Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS

TIEN DAT NGUYEN

Toward concept visualization through image generation

Institute of Formal and Applied Linguistics

Supervisors of the master thesis:	Asst Prof. Pavel Pecina Asst Prof. Raffaella Bernardi Assoc Prof. Marco Baroni
Study programme:	Master of Computer Science
Specialization:	Mathematical Linguistics

Prague 2015

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In .... date .... signature of the author

to me & my family ♡ my love & my friends.
I love you all...

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Asst Prof. Pavel Pecina, Asst Prof. Raffaella Bernardi and Assoc Prof. Marco Baroni for their support and guidance during the completion of this thesis. The work in this thesis could not have been accomplished without their encouragement and kind assistance.

My special thanks go to Angeliki Lazaridou, a researcher in the Language, Interaction and Computation Laboratory (CLIC-CIMEC), who discussed and gave me a lot of invaluable advice.

I would like to thank the committee of The European Masters Program in Language and Communication Technologies (LCT) for giving me such a fantastic opportunity and financial support to study in Europe. Many thanks to Stanislav Veselý, he is very nice friend who helps me a lot during my stay in Prague.

Last but not least, I owe my loving thanks to my family for their constant love and support. I always find my inspiration from all of you. A big hug to all of my friends in Trento and Prague for great times together.

Thanks my love! You always encourage me, advice me, love me and spoil me.

Somewhere in Europe, 04/12/2015
Dat

Title: Toward concept visualization through image generation

Author: Tien Dat Nguyen

Department: Institute of Formal and Applied Linguistics

Supervisors: Pavel Pecina (Charles University in Prague), Angeliki Lazaridou, Raffaella Bernardi, Marco Baroni (University of Trento),

Abstract: Computational linguistic and computer vision have a common way to embed the semantics of linguistic/visual units through vector representation. In addition, high-quality semantic representations can be effectively constructed thanks to recent advances in neural network methods. Nevertheless, the understanding of these representations remains limited, so they need to be assessed in an intuitive way. Cross-modal mapping is mapping between vector semantic embedding of words and the visual representations of the corresponding objects from images. Inverting image representation involves learning an image inversion of visual vectors (SIFT, HOG and CNN features) to reconstruct the original one. The goal of this project is to build a complete pipeline, in which word representations are transformed into image vectors using cross modal mapping and these vectors are projected to pixel space using inversion. This suggests that there might be a groundbreaking way to inspect and evaluate the semantics encoded in word representations by generating pictures that represent it.

Keywords: text2image, Cross-model Mapping, Distributed Semantics, Convolutional Neural Networks, Visual Feature Inversion.

Contents

Chapter 1

Introduction

"The development of full artificial intelligence could spell the end of the human race."
by Stephen Hawking.

1.1 Motivation

There has been an inconclusive debate about whether future artificial intelligence could destroy mankind. Could AI destroy the world? Stephen Hawking told the BBC about the effect of AI development to human and Elon Musk, CEO of SpaceX and Tesla Motors, who has called AI as the "biggest existential threat" of humanity. AI has a great potential for the future, it is important to research how to derive its benefits while voiding unexpected dangers. But many AI researchers say that humanity is nowhere near being able to create strong AI. Demis Hassabis, an AI scientist at Google DeepMind, said at a news conference about AI: "We are still decades away from any technology we need to worry about". He also added that: "it is good to start the conversation now".

People always dream to invent a machine that would do any complex behaviour like humans do. The machine could have an ability to think, act and solve solutions for every problem in real life. Computers today can present any computable problem and algorithm. This was already proved a long time ago since Alan Turing showed that a Universal Turing Machine can solve any computable problem.

With the rapid growth of World Wide Web, now we have enormous various data sources coming from images, text, video and speech, etc... Thus, the main problem is how to teach the machine understand and perceive knowledge from these data sources so that it would think and produce any complex human behaviour. The problem is still very difficult to be solved immediately, although there are currently breakthroughs in machine learning, machine perception and robotics techniques such as: deep learning and decision-theoretic planning. So at the moment, it is too early to think about the effect of AI developing to human race. There has been still a long-term goal to build a complete human-machine system occurring in our dream.

To achieve the success towards an intelligent machine, we have to think of several ways. The general problem of creating AI has been broken down into a number of sub-problems such as: reasoning, knowledge acquisition and representation, planning, human-computer interaction (including: natural language processing and computer vision), perceptions and robotics (the ability to move and manipulate objects). Among these sub-problems, machine learning, a study of computer algorithms, has been central to AI research which can improve performance automatically through experience. In other words, machine learning explores the study and development of algorithms that can learn from data and make predictions through its learning process.

The question: "Can machines think?", which is mentioned in Alan Turing's proposal in his paper "Computing Machinery and Intelligence", can be replaced by this question "Can machines do what humans can do?". Firstly, people did start to build a machine that can have human vision for perceiving and understanding our visual world. A machine can acquire, process and analyze basic shapes and concrete images like digital written and concrete basic objects, then improve it towards more complex visual contents such as human faces, running vehicles and so on, finally creating a machine that can see (recognize and deep understand) the real world.

Another possible way is building a machine which can communicate with us using our natural language. Research in Natural language processing allows the dreamed machine the ability to read, listen and understand the languages that we speak in our everyday activities. Some straightforward applications of NLP include information retrieval, dialogue system, question answering and machine translation. We spend very expensive cost (e.g., time, money, human resource) to build various rule-based language computational systems and develop a number of language models to teach computers to "speak" and understand our natural languages. The dreamed machine needs to have a lexicon of our language, a parser and grammar rules to segment sentences into a machine-understandable representation. The construction of lexical resources requires significant effort. Also, the machine needs to have a semantic theory to deal with reading comprehension and so on.

So now, we already have computational models for vision and language to create our dreamed machine. That dreamed machine has an ability to produce desired behaviors that humans consider intelligent from both modalities (language and vision). As a result, the artificial intelligence community has extended the research area in the interaction between language and vision. For example, if a person is presented a picture and is asked to describe what he/she is seeing or ask someone to imagine a scene, he/she is performing a task which links the two modalities. We dream a machine that is able to think and imagine like humans. Giving that machine an image, it can tell us the caption or predict the events and the contexts depicted in the image. There has been significant growth in the research direction of linking language and vision. Language can contribute to expand the tasks of vision community such as: images to captions, describing events in images, video to text. Conversely, how is vision able to contribute to the linguistic side of the long-term goal of artificial intelligence?

Progress have already been obtained on the task of building a machine that can form a coherent and global understanding of a scene in a picture or a video. What is next? We will teach the machine to imagine a new scene never seen before to step toward computer creativity and imagination.Imagination is constrained by three different things. The first thing is the environment that you are in when you are called upon to use your imagination and this might be the book that you are reading or when we tell you to imagine a dog or something. The second thing is what we understand about the world. Then finally are visual memory and that is everything that we have ever seen during our life. To teach a machine to imagine, we need to provide knowledge of the word and vision to that machine. Then, the machine has an ability to think and draw a new scene based on that knowledge. For example, the machine has an ability to draw a picture of a *"wampimuk"* by knowing this event "we found a cute, hairy wampimuk sleeping behind the tree? that a "wampimuk" will probably look like a small furry animal, even though a *"wampimuk"* has never been seen before. In this thesis, we contribute to the work on machine's imagination, by starting to tack the question of how a machine can draw knowledge that it learns from linguistic and visual environment. More precisely, our idea is to create a machine that can produce a picture of concepts it has rarely encountered or never seen before by generalizing from the semantic information encoded in word embedding. We present a language-driven image generation system which can visualize semantics encoded in word representations induced from text corpora.

1.2 Main Contributions

To achieve our goal, we combine various results in language and vision. Thanks to the latest research in distributed representation, tools such as word2vec [1, 2] and Glove [3] have produced high-quality word embeddings from very large text corpora. Most of the models are based on modern Recurrent Neural Networks

(RNNs). They outperform traditional distributional approaches in many linguistic tasks [4]. Meanwhile, in computer vision, semantic representation of concepts are also extracted from images that they are associated (tagged) with. By using very deep neural networks (e.g., Convolutional Neural Networks [5]), trained on image datasets, Image embeddings can be represented by outputs (visual features) of each neural networks' layer.

Essentially, there are two challenges in this project: 1) translating word vectors to visual feature representations, 2) generating an image based on a translated visual vectors. The first challenge is a problem of cross-model mapping between language and vision domains. At the same time, in computer vision community, there is currently much attention to better evaluate what the layers of CNNs and other deep architectures have really learned. Furthermore, the understand of visual vectors extracted by CNNs remains limited, they need to be assessed in an intuitive way. Therefore, some feature inversion algorithms [6, 7] have been proposed to reconstruct the natural image from visual features (e.g., SIFT, HOG or CNN features).

Our language-driven image generation system[1] takes a word vectors as an input (e.g., word embedding of "car"), translate it into visual feature space defined by the outputs of CNN layer, then generate a natural image from transformed-feature representation by applying the feature inversion technique in the HOGgle framework [6]. We test our system generating images in a zero-shot way, in which knowledge of testing concepts (semantic representations from text and image) is never used during training cross-modal mapping function as well as inducing the feature inversion function. In particular, our system has an ability to project a word vector of "car" onto visual feature space and then reconstruct the image of a "car" without having ever seen any photo of "car".

We next design a series of CrowdFlower studies providing quantitative and qualitative insights into visualizing semantic information encoded in word embedding by allowing subjects inspect generated images based on their visual properties (e.g., shape, color and characteristic environment). The experimental results show that our current system can capture visual properties related to color and the environment in the task of object discrimination. However, it is not good at expressing shape or size of objects because shapes are not often expressed by linguistic means.

[1] Our work is accepted for a presentation at *Multimodal Machine Learning Workshop @NIPS 2015* under the publication: "Unveiling the Dreams of Word Embeddings: Towards Language-Driven Image Generation"

1.3 Structure of the thesis

This thesis is organized as follows:

Chapter 2 is a literature review of extracting semantic representations from text and images and related works. It starts by briefly introducing recent advanced research in word and image embedding. We also describe the task of cross-modal mapping in zero-shot manner. The last section is about recent research in visual feature inversion.

Chapter 3 presents our language-driven image generation system. It describes in detail our pipeline of image generation from word embeddings. The chapter first sketches out the system and then specifies materials which used to do training and evaluation.

Chapter 4 provides a through model selection and experimental evaluation. We carry our pre-experiments to determine the best model and parameters for our system in the first section. Subsequently, evaluation section covers four different experiments which estimate visual properties of the generated images. In each experiment, the task description is first described; hence the experimental results and discussion.

Chapter 5 is a summary of our achievements throughout the previous chapters. Some future research directions are also proposed to continue our recent work of image generation.

Chapter 2

Literature Review

In this thesis, we combine a variety of recent strands of research to perform language-driven image generation. Firstly, we are focusing on the problem of learning semantic representation from both modalities: language and vision. The easiest way to teach computers to understand documents and images is developing a mathematical model of meaning representation. Computers discriminate linguistic/visual units based on their semantic representations. We name semantic representations inducing from text and images as Word Embedding and Image Embedding, respectively. Secondly, we provide a review of cross-modal mapping problem which has received much attention in the machine learning community today. Finally, we introduce the literature on image generation by focusing on works that look at the task as the one of visual feature inversion.

2.1 Semantic Space representation

Learning semantic information has been a hot topic in Natural Language Processing. Semantic information has been represented in a number of ways such as: feature-based representation, semantic networks and semantic space. Feature-based models capture specific aspects of semantics, in which the semantic of a concept can be learned through a list of human-defined features or attributes that are related to the meaning of that concept [8, 9]. On the other hand, semantic networks provide semantic relations between concepts. The most common approach of semantic networks is using directed or undirected graphs to present concepts as vertices and semantic relations as edges. WordNet [10] is an example for such a semantic network, where concepts are linked by semantic relations such as: synonym or hypernym. The semantic similarity of two concepts is measured by the path length between two vertices that denote two concepts.

Another approach for representing the meanings of words is semantic space, in which words are represented by vectors. Each dimension represents some latent categories (e.g. semantic or syntactic features). One of the key benefits of such a representation is that it does not require human-defined features which are used in feature-based approach. On the other hand, distance measures can be applied to estimate semantic similarity between words given their vector representations. See Figure 2.1 for an example.

Similarity metrics: In many NLP tasks, it is necessary to estimate the semantic similarity between concepts. Depending on the model setting and vector normalization, there are two most common measures: Cosine distances and Euclidean distances. The Cosine distance takes into account the angle between two vectors, while the Euclidean distance measures the distance between two points.

$$cosine(A, B) = cos(\theta) = \frac{A.B}{||A||.||B||} \tag{2.1}$$

$$Eucl(A, B) = |A - B| = \sqrt{\sum_{i=1}^{d} A_i - B_i} \tag{2.2}$$

In this section, we introduce Distributional Semantic and Distributed Semantic representation. Both representations are based on the idea that the meaning of a word can be captured form its linguistic environment. The idea of distributional semantic (Section 2.1.1) approaches can be summed up in the distributional hypothesis, while distributed semantic approaches (section 2.1.2) use neural networks to induce the meaning of words from text corpora.

