); Neubert et al. (2021) use the activations of CNNs to form the HVs of images. Mitrokhin et al. (2020) use a deep quantization network (Yue et al. 2016) to form bipolar HVs. Mitrokhin et al. (2020); Neubert et al. (2021); Sutor et al. (2022) combine the activations of different lengths from multiple neural networks using HDC techniques and improve the results in applications.

This paper is another attempt to adopt HDC in deep neural networks (DNNs), specifically using HVs as the input to a DNN. But there are several characteristics that make this paper unique: 1. Previous works view HVs simply as embeddings and use them as the input/output of a neural network. This paper uses HVs as transformations (through binding) which can model conditional relations and bias the prediction of the neural network. 2. In previous work, the HDC encoder is separated from the DNN and the training of the DNN is independent of the HDC encoder. For example, the work of Karlgren and Kanerva (2019) encodes sentences into HVs and uses them as the inputs to a neural network. But the sentence encoding is fixed and not learnable. In our paper, HDC is responsible for disentangling the context, and the DL module can be applied to learn the embedding under the regularization of the HDC module. Such design enables end-to-end training of all the parameters in the system and enhances the performance of the whole system.

Data Fusion in Neural Network

In many applications, neural networks are used to process inputs from more than one resource. For example, images from different views (Ding and Tao 2015; Guo et al. 2016; Yan et al. 2020), images and texts (reviewed in the next paragraph), images and geological information (reviewed in the next paragraph). The input resources can be divided into two categories: 1. raw data, e.g. images, raw texts, videos, sounds; 2. structured data, e.g. parsed texts, geological information. Fusing raw data usually is achieved with data-driven approaches since it is hard to manually encode raw data into structured forms, as surveyed in Wang (2021). Fusing raw data with structured data usually has more flexibility be-

cause better encoding schemes and fusing strategies can be designed to fit the structured data and make the neural network train faster and achieve higher performance. Our paper belongs to the latter category. In the next two paragraphs, we will restrict the literature review to fusing raw data with structured data. The first paragraph focuses on the encoding of the structured data. The second paragraph focuses on the fusion of the two sources of data.

Yang et al. (2018); Wei et al. (2016); Tu et al. (2020); Wang, Yang, and Meinel (2015); Jiang and Li (2017); Song et al. (2018) fuse images and texts, where the text features are extracted based on the bag of word (BoW) representation and the extracted text features are processed by fully-connected layers. Tang et al. (2015); Guo et al. (2018); Salem, Workman, and Jacobs (2020) fuses images and geological information (where the image was taken), where the geological information is used to retrieve the geological statistics, which are processed by fully-connected layers. Christie et al. (2018); Minetto, Segundo, and Sarkar (2019) fuses satellite images with their meta information, where the meta information is encoded by fully-connected layers.

As summarized by Iv, Kapoor, and Ghosh (2021), almost all data fusion techniques use concatenation or deep merge. For concatenation, inputs from different resources are concatenated to form a single feature vector. This technique is usually used when the input data consists of raw features, class likelihood scores, or neural network intermediate outputs. For deep merge, the raw features are transformed by some fully-connected layers, and the transformed features are concatenated and transformed, again by some fully-connected layers. This technique is data-driven and has more learnable parameters, but the structure within the input resources is disrupted. All the works mentioned above use one of the two techniques.

In this paper, we fuse images and attributes, where attributes have disentangled structure. Unlike previous works which do concatenation or deep merge, this paper proposes a novel fusing technique that preserves the disentangled structure through the binding operation. The structure-preserving fusion enables more efficient training of neural networks and achieves higher accuracy. The methodology is not restricted to attribute encoding and can be applied to other structured data without much modification.

Conclusion

We present an efficient attribute encoder that requires very few or even no learnable parameters to train. We also present a fusing pipeline of combining the image encoding and the attribute encoding. We apply the fusion technique to enhance image classification, where we find that the accuracy and the transferability of an image classifier can be improved by fusing it with a pre-trained attribute extractor. We evaluated the fusion technique extensively, where the settings include transfer learning and online learning. This technique achieves better performance in the task of image classification, is more resilient to noise, and is more adaptive to new information compared to conventional methods.

