Shape2Vec: semantic-based descriptors for 3D shapes, sketches and images

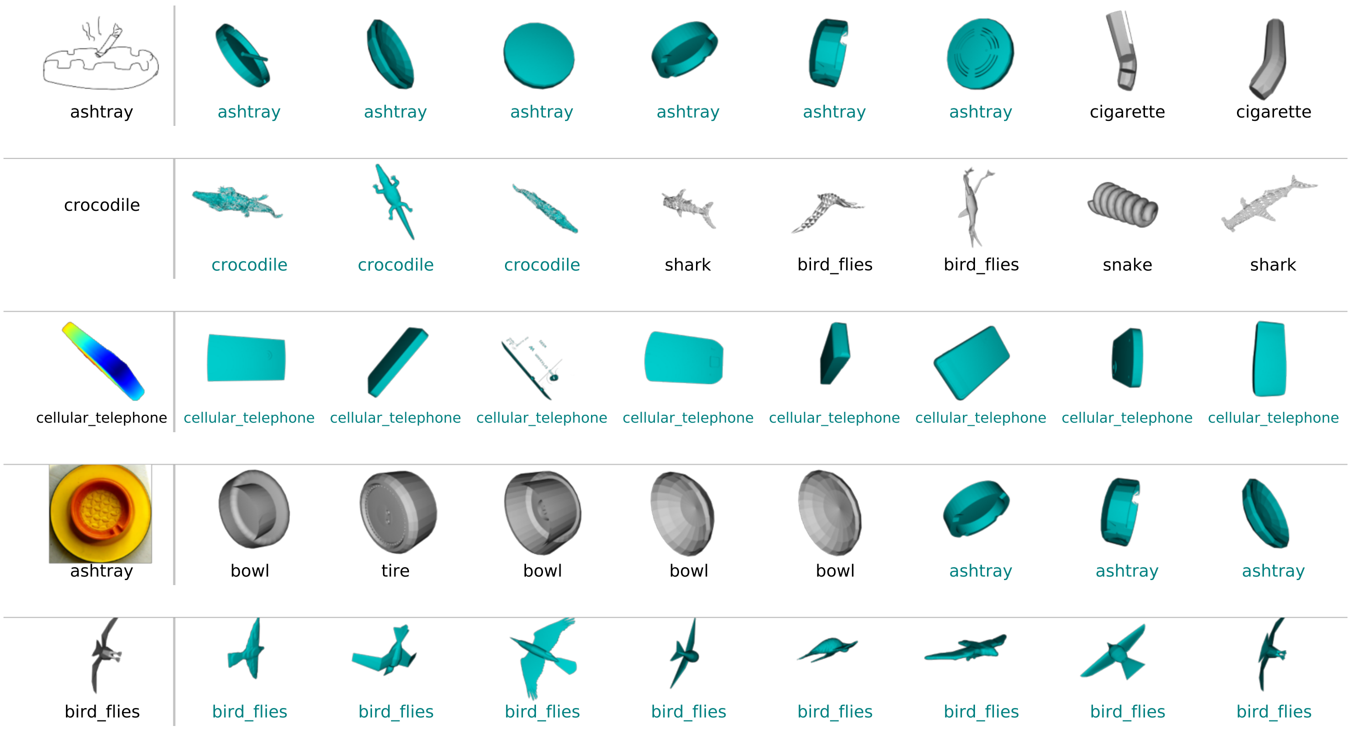


Figure 1: *Cross-modal shape retrieval examples using different input modalities. From the top: a sketch, a word, a synthetic depthmap, a natural image and a 3D model query. Each object has its ground-truth class displayed below it. We represent all these modalities in a common vector space of words, making it possible to assess semantic similarity and perform cross-modal retrieval. Relevant objects are highlighted in dark cyan.*

Abstract

Convolutional neural networks have been successfully used to compute shape descriptors, or jointly embed shapes and sketches in a common vector space. We propose a novel approach that leverages both labeled 3D shapes and semantic information contained in the labels, to generate semantically-meaningful shape descriptors. A neural network is trained to generate shape descriptors that lie close to a vector representation of the shape class, given a vector space of words. This method is easily extendable to range scans, hand-drawn sketches and images. This makes cross-modal retrieval possible, without a need to design different methods depending on the query type. We show that sketch-based shape retrieval using semantic-based descriptors outperforms the state-of-the-art by large margins, and mesh-based retrieval generates results of higher relevance to the query, than current deep shape descriptors.

Keywords: shape descriptor, word vector space, semantic-based, depthmap, 2D sketch, deep learning, CNN

Concepts: •**Computing methodologies** → **Shape representations;** *Image representations;*

SIGGRAPH Asia 2016 Technical Papers, December 5-8, 2016, Macao
ISBN: 978-1-4503-ABCD-E/16/07
DOI: http://doi.acm.org/10.1145/9999997.9999999

1 Introduction

Shape retrieval is increasingly important in light of the recent technological advancements in shape acquisition and the growing online repositories of 3D models. The problem consists of retrieving from a collection of models, shapes most similar to a given query. The underlying challenge is assessing the similarity between the query and objects in the collection. Biasotti et al. [2015] identify shape similarity though descriptors as one of the prevalent approaches in the literature. Shapes are represented by multidimensional vectors called *descriptors* or *signatures*, and a chosen metric over the shape descriptor space is used to assess similarity. We propose Shape2Vec, a method for computing semantic-based descriptors, that can be used to compute semantic similarity between shapes, sketches, images, depth maps, and words. We show that retrieval based on Shape2Vec descriptors outperforms previous sketch-based shape retrieval methods [Wang et al. 2015b] by $\mathbf{49}\%$ better average precision. This impressive improvement in performance is due to capturing semantic features as well as visual features in the descriptors.

Recently, deep convolutional neural networks (CNN) have been tremendously successful for learning discriminative shape descriptors [Wu et al. 2015; Su et al. 2015; Masci et al. 2015]. These networks learn descriptors that minimize the distance between similar shapes, and maximize the distance between shapes from different classes. Other methods embed both 3D shapes and images [Li et al. 2015b], or sketches [Wang et al. 2015b], in the same vector space. This makes it possible to search 3D models given an image query, or a sketch query. This is often referred to as *cross-modal* retrieval. Shape2Vec is a CNN that embeds both shapes and words in a common vector space, and thus learns semantically-meaningful descriptors.

Shape2Vec is inspired by the deep visual-semantic embedding model (DeViSE) [Frome et al. 2013] for image classification. DeViSE addresses two shortcomings of previous classification methods: they attempt to assign images to a small discrete number of selected classes and treat all labels as disconnected. CNN-based shape descriptors share the same limitations. DeViSE addresses these problems in image classification by leveraging both labeled images and semantic information from an unannotated text corpus. The text corpus is used to generate vector representations of words, and a CNN is trained to embed images in the word vector space. This transfers semantic information from the text corpus to visual object recognition, and produces semantically-meaningful image descriptors. We investigate how well leveraging both semantic information and visual information can improve 3D shape descriptors. Moreover, we train an additional CNN to learn similarly described sketches and images, using a fixed word vector space. This allows similarity assessment between all the different modalities, as illustrated by cross-modal shape retrieval results in Figure 1.

This is, to the best of our knowledge, the first attempt to represent such a large number of modalities in a word vector space. DeViSE [Frome et al. 2013] embeds one modality, namely natural images, in a word vector space using one language model. In contrast, we embed several modalities including 3D shapes. We also evaluate two different language models. Semantic-based shape retrieval has been explored in the past by representing shapes based on attributes such as "natural", "flexibility", "fly", "swim", and "rectilinearity" [Gong et al. 2013]. Our work uses word embeddings in a vector space, which provides a continuous representation that encodes semantic information. CNN have been used to embed 3D shapes and images [Li et al. 2015b] or sketches [Wang et al. 2015b] in a common vector space. However, these methods train one or two connected CNN with pairs of semantically similar input from each modality. We take a different approach by fixing a word vector space and training separate, disconnected, CNN to embed each modality in this fixed vector space.

Generating semantic-based descriptors has several benefits beyond cross-modal retrieval. One of them is the ability to support text queries that are not in the small set of classes used for training. This make text-based retrieval more flexible and not restricted to known class labels. Users can use new text queries and, receive relevant results if the query is semantically close to a known shape class.

This paper makes the following contributions:

1. A novel language model for vector representation of words, restricted to physical objects and based on human-labeled semantic relationships between objects (Section 5.2).
2. Embedding of 2D depthmaps, 3D shapes, 2D sketches and natural images in a word vector space (Section 6).
3. Cross-modal shape retrieval with semantic-based embeddings (Section 7).
4. Fine-tuning of a CNN trained over synthetic depthmaps for the embedding of real-world RGB-D images (Section 7.5).

2 Related Work

Shape retrieval has traditionally used view-based global descriptors such as spherical harmonics [Kazhdan et al. 2003], or Bag-of-features (BOF) retrieval systems that represent a shape by encoding local features. These use hand-crafted features to assess similarity. Learning features from training examples can improve this assessment. In that direction, CNN have become increasingly popular for representing shapes.