2.1.1 Distributional Semantic

Distributional semantic models (DSMs) – also known as "word space models" are based on the distributional hypothesis. This hypothesis has been stated in different ways: "Linguistic items with similar distributions have similar meanings" [11]; "words which are similar meaning occurs in similar contexts" [12]; "a representation that captures much of how words are used in natural context will capture much of what we mean by meaning" [13]; and "words that occur in the same contexts tend to have similar meanings" [14]. The basic idea is collecting distributional information in high-dimensional representation. In other words, targeted words (concepts) are represented as vectors of distributional characteristics, especially co-occurrences with other words in the same context from large-scale linguistic data sources [15, 16, 17, 18]. The surrounding context contributes meaning to the targeted words. It can be slide-window surrounding words, a paragraph or a sentence, or even a document that targeted words appear. Consequently, the meanings of concepts can be discriminated by their usages or surrounding contexts.

DSMs typically represent meanings of concepts as context vectors in a high-dimensional space; and it is called as "vector space" or "semantic space". Concepts that are semantically related tend to be closed in the semantic space. For example, the concept *"sun"* may be observed in the same context as the concept *"moon"*. As a result, their vectors are expected to have large cosine distance.

On the other hand, *"sun"* and *"computer"* rarely co-cocurs in a similar contexts; therefore their meaning are not much related.

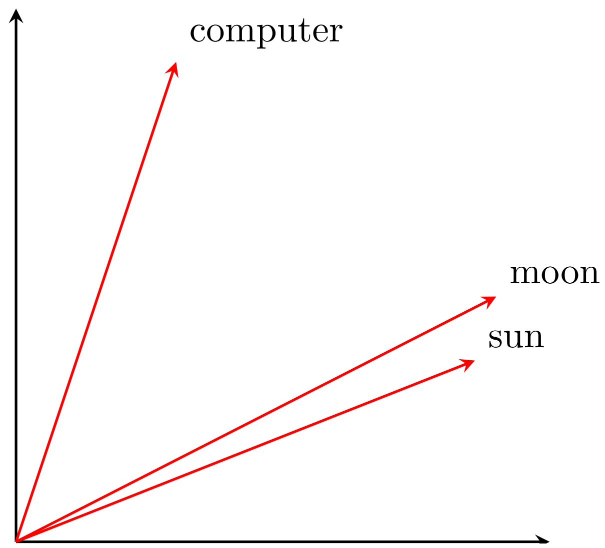


Figure 2.1: Semantic Space representation

Based on semantic space models, semantic similarity between words can be measured in term of vector similarity (e.g, Cosine distance...). These distributional representations have proven very effective in some NLP applications such as word/document clustering [19], lexical ambiguity resolution [20], and cognitive modelling [21]. We list successfully DSM techniques below:

- Latent semantic analysis
- Latent Dirichlet allocation
- Self-orgarnizing map
- Hyperspace Analogue to Language (HAL)
- Independent component analysis
- Random indexing

Extracting distributional representations can lead to very high-dimensional vectors since most of methods are based on word counts. These methods, namely Principal Component Analysis and Factor Analysis, are very useful both to deal with sparsity problem via smoothing as well as to improve the efficiency of subsequent models making use of such representations.

2.1.2 Distributed semantics

The last few years, there has seen an increase of research in distributed semantic representation. In this review, we focus on distributed representations of words induced by neural networks since it was shown that they have performed significantly better than traditional distributional semantic models.

The first neural network language model (NNLM) was introduced by Y. Bengio and his co-authors [22]. Their n-gram feed-forward NNLM has four layers namely input, projection, hidden and output layers. At the input layer, each of the previous n-1 words is encoded using 1-of-V orthogonal representation, where V is the size of the vocabulary. Therefore, every word is associated with a vector with length V, in which only one value corresponding to the index of a work in the vocabulary is 1 and all other values are 0. The input layer is then projected to a projection layer P called also a shared projection matrix. All words in a context share the same projection matrix, more precisely the matrix is the same when projection word w_{t-1}, w_{t-2}, etc. The third layer is a hidden layer with non-linear activation function, where some common functions are applied such as: a *tanh* or a *logistic* sigmoid function. The last layer is an output layer whose size is equivalent to the size of vocabulary V. The output of this layer represents the probability distribution $P(w_t|w_{t-1}, ..., w_{t-n+1})$.

Being inspired by the feed-forward NNLM, Mikolov at el. proposed a different neural network architecture to learn the representation of word sequence, which is called Recurrent Neural Networks language model (RNNLM) [23, 24]. The main difference between two models lies in the representation of the context. While the feed-forward NNLM takes just the previous $n-1$ words as a history, the RNNNM learn the representation of context from data during training. The hidden layer of RNN represents all previous words, thus it can represent long contexts.

Mikolov and his co-authors also prove that the most of the complexity of NNLM and RNNLM is caused by training the non-linear hidden layer. They proposed new neural net architectures that might not only be able to learn word embeddings as precisely as NNLM and RNNLM, but can be trained efficiently on much more data by minimizing the computational complexity. An example of a RNN language model is shown in figure 2.2.

The first architecture is similar to that of the feed-forward NNLM, where the non-linear hidden layer is removed. The projection layer is shared for all words so that all words project into the same position. The word order in the context does not influence the projection, so they call the model Continuous Bag-of-Words model **CBOW**. Unlike standard bag-of-words model, the new model uses continuous distributed representation of the context. In addition, the **CBOW** takes

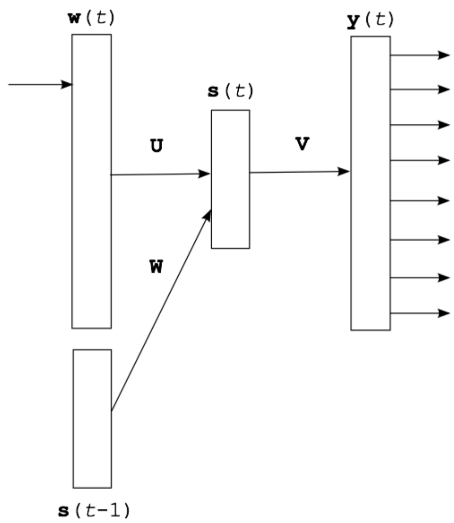


Figure 2.2: A simpler Recurrent Neural Network architecture.
[Source: Mikolov et al. 2012 [24]]

into account words from the future. That means it predicts the current word in the middle of a symmetric window based on the context (sum of vector representations of words in the window). Context window is considered from 2 to 10 words either side of a targeted word. They obtained the best performance on the task of semantic-syntactic Word Relationship[1] by building a log-linear classifier with a context including four future and four history words, where the training criterion is to correctly classify the middle word.

On the other hand, their second architecture **Skip-gram** is useful for predicting surrounding words in a sentence instead of using context to predict the current word. More precisely, they use each current word as an input in a log-linear classifier with continuous project layer and predict other words in a symmetric window context (words before and after the current word).

The newest language model for the unsupervised learning of semantic representation is proposed in *GloVe* framework [3]. They use a specific weighted least square model that trains on a global co-occurrence count matrix. Particularly, their model efficiently leverages statistical information by training on non-zero elements in a co-occurrence matrix, rather than a sparse matrix or small context windows in traditional distributional methods and Skip-gram model proposed in [2]. As a results, both *Word2vec* and *GloVe* models outperform any previous methods on word analogy, word similarity and named entity recognition task.

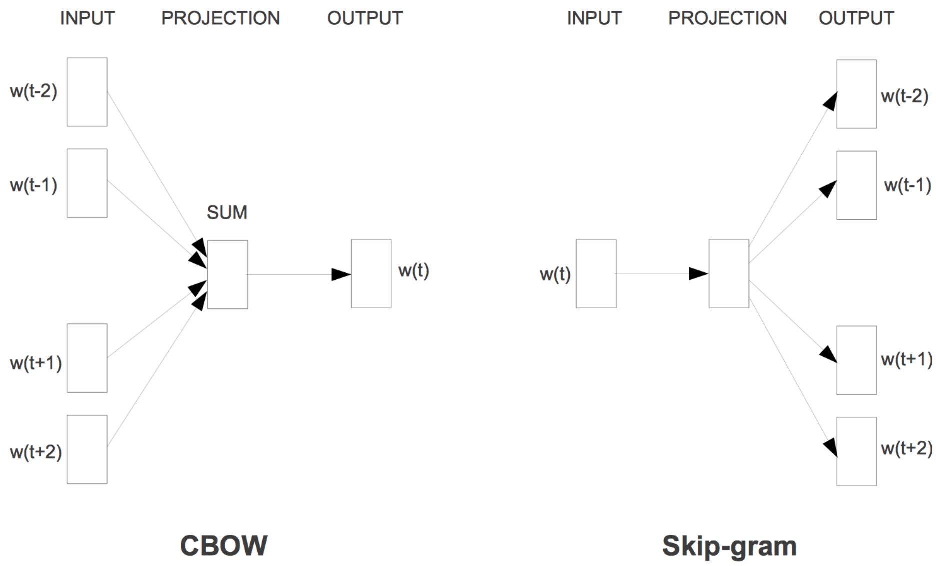


Figure 2.3: Mikolov et al. neural network architectures. The CBOW predicts the current word based on for surrounding words, and the Skip-gram predicts surrounding words given the current word.

[Source: Mikolov et al. 2013 [2]]

2.1.3 Applications

Semantic space representation can easily be plugged into many NLP applications. Distributed representation has improved performance in a wide range of tasks by providing richer semantic information than those of distributional semantics. We list the usefulness of semantic representation across a variety of NLP applications below.

The easy task is synonym or semantic similarity among wordsconcepts [25]. Topic modelling task has been explored by using distributional semantic [26, 27]. There are other applications such as: named entity recognition, word-sense discrimination, document classification, discourse analysis, etc. Word vectors can be used in entity recognition [28], question answering [29], sentiment analysis [30] and parsing [31].

When considering semantic representation in the task of language driven image generation, we only focus on word-level representation only. Having richer semantic representations extracted from text is the prerequisite to succeed in our project. Distributional techniques are memory intensive and not as efficient (not a compact representation) as distributed representation. They are mainly sparse and typical high-dimensional features (based on word counts), thus resulting in interpretable representations. Distributed representations are, on other hand, compact, dense and low-dimensional representations, and thus more difficult to interpret. Our method is primarily developed on dense vectors. Tools such as *word2vec*[1] and *Glove*[2] have shown to capture fine-grained semantic and syntactic regularities to produce very high-quality word embeddings. Thus, we train our word embeddings with *word2vec* toolkit on a language corpus of 2.8 billion words, choosing the CBOW method [1] which produces state-of-the-art performance in many linguistic tasks [4].

2.2 Image Embedding

The previous subsection introduces various approaches of learning word representation from text. Many experimental studies show that human study lexical semantics not only from the linguistic environment (verbal information), but also from our interaction with the world such as: non-verbal experiments (e.g., vision) and representations [32, 33]. So, we can learn semantic representation (we call Image Embedding) from visual side or other data sources (e.g. images, videos, fMRI...). The representation is mostly applied to many tasks in the computer vision community. However, they can be used to open a new research direction of interaction between language and vision such as: image to text, video to text and vice versa. This section reviews the most successful method for learning semantics from images: Convolutional Neural Networks (CNNs).

2.2.1 Deep Convolutional Neural Networks

A convolutional neural network (CNN) is a type of fead-forward artificial neural network, where a individual neuron in a convolutional layer can connect to overlapping regions of the previous layer. CNNs were inspired by biological processes in visual cortex of human brain. The visual cortex has a complex arrangement of cell. These cells are sensitive to small sub-regions of the visual field, called a *receptive field*. The sub-regions are not overlapped and tiled to cover the entire visual field. These cells act as local filters over the input space and they are well-suited to exploit the strong spatially local correlation present in natural images. CNNs are widely used for image and video recognition in computer vision tasks.

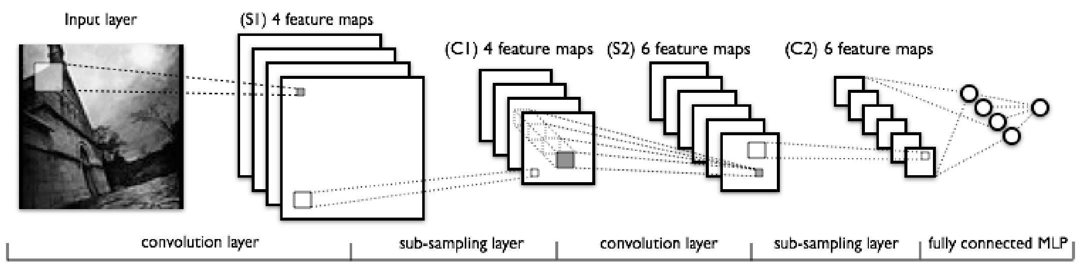


Figure 2.4: The LeNet architecture for hand writing recognition.
[Source: deeplearning.net tutorial]

[1]https://code.google.com/p/word2vec/
[2]http://nlp.stanford.edu/projects/glove/

An CNN architecture consists of various combinations of convolutional layers and fully connected layers:

- **Convolutional layer:** It contains a rectangular grid of neurons, where the output of its previous layer is also a rectangular grid of data. Each neuron is connected only to rectangular sections of the previous layer and the weights for this rectangular section are the same for each neuron. Therefore, a convolutional layer is about to do a convolutional operation of its previous layer, where the weights specify the convolution filter. Each convolutional layer has several grids, each grid takes inputs from all the grids of the previous layer using some different filters.