References

Alonso, P.; Shridhar, K.; Kleyko, D.; Osipov, E.; and Liwicki, M. 2021. HyperEmbed: Tradeoffs between resources and performance in NLP tasks with hyperdimensional computing enabled embedding of n-gram statistics. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–9. IEEE.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Bandaragoda, T.; De Silva, D.; Kleyko, D.; Osipov, E.; Wiklund, U.; and Alahakoon, D. 2019. Trajectory clustering of road traffic in urban environments using incremental machine learning in combination with hyperdimensional computing. In *2019 IEEE intelligent transportation systems conference (ITSC)*, 1664–1670. IEEE.

Christie, G.; Fendley, N.; Wilson, J.; and Mukherjee, R. 2018. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6172–6180.

Ding, C.; and Tao, D. 2015. Robust face recognition via multimodal deep face representation. *IEEE transactions on Multimedia*, 17(11): 2049–2058.

Furlong, M.; and Eliasmith, C. 2022. Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guo, H.; Wang, J.; Gao, Y.; Li, J.; and Lu, H. 2016. Multi-view 3D object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25(12): 5526–5537.

Guo, W.; Wu, R.; Chen, Y.; and Zhu, X. 2018. Deep learning scene recognition method based on localization enhancement. *Sensors*, 18(10): 3376.

Hu, J.; Lu, J.; and Tan, Y.-P. 2017. Sharable and individual multi-view metric learning. *IEEE transactions on pattern analysis and machine intelligence*, 40(9): 2281–2288.

Imani, M.; Huang, C.; Kong, D.; and Rosing, T. 2018. Hierarchical hyperdimensional computing for energy efficient classification. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 1–6. IEEE.

Imani, M.; Morris, J.; Bosch, S.; Shu, H.; De Micheli, G.; and Rosing, T. 2019. Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 1–4. IEEE.

Iv, W. C. S.; Kapoor, R.; and Ghosh, P. 2021. Multimodal Classification: Current Landscape, Taxonomy and Future Directions. *ACM Computing Surveys (CSUR)*.

Jiang, Q.-Y.; and Li, W.-J. 2017. Deep cross-modal hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3232–3240.

Joshi, A.; Halseth, J. T.; and Kanerva, P. 2016. Language geometry using random indexing. In *International Symposium on Quantum Interaction*, 265–274. Springer.

Kanerva, P. 2009. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2): 139–159.

Karlgren, J.; and Kanerva, P. 2019. High-dimensional distributed semantic spaces for utterances. *Natural Language Engineering*, 25(4): 503–517.

Kleyko, D.; Osipov, E.; Senior, A.; Khan, A. I.; and Şekerciogğlu, Y. A. 2016. Holographic graph neuron: A bioinspired architecture for pattern processing. *IEEE transactions on neural networks and learning systems*, 28(6): 1250–1262.

Kleyko, D.; Rachkovskij, D. A.; Osipov, E.; and Rahimi, A. 2021. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part I: Models and Data Transformations. *ACM Computing Surveys (CSUR)*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Ma, Y.; Hildebrandt, M.; Tresp, V.; and Baier, S. 2018. Holistic Representations for Memorization and Inference. In *UAI*, 403–413.

Manabat, A. X.; Marcelo, C. R.; Quinquito, A. L.; and Alvarez, A. 2019. Performance analysis of hyperdimensional computing for character recognition. In *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*, 1–5. IEEE.

Minetto, R.; Segundo, M. P.; and Sarkar, S. 2019. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9): 6530–6541.

Mirus, F.; Blouw, P.; Stewart, T. C.; and Conradt, J. 2019. An investigation of vehicle behavior prediction using a vector power representation to encode spatial positions of multiple objects and neural networks. *Frontiers in neurorobotics*, 13: 84.