Deep shape descriptors 3D Shapenets [Wu et al. 2015] represent shapes as probability distributions of binary variables on a voxel grid, by training a convolutional deep belief network. Retrieval based on these descriptors outperforms previous hand-crafted view-based shape descriptors such as Spherical harmonics [Kazhdan et al. 2003]. One of the limitations of using 3D volumes as input is the loss in detail when shapes are voxelised. Su et al. [2015] propose a Multi-view CNN (MVCNN) which consists of learning descriptors from 2D rendered views and learning how to integrate these image-based descriptors in a single shape descriptor. They outperform 3D Shapenets by a large margin (49.2% to 80.2% average precision). Our work on 3D shape description is similar to MVCNN in that we use rendered depthmaps to generate image-based descriptors. It differs by the fact our descriptors are embedded in a word vector space while MVCNN image descriptors encode only visual features. Generating shape descriptors based on multiple views can be time-consuming and challenging for real-time retrieval. Bai et al. [2016] propose real-time shape retrieval, using GPU acceleration and two inverted files (GIFT). Their reported results show that GIFT outperforms hand-crafted methods and MVCNN on datasets with about $10K$ shapes divided into classes. However, MVCNN outperforms GIFT on a larger dataset, ShapenetCore, of about $51,300$ models from 55 classes subdivided into subclasses. We show that Shape2Vec outperforms GIFT on ShapenetCore, across all performance metrics, and retrieves results with higher relevance than MVCNN. Geodesic CNN [Boscaini et al. 2016; Masci et al. 2015] extends CNN to non-Euclidean manifolds and generates intrinsic shape descriptors, invariant to pose changes. However the use of a geodesic local coordinate system means it has limited support for noisy shapes like range scans.

The above methods learn shape descriptors for mesh-based retrieval. Another class of CNN in shape understanding embed models and other modalities in a joint vector space for cross-modal retrieval applications.

Joint embedding of shapes and other modalities Wang et al. [2015b] jointly train two connected CNN (*Siamese* networks), one for 2D rendered views and the other for hand-drawn sketches. They feed the networks with pairs of views and sketches from the same class and use a loss function based on within-domain as well as cross-domain similarity. They outperformed previous state-of-art in the SHREC'14 Large-scale Sketch-based Shape Retrieval Challenge [Li et al. 2014b]. We show (Section 7.2) that sketch-based retrieval using Shape2Vec descriptors for sketches and shapes achieves a better performance (22.8% to 72% AP). Li et al. [2015b] embed natural images of objects in a shape embedding space by training a CNN using realistic rendered images of shapes. The embedding space is constructed using non-linear Multi-Dimensional Scaling (NMDS) on pairwise similarities of training $3D$ models. A CNN is then trained to embed images in this embedding space. Our method shares some similarities with this approach: we use a fixed embedding space based on a single modality (text in our case), and one of our language models is based on NMDS over pairwise semantic similarities between words. On the other hand, we embed more modalities than images, making Shape2Vec applicable to a wide variety of tasks. Wang et al. [2015a] learn a joint embedding of depth and color images for RGB-D object recognition. Their results on multi-modal classification show 10% improvement in accuracy over using only RGB channels or depth images. We analyse retrieval on a challenging dataset consisting of RGB-D images, taken by normal users in uncontrolled settings (Section 7.5).

Convolutional Neural Networks Deep learning for shape representation has been inspired by the recent success of CNN in image classification [Krizhevsky et al. 2012]. CNN are composed of several layers of linear and non-linear operators that are learned jointly to perform a given task such as classification and feature extraction [Karpathy 2015]. Through these layers, CNN automatically learn increasingly complex feature maps. The main building blocks of modern CNN are: convolution layers (Conv) based on banks of learnable filters, an activation function such as the rectifier linear unit (ReLU), pooling layers (MaxPool) to reduce the spatial size of feature maps, and fully-connected layers (FC) that correspond to traditional single-hidden-layer neural network common for logistic regression. Dropout [Srivastava et al. 2014] is often used to overcome overfitting due to a large number of parameters, by turning off or on neurons during a training iteration based on a given probability.

There are several deep learning frameworks that efficiently implement the above building blocks, such as Berkeley Caffe [Jia et al. 2014] and Google Tensorflow [Abadi et al. 2015]. We use Tensorflow.

The next sections describe how we use CNN to compute semantic-based descriptors.

3 Shape2Vec overview

Shape2Vec generates semantic-based shape descriptors that correspond to vector representations of the shape class label. In this section, we provide an overview of Shape2Vec and present the datasets that are used for training and testing in the rest of the paper.

3.1 Shape2Vec

We generate shape descriptors as follows. Descriptors are first generated for depthmaps taken from multiple viewpoints. These depthmap descriptors are averaged to obtain a 3D shape descriptor (descriptors for images and sketches are discussed in Section 4.3). Assuming a known method for converting words to their vectorial representation (we use Word2Vec and WordNet, see Section 5), we generate depthmap descriptors in two stages: *classification* to predict depthmap labels and *encoding* to produce semantically-meaningful descriptors.

Classification The first stage trains a CNN to predict depthmaps labels, similarly to the DeViSE model for natural images [Frome et al. 2013]. This CNN learns class-specific visual features in depthmaps. The softmax function is applied to the final layer of the CNN to output vectors that represent class probabilities. We will refer to this CNN as the *Softmax classifier*.

Encoding This stage fine-tunes the parameters learned in the Softmax classifier by training it to generate, in the final layer, depthmap descriptors similar to vector representations of depthmap labels. Only the parameters in the FC layers are updated during this second training, to preserve the visual features learned in the Conv layers. This CNN, which we will often referred to as the *encoder*, can be evaluated as a classifier by returning the nearest word to a depthmap descriptor as the predicted class. We will use this approach to compare the classification accuracy of the Softmax classifier and the encoder.

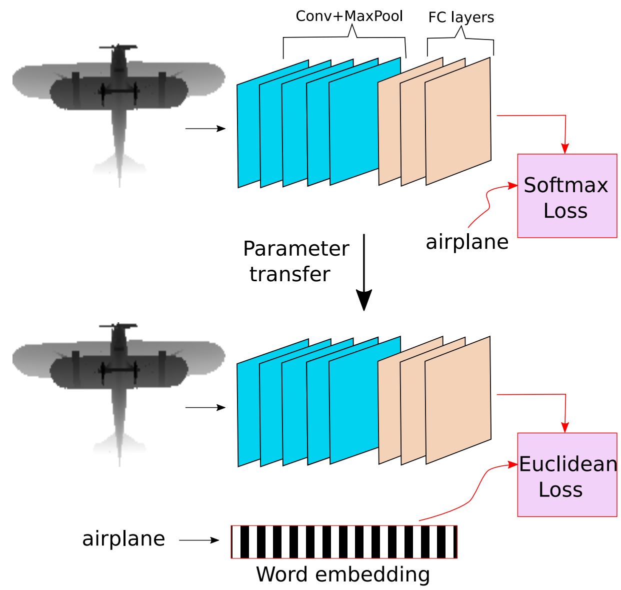


Figure 2: *Overview of the system. Assuming a known vector space of words, Shape2Vec generates semantic-based depthmap descriptors in two steps. Top: class-specific visual features in depthmaps are learned by training a Softmax classifier to predict depthmap classes. In this case,a class is represented by an index between* 0 *and* $K-1$*, where* K *is the number of classes. The classifier outputs class probabilities. Bottom: the parameters learned for object classification are fine-tuned in a second CNN, which is trained to generate a depthmap descriptor close to the word embedding of the depthmap class. The output is a vector similar to a word embedding. There are three differences between the two CNNs: the representation of the class label (an index* vs *a vector), the output layer (class probabilities* vs *descriptors), and the loss function.*

Word embeddings in a vector space The previous step assumes a known method for computing vector representations of words. Such methods are often referred to as *language models*. We select two language models and evaluate how they affect semantic-based descriptors. The first is based on *Word*2*Vec* [Mikolov et al. 2013], an unsupervised encoder for words, trained using words contexts in a large text corpus. The embedding space generated by Word2Vec often contains millions of words including concepts and verbs. To obtain an embedding space restricted to objects, we propose also a novel language model based on *Wordnet*, a hierarchy of synonym sets (*synsets*). We select a subset of synsets representing physical entities and learn vector representations of these synsets using non-linear Multidimensional Scaling (NMDS) on their pairwise semantic similarities. Despite our initial hypothesis that the Wordnet approach would be better, our results show that Word2Vec is superior to WordNet.

To train a CNN for shapes, sketches and images, large training datasets are needed. The next section describes the dataset sources used for the results presented in this paper.

3.2 Datasets

Deep CNN require large amounts of data for training that will not overfit. In order to evaluate cross-modal retrieval, our choice of datasets is limited to those with more than one modality.