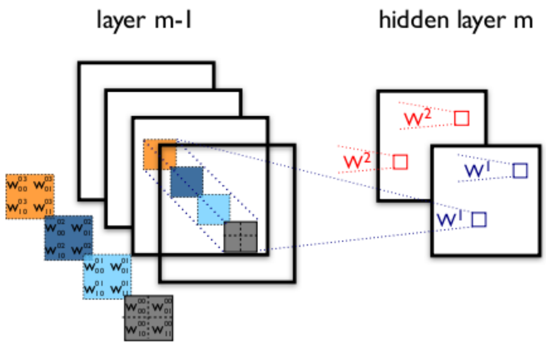


Figure 2.5: An example of a convolutional layer.
[Source: deeplearning.net tutorial]

- **Pooling Layer:** Another important term of CNNs is pooling. There may be a pooling layer after each convolutional layer. The pooling layer partitions the convolutional layer into a set of small non-overlapping rectangles and for each sub-region, it produces a single output. There are several ways to subsample such as: computing the average or the maximum or a learned linear combination of the neutrons in a sub-region. Most CNNs architectures include max-pooling layers, where they compute the maximum of the sub-regions they are pooling. Max-pooling layers eliminate non-maximal values, then reduce computation for upper layers (reducing the dimensionality of representations). These layers also reduce variance since outputs of convolutional layers have small translations. This is an important operation for object classification and detection.

- **Fully-Connected layers:** Upper layers of a CNN are high-level semantic layers. A fully connected layer connects every single neuron to all neurons in the previous layer (not doing convolutional operation).

- **Loss layer:** It is the last fully-connected layer of CNNs, where a softmax loss classification is applied to produce a distribution over K mutually exclusive classes. There are some types of loss functions: a sigmoid cross-entropy loss predicts K independent probability values in *[0,1]* or an Euclidean loss predict to real-valued labels *[-inf,inf]*.

CNNs are trained with the **Backpropagation** algorithm which is a common method for training artificial neural networks used along with an optimization technique such as gradient descent. The method compute the gradient of a loss function with all parameters in the neural network. Then, the gradient is used to update the weights to minimize the loss function. Thanks to recent advance in GPU computing, it has become possible to train larger neural networks. To increase efficiently training computation, these below techniques are also used in CNNs nets:

ReLU stands for Rectified Linear Units:

It is applied to the output of every convolutional and fully-connected layers of a CNN network. Typically, we compute a neuron's output f as a function of its input x with a saturating hyperbolic tangent function $f(x) = tanh(x)$, $f(x) = |tanh(x)|$ or with a sigmoid function $f(x) = (1 + e^{-x})^{-1}$. However, in terms of training time with gradient descent, these saturating nonlinearities are much slower than the non-saturating activation function $f(x) = max(0, x)$. This is already proved in the paper Krizhevsky et al. [5]. CNNs train several time faster with ReLU than their equivalents with $tanh$ functions [5].

Dropout:

A CNN net has a large number of parameters (typically it contains at least 2 or 3 fully-connected layers), it is prone to overfitting. Deep and big neural nets are also slow to train. It is too expensive for a neural network that already takes several days to train. Recently, there is a technique addressing this problem, namely Dropout. The key idea is randomly to set to 0 the output of some hidden neurons with probability p (normally, $p = 0.5$). The neurons which are dropped out do not contribute to the forward and back-propagation computation. It means that every time an input is presented, the neural network samples a different architecture, but all these architectures share weights. By avoiding training all neurons in neural nets, Dropout decreases overfitting in neural nets and improve training speed. Dropout is already shown to improves neural nets performance on supervised learning tasks in vision, speech recognition, document classification, producing state-of-the-art results on many benchmark datasets [34].

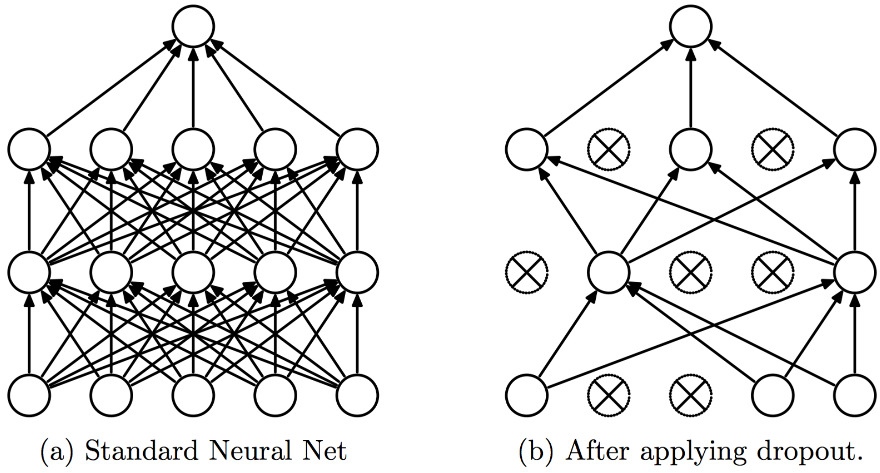


Figure 2.6: An example of Dropout method. Left: A standard neural net with 2 hidden layers. Right: Applying dropout method to the network on the left. Crossed neurons have been dropped.

[Source: Srivastava et al. 2012[34]]

2.2.2 The state of the art CNN model

Krizhevsky's architecture: In this project, we used pre-trained CNN model introduced in Krizhevsky et al. [5], which produces state-of-the-art performance in the task of object recognition. More precisely, performing on the test data of ImageNet LSVRC-2010 contest, they achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably state-of-the-art. They train a very deep convolutional neural network to classify 1.2 million images from ImageNet LSVRC-2012[3] dataset into 1000 different classes. The rest of this section is a brief review of the architecture of this neural network model.

An illustration of the architecture of Krizhevsky's CNN is depicted in Figure 2.7, it includes eight layers. The first five are convolutional and the remaining three are fully-connected. The output layer is fed to a 1000-way softmax classified into 1000 class labels.

The input data is $224 \times 224 \times 3$ images which 96 kernels of size $11 \times 11 \times 3$. The output (response-normalized and pooled) of the first layer is the input of the second convolutional layer which 256 kernels of size $5 \times 5 \times 48$. The next three convolutional layers are connected to one other without any intervening pooling or normalization. The third layer has 384 kernels of size $3 \times 3 \times 256$ connected to the (normalized, pooled) output of second layer. The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 92$, and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 192$. The fully-connected two layers have 4096 neurons each.

[3]http://www.image-net.org/challenges/LSVRC/2012/

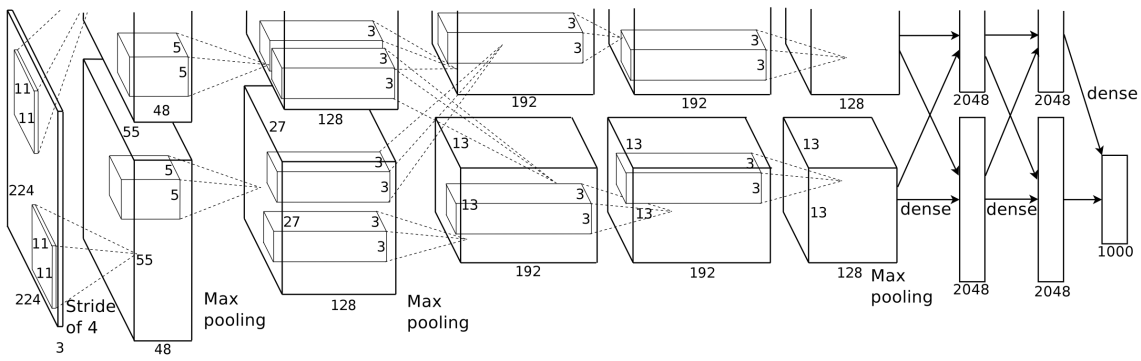


Figure 2.7: The Krizhevsky's Convolutional Neural Network architecture. *[Source: Krizhevsky et al. 2012 [5]]*

Each output of each layer in CNN can be considered as an image embedding or a visual feature. The last layer output is a vector of 1000 dimensions. CNNs trained on natural images learn a hierarchy of increasingly more abstract properties: the features in the bottom layers resemble Gabor filters, while features in the top layers capture more abstract properties of the dataset or tasks the CNN is trained for [35] (e.g., the topmost layer captures a distribution over training labels).

In our project, using pre-trained CNN model[4], we extract visual features from ImageNet dataset. We experiment on outputs of *pool5* (6x6x256=9216 dimensions)and *fc7* (1x4096 dimensions) layer as semantic representations of images. Configuration details are mentioned in Section 3.3. *Pool-5* is an intermediate pooling layer that should capture object commonalities and *fc-7* is a fully-connected layer just below the topmost one, and as such it is expected to capture high-level discriminative features of different object classes.

2.3 Cross-modal Mapping vs. Zero-Shot learning

Recently, the problem of zero-shot learning in cross-modal mapping has received much attention in machine learning community. The key of cross-modal mapping is the use of a set of semantic embedding vectors associated with class labels. These embedding vectors might be obtained from human-designed object attributes, text corpora, fMRI signal or outputs of layers in neural network. The goal of zero-shot learning is to learn a cross-modal classifier $f : X \rightarrow Y$ that predicts novel class of Y which is not included in the training set.

Zero-data learning was firstly introduced by Larochelle et al. [36], in which

[4] http://www.caffe.berkeleyvision.org/model_zoo.html

they has shown an ability to predict novel classes of digits that are not included in training set. Additionally, zero-shot learning is very common in vision community since object classes generally share common visual attributes. Most existing zero-shot models have a two-stage classification: given an image, firstly visual features are predicted; hence predicting its class label based on those features. For example, an image is represented by a binary indicator vector. The image is mapped to an unseen class which is the most similar to its visual vector prediction [37, 38, 39]. Another zero-shot strategy is using class relationship to classify unseen objects. For instance, an unseen object's class can be estimated by the nearest classifiers trained with ImageNet hierarchy [40, 41] or the other ones based on label co-occurrences [42].

In the neuroscience community, cross-modal mapping with zero-shot learning has been applied in a variety of applications. Mitchell et al.'s approach uses semantic features induced from text corpora to generate a neural activity pattern for any noun in English [43]. By contrast, Palatucci et al. [44] focus on word decoding, they desire to predict a word from a large set of possible words given it novel neural image. They also consider manually designed semantic features in addition to feature learned from text corpora. They develop a semantic classifier (Semantic Output Codes) for a neural decoding task. The experimental results has shown that it is possible to predict words that people are thinking about from fMRI signal of their neural activity, even without seeing those words in training set.

In recent years, there has been much attention to cross-modal mapping in language and vision domain. Socher et al. 's approach learns to map images to a lower dimensional semantic vector space using a neural network architecture. They link the image representation space to the word vector space by representing 8 classes for which they had labeled images. Firstly, a testing image can be distinguish whether it belongs to seen or unseen classes. Then, if the model determines a testing image to be in the set of 8 seen classes, a separately trained softmax model is used to perform the 8-way classification; otherwise the model predicts the nearest class in the word semantic space.

Meanwhile, the deep visual-semantic embedding model (DeViSE) [46] trains a mapping function from two pre-trained neural network models. Word embeddings are induced from a text corpus by implementing the Skip-gram method proposed in [2]. Image embeddings are low-level features of a deep neural network successfully trained on a supervised object recognition task [5]. As a result, their model can exploit semantic information to make predictions about ten of thousand of images labels not appeared in training data. ConSE [47] is inspired by the idea of DeViSE framework, but there is an important difference between the neural network architectures when constructing semantic vectors of concepts from image dataset. The ConSE model keeps the last layer of the convolutional

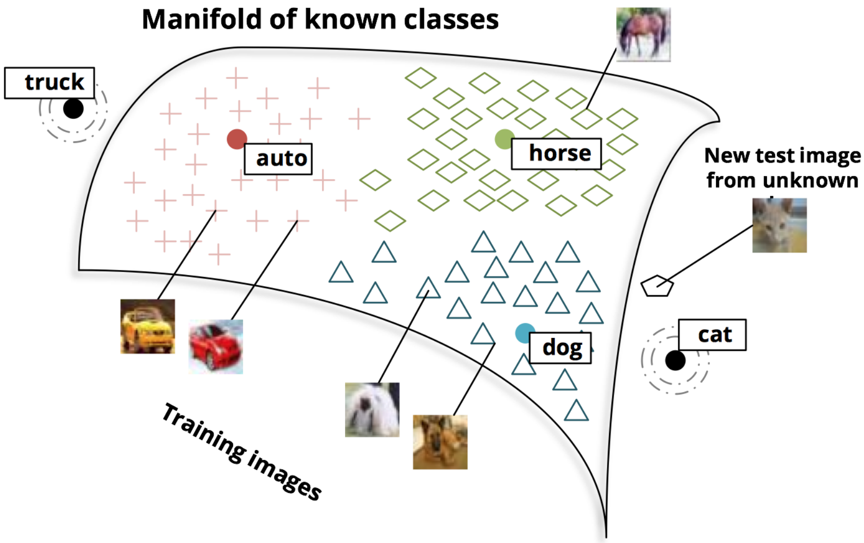


Figure 2.8: Overview of Socher et al. cross-modal zero-shot model. Firstly, mapping each new testing image into a lower dimensional semantic word vector space; then, determine whether it is on the manifold of seen images. If the image is not on the manifold, they classify it by unsupervised semantic word vectors. In this example, the unseen classes are "truck" and "cat".
[Source: Socher et al. 2013 [45]]

net, the softmax layer, by contrast the DeViSE model replaces it with a linear transformation layer.

