A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels

Marcus Klasson[1]* Cheng Zhang[2] Hedvig Kjellström[1]

[1] KTH Royal Institute of Technology, Stockholm, Sweden, {mklas,hedvig}@kth.se
[2] Microsoft Research, Cambridge, United Kingdom, cheng.zhang@microsoft.com

Abstract

Image classification models built into visual support systems and other assistive devices need to provide accurate predictions about their environment. We focus on an application of assistive technology for people with visual impairments, for daily activities such as shopping or cooking. In this paper, we provide a new benchmark dataset for a challenging task in this application – classification of fruits, vegetables, and refrigerated products, e.g. milk packages and juice cartons, in grocery stores. To enable the learning process to utilize multiple sources of structured information, this dataset not only contains a large volume of natural images but also includes the corresponding information of the product from an online shopping website. Such information encompasses the hierarchical structure of the object classes, as well as an iconic image of each type of object. This dataset can be used to train and evaluate image classification models for helping visually impaired people in natural environments. Additionally, we provide benchmark results evaluated on pretrained convolutional neural networks often used for image understanding purposes, and also a multi-view variational autoencoder, which is capable of utilizing the rich product information in the dataset.

1. Introduction

In this paper, we focus on the application of image recognition models implemented into assistive technologies for people with visual impairments. Such technologies already exist in the form of mobile applications, e.g. Microsoft's Seeing AI [2] and Aipoly Vision [1], and as wearable artificial vision devices, e.g. Orcam MyEye [3] and the Sound of Vision system introduced in [6]. These products have the ability to support people with visual impairments in many different situations, such as reading text documents, describing the user's environment and recognizing people the user may know.

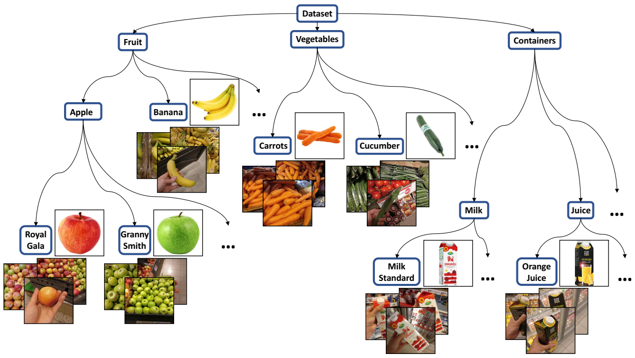


Figure 1. The primary contribution of this paper is a dataset of grocery items, for the purpose of training a visual recognition system to aid visually impaired people. The dataset is organized according to a hierarchical class structure, as illustrated above. A novel aspect of the dataset is that each class, apart from the semantic label, also has a visual label in the form of an iconic image.

We here address a complementary scenario not handled by current systems on the market: visual support when shopping for grocery items considering a large range of eatable objects, including fruits, vegetables, milk, and juices. In the case of fruits and vegetables, these are usually stacked in large bins in grocery stores as shown in Figure 2(a-f). A common problem in grocery stores is that similar items are often stacked next to each other; therefore, items are often misplaced into neighboring bins. Figure 2(a) shows a mix of red and green apples, where it might be difficult for the system to determine which kind of apple is the actual target. Humans can distinguish between groceries without vision to some degree, e.g. by touching and smelling them, but it requires prior knowledge about texture and fragrance of food items.

Moreover, in addition to raw grocery items, there are also items that can only be differentiated with the help of visual information, e.g. milk, juice, and yogurt cartons, see Figure 2(g-i). Such items usually have barcodes, that are readable using the existing assistive devices described above. However, the barcodes are not easily located by visually impaired persons. Thus, an assistive vision device that fully

*This research is funded by the Promobilia foundation and the Swedish e-Science Research Centre.

relies on natural image understanding would be of significant added value for a visually impaired person shopping in a grocery store.

Image recognition models used for this task typically require training images collected in similar environments. However, current benchmark datasets, such as ImageNet [7] and CIFAR-100 [18], do contain images of fruits and vegetables, but are not suitable for this type of assistive application, since the target objects are commonly not presented in this type of natural environments, with occlusion and cluttered backgrounds. To address this issue, we present a novel dataset containing natural images of various raw grocery items and refrigerated products, e.g. milk, juice, and yogurt, taken in grocery stores. As part of our dataset, we collect images taken with single and multiple target objects, from various perspectives, and with noisy backgrounds.