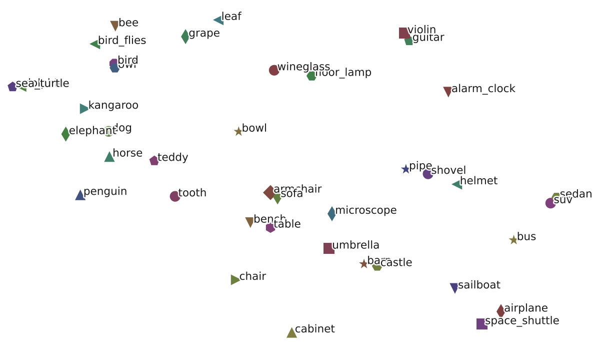


Figure 3: *2D visualisation of selected class label embeddings in Word2Vec. Embeddings are projected in 2D using parametric t-SNE [van der Maaten 2009].*

SHREC'14 Large Sketch-based Shape Retrieval Challenge This dataset [Li et al. 2014b; Li et al. 2015a] is the largest available that contains both labeled sketches and 3D shapes. It consists of data from previous datasets of shapes [Li et al. 2014a] and hand-drawn sketches [Eitz et al. 2012]. The collection has an unbalanced set of $8,987$ 3D models and a balanced set of $13,180$ sketches from 171 classes. We denote the set of shapes by **SHREC14-3D** and the set of sketches by **SHREC14-Sketch**. For each 3D model, we generate depth images from 12 views located at the vertices of a bounding icosahedron, for fast computation. We aggregate depthmaps class predictions or semantic-based descriptors by averaging. An alternative method consisting of assigning more weights to views that show more area of the shape (*view entropy*) does not impact classification or retrieval. We denote the set of depthmaps by **SHREC14-Depth**.

ImageNet subset ImageNet [Russakovsky et al. 2015] is a large database of images organized according to the Wordnet hierarchy [Fellbaum 1998]. Wordnet itself is database of words grouped into sets of synonyms or synsets. ImageNet contains about $21,841$ synsets, with an average of 500 images per synset. Subsets of ImageNet are commonly used for Computer Vision challenges such as image classification [Krizhevsky et al. 2012]. From the 171 classes in the SHREC14-3D dataset, only 144 had matching synsets in Imagenet. For computational purposes, we download at most 100 images per matching synset. We refer to the resulting dataset as **IMAGENET-Sub**.

We split the datasets above for training, validation and testing. First we set aside 20% of each dataset for testing. SHREC14-$Sketch$ was already divided into a training and a testing dataset. To decide on the CNN configurations and hyperparameters, we use a small validation set: 20% of the training dataset. The assignment of an object to a split is random. We will attach the terms *-Train*, *-Val*, *-Test*, or *-All* to the dataset name to refer to a particular split or the complete dataset. For instance, to generate depthmap descriptors, we train the Softmax classifier and the encoder on SHREC14-Depth-Train.

We later show results on a dataset of real RGB-D images [Choi et al. 2016] (Section 7.5) and ShapeNetCore, which is the largest academic shape dataset to date [Chang et al. 2015] (Section 8). The next sections describe each of the building blocks of Shape2Vec.

4 Learning shape classes

This section describes classification of depthmaps using CNN, as well as results of similar CNN classifiers for other modalities such as sketches.

4.1 Depthmaps

We train a CNN for depthmap classification, similarly to DeViSE. The CNN parameters will be fine-tuned later to learn semantic embeddings of depth images. The chosen network architecture is based on AlexNet [Krizhevsky et al. 2012], consisting of about 60 million parameters. AlexNet has been successfully used for a wide range of computer vision tasks such as image classification [Krizhevsky et al. 2012], shape retrieval [Su et al. 2015] and sketch recognition [Yu et al. 2015].

AlexNet is a multi-layer network consisting of one input layer, a combination of 5 Conv+MaxPool layers and 3 FC layers. The classical AlexNet has Local Response Normalization (LRN) layers applied at the end of the first two Conv+MaxPool layers. LRN is supposed to provide lateral inhibition present in real neurons, but in practice, there was no improvement in the depthmap classification accuracy with LRN added. On the other hand, removing it improves learning speed. Setting initial parameters of the neural net using parameters optimized for image classification has been successful for shape recognition [Su et al. 2015]. We use the same scheme here, and initialize the CNN with parameters learned for image classification in the ImageNet challenge [Krizhevsky et al. 2012] and made available by Caffe [Jia et al. 2014]. Parameters are updated during training to minimize the Softmax loss, which was also used in AlexNet. The Softmax loss or cross-entropy loss given input depthmap i is

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_{j=0}^{K-1} e^{f_j}}\right) \tag{1}$$

where y_i is the true label of input i, K is the number of classes, and f is the Softmax function. This function is defined by:

$$f_j = \frac{e^{z_j}}{\sum_{k=0}^{K-1} e^{z_k}}, \tag{2}$$

where z_i is an output of the last FC layer i.e. a score for each class given the input depthmap. Softmax takes a vector of real-valued class scores, and normalises them so that they sum up to 1.0. The output can be interpreted as unnormalized log probabilities for each class. The total loss L is the mean of individual losses L_i over a batch of training input, plus regularization terms such as L2 regularization that encourages parameters to be small. We use Adagrad [Duchi et al. 2011] for the optimisation. Adagrad is an adaptive learning rate method that adaptively determine how much individual parameters should be updated based on the previous behaviour of their gradients.

The method above trains a network to output class probabilities, given an input depthmap. Parameter optimization converges after 100 epochs (epoch=number of times the whole training dataset is processed). Note that SHREC14-Depth-Train consists of $107,844$ views. Classification accuracy on SHREC14-Depth-Test is $\mathbf{77.9}\%$. This is the *top-1* or *nearest-neighbour* accuracy, where the classifier returns the correct class as the best match.

4.2 3D models

Class probabilities of all 12 depthmaps of a shape are averaged to predict its class. Assigning weights according to view entropy

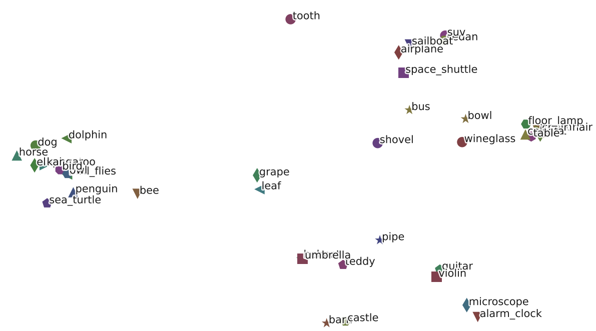


Figure 4: *2D visualisation of selected class label embeddings in Wordnet-based vector space (WN).*

does not affect performance. The Softmax classifier recognises 3D shape classes with an accuracy of $\mathbf{87.7}\%$ on SHREC14-3D-Test and $\mathbf{96.5}\%$ on SHREC14-3D-All. In contrast, Tatsuma et al. generate shape descriptors using Super Vector encoding of view-based features and achieve an accuracy of 86.8% on SHREC14-3D-All when the nearest neighbour class is returned as the predicted class [Li et al. 2015a]. This indicates that the Softmax classifier is better on average at predicting shape classes.

4.3 2D sketches and natural images

We also train two separate CNN using the same architecture to classify sketches and natural images.

The sketch classifier achieves an accuracy rate of $\mathbf{72.6}\%$ on SHREC14-Sketch-Test, lower than the state-of-the-art SketchNet [Yu et al. 2015] accuracy of 74.9% on a larger dataset of sketches from 250 classes. Thus SketchNet, which uses an ensemble of CNNs with a similar architecture to our Softmax classifier, performs slightly better on average.

The image Softmax classifier achieves $\mathbf{43.2}\%$ accuracy, which is significantly lower compared to other modalities. This is because, contrary to depthmaps and sketches, an image can contain multiple objects. The input data is more complex, and our classifier overfits on the training data. With larger training data, the classifier may learn invariance to background. Note that the original AlexNet network won the 2012 ImageNet image classification task [Russakovsky et al. 2015] (1 million images from 1000 classes) with a top-5 accuracy rate of 83.5%. In contrast, we achieve a top-5 accuracy of 70.2%, using IMAGENET-Sub which has $14,100$ images from 144 classes.

We report the accuracy results above and compare them with classification based on the semantic-based encoder in Section 6. Once trained for classification, the CNN are ready to be fine-tuned to generate semantic-based descriptors close to word embeddings.

5 Learning word embeddings

This section focuses on learning a language model, that maps words in a text corpus to vectors in the Euclidean space. We present one language model from the natural language processing literature and propose a new language model.

5.1 Word2Vec

Word2Vec [Mikolov et al. 2013] belongs to the class of vector space models that map words to a continuous vector space, such that semantically similar words correspond to nearby points. In particular, the Word2Vec neural network efficiently learns word embeddings from unannotated text, such that words that occur in the same context are mapped to vectors with a small cosine distance. It captures both semantic and syntactic relationships, and supports basic algebraic operations such as "$king - man + woman = queen$". Word2Vec propose two architectures to learn word vector representations: Continuous Bag-Of-Words model (CBOW) and Skip-Gram models. CBOW predicts a word (e.g. "mat") given its context ("the cat sits on the"). The number of words used to determine a context is based on a window size. On the other hand, Skip-Gram predicts source context words from a target word. CBOW is faster while Skip-Gram performs better on small training data.