In our language-driven image generation pipeline, the first step is about mapping word vectors to visual feature space. Our cross-modal mapping is inspired by the work of Lazaridou et al. [48, 49]. They first introduce a simple regression method to project BoVW semantic vectors to word embeddings [48]. Their framework will predict an object occurring in a given input image. In their experiments, the search for the correct label of an input image is performed in full concept space. The experimental results are not promising because of lack of data for training cross-modal function. Lately, they delve into zero-shot learning problem by using the max-margin method to deal with intrinsic property of least-squares estimation and sparse matrix [49]. They also take into account visual semantic vectors extracted from ImageNet using CNN architecture. Unlike them, our first step of image generation system implements a variety of regression methods, especially lasso and elastic net with a purpose to solve a problem of sparse learning. More details of our cross-modal implementation are shown in Section 3.3.

2.4 Visual Feature Inversion

In this section, we will describe recent approaches in reconstructing an image from its visual feature. Most image understanding and object recognition systems build on image representations from histogram of oriented gradients (SIFT[50] and HOG[51] feature) and Bag of Visual Words (BoW)[52] to lately deep neural networks, especially the CNN variety [5, 53, 35]. This leads to an active area of research in visualizing features[54, 7]. In other words, feature visualization can be considered as a feature inversion problem which recovers the natural image that the feature is extracted from. It shows us a deeper and more intuitive understanding of how object recognition systems work.

Feature inversion was first introduced by Weinzaepfel et al. [55], in which they approximately recover images from its local descriptions (e.g, SIFT or PHOW features). However, there is a problem of privacy information since local descriptions of photos or videos are often used in many search purposes and image retrieval systems. Kato et al. [56] tackle the problem of recover natural images from Bag-of-Visual-Words (BoVWs) representations. BoVW representation is defined as a histogram of local feature descriptors extracted on a regular grid at a particular scale. There are two obvious challenges: 1) Local descriptions are assigned to visual words, so BoVW has quantization errors 2) It lacks of spatial information of local descriptors because we count their occurrences as semantic representations of images. To solve this problem, they use a very large image database to estimate the arrangement of local descriptions and quantization errors. In their experiments, they reconstructed 101 different images of objects, and showed that original images can be recovered from their BoVW representations.

Deep Convolutional Neural Networks allow us to produce state-of-the-art image embeddings. [5]. It is an important component in almost computer vision applications today. However, the understand of CNN features is still limited. Mahendran et al. [7] propose a general method for inverting CNN visual features by integrating natural image priors. The experimental results showed that their inversion algorithm outperforms any alternatives in the frame of reconstructing images from HOG and SIFT features. It is also reasonable and applicable for CNN features. They also conclude that some deep layers (e.g., pool-5 and fc-7) near the last layer of a CNN still retain rich semantic information of objects.

Finally, Cark Vondrick develops HOGgle framework[6, 57] that solves the problem of visual feature inversion as paired dictionary learning. It was first introduced to invert HOG visual features to natural images. However, it is customized to train on CNN feature later. We implement some pre-experiments for both latest methods to choose the most suitable one for our image generation pipeline. As a result, the feature inversion method proposed by Mahendran et al. [7] can invert CNNs features more accurately (shape and edge) than those of HOGgles. However, the inverted images are too much "blue" compared to those of HOGgles. In addition, HOGgles seems to recover more image contents such as: colors and background. Therefore, we decide to integrate the HOGgles framework into the second step of our image generation pipeline. The output of the first step is visual feature representations learned from cross-modal from word embedding to visual semantic space. We desire to invert those learned features back to pixel space to generate natural images of input concepts. Section 3.2 describes the HOGgles framework in more details, and training data and learning process is introduced in Section 3.3.

Chapter 3

Language-Driven Image Generation System

This chapter provides some details of our language-driven image generation model. The first section describes the task of cross-modal mapping (using regression to translate word embeddings to image embeddings), while the second is about image generation process where we apply the method HOGles to inverted visual feature back to pixel space (image generation). The last section is a description of all materials we used in this project.

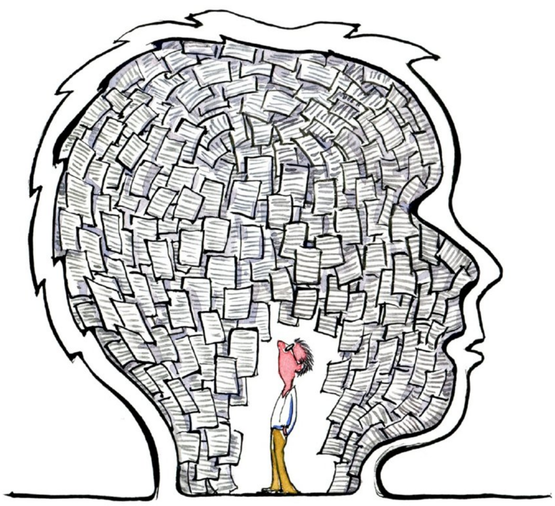


Figure 3.1: In the spirit of the English Idiom: "A picture is worth a thousand words", referring to the notion that a complex idea can be conveyed with just a single image or a picture tells a story just as well as a large amount of descriptive text. It also expresses one of the main goals of visualization, namely making it possible to absorb large amounts of data quickly.
[Source: Wikipedia]

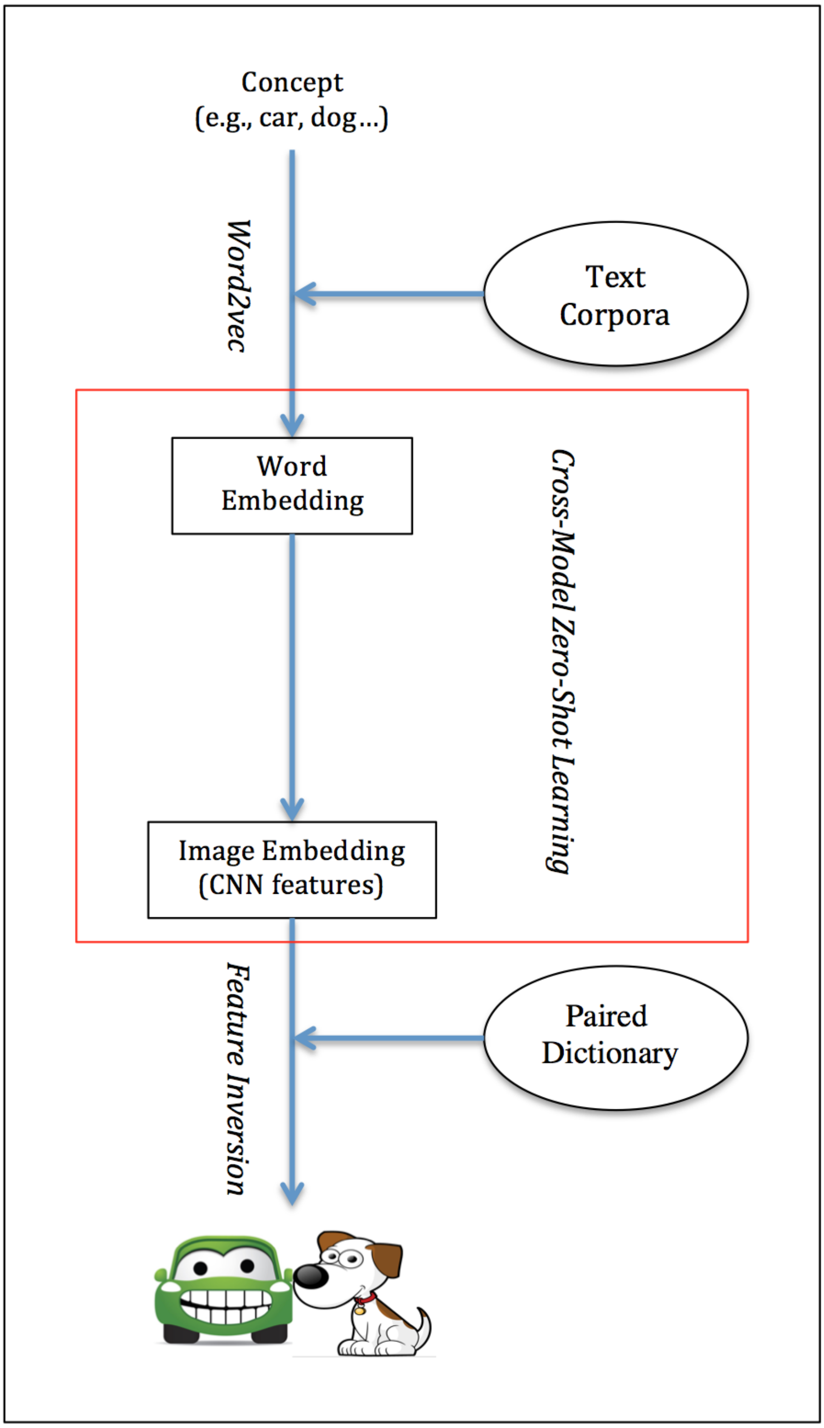


Figure 3.2: Overview of our language-driven image generation system

3.1 From linguistic space to visual feature space

This step is to learn a cross-modal mapping function from word embeddings to image embeddings (visual feature vector). In particular, the mapping is performed by inducing a function $f : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$ from data points (w_i, v_i) where $w_i \in \mathbb{R}^{d_1}$: word representation, $v_i \in \mathbb{R}^{d_2}$: corresponding visual feature representation. The translation of a given word vector w_j into visual feature space is obtained by applying the mapping function $\hat{v}_j = f(w_j)$. We assume that the cross-modal mapping is linear.

$$\hat{\mathbf{M}} = argmin_{M \in R^{d1 \times d2}} ||\mathbf{WM} - \mathbf{V}||_F + \lambda_1 ||\mathbf{M}||_1 + \lambda_2 ||\mathbf{M}||_F \tag{3.1}$$

From the above equation, we implement various regression methods by modifying λ_1 and λ_2.

- **Plain Regression**: $\lambda_1 = 0$ and $\lambda_2 = 0$, we estimate the coefficients $\hat{M} \in \mathbb{R}^{d1 \times d2}$ by least squares:

$$\hat{\mathbf{M}} = \underset{\hat{M} \in \mathbb{R}^{d1 \times d2}}{\operatorname{argmin}} ||\mathbf{V} - \mathbf{WM}||_2^2 \tag{3.2}$$

- **Ridge Regression**: $(\lambda_1 = 0, \lambda_2 \neq 0)$, has better prediction error than plain regression in a variety of scenarios, depending on the choice of λ_2. Ridge regression never set coefficients to zero exactly, and therefore it cannot perform variable section in the linear model. This is not desirable in our case as the number of variables (the dimension of word vectors and visual features) is quite huge (a pool5 feature vector has 9216 dimensions).

$$\hat{\mathbf{M}} = \underset{\hat{M} \in \mathbb{R}^{d1 \times d2}}{\operatorname{argmin}} ||\mathbf{V} - \mathbf{WM}||_2^2 + \lambda_2 ||M||_2^2 \tag{3.3}$$

- **Lasso**: $(\lambda_1 \neq 0, \lambda_2 = 0)$ is actually an acronym for: Least Absolute Selection and Shrinkage Operator [58] Lasso problem uses a $L1$ penalty $||M||_1$ while ridge regression adds a (squared) $L2$ penalty $||M||_2^2$ to least squares error.

$$\hat{\mathbf{M}} = \underset{\hat{M} \in \mathbb{R}^{d1 \times d2}}{\operatorname{argmin}} ||\mathbf{V} - \mathbf{WM}||_2^2 + \lambda_1 ||M|| \tag{3.4}$$

The tuning parameter λ_1 controls the strength of the penalty (like λ_2 in ridge regression) and generates a sparse model. Implementing Lasso will balance two ideas: fitting a linear model of V on W, and shrinking the coefficients. $L1$ penalty causes some coefficients to be shrunken to zero exactly. This is our expectation because there are many dimensions of the CNN visual feature extracted from images are equal to zero. This is the substantially different from ridge regression: it is able to perform variable selection in the linear model. Increasing λ_1 will turn more coefficients are set to zero (less variables are selected), and more shrinkages are employed in the non-zero coefficients.

- **Symmetric Elastic Net**: ($\lambda_1 \neq 0$,$\lambda_2 = \lambda_1$) is a regularized regression method that linearly combines the $L1$ and $L2$ penalties of the lasso and ridge method. More precisely, symmetric elastic finds an estimator for coefficients in a two-stage procedure: finding the ridge regression coefficients for each fixed λ_2, then doing a lasso shrinkage by tuning λ_1. However, this estimation makes double amount of shrinkage, which leads to increased bias and poor predictions. To solve this problem, we may rescale the estimated coefficients by multiplying them by $(1 + \lambda_2)$.