In computer vision, previous studies have shown that model performance can be improved by extending the model to utilize other data sources, e.g. text, audio, in various machine learning tasks [11, 12, 16, 24]. Descriptions of images are rather common to computer vision datasets, e.g. Flickr30k [29], whereas the datasets in [12, 20] includes both descriptions and a reference image with clean background to some objects. Therefore, in addition to the natural images, we have collected iconic images with a single object centered in the image (see Figure 3) and a corresponding product description to each grocery item. In this work, we also demonstrate how we can benefit from using additional information about the natural images by applying the multi-view generative model.

To summarize, the contribution of this paper is a dataset of natural images of raw and refrigerated grocery items, which could be used for evaluating and training image recognition systems to assist visually impaired people in a grocery store. The dataset labels have a hierarchical structure with both coarse- and fine-grained classes (see Figure 1). Moreover, each class also has an iconic image and a product description, which makes the dataset applicable to multimodal learning models. The dataset is described in Section 3.

We provide multiple benchmark results using various deep neural networks, such as Alexnet [19], VGG [34], DenseNet [15], as well as deep generative models, such as VAE [27]. Furthermore, we adapt a multi-view VAE model to make use of the iconic images for each class (Section 4), and show that it improves the classification accuracy given the same model setting (Section 5). Last, we discuss possible future directions for fully using the additional information provided with the dataset and adopt more advanced machine learning methods, such as visual-semantic embeddings, to learn efficient representations of the images.

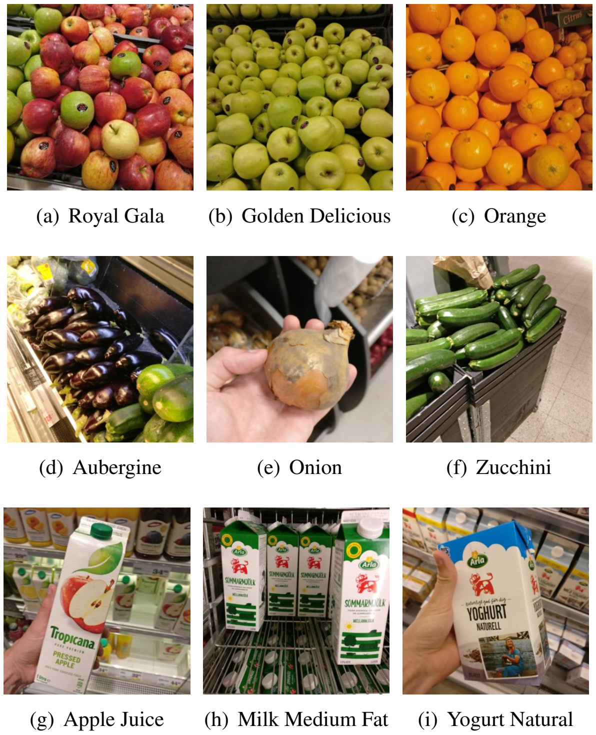
2. Related Work

Many popular image datasets have been collected by downloading images from the web [7, 10, 12, 14, 18, 20, 36, 42, 43]. If the dataset contains a large amount of images, it is convenient to make use of crowdsourcing to get annotations for recognition tasks [7, 18, 21]. For some datasets, the crowdsourcers are also asked to put bounding boxes around the object to be labeled for object detection tasks [10, 12, 42]. In [14] and [18], the target objects are usually centered and takes up most content of the image itself. Another significant characteristic is that web images usually are biased in the sense that they have been taken with the object focus in mind; they have good lighting settings and are typically clean from occlusions, since the collectors have used general search words for the object classes, e.g. *car*, *horse*, or *apple*.

Some datasets include additional information about the images beyond the single class label, e.g. text descriptions of what is present in the image and bounding boxes around objects. These datasets can be used in several different computer vision tasks, such as image classification, object detection, and image segmentation. Structured labeling is another important property of a dataset, which provides flexibility when classifying images. In [12, 20], all of these features exist and moreover they include reference images to each object class, which in [20] is used for labeling multiple categories present in images, while in [12] these images are used for fine-grained recognition. Our dataset includes a reference image, i.e. the iconic image, and a product description for every class, and we have also labeled the grocery items in a structured manner.

Other image datasets of fruits and vegetables for classification purposes are the FIDS30 database [39] and the dataset in [23]. The images in FIDS30 were downloaded from the web and contain background noise as well as single or multiple instances of the object. In [23], all pixels belonging to the object are extracted from the original image, such that all images have white backgrounds with the same brightness condition. There also exist datasets for detecting fruits in orchards for robotic harvesting purposes, which are very challenging since the images contain plenty of background and various lighting conditions, and the targeted fruits are often occluded or of the same color as the background [4, 33].