We chose CBOW for fast computation and use an open source implementation of Word2Vec [Mikolov et al. 2013] that generates a large model from a public corpus of 8 billion words tokenized into a set of $1,111,684$ single- and multi-word terms. The model produces 500-dimensional word embeddings, based on CBOW, using a 10-word window size. Figure 3 visualizes vector representations of a subset of SHREC14-3D labels in 2D. Note how mammals are grouped together, as well as vehicles. The visualization indicates that Word2Vec learns semantic relationships between words.

Although Word2Vec seems to accurately capture semantic similarities between words, it contains more than 1 million words, a large fraction of which are not nouns and even fewer are names of physical objects. We propose a second language model, restricted to physical entities and based on ground-truth semantic relationships labeled by humans.

5.2 Non-linear multi-dimensional scaling using Wordnet

Wordnet [Fellbaum 1998] is a taxonomy curated by humans, that establishes how synsets (sets of synonyms) are related in a hierarchical structure. For instance "carnivore" has as children "dog" and "cat", and each has their own children which are different dog and cat species. In this taxonomy, semantic similarity between two words is based on the shortest path between them. One of the widely used metrics in Wordnet is the wup similarity [Wu and Palmer 1994]. wup measures the relatedness between two synsets by considering their depth in the taxonomy and the depth of their lowest common subsumer (most specific ancestor) lcs:

$$wup(A, B) = 2\frac{depth(lcs(A, B))}{depth(A) + depth(B)}. \tag{3}$$

wup provides an implicit representation of the space of synsets, but it can not be plugged directly into the CNN described in Section 4. A vector representation of words is required. We learn these wup-based vector representations using non-linear Multidimensional scaling (NMDS) [Kruskal 1964]. Given pairwise wup distances between a set of words, we use NMDS to generate 100-D vectors for each word, such that Euclidean distance between two word vector representations is close to the original wup distance between the words. Wordnet contains $155,287$ words organized in $117,659$ synsets. We reduce this number since computing pairwise wup similarities is expensive.

We restrict the list of synsets to those that are within $r = 5$ edges in the Wordnet tree, to classes in the training dataset. We set $r = 5$, after preliminary experiments with r ranging from 3 to 8.

The selected value of the parameter r is a compromise between computational cost and vocabulary size. This not only restricts the vocabulary to words representing physical objects, but reduces the complexity of pairwise similarity comparisons and NMDS. The final vocabulary consists of $12,008$ words, from 171 classes present in training. We use 100 dimensions in this language model, as opposed to 500 used for Word2Vec because the vocabulary size is 3 orders of magnitude smaller, compared to 1 million words vocabulary in the Word2Vec model. Preliminary 2D visualization of 500-D embeddings of the SHREC14-3D class labels showed poor performance. To visualise the embeddings in 2D, we compute a matrix of pairwise cosine distances between label vectors. The matrix is used to learn 2D embeddings using t-SNE [van der Maaten 2009], which is the standard method for mapping high-dimensional vectors to 2D for visualisation purposes. Figure 4 shows a visualization of selected class labels using 100-D embeddings. Similar classes such as mammals are tightly grouped and far away from unrelated classes such as vehicles. We denote the proposed language model by **WN**, for Wordnet.

We manually create a one-to-one mapping of SHREC14-3D class labels between the Wordnet and Word2Vec vocabularies, so that either language model can be used. Given these two vector representations of words in a vector space, the Softmax classifier is modified and fine-tuned to generate shape embeddings that lie in the same vector space.

6 Learning semantic-based shape descriptors

We present how the Softmax classifier, described in Section 4, is modified to generate semantic-based descriptors.

6.1 Depthmaps

The last layer of the Softmax classifier outputs class probabilities for each class in SHREC14-Depth. We change this layer, and the loss function to obtain an encoder that learn depthmap embeddings. The penultimate layer now outputs a L2-normalized descriptor with the same dimensionality as the word vector space. The loss function is selected such that the network is trained to output descriptors that are close to the vector representation of the depthmap class label. We investigate the influence of three loss functions:

- L_2 loss: Often referred to as the Euclidean loss, it generates descriptors that are as close to the class vector representations as possible, according to the L_2 norm. Let $v(y_i)$ be the vector representation of the class y_i then the loss associated with the input i is:
$$L_i^l = ||s_i - v(y_i)||_2 \tag{4}$$
where s_i is the generated shape descriptor.
- Cosine Distance: This minimizes cosine distance between shape descriptors, and their associated class. We investigate this loss function because words in the both language models are compared using cosine similarity.
$$L_i^c = 1 - s_i.v(y_i) \tag{5}$$
- Rank hinge loss: The above loss functions only attempt to select shape descriptors close to correct or positive class, without taking into account negative classes. The hinge loss was successfully used in the visual-semantic model of images [Frome et al. 2013], to ensure that image descriptors were far from negative classes with a given margin. The loss function is
$$L_i^h = \sum_{j \neq y_i} \max(0, \alpha - s_i.v(y_i) + s_i.v(j)) \tag{6}$$
where α is the margin, set in our implementation to 0.3 based on empirical results on a small validation dataset.

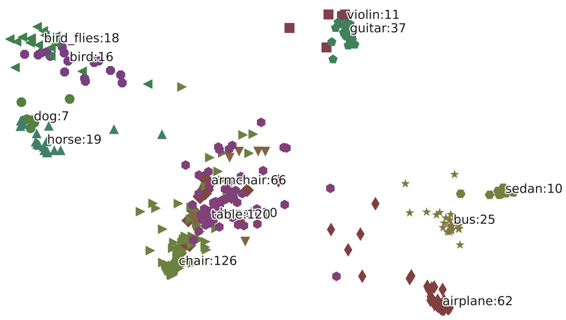
The Conv layers in the neural net are fixed and only parameters of FC layers are updated to minimize the selected loss function. Thus, visual features learned during classification are preserved. We chose the same optimization method, Adagrad, used for training classifiers in Section 4.

A 3D shape descriptor is obtained by averaging its depthmap descriptors, similarly to how class probabilities were aggregated. We refer to CNN based on L2 loss, Cosine Distance loss and Hinge loss as **L2-W2V**, **CosineDist-W2V**, and **HingeLoss-W2V** respectively when Word2Vec is used. We replace -W2V with **WN** when referring to an encoder based on WN embeddings. Classification and retrieval accuracy are reported on all six methods, in addition to the Softmax classifier described in Section 4 when applicable.

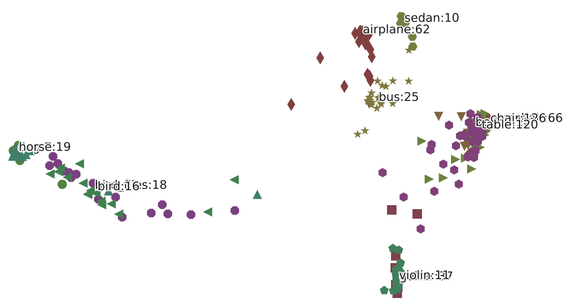
Shape embedding visualisation Figure 5 shows 2D visualisations of shapes from a subset of classes. Note that for the purpose of visualisation, we choose parametric t-SNE [van der Maaten 2009] for all 2D projections in this paper, as opposed to the traditional t-SNE, so that parameters can be learned for projecting word embeddings to 2D, and then used for shape embeddings. Figure 5 shows two projections of shape descriptors trained with the L2 loss, using the Word2Vec and the WN models. Note how with an encoder based on Word2Vec, shapes from the same class form clusters, indicating that the distance between them is small as expected. In contrast WN does not discriminate between shapes from similar classes such as "chair", "bench", and "table", while shapes from unrelated classes are clearly separated.

Semantic-based classification We compare how well the encoder classifies depthmaps, by selecting the word whose vector representation is closest to a depthmap descriptor, as the predicted class. Top-k accuracy retrieves the first k most confident classes for a depthmap and returns 1 if one of them is correct. Table 1 shows classification results per loss function and language model, compared to the Softmax classifiers in Section 4. Note that by returning words close to embeddings as predicted classes, the size of possible results is no longer limited to the 171 labels in the SHREC-3D dataset. It is expanded to the whole vocabulary of the underlying language model. In Table 1, we include accuracy results where predicted classes are restricted to classes in the dataset.

Results show that L2-W2V and CosineDist-W2V outperform the other four encoders, with SHREC14-Depth-Test top-1 accuracy of $\mathbf{77.7}\%$ and SHREC14-3D-Test top-1 accuracy of $\mathbf{87.4}\%$. It has similar top-1 accuracy to the Softmax classifier even though the size of possible guessed classes is 1 million when using Word2Vec. HingeLoss-W2V performs slightly worse at $\mathbf{86}\%$ on 3D shapes compared to the former, but HingeLoss-WN has a dramatic drop in performance at $\mathbf{21.6}\%$. This suggests that the hinge loss or hinge margin is not appropriate for the WN language model. Note however, that HingeLoss-WN shows a significant improvement, especially in top-10 accuracy when the predicted class is restricted to one of the 171 classes in the training dataset. Also note that irrespective of the loss function, WN-based embeddings perform worse than Word2Vec embeddings, which supports our interpretation of 2D visualisation of shape embeddings in both vector spaces.