3.2 Image Generation

Most object recognition systems are built on image representations from histogram of oriented gradients (SIFT[50] and HOG[51] feature) and Bag of Visual Words (BoW)[52] to lately deep neural networks, especially the Convolutional Net variety [5, 53, 35]. This leads to an active area of research in visualizing features[54, 7]. In other words, feature visualization can be considered as a feature inversion problem which recover the natural image that the feature is extracted from. It shows us a deeper and more intuitive understanding of how object recognition systems work.

We are inspired by Carl Vondrick's framework[6] that solves the problem of visual feature inversion as *paired dictionary learning*. A visual feature can be various standards of image representations such as: SIFT, HOG, CNN features, etc. In particular, given an image $x_0 \in \mathbb{R}^D$ and its visual feature $y = \phi(x_0) \in \mathbb{R}^d$, find an image x^* so that minimizes the reconstruction error:

$$x^* = argmin_{x \in \mathbb{R}^D} ||\phi(x) - y||_2^2$$

Suppose that x and y are written in terms of bases $U \in \mathbb{R}^{DxK}$ and $V \in \mathbb{R}^{dxK}$ respectively, but they have paired representation through shared coefficients $\alpha \in \mathbb{R}^K$:

$$x = U\alpha \text{ and } y = V\alpha$$

To recover the original image, the visual feature y is firstly projected onto basis V, then projecting α in the basis U. Therefore, inversion of visual feature y is computed by the following formula:

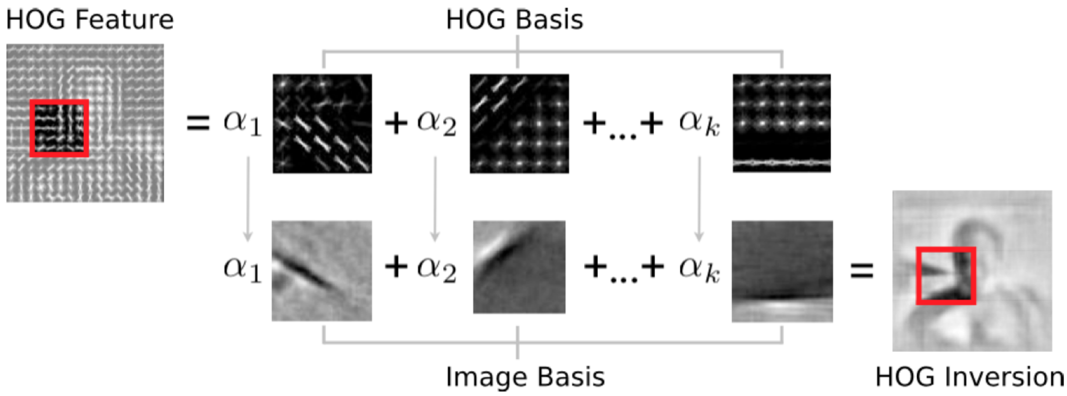


Figure 3.3: Inverting HOG feature using paired dictionary learning: first, project the HOG vector onto a HOG basis. By jointly learning a coupled basis of HOG features and natural images, then transfer the coefficients to the image basis to recover the natural image.

[Source: Vondrick et al. 2013 [6]]

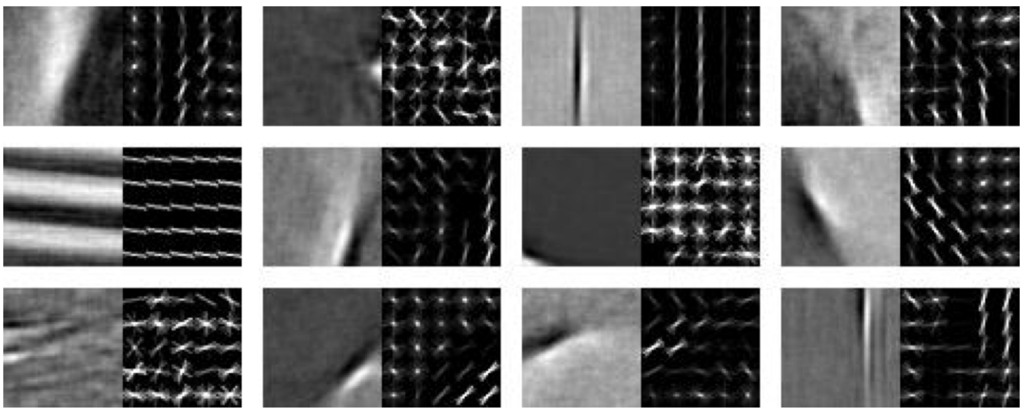


Figure 3.4: Some pairs of dictionaries for U (the left of every pair) and V (the right). Notice the correlation between dictionaries.

[Source: Vondrick et al. 2013 [6]]

$$\theta_D^{-1}(y) = U\alpha^*$$
$$\text{where } \alpha^* = argmin_{\alpha \in \mathbb{K}} ||V\alpha - y||_2^2 \quad \text{s.t.} \quad ||\alpha_1|| < \lambda$$

The algorithm has to estimate coefficients α and find appropriate bases U and V (paired dictionaries) to that minimize the reconstruction error. This is solved by applying recent advances in sparse coding method [59, 60]. HOGgles optimises SPAMS[61] to solve a standard sparse coding and dictionary learning problem with concatenated dictionaries:

$$argmin_{U,V,\alpha} \sum_{i=1}^{N}(||x_i - U\alpha_i||_2^2 + ||\theta(x_i) - V\alpha_i||_2^2)$$
$$\text{s.t} \quad ||\alpha_i||_1 \leq \lambda \forall i, ||U||_2^2 \leq \gamma_1, ||V||_2^2 \leq \gamma_2$$

In the first place, HOGgles was introduced to invert specific HOG features to natural images, but learned paired dictionary algorithm does not depend on what type of visual features is. So, it has also been used to invert CNN features back to multiple natural images to gain new insight into the failure of object detection systems[62, 57]. See the next section for details of our training data for CNN features.

3.3 Pipeline's Materials

3.3.1 Unseen Concepts

Unseen concepts are the words we generate images for. The set of unseen concepts (testing concept, dreamed concept) contains concepts coming from a study of McRae [9] in the context of property norm generation. McRae and his colleagues provide a set of feature norms collected from approximately 725 participants for 541 living (e.g., *cat, dog, etc.*) and nonliving (e.g., chair, car, boat etc) concepts. In many theories of word meaning and concept categorization, semantic feature vectors have been a key factor. They are also used in many vector space models of memory, object recognition and semantic memory. McRaer and his colleagues have been collected semantic feature production norms since 1990. The major goal of this work is to construct vector representations of concepts that can be used to test theories of semantic computation.

Each of 541 normed concepts corresponds to a basic concrete English noun. They are chosen from those used in various experiments about semantic memory tasks. Participants are asked to list semantic features they think are important for each concept. Each feature of each concept is assigned with a production frequency, which means the number of participants who produce that feature belonging to that concept (ranging from 1 to 30). For example, the semantic representation of concept *knife* has some attributes such as: *has-a-handle, has-a-blade, made-of-steel, used-for-cutting, is-sharp, etc...* Appendix B provides more details of feature representations derived from such concepts. This collection of feature norms [1] are also available for other research in: neuroscience and com-

[1] https://sites.google.com/site/kenmcraelab/norms-data

puter science, etc.

The set includes 541 base-level concrete concepts (e.g., cat, apple, car etc.) that span across 20 general and broad categories (e.g., animal, fruit/vegetable, vehicle etc). For some of the reasons (concepts are ambiguous or technical reasons), we exclude 69 McRae concepts, resulting in **472** test concepts (see in the Appendix A). We aim at generating natural images for all these concepts from their word embeddings. In one of the experiments we implement to evaluate our system performance, given a generated image, we ask participants to choose the right concept in a pair of an unseen concept and its confounder. The confounder is computed as the nearest semantic neighbor of that unseen concept in MacRae's conceptual distance space (Chapter 4, Experiment 2).

3.3.2 Seen Concepts

Seen Concepts refer to the set of training words associated with real images which are used for training cross-modal mapping functions. A set of 5K distinct concepts labelling 480K images from ImageNet [63]. Note that Unseen Concepts and Seen Concepts do *not* overlap, we want to stress again that our system is generating images in a zero-shot manner.

ImageNet is a WordNet-based image dataset, where images are organized according to the WordNet hierachy. Each "synset", described by a word or a word phrase, is a meaningful concept in WordNet. ImageNet provides on average 500-1000 clean and full resolution images for each synset. This results in ten of millions of annotated images organized by 80,000 synsets of WordNet. Recently, ImagetNet has become a central resource for the computer vision community following possible applications: a training dataset, a benchmark dataset, visual semantic modelling and human vision research. The figure 3.5 illustrates an example of ImageNet dataset.

3.3.3 Word Representation

Using *word2vec* toolkit[2] to produce 300-dimemesional semantic vectors for Unseen and Seen concepts from a very large text corpus of 2.8 billion words (e.g., BNC, Wikipedia...)[3]. In more details, we implement the *CBOW* method [2, 1] which predicts the semantic vector of the targeted word from the ones surrounding it, perform state-of-the-art results in many linguistic tasks [4].

[2]https://code.google.com/p/word2vec/

[3] Corpus sources: http://wacky.sslmit.unibo.it, http://www.natcorp.ox.ac.uk

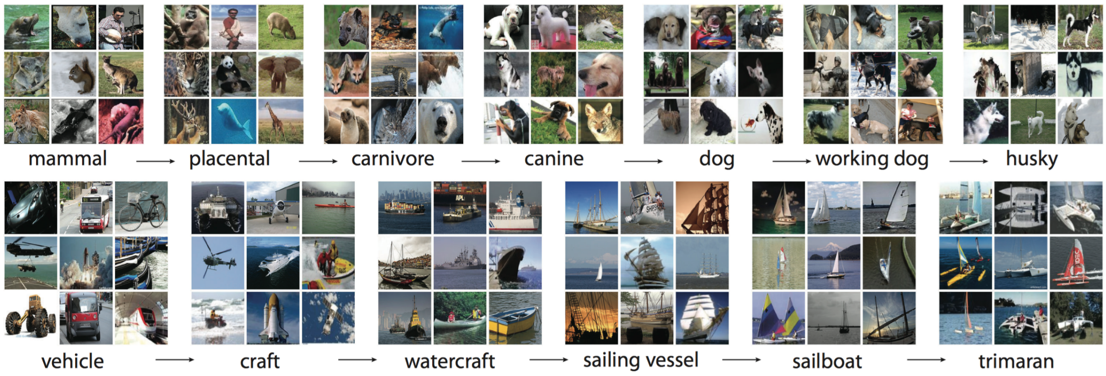


Figure 3.5: A snapshot of two root-to-leaf branches of ImageNet: the top row is from the mammal subtree; the bottom row is from the vehicle subtree. For each synset, 9 randomly sampled images are presented.
[Source: ImageNet.net]

3.3.4 Visual Representation

We use the pre-train CNN model through the Caffe toolkit[4] for extracting visual representations of 480K seen concept images. In this work, we experiment with visual features from two layers: *pool-5* and *fc-7*

- Pool-5 (6x6x256=9216 dimensions): an intermediate pooling layer that should capture object commonalities.
- Fc-7 (1x4096 dimensions): a fully-connected layer just below the topmost one, and as such it is expected to capture high-level discriminative features of different object classes.

Because each seen concept labels many images, we attempt to compute a unique visual representation in two ways. Inspired of from categorization schemes in cognitive science, we compute a visual representation by *prototype* [64] and *exemplar* methods [65]. The exemplar visual vector of a concept is a certain single vector that is the centroid vector among all visual features (either pool-5 or fc-7) extracted from the images it labeled. The prototype vector of a concept, on the other hand, does not actually depict the concept, is constructed by average the visual features of images tagged in ImageNet with the concept.

3.3.5 Feature Inversion Training

Training data for the second phrase feature inversion" are created by using the PASCAL VOC 2011 dataset. We want our system generates an image for a concept that it has never encountered before, so we pick 20 PASCAL objects (labelling 15k images) which do not occur in our Unseen Concepts. At this point,

[4]http://caffe.berkeleyvision.org

the feature inversion is also performed in a zero-shot manner. In order to increase the size of the training data, from each image we divided several image patches x_i associated with different parts of the image and paired them with their equivalent visual representations y_i. We trained paired dictionary learning for both features (pool-5 and fc-7) using the HOGgles software with default parameters[5].

[5] https://github.com/CSAILVision/ihog

Chapter 4

Experiments

We have already introduce our Language-driven image generation system in Chapter 3. Our system performs the task of visualizing semantics encoded in word embeddings. Given a text-based vector of concept "boat" , our system will generate a natural image of a "boat". But how do we evaluate the quality of semantic visualization? In this chapter, we design a series of ClowFlower studies to evaluate our model. We also provide quantitative and qualitative insights into the semantic information encoded in word embedding by allowing subjects judge and inspect generated images based on their visual properties (e.g., shape, color, characteristic environment).

4.1 Model Selection and Parameter Estimation

Recall from the last section, our system implements cross-modal mapping functions, namely four regression functions, to translate word embeddings to visual space. We also take into account two types of visual feature, which are outputs of two CNN layers: *pool5* and *fc7* and also two methods of concept representation (*prototype* and *exemplar*). In overall, our system contains 16 settings for each run. Therefore, to estimate quality of semantic visualization, we firstly need to determine the optimal setting that produces the best performance of image generation.