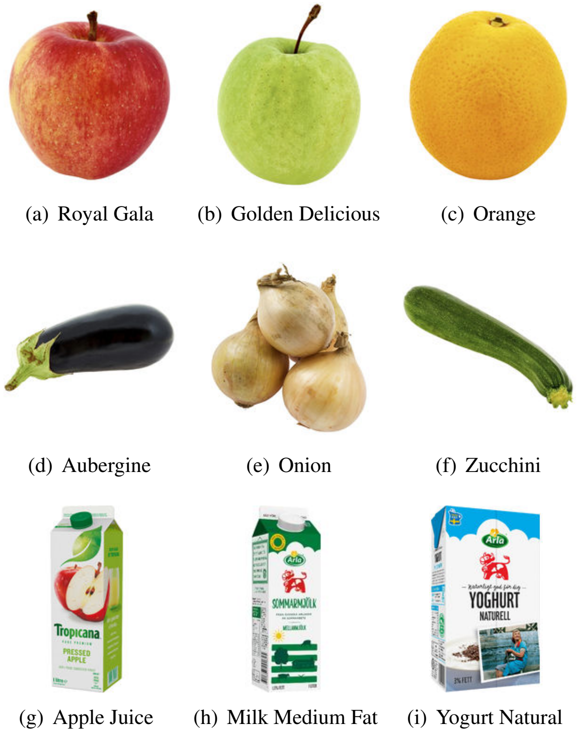
Another dataset that is highly relevant to our application need is presented in [40]. They collected a dataset for training and evaluating the image classifier by extracting images from video recordings of 23 main classes, which are subdivided into 98 classes, of raw grocery items (fruits and vegetables) in different grocery stores. Using this dataset, a mobile application was developed to recognize food products in grocery store environments, which provides the user with details and health recommendations about the item along


(a) Royal Gala (b) Golden Delicious (c) Orange

(d) Aubergine (e) Onion (f) Zucchini

(g) Apple Juice (h) Milk Medium Fat (i) Yogurt Natural

Figure 2. Examples of natural images in our dataset, where each image have been taken inside a grocery store. Image examples of fruits, vegetables and refrigerated products are presented in each row respectively.


(a) Royal Gala (b) Golden Delicious (c) Orange

(d) Aubergine (e) Onion (f) Zucchini

(g) Apple Juice (h) Milk Medium Fat (i) Yogurt Natural

Figure 3. Examples of iconic images downloaded from a grocery shopping website, which corresponds to the target items in the images in Figure 2.

with other proposals of similar food items. For each class, there exists a product description with nutrition values to assist the user in shopping scenarios. The main difference between this work and our dataset is firstly the clean iconic images (visual labels) for each class in our dataset, and secondly that we have also collected images of refrigerated items, such as dairy and juice containers, where visual information is required to distinguish between the products.

3. Our Dataset

We have collected images from fruit and vegetable sections and refrigerated sections with dairy and juice products in 18 different grocery stores. The dataset consists of 5125 images from 81 fine-grained classes, where the number of images in each class range from 30 to 138. Figure 4 displays a histogram over the number of images per class. As illustrated in Figure 1, the class structure is hierarchical, and there are 46 coarse-grained classes. Figure 2 shows examples of the collected natural images. For each fine-grained class, we have downloaded an iconic image of the item and also a product description including origin country, an appreciated weight and nutrient values of the item from a grocery store website. Some examples of downloaded iconic images can be seen in Figure 3.

Our aim has been to collect the natural images under the same condition as they would be as part of an assistive application on a mobile phone. All images have been taken with a 16-megapixel Android smartphone camera from different distances and angles. Occasionally, the images include other items in the background or even items that have been misplaced in the wrong shelf along with the targeted item. It is important that image classifiers that are used for assisting devices are capable of performing well with such noise since these are typical settings in a grocery store environments. The lighting conditions in the images can also vary depending on where the items are located in the store. Sometimes the images are taken while the photographer is holding the item in the hand. This is often the case for refrigerated products since these containers are usually stacked compactly in the refrigerators. For these images, we have consciously varied the position of the object, such that the item is not always centered in the image or present in its entirety.