Shape embeddings in Word2Vec (L2-W2V)



Shape embeddings in WN (L2-WN)

Figure 5: *2D projections of 3D shape descriptors embedded in word vector spaces.*

	Depthmaps			3D models		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Softmax classifier	**0.779**	**0.936**	**0.962**	**0.877**	**0.962**	**0.980**
L2-W2V	0.774	0.832	0.837	0.874	0.922	0.928
L2-W2V*	0.774	0.862	0.866	0.874	0.943	0.950
L2-WN	0.537	0.633	0.668	0.583	0.701	0.756
L2-WN*	0.655	0.811	0.845	0.723	0.887	0.910
CosineDist-W2V	0.777	0.835	0.843	0.874	0.926	0.932
CosineDist-W2V*	0.777	0.867	0.872	0.874	0.945	0.950
CosineDist-WN	0.538	0.639	0.676	0.592	0.709	0.767
CosineDist-WN*	0.654	0.813	0.849	0.727	0.885	0.911
HingeLoss-W2V	0.732	0.856	0.871	0.860	0.936	0.944
HingeLoss-W2V*	0.734	0.903	0.913	0.861	0.961	0.966
HingeLoss-WN	0.185	0.247	0.289	0.216	0.284	0.336
HingeLoss-WN*	0.504	0.795	0.842	0.579	0.878	0.916

Table 1: *Top-k classification accuracy for depthmaps (SHREC14-Depth-Test) and 3D models (SHREC24-3D-Test). Classification based on an encoder can output any of the words in the language model vocabulary. The number of possible classes is $1,000,000$ (Word2Vec) or $12,000$ (WN). A star (*) indicates results where output classes were restricted to one of the 171 class labels in the training dataset. This provides a fairer comparison against the Softmax classifier. Note how this restriction does not affect top-1 accuracy for encoders based on Word2Vec, but significantly improves the accuracy of WN-based encoders. It also improves top-5 and top-10 accuracies for all encoders.*

	Hand-drawn sketches			Natural images		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
Softmax classifier	**0.726**	**0.923**	**0.959**	**0.430**	**0.702**	**0.800**
L2-W2V	0.723	0.780	0.787	0.381	0.444	0.454
L2-W2V*	0.723	0.818	0.823	0.382	0.505	0.527
L2-WN	0.585	0.679	0.713	0.261	0.341	0.382
L2-WN*	0.664	0.804	0.833	0.321	0.475	0.534
CosineDist-W2V	0.725	0.776	0.785	0.387	0.446	0.455
CosineDist-W2V*	0.725	0.818	0.825	0.388	0.511	0.533
CosineDist-WN	0.587	0.681	0.715	0.275	0.354	0.392
CosineDist-WN*	0.667	0.799	0.835	0.338	0.485	0.547
HingeLoss-W2V	0.544	0.713	0.752	0.427	0.567	0.596
HingeLoss-W2V*	0.596	0.835	0.860	0.435	0.667	0.687
HingeLoss-WN	0.345	0.437	0.501	0.175	0.222	0.268
HingeLoss-WN*	0.581	0.784	0.823	0.308	0.481	0.544

Table 2: *Top-k classification accuracy for 2D sketches (SHREC14-Sketch-Test) and natural images (IMAGENET-Sub-Test). The star refers to results where the predicted classes are restricted to those used in training.*

6.2 Hand-drawn sketches and natural images

We generate semantic-based descriptors for 2D sketches using the above methods. The sketch Softmax classifier is fine-tuned to learn sketch embeddings in a word vector space. On SHREC14-Sketch-Test, the classifier achieves a top-1 accuracy rate of $\mathbf{72.6}\%$. Once the sketches are embedded in a vector space, L2-W2V and CosineDist-W2V semantic-based classification has a similar top-1 accuracy of $\mathbf{72}\%$. Table 2 shows accuracy results per loss function and language model. The observations made for 3D shapes, regarding accuracy per loss function and language model, also hold for sketches.

In contrast to both shapes and sketches, semantic-based image classification accuracy significantly drops compared to the image Softmax classifier (a drop of 5% in top-1 accuracy and 35% in top-10 accuracy). This indicates a loss of visual information when embedding images in a word vector space. DeViSE, which inspired Shape2Vec, reports a drop of performance of 2% accuracy compared to the Softmax classifier [Frome et al. 2013]. This suggests the larger drop of performance here is due to the small training dataset for images and the complex nature of images. DeViSE uses Word2Vec as its language model, but differs from Shape2Vec in one significant aspect. In Shape2Vec, the language model remains fixed throughout the encoder training, whereas DeViSE updates the weights of the neural network that generates vector representations of words. Thus, its final vector space is adapted to the image dataset. This could help limit the drop in image classification accuracy when images are embedded. We chose neither to update the language model nor to adapt it to a specific dataset, so that similarity can be assessed between descriptors generated from different CNN. Fine-tuning the language model to a dataset might otherwise have a negative impact on cross-modal retrieval.

7 Retrieval applications

We investigate shape retrieval performance on five types of queries: 3D shape, 2D sketch, natural image, text and natural RGB-D images. Performance is evaluated using these standard criteria: Precision-recall curve (PR), Average mean precision (AP), Nearest Neighbor (NN), First/SecondTier (FT/ST) and normalised Discounted Cumulative Gain (DCG). We also report results on one additional metric, the E-Measure (E). E is the harmonic mean of precision and recall for the top $K = 32$ retrieval and has been reported by previous retrieval methods on the datasets used here. We denote this additional metric by $\mathbf{E@32}$.

	NN	FT	ST
Ours (L2-W2V)	0.953	0.916	0.952
Ours (L2-W2V) SHREC14-3D-Test	**0.998**	0.849	0.898
Ours (L2-WN)	0.894	0.749	0.864
Ours (CosineDist-W2V)	0.954	**0.917**	**0.953**
Ours (CosineDist-WN)	0.895	0.746	0.862
Ours (HingeLoss-W2V)	0.920	0.747	0.878
Ours (HingeLoss-WN)	0.901	0.773	0.882
Bai (GIFT)	0.889	0.567	0.689
Tatsuma (LCDR-DBSVC)	0.865	0.528	0.661
	E@32	DCG	AP
Ours (L2-W2V)	0.373	**0.975**	**0.937**
Ours (L2-W2V) SHREC14-3D-Test	0.333	0.935	0.866
Ours (L2-WN)	0.326	0.929	0.788
Ours (CosineDist-W2V)	**0.374**	0.975	0.937
Ours (CosineDist-WN)	0.326	0.928	0.785
Ours (HingeLoss-W2V)	0.314	0.933	0.791
Ours (HingeLoss-WN)	0.329	0.937	0.810
Bai (GIFT)	N/A	N/A	N/A
Tatsuma (LCDR-DBSVC)	0.255	0.823	0.541

Table 3: *Comparison of mesh-based retrieval on SHREC14-3D-All. Although previous methods report results on the complete dataset and use machine learning techniques, none of them uses class assignments in SHREC14-3D. For a fairer comparison, we present the top retrieval performance of Shape2Vec, using only shapes never seen during training as queries.*

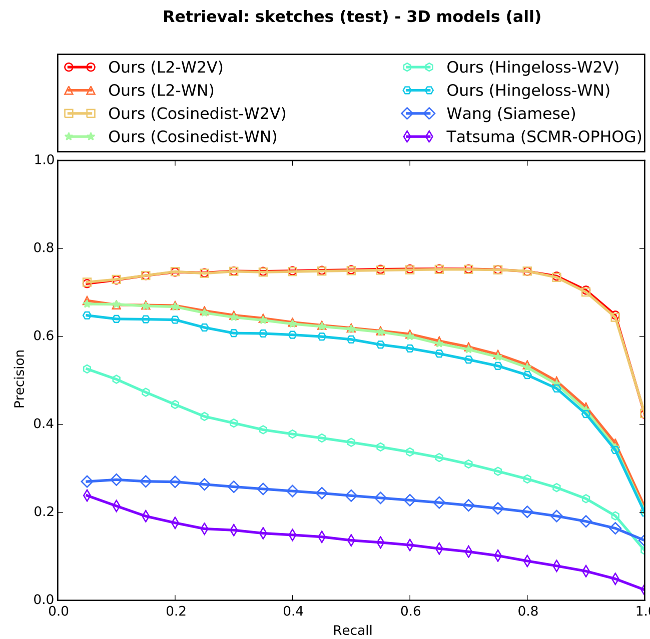


Figure 6: *Precision recall of mesh-based retrieval and sketch-based retrieval on the SHREC14 dataset.*