4.1.1 Visual feature type and concept representations

We set up a human study through CrowFlower[1] to identify the ideal visual feature type (*pool-5* or *fc-7*) and which concept representation method (*prototype* and *exemplar*) is better. In this experiment, our system does not perform cross-modal mapping, instead generate images from gold-standard visual vectors of unseen concepts (inverted pool5 and fc7 features of a concept back to its natural images.). We randomly choose 50 from unseen concepts and generate 4 different images for each concept. Each obtained by inverting visual vector computed by a combination of a feature type with a concept representation method. Table 1

[1] http://www.crowdflower.com/

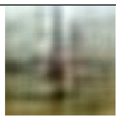
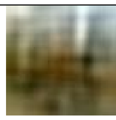
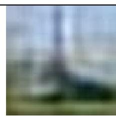
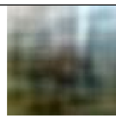
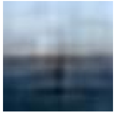
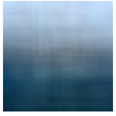
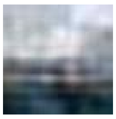
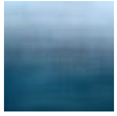
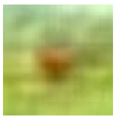
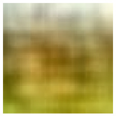
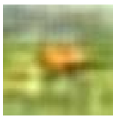
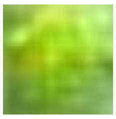
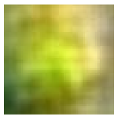
	Prototype		Examplar	
	pool-5	fc-7	pool-5	fc-7
Catapult				
Sailboat				
Elk				
Cucumber				

Table 4.1: Inverted images from different visual type and concept representation

show an example of system's outputs for each setting.

Task: Participants were asked to judge which one in 4 generated images is more likely to denote the concept. We collected 20 judgements for each concept. Table 4.1 provides generated images of four concepts with various feature types and concept representation methods.

Result: Overall, there is no surprise that the strongest significant preference of visual feature type is for *pool-5* (28/50) because the reconstruction from the *pool-5* features keeps more rich information than that of *fc-7* . Meanwhile, while judging image generation quality from *pool-5* visual feature, participants show the best preference for the exemplar protocol (18/50)[2]. Therefore, all following experiments are conducted using the *pool-5+exemplar* setting.

4.1.2 Cross-modal mapping function

After determining the best visual feature and concept representation setting, we carry out another experiment to firmly decide which is the optimal learning regression method among: plain regression, ridge regression, lasso and elastic net. To do so, we need first to train mapping functions corresponding to four different types of regression. Training data is described in Section 3.1.
Task: We set *pool-5+exemplar* space to run our system. For each in 50 above concepts, our system produces four images for each mapping function. We asked participants to decide which image is more likely to denote the unseen concept

[2] Throughout this thesis, statistical significance is assessed with two-tailed exact binomial tests with threshold $\alpha < 0.05$, corrected for multiple comparisons with the false discovery rate method.

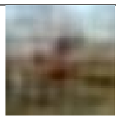
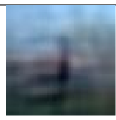
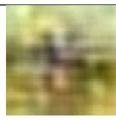
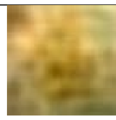
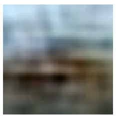
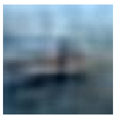
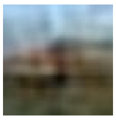
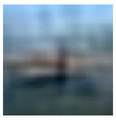
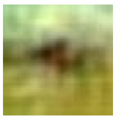
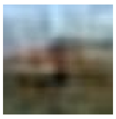
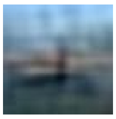
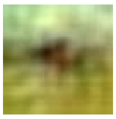
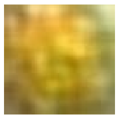
	Catapult	Sailboat	Elk	Cucumber
Plain Regression				
Ridge				
Lasso				
Elastic Net				

Table 4.2: Images inversion from four regression methods

and collected 20 judgements for each concept. Table 4.2 provides some generated images in this task.

Result: Surprisingly, participants showed a preference for plain regression (9/50 significant tests in favor of this model). Consequently, we adopt plain regression and *pool-5+exemplar* space for the rest of the experiments.

4.2 Inspecting Visual Properties of Generated Images

The optional model and setting allow system produce the best quality generated images of Unseen concepts. Bear in mind that Unseen concepts were never used in any step of the pipeline of our system (cross-modal training or paired dictionary learning for feature inversion), so they are generated in a zero-shot manner. That means our language-driven image generation system leverages linguistic associations between Unseen concept and seen concepts to generate natural images. To summary generated images of Unseen concepts, we randomly select 10 concepts from each of 20 McRae categories, then plot them in Figure 4.1

Unexpectedly, these images do not clearly denote objects compared to those real ones. However, the figure shows that most of images belonging to different categories are clearly distinguished, but concepts belonging to *food* and *fruit/vegetable* group look pretty much the same. Food, fruit and vegetable are eatable thing, they are likely co-occur in the same context. Modern object recognition systems do not perform well on a task of food recognition and classification. Overall, from the figure we can conclude that concepts belonging to different categories

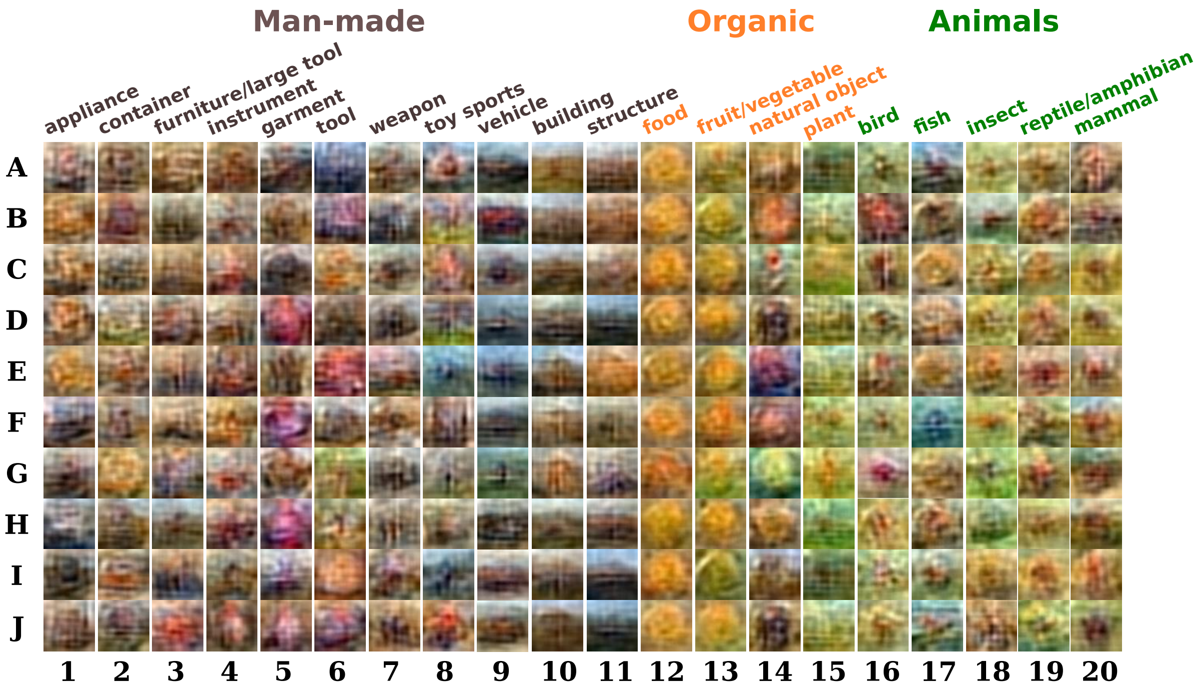


Figure 4.1: Generated images of 10 concepts per category for 20 basic categories grouped by macro-category

(MAN-MADE and ORGANIC, ANIMAL) are clearly distinguished by color and environment visual properties.

4.2.1 Experiment 1: Correct word vs. random confounder

This experiment allows us to examine whether visual properties help participants distinguished between an Unseen concept and a random alternative.

Experiment Description: We present to participants the generated images of all 472 Unseen concepts. For each image, participants have to judge which concept (the unseen concept or a cofounder choosing randomly from the seen concept) it denotes. We collected 20 votes for each trail.

Results and Discussion: Participants showed a strong preference for the correct word (Unseen concept) (medium of the vote in favor: 75%). Preference for the correct concept is different from chance in 211/472 trails. Moreover, there are 10 cases in favor of cofounders which participants expressed significant preferences. Almost in these cases, unseen concepts and their cofounders have some properties and attributes in common: *cape-tabletop* (both made of textile), *zebra-baboon* (they are mammals), oak-boathouse (living in similar natural environments). To conclude, our method is able to capture at least those visual properties encoding in word representations of Unseen concepts that distinguish them from visually dissimilar random alternative.

4.2.2 Experiment 2: Correct image vs. image of similar concept

In Experiment 1, both cofounders and Unseen concepts are basic and concrete objects. In fact, the cofounder of the correct word is a randomly selected from Seen concepts (Training concepts); hence they are related to each other by chance. Consequently, the judgement task in Experiment 1is relatively simple. For an example, participants easily guess correct concept "turtle" over a random cofounder *"bike"* because they are totally not in the same class and not related to each other. However, if we present to participants the correct image of test concept and the generated image of its relevant concept. For example, *"turtle"* and *"tortoise"* are presented. Is it still easy for subjects to discriminate which image labelling *"turtle"*. The Experiment 2 is conducted to discover those visual properties in the generated images are informative enough for subjects to judge an Unseen concept over its closely related concept.

Experiment Description: For each test concept, we need to find a related concept as its cofounder. Thanks to the MacRae's statistics of subject-based conceptual distance [9], we consider a confounder of a concept as the closed semantic neighbor of the concept in MacRae's conceptual distance space. For more details, we found 379/472 cofounders belong to the set of Unseen concept. Judging two semantic close concepts given an image is thus more challenging (e.g. mandarin vs. pumpkin, turtle vs. tortoise, bowl vs. dish etc.). Participants were presented with two images generated from an unseen concept and its cofounder, and they were also asked to guess which image is more likely to depict the unseen concept. The set up of the experiment is the same as that of Experiment 1.

Results and Discussion: Unexpectedly, in many cases (409/472) participants did not express a strong preference for either the correct image or the cofounder. This means that our current image generation system does not capture enough yet visual properties from word embedding that could allow within-category discrimination.

Observing within the subset of 63 cases expressed significant by participants, we notice a clear trend in favor of the correct image (41 vs.22). Some visual properties (e.g., *edge* and *angle*) are not good enough to discriminate two images of the correct word and its cofounder although color and environment seem to be the fine-grained properties which help subjects decided right or wrong choice within this subset. In 63 pairs of images, there are 14 concepts coming from different categories, and 49 concepts within the same-categories. In the different group, 11 cases were for the right images. In two of the wrong cases, images of correct concept and cofounder have similar color (*emerald* vs. *parsley*, *bowl* vs. *dish*), while (*thermometer* vs. *marble* has a typically *"brown"* color. For the 49 pairs in the same-category group, the task is very challenging, but participants expressed 30/49 right choices as a strong preference. Again, color and living environment

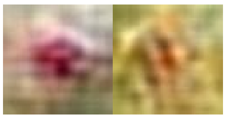
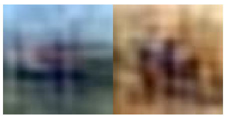
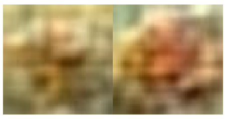
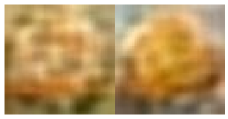
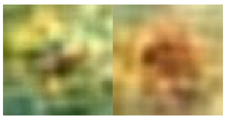
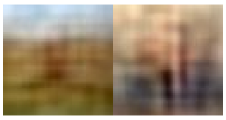
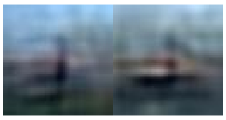
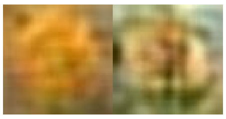
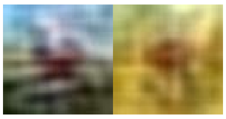
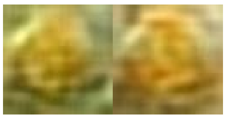
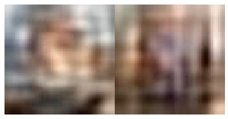
Same category	Different category	Same category	Different category
flamingo partridge	helicopter shotgun	alligator crocodile	bowl dish
			
turtle tortoise	barn cabinet	sailboat boat	emerald parsley
			
pumpkin mandarin	whale bison	asparagus spinach	thermometer marble
			
In favor of Unseen concept 8.6% (41/472)		In favor of confounder 4.6% (22/472)	

Table 4.3: Examples of cases where subjects significantly preferred the dreamed concept image (left) or the confounder (right)

play an important role in this preference. The judgement was mostly based on different colors (e.g., *flamingo* vs. *partridge*), or living in different environments (e.g., *turtle* vs. *tortoise*).