We also split the data into a training set and test set based on the application need. Since the images have been taken in several different stores at specific days and time stamps, parts of the data will have similar lighting conditions and backgrounds for each photo occasion. To remove any such

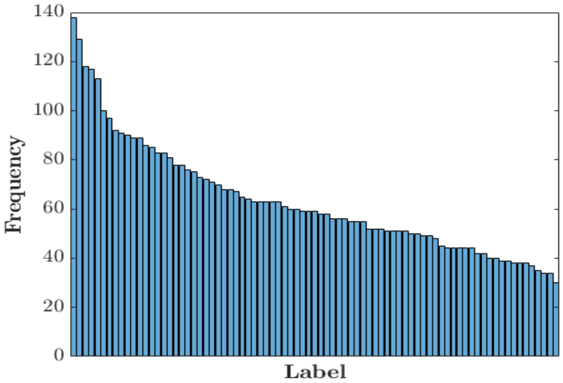


Figure 4. Histogram over the number of images in each class in the dataset.

biasing correlations, all images of a certain class taken at a certain store are assigned to either the test set or training set. Moreover, we balance the class sizes to as large extent as possible in both the training and test set. After the partitioning, the training and test set contains 2640 and 2485 images respectively. Predefining a training and test set also makes it easier for other users to compare their results to the evaluations in this paper.

The task is to classify natural images using mobile devices to aid visually impaired people. The additional information such as the hierarchical structure of the class labels, iconic images, and product descriptions can be used to improve the performance of the computer vision system. Every class label is associated with a product description. Thus, the product description itself can be part of the output for visually impaired persons as they may not be able to read what is printed on a carton box or a label tag on a fruit bin in the store.

The dataset is intended for research purposes and we are open to contributions with more images and new suitable classes. Our dataset is available at `https://github.com/marcusklasson/GroceryStoreDataset`. Detailed instructions on how to contribute to the dataset can be found on our dataset webpage.

4. Classification Methods

We here describe the classification methods and approaches that we have used to provide benchmark results to the dataset. We apply both deterministic deep neural networks as well as a deep generative model used for representation learning to the natural images that we have collected. Furthermore, we utilize the additional information – iconic images – from our dataset with a multi-view deep generative model. This model can utilize different data sources and obtain superior representation quality as well as high interpretability. For a fair evaluation, we use a linear classifier with the learned representation from the different methods.

Deep Neural Networks. CNNs have been the state-of-the-art models in image classification ever since AlexNet [19] achieved the best classification accuracy in ILSVRC in 2012. However, in general, computer vision models require lots of labeled data to achieve satisfactory performance, which has resulted in interest for adapting CNNs that have already been trained on a large amount of training data to other image datasets. When adapting pretrained CNNs to new datasets, we can either use it directly as a feature extractor, a.k.a use the off-the-shelf features, [9, 31], or fine-tune it [13, 26, 28, 44, 47]. Using off-the-shelf features, we need to specify which feature representation we should extract from the network and use these for training a new classifier. Fine-tuning a CNN involves adjusting the pretrained model parameters, such that the network can e.g. classify images from a dataset different from what the CNN was trained on before. We can either choose to fine-tune the whole network or select some layer parameters to adjust while keeping the others fixed. One important factor on deciding which approach to choose is the size of the new dataset and how similar the new dataset is to the dataset which the CNN was previously trained on. A rule of thumb here is that the closer the features are to the classification layer, the features become more specific to the training data and task [44].

Using off-the-shelf CNN features and fine-tuned CNNs have been successfully applied in [9, 31] and [13, 26, 47] respectively. In [9, 31], it is shown that the pretrained features have sufficient representational power to generalize well to other visual recognition tasks with simple linear classifiers, such as Support Vector Machines (SVMs), without fine-tuning the parameters of the CNN to the new task. In [13, 47], all CNN parameters are fine-tuned, whereas in [26] the pretrained CNN layer parameters are kept fixed and only an adaptation layer of two fully connected layers are trained on the new task. The results from these works motivate why we should evaluate our dataset on fine-tuned CNNs or linear classifiers trained on off-the-shelf feature representations instead of training an image recognition model from scratch.

Variational Autoencoders with only natural images. Deep generative models, e.g. the variational autoencoder (VAE) [27, 32, 45], have become widely used in the machine learning community thanks to their generative nature. We thus use VAEs for representation learning as the second benchmarking method. For efficiency, we use low-level pretrained features from a CNN as inputs to the VAE.