	NN	FT	ST
Ours (L2-W2V)	**0.714**	**0.697**	**0.748**
Ours (L2-WN)	0.599	0.523	0.598
Ours (CosineDist-W2V)	0.713	0.696	0.742
Ours (CosineDist-WN)	0.594	0.517	0.594
Ours (HingeLoss-W2V)	0.388	0.303	0.431
Ours (HingeLoss-WN)	0.557	0.506	0.590
Wang (Siamese)	0.239	0.212	0.316
Tatsuma (SCMR-OPHOG)	0.160	0.115	0.170
	E@32	DCG	AP
Ours (L2-W2V)	**0.360**	**0.811**	**0.720**
Ours (L2-WN)	0.306	0.707	0.546
Ours (CosineDist-W2V)	0.359	0.810	0.718
Ours (CosineDist-WN)	0.304	0.705	0.540
Ours (HingeLoss-W2V)	0.200	0.576	0.326
Ours (HingeLoss-WN)	0.292	0.696	0.529
Wang (Siamese)	0.140	0.496	0.228
Tatsuma (SCMR-OPHOG)	0.079	0.376	0.131

Table 4: *Comparison of sketch-based retrieval on SHREC14-Sketch-Test and SHREC14-3D-All.*

7.1 Mesh-based shape retrieval

We evaluate shape retrieval on SHREC14-3D-All, as done in previous retrieval methods on the same dataset. Table 3 presents the results of this evaluation against LCDR-DBSVC [Li et al. 2014a] and GIFT [Bai et al. 2016]. GIFT had the best reported performance on the shape dataset. L2-W2V improves on LCDR-SBSV by a 40% AP difference (54.1% to 93.7%) and on GIFT by a 29.1% ST difference. In fact, all semantic-descriptors outperform state-of-the-art, including HingeLoss-WN at 81% AP. Note however that when evaluating retrieval on the complete dataset, shapes that were used for training are included, which produces a biased result. When we restrict evaluation of our method to SHREC14-3D-Test, L2-W2V has a $\mathbf{86.6}\%$ AP, still outperforming LCDR-DBSVC by a difference of 32% AP and GIFT by a 23.7% ST difference.

The significant improvement is partly due to deep learning, which automatically learns class-specfic descriptors from the 3D shapes. In contrast LCDR-DBSVC learns how to encode hand-crafted local features from a separate set of unclassified models. GIFT generates view descriptors by training a CNN on 54,728 unrelated models from ModelNet [Wu et al. 2015] divided into 461 categories. In Section 8, we compare Shape2Vec against GIFT and MVCNN, using training and testing splits of the same dataset, ShapenetCore.

7.2 Sketch-based Shape retrieval

We report evaluation of retrieval from the complete shape dataset using SHREC14-Sketch-Test [Li et al. 2014b], upon which all state-of-the-art sketch-based methods report retrieval results. Figure 6 and Table 3 show comparison of performance against two recent sketch-based shape retrieval methods, including recent work on joint embedding of sketches and 3D models denoted by Wang (Siamese) [Wang et al. 2015b]. Again, Shape2Vec outperforms state-of-the-art by large margins, with L2-W2V improving on Wang (Siamese) by a 49% AP performance (22.8% to $\mathbf{72.0}\%$).

We believe that Shape2Vec shows significant improvement over previous work because it does not rely on learning distance metrics across modalities. Rather, it finds embeddings of these modalities in a common vector space, and as long as the embedding of each modality maps to similar points, high retrieval performance will be achieved. The results of semantic-based classification on both sketches and shapes showed comparable performance to the Softmax classifier (Section 6), indicating that even when embedded in a word vector space, little information on visual features is lost. Thus Shape2Vec is able to achieve these two tasks across domains with little trade-off: capture discriminative visual features and provide a common embedding.

	NN	FT	ST
Ours (L2-W2V)	**0.376**	0.374	0.420
Ours (L2-WN)	0.300	0.259	0.324
Ours (CosineDist-W2V)	0.375	**0.379**	**0.426**
Ours (CosineDist-WN)	0.304	0.264	0.328
Ours (HingeLoss-W2V)	0.334	0.279	0.376
Ours (HingeLoss-WN)	0.288	0.262	0.331
	E@32	DCG	AP
Ours (L2-W2V)	0.199	0.570	0.394
Ours (L2-WN)	0.162	0.504	0.283
Ours (CosineDist-W2V)	**0.202**	**0.574**	**0.399**
Ours (CosineDist-WN)	0.161	0.508	0.286
Ours (HingeLoss-W2V)	0.176	0.539	0.306
Ours (HingeLoss-WN)	0.161	0.502	0.285

Table 5: *Comparison of image-based shape retrieval on IMAGENET-Sub-Test and SHREC14-3D-All.*

	NN	FT	ST
Ours (L2-W2V)	**0.754**	**0.659**	**0.715**
Ours (L2-WN)	0.673	0.491	0.552
Ours (CosineDist-W2V)	0.749	0.659	0.710
Ours (CosineDist-WN)	0.655	0.476	0.543
Ours (HingeLoss-W2V)	0.743	0.611	0.690
Ours (HingeLoss-WN)	0.480	0.407	0.511
	E@32	DCG	AP
Ours (L2-W2V)	**0.283**	**0.742**	**0.689**
Ours (L2-WN)	0.245	0.632	0.530
Ours (CosineDist-W2V)	0.283	0.741	0.690
Ours (CosineDist-WN)	0.245	0.623	0.519
Ours (HingeLoss-W2V)	0.281	0.726	0.662
Ours (HingeLoss-WN)	0.226	0.554	0.424

Table 6: *Comparison of text-based shape retrieval on the* 171 *class labels and SHREC14-3D-All.*

7.3 Image-based Shape retrieval

For each image in IMAGENET-Sub-Test, we retrieve similar 3D shapes according to their embeddings. Table 5 summarizes the image-based retrieval performance. Note the low performance of the top-performing method CosineDist-W2V, which shows $\mathbf{39.9}\%$ AP. This is expected, based on the poor accuracy of the Softmax classifier and semantic-based classifier on images.

Because we constructed our own image dataset to match the classes in the shape dataset, we do not perform quantitative comparison against competing image-based shape retrieval. Also note that current methods on joint image-shape embedding are trained using synthetic images obtained by realistic rendering of 3D shapes in selected scenes [Li et al. 2015b]. In contrast, we use real images which are more diverse and complex.

7.4 Text-based Shape retrieval

One of the main motivations behind embedding shape descriptors in a space of words is the ability to use text queries not yet seen during training. However, this is difficult to show empirically, without attaching multiple classes to each shape for testing. We show retrieval performance of 3D shapes, based on text queries in the dataset. Given each of the 171 classes, we retrieve the most similar 3D shapes, based on their embeddings. Intuitively, the first result is the 3D shape most representative of that word. Table 6 summarizes our findings. L2-W2V has 75.4% NN performance, representing the probability of finding a 3D representative shape in the first result. Text-based retrieval shows lower performance compared to sketch-based and mesh-based retrieval, which may suggest that not only do shape and sketch embeddings capture semantics, they also contain additional information such as visual features.

	NN	FT	ST
Ours (L2-W2V)	0.640	0.658	**0.724**
Ours (L2-WN)	0.555	0.516	0.645
Ours (CosineDist-W2V)	**0.654**	**0.660**	0.724
Ours (HingeLoss-W2V)	0.603	0.590	0.708
Ours (HingeLoss-WN)	0.467	0.446	0.641
Ours (CosineDist-WN)	0.563	0.546	0.662
	E@32	DCG	AP
Ours (L2-W2V)	**0.147**	**0.848**	**0.693**
Ours (L2-WN)	0.107	0.804	0.536
Ours (CosineDist-W2V)	0.142	0.846	0.692
Ours (HingeLoss-W2V)	0.131	0.830	0.627
Ours (HingeLoss-WN)	0.084	0.760	0.457
Ours (CosineDist-WN)	0.113	0.815	0.567

Table 7: *Comparison of RGBD-based retrieval on the Stanford test dataset.*

7.5 Range scan-based shape retrieval

We trained the depthmap encoders on clean synthetic depthmaps. These depthmaps are different from real-world depth images since the latter often contain cluttered scenes, including background. We fine-tune the depthmap encoder on real-world range scans using a dataset of RGB-D images taken in realistic conditions.

Large Stanford RGB-D Dataset This is a recent dataset of RGB-D images [Choi et al. 2016], where human participants were given scanning devices and asked to scan objects in their everyday life, with no supervision and no control over what objects will be selected or from which distance they will be scanned. The authors identified 398 sequences from nine classes they use for mesh reconstruction. The set of labels we have used so far has 6 matching labels in the RGB-D dataset, namely: "bench", "table", "chair", "minibike", "sofa" and "pot". From each sequence we extract a frame every 10 seconds in the first 5 minutes. This provides a dataset of $2,704$ RGB-D images. Note that this dataset is particularly challenging due to thte variety of scanning environments, cluttered scenes and background. Directly testing the depth images using the CNN trained on synthetic depthmaps is not appropriate since they are different modalities. Thus, we fine-tune the synthetic depthmap encoder using 80% of the RGB-D dataset for training and 20% for testing. This dataset was recently released and has not yet being used for classification or retrieval, thus we cannot report comparison against state-of-the-art.