4.2.3 Experiment 3: Gold-standard Image Judgement

As far as we investigate, our current image generation system performs not quite well on Unseen concepts. There might be some reasons we want to examine: the quality of word embedding (cannot capture semantic information enough from text corpus) or the poorly feature inversion algorithm. Although recently tool Glove[3] is the best method to extract word representation from text, its performance in many linguistic tasks is not significantly improved compared to that of *CBOW* method that we implement in our approach. We suspect the main reason behind this is a lack of HOGgle algorithm. To prove this, we carry out a "gold-standard" image judgement flowing the Experiment 2.

Particularly, we generate images for each Unseen concept and its cofounder using the "gold-standard" visual features extracting from real existing images. That means we do not take into account running cross-modal mapping in this experiment. "Gold-standard" visual features are also extracted using the consistent setting *(pool-5+exemplar)*. We replace the translated visual vectors with these images and then repeat the Experiment 2 with the same setup.

The results are better than the case that visual features are translated from word embedding, but it is not much different. Comparing to the earlier experiment, the number of pairs for which no consistent preference appeared was 75% (365/472 cases) and the strong preferences for the correct images and the cofounders were 17.6% (83/472 cases) and 7% (33/472). So our suspicion is right. Accordingly, we hope to improve our system's performance by applying upcoming advances in image inversion in the future, such as: a CNN feature inversion approach is proposed in [54, 7].

4.2.4 Experiment 4: Judging macro-categories of objects

The previous experiments have shown that our language-driven image generation system visualizes semantic information that are salient and enough to discriminate unrelated concepts (Experiment 1), but not closely related ones (Experiment 2). We also prove that the inaccuracy of our image inversion algorithm can be attributed to the matter of poor image generation (Experiment 3). In this experiment, we want to check captured visual properties, whether allow subjects classify generated images into high-level semantics groups.

Experiment Description: Unseen concepts are manually divided into three macro- categories groups, namely ANIMAL vs. ORGANIC vs. MAN-MADE. These groups are essential and unambiguous in cognitive science [9]. Subjects were asked to categorize a given generated image under three macro-category groups.

Results and Discussion: Table 4.4 denotes a confusion matrix of choices across the macro-categories. Participants show a significant preference for ORGANIC group (49/68 cases) while there are expectable preferences for images of MAN-MADE and ANIMAL group: 142/266 cases and 56/128 cases respectively. However, looking at a subset of images which participants preferred, most of the cases are in favor of the correct macro- category: 98% of the ORGANIC images (70,5% of total), 90% of the MAN-MADE images (48% of total), and 59% of the ANIMAL ones (25.7% of total). The table also shows that images of both MAN-MADE and ANIMAL macro-categories are more often confused than that of ORGANIC one.

	man-made	organic	animal	pref	nopref	total
man-made	128	9	5	142	124	266
organic	0	47	1	49	19	68
animal	9	14	33	56	72	128

Table 4.4: Confusion Matrix of significant choices of categories.

Color is till the most important property that allow subjects to classify objects in three macro-categories. It is easy to see in the Figure 5.1 that orange, green and a darker color characterizes objects in ORGANIC, ANIMALS, and MAN-

MADE group respectively. Images which do not have these colors are harder to be classified.

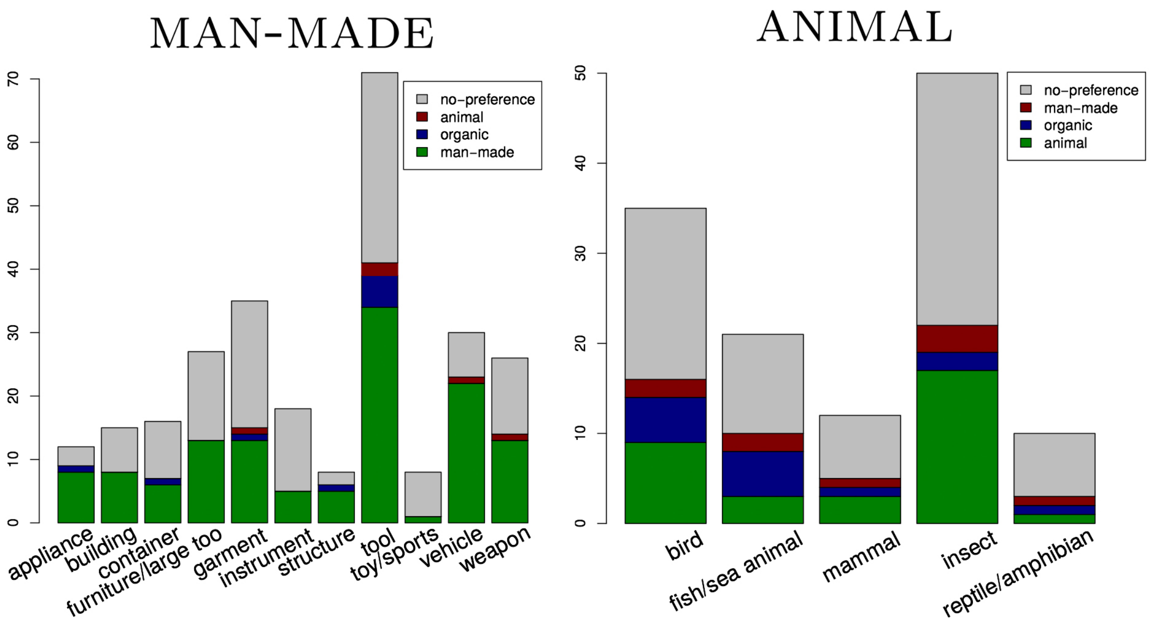


Figure 4.2: Distribution of macro-category preferences across the gold concepts of the MAN-MADE and ANIMAL categories.

In the MAN-MADE macro-category (Figure 4.2, left), the generated images of a building are easier to recognize because they share the same pattern: blue and dark background of sky and ground and a polygon structure in the middle (the building itself). Similarly, vehicles are mainly judged by colors visual properties. For example, generated image of "ambulance" is red, dark blue color of see environment allow subjects judge images of "ship" and "boat". Beside, vehicles display two layers with small horizontal structure crossing frames of images, they are almost always correctly classified.

Within the ANIMAL macro-category (Figure 4.2, right), birds and fishes are more often misclassified than other animals , with their typical environment probably playing a role in the confusion. However, insects share the same environment pattern with birds, but they are easily recognizable because of their small sizes. So, our system can capture a little information to shape or size of objects. Obviously, looking at generated images of *"camel"* and calf, they depict animal leg shape crossing land background. There is also the strongest preference for sub-class mammal.

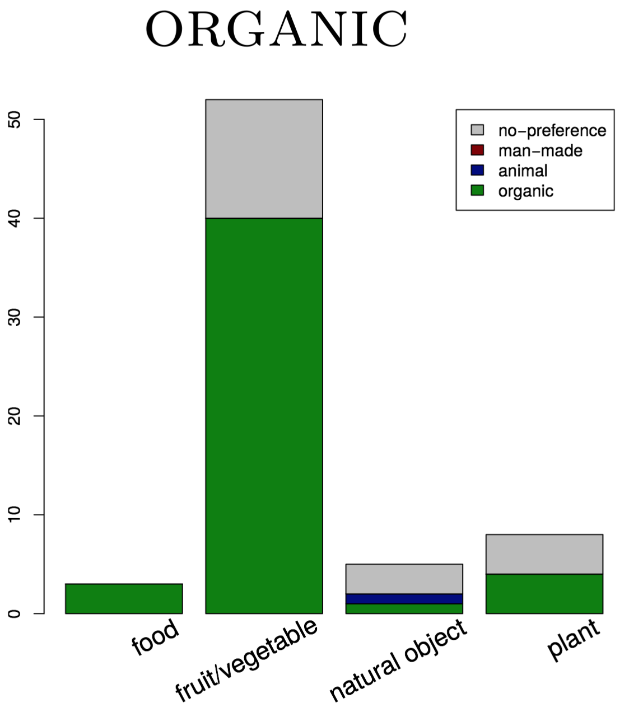


Figure 4.3: Distribution of macro-category preferences across the gold concepts of the ORGANIC categories.

Finally, the ORGANIC group is the most misclassified (Figure 4.3). Images of food and fruits look like the same (color and size). This might be due to quality of word embedding inducing. Word2vec tool cannot capture enough semantic information from our text corpus to strongly distinguish between food and fruits. However, it is still easier for participants to judge based on color properties In the other hand, there are few mistakes for ORGANIC images belong to the natural object category (e.g., stone, rock...). They are mainly classified under the ANIMAL macro-category.

4.2.5 Experiment 5: Automatic Evaluation

We have already introduced a series of CrowdFlower to examine our system performance. Then, we provide quantitative and qualitative insights into the information that subjects can extract from the visual properties of the generated images. In this experiment, we automatically evaluate the quality of generated images of unseen concepts. Instead of calculating error per pixel, we turn to The Structural Similarity (SSIM) algorithm, which is a method for measuring the similarity between two images. The SSIM method can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality [66].

Experiment Description: Due to independence property of unseen concepts, we sample one hundred images for each unseen concept in order to calculate SSIM. For each unseen concept, we calculate the SSI scores of each sample image with the generated one[3] Then, we average the 100 values as a SSIM score between the generated image with 100 gold-standard images. We pick the top 15 highest and lowest SSIM scores among 472 unseen concepts (lower is better). They are shown in Table 4.5 and Table 4.6

Results and Discussion: Most of the best case (the lower scores) are concepts belonging MAN-MADE macro-category. This validates our conclusion from the Experiment 4 since images of the MAN-MADE concepts are easy to recognize because their color and backgrounds. There are two ANIMAL concepts: *"otter"*, *"blackbird"* whose the quality of generated images are quite high. On the contrary, our system is not good at producing images for ORGANIC concepts which is the most of cases in the top 15 worst performance. This is the reason why the ORGANIC marco-category is the most misclassified group in the Experiment 4.

[3]https://github.com/pornel/dssim

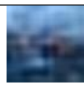
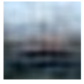
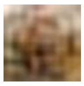
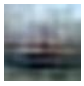
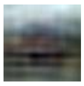
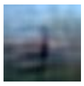
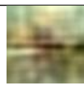
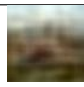
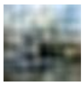
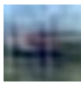
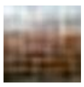
Concept	Score	Gen Image	Macro-Ca.
jet	0.510174		Man-Made
ship	0.540007		
armour	0.573253		
yacht	0.573513		
skyscraper	0.574131		
bus	0.597194		
sailboat	0.598178		
otter	0.608347		Animal
cottage	0.617131		Man-Made
harpoon	0.617627		
submarine	0.620575		
helicopter	0.622526		
blackbird	0.624216		Animal
cannon	0.625781		Man-Made
apartment	0.627062		

Table 4.5: Top 15 best generated images of our system comparing to 100 gold standard images by Structural Similarity distance

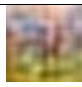
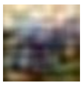
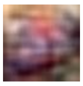
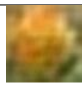
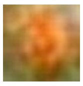
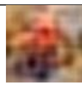
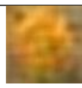
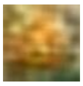
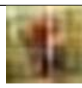
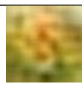
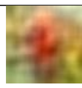
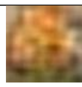
Concept	SSMI	Gen Image	Macro-Ca.
ball	1.419652		Man-Made
pencil	1.262961		
crayon	1.251465		
eggplant	1.241847		Organic
rhubarb	1.201448		
radish	1.188495		
toy	1.187393		Man-made
pumpkin	1.186196		Organic
banana	1.178495		
nectarine	1.174497		
budgie	1.173635		Animal
strawberry	1.160650		Organic
tomato	1.158309		
fawn	1.149805		Animal
plum	1.145742		Organic

Table 4.6: Top 15 worst generated images of our system comparing to 100 gold standard images by Structural Similarity distance

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we introduce a new task of images visualizing the semantic information as encoded in word representation. We also review various research directions related to our project such as: learning semantic representations from text and image, cross-modal mapping and visual feature inversion. We develop a completely language-driven image generation pipeline. Our pipeline takes a text-based semantic vector of a concept as its input, then projects the vector to the visual feature space. The visual features are thus inverted into pixel space representing a natural image of the input concept.

Experimental results show that our current system seems capable to visualize color properties of object classes and aspects of their characteristic environment. However, the experimental results show that our pipeline might be used to enrich the semantics encoding in word representations, by highlighting aspects of concepts that are not salient in language such as characteristic environment. In this sense, language-driven image generation is a tool to evaluate semantic quality of word representation. In other words, our system take a word embedding as an input, if we can check whether the representation is good or not by looking at the generated image, where it depicts roughly visual properties of the concept. In addition, when interpreting and evaluate dense vectors, there is only the possibility of looking at their neighbours. Our method can be seen as a complementary since intuitively visualizing those vectors by generating their natural images. Nevertheless, our system is not good enough to capture visual properties related to shape. The reason might be that there is often a lack of expressions about shape in linguistic. (Although we all know that an *"elk"* is a mammal, it is very difficult to describe it in shape to discriminate to other mammal), but in the same way we can easily recognize the *"elk"* based on color and the environment. Effective cross-modal learning functions or feature inversion algorithms might lead us in the future by better visualizing typical shapes to shape-blind linguistic representations.