The latent representations from VAEs are encodings of the underlying factors for how the data are generated. VAEs belongs to the family of latent variable models, which commonly has the form $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, where $p(\mathbf{z})$ is a prior distribution over the latent variables $\mathbf{z}$ and $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the likelihood over the data $\mathbf{x}$ given $\mathbf{z}$. The prior distribution is often assumed to be Gaussian, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, \mathbf{0}, \mathbf{I})$, whereas the likelihood distribution depends on the values of $\mathbf{x}$. The likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is referred to as a decoder represented as a neural network parameterized by $\boldsymbol{\theta}$. An encoder network $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ parameterized by $\boldsymbol{\phi}$ is introduced as an approximation of the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, which is intractable since it requires computing the integral $p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$. When the prior distribution is a Gaussian, the approximate posterior is also modeled as a Gaussian, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \,|\, \boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}^2(\mathbf{x}) \odot \mathbf{I})$, with some mean $\boldsymbol{\mu}(\mathbf{x})$ and variance $\boldsymbol{\sigma}^2(\mathbf{x})$ computed by the encoder network. The goal is to maximize the marginal log-likelihood by defining a lower bound using $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$:

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) =& \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \right] \\ &- D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})).\end{aligned} \tag{1}$$

The last term is the Kullback-Leibler (KL) divergence of the approximate posterior from the true posterior. The lower bound $\mathcal{L}$ is called the evidence lower bound (ELBO) and can be optimized with stochastic gradient descent via backpropagation [8, 27]. VAE is a probabilistic framework. Many extensions such as utilizing structured priors[5] or using continual learning [25] have been explored. In the following method, we describe how to make use of the iconic images while retaining the unsupervised learning setting in VAEs.

Utilizing iconic images with multi-view VAEs. Utilizing extra information has shown to be useful in many applications with various model designs [5, 37, 38, 41, 46]. For computer vision tasks, natural language is the most commonly used modality to aid the visual representation learning. However, the consistency of the language and visual embeddings has no guarantee. As an example with our dataset, the product description of a Royal Gala apple explains the appearance of a red apple. But if the description is represented with word embeddings, e.g. word2vec [22], the word 'royal' will probably be more similar to the words 'king' and 'queen' than 'apple'. Therefore, if available, additional visual information about objects might be more beneficial for learning meaningful representations instead of text. In this work, with our collected dataset, we propose to utilize the iconic images for the representation learning of natural images using a multi-view VAE. Since the natural images can include background noise and grocery items different from the targeted one, the role of the iconic image will be to guide the model to which features that are of interest in the natural image.

The VAE can be extended to modeling multiple views of data, where a latent variable $\mathbf{z}$ is assumed to have generated the views [37, 41]. Considering two views $\mathbf{x}$ and $\mathbf{y}$, the joint distribution over the paired random variables ($\mathbf{x}$, $\mathbf{y}$) and latent variable $\mathbf{z}$ can be written as $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{z})p_{\boldsymbol{\theta}^{(1)}}(\mathbf{x} \,|\, \mathbf{z})p_{\boldsymbol{\theta}^{(2)}}(\mathbf{y} \,|\, \mathbf{z})$, where both $p_{\boldsymbol{\theta}^{(1)}}(\mathbf{x} \,|\, \mathbf{z})$ and $p_{\boldsymbol{\theta}^{(2)}}(\mathbf{y} \,|\, \mathbf{z})$ are represented as neural networks with parameters $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$. Assuming that the latent variable $\mathbf{z}$ can reconstruct both $\mathbf{x}$ and $\mathbf{y}$ when only $\mathbf{x}$ is encoded into $\mathbf{z}$ by the encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, then the ELBO is written as

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}, \mathbf{y}) \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}^{(1)}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}^{(2)}}(\mathbf{y}|\mathbf{z}) \right] \\ &\quad - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})).\end{aligned} \tag{2}$$