Depth-based retrieval Figure 7 shows examples of retrieval using depth images. It shows that even in ambiguous scenes, such as the depth scan of a minibike (middle), relevant results are still retrieved. The last example in Figure 7 shows a table scan confused with chair models. On average Shape2Vec achieves AP performance of $\mathbf{69.3}\%$ on real-world depth images. This is impressive considering the complexity and variations in the depth images.

Our evaluation of RGBD-based retrieval indicates that Shape2Vec can be fine-tuned to generate embeddings for new shape types.

This section described different types of queries supported by Shape2Vec. To our knowledge, this is the first method adaptable to such a wide range of cross-modal retrieval tasks. Note that although

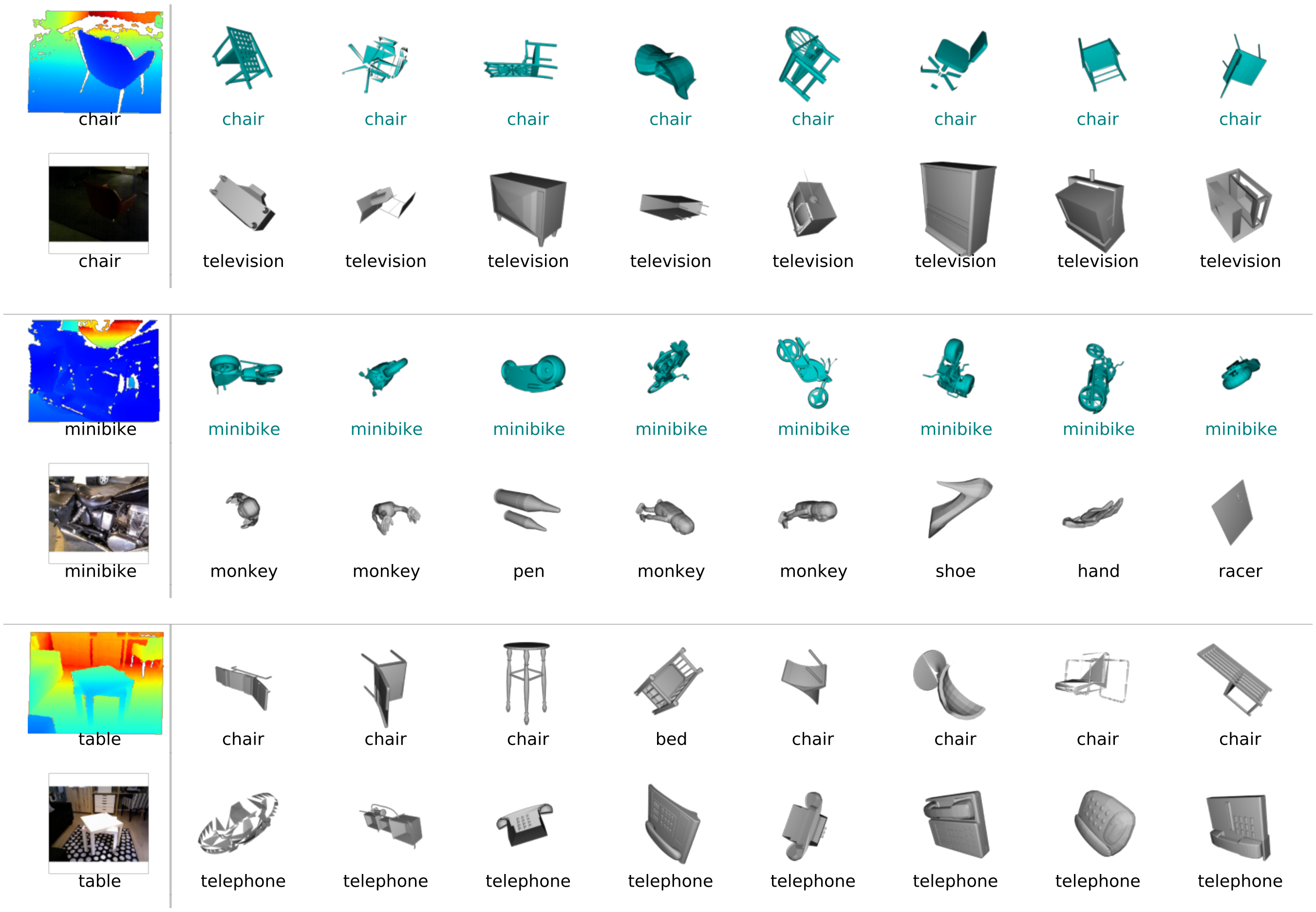


Figure 7: *Results of shape retrieval based on RGB-D images. Each row shows the top eight results for a query, with a depthmap (top) or an image (bottom). This illustrates how image-based retrieval can underperforms compared to depthmap queries.*

we use the same architecture for each modality, it is not necessary. Distinctive CNN could be trained to generate shape embeddings, as long as the word vector space remains fixed and the loss function is selected to reduce the distance between the input descriptor and its label embedding. Our comparison against previous mesh-based retrieval has been limited to those methods who have reported results on SHREC14-3D. It did not include state-of-art methods that use deep learning on larger datasets. For completeness, the next section compares Shape2Vec against other CNN-based shape descriptors.

8 Comparison against other deep shape descriptors

Savva et al. [2016] present results of the SHREC'16 Large-Scale 3D Shape Retrieval using ShapeNetCore. This dataset was collected by Chang et al. [2015], for the specific purpose of deep learning. It is five times larger than SHREC14-3D, and contains about $51,300$ shapes from 55 classes, each subdivided into subclasses. The competing methods in the SHREC'16 challenge are based on deep neural networks and the top performing method is Multi-view CNN (MVCNN) [Su et al. 2015], which was presented in Section 2. MVCNN trains one CNN to generate descriptors of 2D rendered views and use a second CNN to aggregate view descriptors into shape descriptors. We are interested in how Shape2Vec compares to MVCNN, since the latter is the most related work in the 3D domain. The authors publicly released the rendered images used to generate their reported results. To provide a fair comparison against their method, we use their dataset of 12 rendered views per shape. The viewpoints used by MVCNN for rendering are based on the assumption that the shapes in the dataset are consistently aligned, which is the case for ShapenetCore. We use the same split of training, validation, and testing sets used in the challenge.

To generate shape descriptors, we follow the approach described in Sections 4–6, and focus on L2-W2V which has shown better performance than alternative encoders. More specifically, we generate view descriptors in two steps: a Softmax classifier is trained to learn view subclasses and, then, it is modified to learn view embeddings in the Word2Vec vector space. View descriptors are averaged to form a shape descriptor. We report retrieval results when only shapes in ShapenetCore-Test are used for querying and retrieval, as done by other methods in the SHREC'16 challenge.

Table 8 shows performance metrics generated with evaluation code provided by the contest organisers. The table shows additional retrieval metrics than the ones we have used so far: precision (P), recall (R) and the F-score (F) at N, where N is the number of retrieved objects. We report unweighted averages (microALL) and weighted averages (macroALL) to adjust for differences in class sizes, as done by Savva et al. [2016]. The DCG metric is the only performance metric that takes subclasses into consideration, by assigning higher relevance to results that match both the main class and subclass of the query.

	P@N	R@N	F@N	DCG	AP
microALL					
Ours (L2-W2V)	**0.778**	0.698	0.709	**0.915**	0.871
Su (MVCNN)	0.770	**0.770**	**0.764**	0.899	**0.873**
Bai (GIFT)	0.706	0.695	0.689	0.896	0.825
macroALL					
Ours (L2-W2V)	0.565	0.615	0.545	**0.878**	0.792
Su (MVCNN)	**0.571**	**0.625**	**0.575**	0.865	**0.817**
Bai (GIFT)	0.444	0.531	0.454	0.850	0.740

Table 8: *Comparison against other CNN-based shape retrieval methods, on ShapenetCore-Test. Micro-averaged results (top) present performance metrics averaged over classes, and macro-averaged results (bottom) show unweighted average over the dataset. The normalised DCG metric uses a graded relevance that assigns more weight to retrieved results that match both the main class and the subclass of the query.*

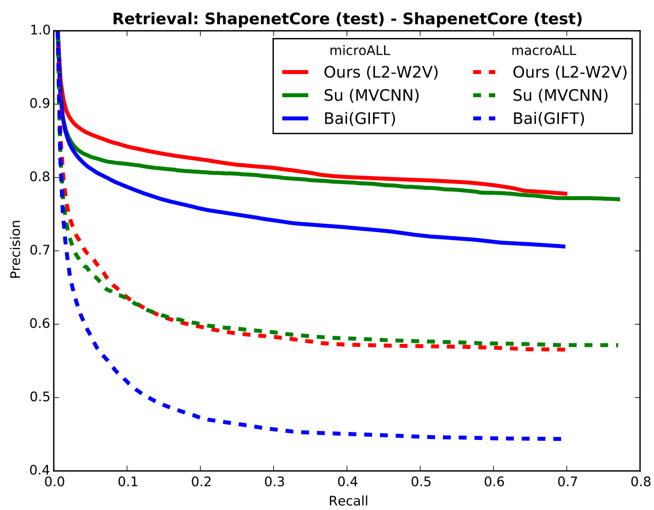


Figure 8: *Precision-recall curves of selected CNN-based methods on ShapenetCore-Test. This indicates that our method, Shape2Vec has comparable results to MVCNN when relevance is not graded.*

Results show that Shape2Vec has comparable performance to MVCNN [Su et al. 2015], when subclasses are not taken into account. On microALL, MVCNN has an AP of 87.3%, comparable with Shape2Vec 87.2% AP. This is best illustrated by the PR curves in Figure 8. However Shape2Vec generate results with higher relevance, as indicated by the improvement in DCG performance (89.9% to **91.5**% on microALL and 86.5% to **87.8**% on macroALL).