Mirus, F.; Stewart, T. C.; and Conradt, J. 2020. The importance of balanced data sets: Analyzing a vehicle trajectory prediction model based on neural networks and distributed representations. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Mitrokhin, A.; Sutor, P.; Summers-Stay, D.; Fermüller, C.; and Aloimonos, Y. 2020. Symbolic representation and learning with hyperdimensional computing. *Frontiers in Robotics and AI*, 7: 63.

Neubert, P.; Schubert, S.; Schlegel, K.; and Protzel, P. 2021. Vector Semantic Representations as Descriptors for Visual Place Recognition. In *Robotics: Science and Systems*.

Nunes, I.; Heddes, M.; Givargis, T.; Nicolau, A.; and Veidenbaum, A. 2022. GraphHD: Efficient graph classification using hyperdimensional computing. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1485–1490. IEEE.

Pham, K.; Kafle, K.; Lin, Z.; Ding, Z.; Cohen, S.; Tran, Q.; and Shrivastava, A. 2021. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13018–13028.

Rahimi, A.; Kanerva, P.; and Rabaey, J. M. 2016. A robust and energy-efficient classifier using brain-inspired hyperdimensional computing. In *Proceedings of the 2016 international symposium on low power electronics and design*, 64–69.

Russakovsky, O.; and Fei-Fei, L. 2010. Attribute learning in large-scale datasets. In *European Conference on Computer Vision*, 1–14. Springer.

Salem, T.; Workman, S.; and Jacobs, N. 2020. Learning a dynamic map of visual appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12435–12444.

Song, L.; Liu, J.; Qian, B.; Sun, M.; Yang, K.; Sun, M.; and Abbas, S. 2018. A deep multi-modal CNN for multi-instance multi-label image classification. *IEEE Transactions on Image Processing*, 27(12): 6025–6038.

Sutor, P.; Yuan, D.; Summers-Stay, D.; Fermuller, C.; and Aloimonos, Y. 2022. Gluing Neural Networks Symbolically Through Hyperdimensional Computing. *arXiv preprint arXiv:2205.15534*.

Tang, K.; Paluri, M.; Fei-Fei, L.; Fergus, R.; and Bourdev, L. 2015. Improving image classification with location context. In *Proceedings of the IEEE international conference on computer vision*, 1008–1016.

Tu, R.-C.; Mao, X.-L.; Ma, B.; Hu, Y.; Yan, T.; Wei, W.; and Huang, H. 2020. Deep cross-modal hashing with hashing functions and unified hash codes jointly learning. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, C.; Yang, H.; and Meinel, C. 2015. Deep semantic mapping for cross-modal retrieval. In *2015 IEEE 27th International conference on tools with artificial intelligence (ICTAI)*, 234–241. IEEE.

Wang, Y. 2021. Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1s): 1–25.

Wei, Y.; Zhao, Y.; Lu, C.; Wei, S.; Liu, L.; Zhu, Z.; and Yan, S. 2016. Cross-modal retrieval with CNN visual features: A new baseline. *IEEE transactions on cybernetics*, 47(2): 449–460.

Yan, C.; Gong, B.; Wei, Y.; and Gao, Y. 2020. Deep multi-view enhancement hashing for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4): 1445–1451.

Yang, Y.; Wu, Y.-F.; Zhan, D.-C.; Liu, Z.-B.; and Jiang, Y. 2018. Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2594–2603.

Yilmaz, O. 2015. Symbolic computation using cellular automata-based hyperdimensional computing. *Neural computation*, 27(12): 2661–2692.

Yue, C.; Long, M.; Wang, J.; Han, Z.; and Wen, Q. 2016. Deep quantization network for efficient image retrieval. In *Proc. 13th AAAI Conf. Artif. Intell.*, 3457–3463.

Zhang, P.; Goyal, Y.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2016. Yin and Yang: Balancing and Answering Binary Visual Questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhen, L.; Hu, P.; Wang, X.; and Peng, D. 2019. Deep supervised cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10394–10403.