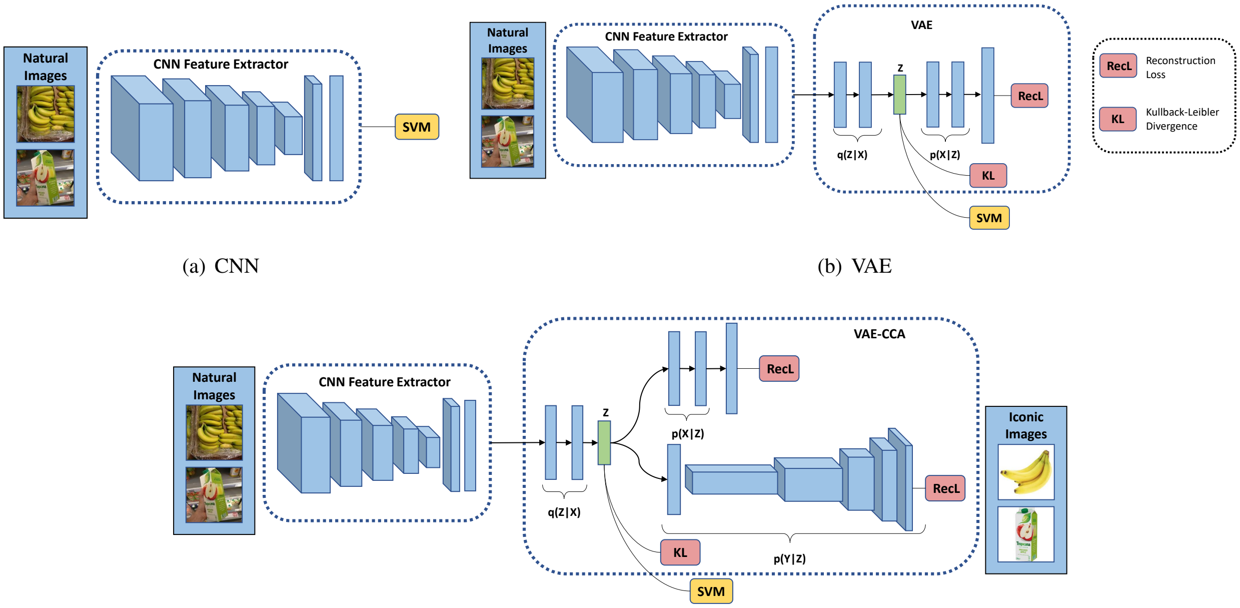
This model is referred to as variational autoencoder canonical correlation analysis (VAE-CCA) and was introduced in [41]. The main motivation for using VAE-CCA is that the latent representations need to contain information about reconstructing both natural and iconic images. The main motivation for using VAE-CCA is that the latent representation needs to preserve information about how both the natural and iconic images are reconstructed. This also allows us to produce iconic images from new natural images to enhance the interpretability of the latent representation of VAE-CCA (see Section 5) [37].

5. Experimental Results

We apply the three different types of models described in Section 4 to our dataset and evaluate their performance. The natural images are propagated through a CNN pretrained on ImageNet to extract feature vectors. We experiment with both the off-the-shelf features as well as fine-tuning the CNN. When using off-the-shelf features, we simply extract feature vectors and train an SVM on those. For the fine-tuned CNN, we report both results from the softmax classifier used in the actual fine-tuning procedure and training an SVM with extracted fine-tuned feature vectors.

These extracted feature vectors are also used for VAE and VAE-CCA which makes further compression. We perform classification for those VAE based models by training a classifier, e.g. an SVM, on the data encoded into the latent representation. We use this classification approach for both VAE and VAE-CCA. In all classification experiments, except when we fine-tune the CNN, we use a linear SVM trained with the one-vs-one approach as in [31].

We experiment with three different pretrained CNN architectures, namely AlexNet [19], VGG16 [34] and DenseNet-169 [15]. For AlexNet and VGG16, we extract feature vectors of size 4096 from the two last fully connected (FC) layers before the classification layer. The features from the n^{th} hidden layer are denoted as AlexNet$_n$ and VGG16$_n$. As an example, the last hidden FC layer in



(a) CNN

(b) VAE

(c) VAE-CCA

Figure 5. The architectures for the classification methods described in Section 4. In this paper, we use either a pretrained AlexNet, VGG16 or DenseNet-169 as the CNN feature extractor, but it may be replaced with any CNN architecture. Note that the pretrained CNN can be fine-tuned. The encoder and decoder of the VAE in 5(b) consist of two fully-connected layers. VAE-CCA in 5(c) uses the DCGAN architecture as an iconic image decoder and the same encoder and feature vector decoder as the VAE.

AlexNet is denoted as AlexNet$_7$, the input of which is output from AlexNet$_6$. For DenseNet-169, we extract the features of size 1664 from the average pooling layer before its classification layer.