Shape2Vec ability to generate results with higher relevance is due to the fact that it leverages semantic information and thus, retrieves results that are semantically close to the query.

GIFT [Bai et al. 2016] was described in Section 2 as the state-of-the-art in real-time shape retrieval. GIFT generates multi-view descriptors, similarly to MVCNN and Shape2Vec, but uses an index structure for multi-view matching to achieve fast retrieval. Results show that both Shape2Vec and MVCNN outperform GIFT. This suggests that the CNN-based aggregator in MVCNN and Shape2Vec semantic-based descriptors are useful for better similarity assessment.

9 Discussion

This section discusses observations made for different stages of training and evaluation, as well as possibilities for future work.

Language model choice Word2Vec outperforms the Wordnet-based WN vector space for each cross-modal retrieval task. Note that WN only uses 100 dimensions compared to the 500 used by Word2Vec. A larger vector space may capture more information and explain the performance difference. Furthermore, Word2Vec captures both syntactic and semantic relationships, while WN is only based on semantic similarity. Furthermore, we only explored one manifold learning technique, NMDS, for learning a vector space based on semantic relatedness. Other techniques could be investigated, that learn embeddings from semantic similarities.

Loss function We investigated training of shape embeddings using three different loss functions. L2 loss consistently performed the best, closely followed by Cosine distance loss and finally hinge loss. Hinge loss with WN had significantly poorer performance compared to the rest. This may be related to the choice of the margin parameter. It will be interesting to see how this parameter affects retrieval based on the language model used.

Fusing depthmap descriptors Shape descriptors are obtained by averaging depthmap descriptors. MVCNN indicates that better performance can be achieved by training a CNN to aggregate view descriptors. We expect such a learning approach to improve shape description. Other architectures beyond AlexNet could be explored. Different network models may be more appropriate for some modalities. In particular, Geodesic CNN [Masci et al. 2015] could be used to generate pose-invariant shape embeddings.

Multi-modal retrieval Learning approaches could be used to extract the most useful features from multiple modalities.

Shape2Vec algebraic operations. One of the main advantages of Word2Vec is its ability to perform basic algebraic operations such as additions and subtractions in the vector space, that correspond to semantically meaningful results. An interesting avenue for future work would explore whether shape embeddings based on this language model share these properties and if not, how the CNN architecture could be modified to support such algebraic operations.

10 Conclusion

We have explored learning of semantic-based shape descriptors from training data. More specifically, we propose a supervised method for generating shape descriptors that are embedded in a word vector space, making it possible to perform shape-based and text-based queries. We showed that the same technique could be used for sketches, images and RGB-D images, making it possible to compare all these modalities with one another. Using these semantic-based embeddings, we reported results on a sketch-based shape retrieval benchmark. Shape2Vec outperformed state-of-the-art by an AP difference of **49**%. This suggests that the proposed method is particularly suited for cross-modal shape retrieval. The substantial improvement on previous work is due to the leverage of semantic information in language models. Thus, similarity assessment is based on both semantic and visual features. We showed that the proposed method could also be used to perform shape retrieval using RGB-D images taken by normal users in uncontrolled settings. Our research raises several questions that will be interesting to investigate in future work.

References

ABADI, M., ET AL., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

BAI, S., BAI, X., ZHOU, Z., ZHANG, Z., AND LATECKI, L. J. 2016. Gift: A real-time and scalable 3d shape search engine. In *CVPR 2016*. To appear.

BIASOTTI, S., CERRI, A., BRONSTEIN, A., AND BRONSTEIN, M. 2015. Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum*.

BOSCAINI, D., MASCI, J., RODOLÀ, E., BRONSTEIN, M. M., AND CREMERS, D. 2016. Anisotropic diffusion descriptors. In *Eurographics 2016*.

CHANG, A. X., FUNKHOUSER, T., GUIBAS, L., HANRAHAN, P., HUANG, Q., LI, Z., SAVARESE, S., SAVVA, M., SONG, S., SU, H., XIAO, J., YI, L., AND YU, F. 2015. ShapeNet: An information-rich 3D model repository. In *arXiv*.

CHOI, S., ZHOU, Q.-Y., MILLER, S., AND KOLTUN, V. 2016. A large dataset of object scans. *arXiv:1602.02481*.

DUCHI, J., HAZAN, E., AND SINGER, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res. 12* (July), 2121–2159.

EITZ, M., HAYS, J., AND ALEXA, M. 2012. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4, 44:1–44:10.

FELLBAUM, C. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

FROME, A., CORRADO, G. S., SHLENS, J., BENGIO, S., DEAN, J., RANZATO, M., AND MIKOLOV, T. 2013. DeViSE: A deep visual-semantic embedding model. In *NIPS'13*, 2121–2129.

GONG, B., LIU, J., WANG, X., AND TANG, X. 2013. Learning semantic signatures for 3d object retrieval. *Trans. Multi. 15*, 2 (Feb.), 369–377.

JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., AND DARRELL, T. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

KARPATHY, A., 2015. "CS231n: Convolutional Neural Networks for Visual Recognition". http://cs231n.github.io/.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 1097–1105.

KRUSKAL, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1, 1–27.

LI, B., LU, Y., LI, C., GODIL, A., SCHRECK, T., AONO, M., CHEN, Q., CHOWDHURY, N., FANG, B., FURUYA, T., JOHAN, H., KOSAKA, R., KOYANAGI, H., OHBUCHI, R., AND TATSUMA, A. 2014. SHREC'14 track: Comprehensive 3d shape retrieval. In *Proc. EG Workshop on 3D Object Retrieval*.

LI, B., LU, Y., LI, C., GODIL, A., SCHRECK, T., AONO, M., FU, H., FURUYA, T., JOHAN, H., LIU, J., OHBUCHI, R., TATSUMA, A., AND ZOU, C. 2014. SHREC'14 track: Extended large scale sketch-based 3d shape retrieval. In *Proc. EG Workshop on 3D Object Retrieval, 2014*.

LI, B., LU, Y., LI, C., GODIL, A., SCHRECK, T., AONO, M., BURTSCHER, M., CHEN, Q., CHOWDHURY, N. K., FANG, B., FU, H., FURUYA, T., LI, H., LIU, J., JOHAN, H., KOSAKA, R., KOYANAGI, H., OHBUCHI, R., TATSUMA, A., WAN, Y., ZHANG, C., AND ZOU, C. 2015. A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *Computer Vision and Image Understanding*.

LI, Y., SU, H., QI, C. R., FISH, N., COHEN-OR, D., AND GUIBAS, L. J. 2015. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph. 34*, 6 (Oct.), 234:1–234:12.

MASCI, J., BOSCAINI, D., BRONSTEIN, M. M., AND VANDERGHEYNST, P. 2015. Geodesic convolutional neural networks on riemannian manifolds. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.

MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop*.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C., AND FEI-FEI, L. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision 115*, 3, 211–252.

SAVVA, M., YU, F., SU, H., AONO, M., CHEN, B., COHEN-OR, D., DENG, W., SU, H., BAI, S., BAI, X., FISH, N., HAN, J., KALOGERAKIS, E., LEARNED-MILLER, E. G., LI, Y., LIAO, M., MAJI, S., WANG, Y., ZHANG, N., AND ZHOU, Z. 2016. Large-Scale 3D Shape Retrieval from ShapeNet Core55. In *Proc. EG Workshop on 3D Object Retrieval*.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 1929–1958.

SU, H., MAJI, S., KALOGERAKIS, E., AND LEARNED-MILLER, E. G. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*.

VAN DER MAATEN, L. 2009. Learning a parametric embedding by preserving local structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR W&CP*, vol. 5, 384–391.

WANG, A., LU, J., CAI, J., CHAM, T. J., AND WANG, G. 2015. Large-margin multi-modal deep learning for rgb-d object recognition. *IEEE Transactions on Multimedia 17*, 11, 1887–1898.

WANG, F., KANG, L., AND LI, Y. 2015. Sketch-based 3D shape retrieval using convolutional neural networks. In *CVPR 2015*.

WU, Z., AND PALMER, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, 133–138.

WU, Z., SONG, S., KHOSLA, A., YU, F., ZHANG, L., TANG, X., AND XIAO, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR 2015*, 1912–1920.

YU, Q., YANG, Y., SONG, Y., XIANG, T., AND HOSPEDALES, T. 2015. Sketch-a-net that beats humans. In *BMVC15*, 7.