5.2 Future Work

Currently we propose a language-driven image generation pipeline as a two-step process. As mentioned above, it still does not capture shape property information enough. Meanwhile, in the work of [7], their inversion algorithm is better at visualizing objects with shape properties, but not for expressing color of objects. We plan to add more functionality to our system, in which it can generate black-white image from word embeddings to effectively estimate in depth shape/size expression. In addition, inspired from recent work on a generative model of text conditioned on images by extending the Deep Recurrent Attention Writer (DRAW) [67], we plan to build a generative model of images from captions. More precisely, the future system will take textual descriptions as input and use them to generate relevant images. For example, the future system will have an ability to generate a scene given a caption such as: "a car is running on a high-way."

Bibliography

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, 2013.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[3] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove:global vectors for word representation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014.

[4] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, Maryland, June 2014.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceeding of Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[6] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2013.

[7] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[8] M. Andrews, G. Vigliocco, and D. Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463–498, 2009.

[9] Ken Mcrae, George S. Cree, Mark S. Seidenberg, and Chris Mcnorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 2005.

[10] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[11] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[12] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October 1965.

[13] Hinrich Schütze and Jan Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA, 1995.

[14] Patrick Pantel. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 125–132, Stroudsburg, PA, USA, 2005.

[15] Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, December 2010.

[16] Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA, 2008.

[17] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, January 2010.

[18] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, June 2007.

[19] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 208–215, New York, NY, USA, 2000. ACM.

[20] Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, March 1998.

[21] Thomas K Landauer and Susan T. Dutnais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[22] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning*, 3:1137–1155, March 2003.

[23] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

[24] Tomas Mikolov. Statistical language models based on neural networks. In *PhD thesis. Brno University of Technology*, 2012.

[25] Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of the 46nd Annual Meeting of the Association for Computational Linguistics (ACL-08: HLT)*, pages 236–244, 2008.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[27] M. Steyvers and T. Griffiths. Probabilistic topic models. In *Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum.*, 2005.

[28] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010.

[29] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 41–47, New York, NY, USA, 2003. ACM.

[30] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013.

[31] Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August 2013.

[32] Max M. Louwerse. Symbol interdependence in symbolic and embodied cognition. In *Topics in Cognitive Science*, pages 273–302, 2011.

[33] Mh Bornstein, Lr Cote, S Maital, K Painter, Sy Park, L Pascual, Mg Pecheux, J Ruel, P Venuti, and André Vyt. Cross-linguistic analysis of vocabulary in young children: Spanish, dutch, french, hebrew, italian, korean, and american english. *Child Development*, 75(4):1115–1139, 2004.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, January 2014.

[35] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceeding of European Conference on Computer Vision (ECCV)*, pages 818–833, 2014.

[36] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In Dieter Fox and Carla P. Gomes, editors, *Association for the Advancement of Artificial Intelligence*, pages 646–651. AAAI Press, 2008.

[37] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[38] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 609–623, Berlin, Heidelberg, 2012. Springer-Verlag.

[39] Felix X. Yu, Liangliang Cao, Rogério Schmidt Feris, John R. Smith, and Shih-Fu Chang. Designing category-level attributes for discriminative visual recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 771–778, 2013.

[40] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1641–1648, Washington, DC, USA, 2011. IEEE Computer Society.

[41] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost. In *Proceeding of European Conference on Computer Vision (ECCV)*, 2012.

[42] Thomas Mensink, Efstratios Gavves, and Cees G. M. Snoek. COSTA: co-occurrence statistics for zero-shot classification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2441–2448, 2014.

[43] Mitchell TM1, Shinkareva SV, Carlson A, Chang KM, Malave VL, Mason RA, and Just MA. Predicting human brain activity associated with the meanings of nouns. In *Science*, pages 1191–1195, 2008.

[44] Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom Mitchell. Zero-shot learning with semantic output codes. In *Proceeding of Annual Conference on Neural Information Processing Systems (NIPS)*, December 2009.

[45] Richard Socher, Milind Ganjoo, Christopher Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of Annual Conference on Neural Information Processing Systems NIPS*, pages 935–943, Lake Tahoe, NV, 2013.

[46] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013.*, pages 2121–2129, 2013.

[47] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *CoRR*, abs/1312.5650, 2013.

[48] Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414, Baltimore, Maryland, June 2014.

[49] Angeliki Lazaridou, Dinu Georgiana, and Marco Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2015.

[50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[51] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005.

[52] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[53] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2014.

[54] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

[55] P. Weinzaepfel, H. Jegou, and P. Perez. Reconstructing an image from its local descriptors. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 337–344, Washington, DC, USA, 2011. IEEE Computer Society.

[56] Hiroharu Kato and Tatsuya Harada. Image reconstruction from bag-of-visual-words. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[57] Carl Vondrick, Aditya Khosla, Hamed Pirsiavash, Tomasz Malisiewicz, and Antonio Torralba. Visualizing object detection features. *CoRR*, abs/1502.05461, 2015.

[58] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society Series B*, 73(3):273–282, 2011.

[59] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. *Transactions on Image Processing*, 19(11):2861–2873, November 2010.

[60] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2216–2223, 2012.

[61] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.

[62] Carl Vondrick, Hamed Pirsiavash, Aude Oliva, and Antonio Torralba. Acquiring visual classifiers from human imagination. *CoRR*, abs/1410.4627, 2014.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 248–255, 2009.

[64] Gregory L. Murphy. *The Big Book of Concepts*. MIT Press, Boston, Mass., 2002.

[65] F. Gregory Ashby and Leola A. Alfonso-Reese. *Categorization as probability density estimation*. Journal of Mathematical Psychology, 1995.

[66] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Transactions on Image Processing*, 13(4):600–612, April 2004.

[67] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning (ICML), 2015.*, 2015.

List of Figures

List of Tables

Attachments

We provide lists of concepts names (Unseen concepts) of word embeddings as input of language-driven image generation. Each table shows concepts names for each category in Experiment 4.

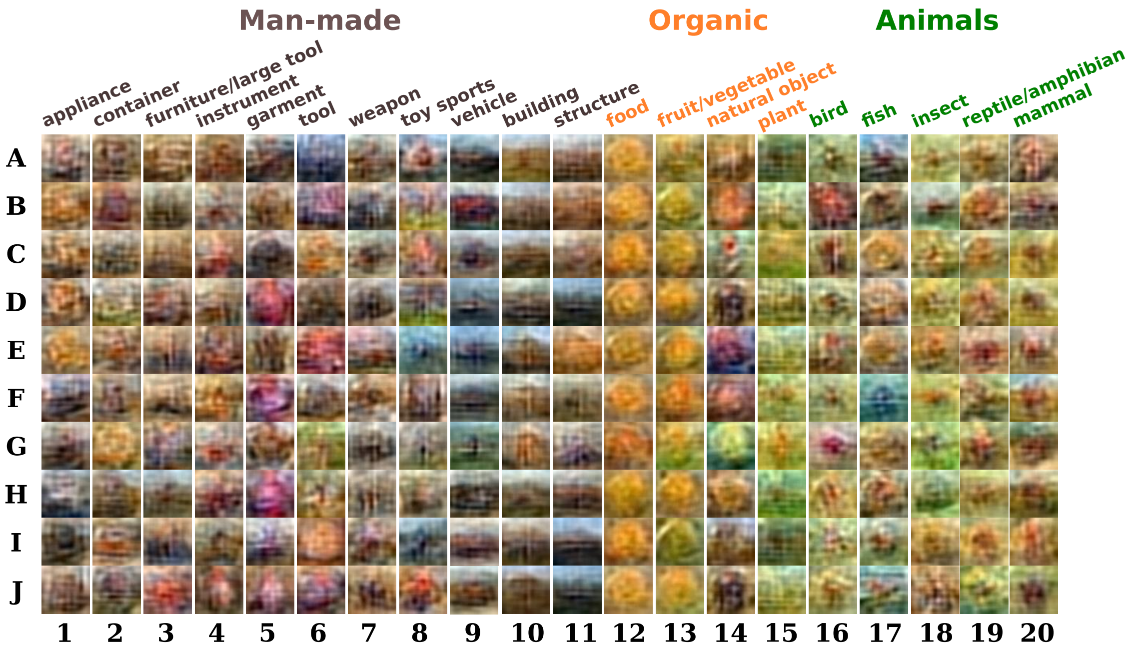


Figure 5.1: Generated images of 10 concepts per category for 20 basic categories grouped by macro-category

	12	13	14	15
A	biscuit	apple	beehive	birch
B	bread	asparagus	bouquet	cedar
C	cake	avocado	emerald	dandelion
D	cheese	banana	muzzle	oak
E	pickle	beans	pearl	pine
F	pie	beets	rock	prune
G	raisin	blueberry	seaweed	vine
H	rice	broccoli	shell	willow
I	cake	cabbage	stone	birch
J	biscuit	cantaloupe	muzzle	pine

Table 5.1: Concept names of word embeddings used to generate ORGANIC images

	1	2	3	4	5	6
A	dishwasher	ashtray	bed	accordion	apron	anchor
B	freezer	bag	bench	bagpipe	armour	banner
C	fridge	barrel	bookcase	banjo	belt	blender
D	microwave	basket	bureau	cello	blouse	bolts
E	oven	bathtub	cabinet	clarinet	boots	book
F	projector	bottle	cage	drum	bracelet	brick
G	radio	bowl	carpet	flute	buckle	broom
H	sink	box	catapult	guitar	camisole	brush
I	stereo	bucket	chair	harmonica	cape	candle
J	stove	cup	sofa	harp	cloak	crayon

Table 5.2: Concept names of word embeddings used to generate MAN-MADE images

	7	8	9	10	11
A	axe	balloon	airplane	barn	apartment
B	baton	ball	ambulance	building	basement
C	bayonet	doll	bike	bungalow	bedroom
D	bazooka	football	boat	cabin	bridge
E	bomb	kite	buggy	cathedral	cellar
F	bullet	marble	bus	chapel	elevator
G	cannon	racquet	canoe	church	escalator
H	crossbow	rattle	cart	cottage	garage
I	dagger	skis	car	house	pier
J	shotgun	toy	helicopter	hut	bridge

Table 5.3: Concept names of word embeddings used to generate MAN-MADE images (cont.)

	16	**17**	**18**	**19**	**20**
A	blackbird	whale	grasshopper	alligator	beer
B	bluejay	octopus	hornet	crocodile	beaver
C	budgie	clam	moth	frog	bison
D	buzzard	cod	snail	iguana	buffalo
s **E**	canary	crab	ant	python	bull
F	chickadee	dolphin	beetle	rattlesnake	calf
G	flamingo	eel	butterfly	salamander	camel
H	partridge	goldfish	caterpillar	toad	caribou
I	dove	guppy	cockroach	tortoise	cat
J	duck	mackerel	flea	cheetah	cheetah

Table 5.4: Concept names of word embeddings used to generate ANIMAL images

Concept Name	Feature	Production Frequency	Brain Region Classification
Moose	is large	27	visual-form and surface
	has antlers	23	visual-form and surface
	has legs	14	visual-form and surface
	has four legs	12	visual-form and surface
	has fur	7	visual-form and surface
	has hair	5	visual-form and surface
	has hooves	5	visual-form and surface
	is brown	10	visual-color
	hunted by people	17	function
	eaten as meat	5	function
	lives in woods	14	encyclopedic
	lives in wilderness	8	encyclopedic
	an animal	17	taxonomic
	a mammal	9	taxonomic
	an herbivore	8	taxonomic
Knife	has a handle	14	visual-form and surface
	has a blade	11	visual-form and surface
	made of steel	8	visual-form and surface
	made of metal	7	visual-form and surface
	made of stainless stell	5	visual-form and surface
	is shiny	5	visual-form and surface
	used for cutting	25	function
	used for killing	7	function
	used by butchers	5	function
	is sharp	29	tactile
	is serrated	8	tactile
	is dangerous	14	encyclopedic
	found in kitchens	18	encyclopedic
	used with forks	16	encyclopedic
	a weapon	11	taxonomic
	a utensil	19	taxonomic
	a cutlery	15	taxonomic

Table 5.5: Feature Representations for Moose and Knife

	fc7+examplar	fc7+prototype	pool5+examplar	pool5+prototype
Min.	0.00	0.00	0.00	1.00
1st Qu.	1.00	2.00	4.00	4.00
Median	2.00	2.00	8.00	5.00
Mean	2.66	3.12	7.36	6.86
3rd Qu.	3.00	4.00	11.00	9.00
Max.	16.00	9.00	16.00	19.00

Table 5.6: Model selection experiment

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.1000	0.5500	0.7500	0.7061	0.8500	1.000

Table 5.7: Distribution of proportion of times when the correct label was picked in Exp 1

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.3500	0.5500	0.5299	0.7000	0.9500

Table 5.8: Distribution of proportion of times when the correct image was picked in Exp 2

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0500	0.4000	0.6000	0.5679	0.7500	1.000

Table 5.9: Distribution of proportion of times when the correct image was picked in Exp 3

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	7.00	11.00	10.62	14.00	20.00

Table 5.10: Distribution of preferences for the right category Exp 4