5.1. Experimental Setups

The following setups were used in the experiments:

Setup 1. Train an SVM on extracted off-the-shelf features from a pretrained CNN, which is denoted as SVM in the results. We also fine-tune the CNN by replacing the final layer with a new softmax layer and denote these results as Fine-tune. We denote training an SVM on extracted finetuned feature vectors as SVM-ft.

Setup 2. Extract feature vectors with a pretrained CNN of the natural images and learn a latent representation $\mathbf{z}$ with a VAE. Then the data is encoded into the latent space and we train an SVM with these latent representations, which used for classification. We denote the results as VAE+SVM when using off-the-shelf feature vectors, whereas using the fine-tuned feature vectors are denoted as VAE+SVM-ft. In all experiments with the VAE, we used the architecture from [35], i.e. the latent layer having 200 hidden units and both encoder and decoder consisting of two FC layers with 1,000 hidden units each.

Setup 3. Each natural image is paired with its corresponding iconic image. We train VAE-CCA similarly as the VAE, but instead, we learn a joint latent representation that is used to reconstruct the extracted feature vectors $\mathbf{x}$ and the iconic images $\mathbf{y}$. The classification is performed with the same steps as in Setup 2 and denotes the results similarly with VAE-CCA+SVM and VAE-CCA+SVM-ft. Our VAE-CCA model takes the feature vectors $\mathbf{x}$ as input and encodes them into a latent layer with 200 hidden units. The encoder and the feature vector decoder uses the same architecture, i.e. two FC layers with 512 hidden units, whereas the iconic image decoder uses the DCGAN [30] architecture.

Figure 5 displays the three experimental setups described above. We report both fine-grained and coarse-grained classification results with an SVM in Table 1 and 2 respectively. In Table 3, we report the fine-grained classification results from fine-tuned CNNs.

When fine-tuning the CNNs, we replace the final layer with a softmax layer applicable to our dataset with randomly initialized weights drawn from a Gaussian with zero mean and standard deviation 0.01 [47]. For AlexNet and VGG16, we fine-tune the networks for 30 epochs with two different learning rates, 0.01 for the new classification layer and 0.001 for the pretrained layers. Both learning rates are reduced by half after every fifth epoch. The DenseNet-169 is fine-tuned for 30 epochs with momentum of 0.9 and an

Table 1. Fine-grained classification (81 classes) accuracies with the methods described in Section 5.1. Each row displays from which network architecture and layer that we extracted the feature vectors of the natural images. The columns show the result from the classifiers that we used (see Section 5.1).

	SVM	SVM-ft	VAE+SVM	VAE+SVM-ft	VAE-CCA+SVM	VAE-CCA+SVM-ft
AlexNet$_6$	69.2	72.6	65.6	70.7	67.8	71.5
AlexNet$_7$	65.0	70.7	63.0	68.7	65.0	70.9
VGG16$_6$	62.1	73.3	57.5	71.9	60.7	73.0
VGG16$_7$	57.3	71.7	56.8	67.8	56.8	71.3
DenseNet-169	72.5	85.0	65.4	79.1	72.6	80.4

Table 2. Coarse-grained classification (46 classes) accuracies with an SVM for the methods described in Section 5.1 that uses off-the-shelf feature representations. Each row displays from network architecture and layer that we extracted the feature vectors of the natural images and the columns show the result for the classification methods.

	SVM	VAE+SVM	VAE-CCA+SVM
AlexNet$_6$	78.0	74.2	76.4
AlexNet$_7$	75.4	73.2	74.4
VGG16$_6$	76.6	74.2	74.9
VGG16$_7$	72.8	71.7	72.3
DenseNet-169	85.2	79.5	82.0

Table 3. Fine-grained classification accuracies from fine-tuned CNNs pretrained on ImageNet, where the column shows which architecture that has been fine-tuned. A standard softmax layer is used as the last classification layer.

	AlexNet	VGG16	DenseNet-169
Fine-tune	69.3	73.8	84.0

initial learning rate of 0.001, which decays with 10^{-6} after each epoch. We report the classification results from the softmax activation after the fine-tuned classification layer. We also report classification results from an SVM trained with feature representations from a fine-tuned CNN, which are extracted from FC6 and FC7 of the AlexNet and VGG16 and from the last average pooling layer in DenseNet-169.

The VAE and VAE-CCA models are trained for 50 epochs with Adam [17] for optimizing the ELBOs in Equation 1 and 2 respectively. We use a constant learning rate of 0.0001 and set the minibatch size to 64. The extracted feature vectors are rescaled with standardization before training the VAE and VAE-CCA models to stabilize the learning.

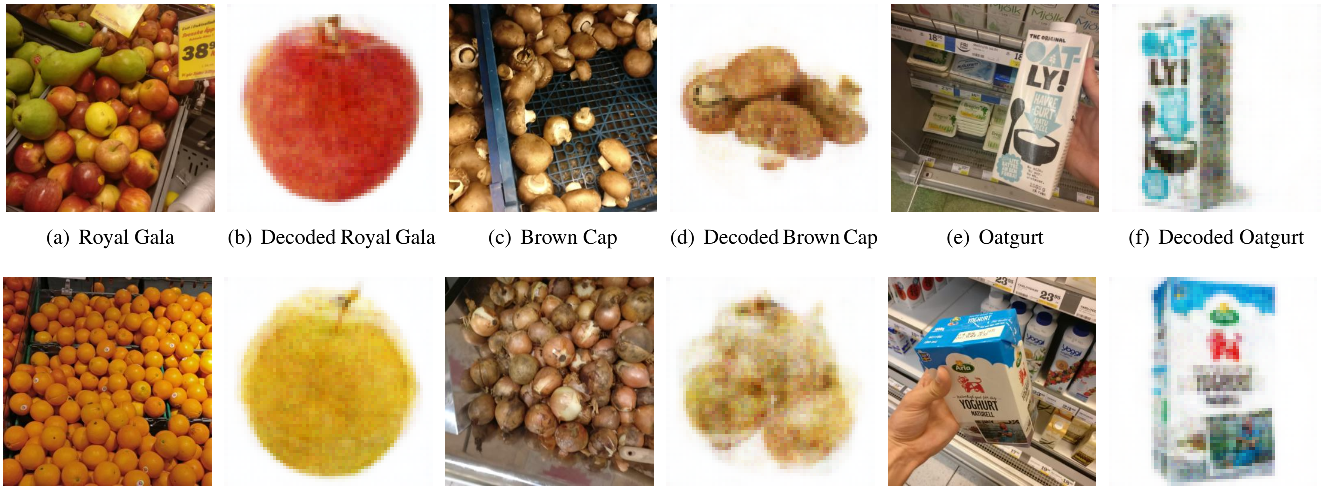
5.2. Results

The fine-grained classification results for all methods using an SVM as classifier are shown in Table 1. We also provide coarse-grained classification results for some of the methods in Table 2 to demonstrate the possibility of hierarchical evaluation that our labeling of the data provides (see Figure 1). The accuracies in the coarse-grained classification are naturally higher than the accuracies in the corresponding columns in Table 1. Table 3 shows fine-grained classification accuracies from a softmax classifier in the fine-tuned CNNs. We note that fine-tuning the networks gives consistently better results than training an SVM on off-the-shelf features (see Table 1).

Fine-tuning the entire network results improves the classification performance consistently for each method in Table 1. The performance is clearly enhanced for features extracted from fine-tuned VGG16 and DenseNet-169, which improves the classification accuracy by 10% in most cases for SVM-ft, VAE+SVM-ft, and VAE-CCA+SVM-ft. For AlexNet and VGG16, we see that the performance drops when extracting the features from layer FC7 instead of FC6. The reason might be that the off-the-shelf features in FC7 are more difficult to transfer to other datasets since the weights are biased towards classifying objects in the ImageNet database. The performance drops also when we use fine-tuned features, which could be due to the small learning rate we use for the pretrained layers, such that the later layers are still ImageNet-specific. We might circumvent this drop by increasing the learning rate for the later pretrained layers and keeping the learning rate for earlier layers small.

The VAE-CCA model achieves mostly higher classification accuracies than the VAE model in both Table 1 and 2. This indicates that the latent representation separates the classes more distinctly than the VAE by jointly learning to reconstruct the extracted feature vectors and iconic images. However, further compressing the feature vectors with VAE and VAE-CCA will lower the classification accuracy compared to applying the feature vectors to a classifier directly. Since both VAE and VAE-CCA compresses the feature vectors into the latent representation, there is a risk of losing information about the natural images. We might receive better performance by increasing the dimension of the latent representation at the expense of speed in both training and classification.

In Figure 6, we show results from the iconic image decoder $p_{\boldsymbol{\theta}^{(2)}}(\mathbf{y} \mid \mathbf{z})$ when translating natural images from



(a) Royal Gala (b) Decoded Royal Gala (c) Brown Cap (d) Decoded Brown Cap (e) Oatgurt (f) Decoded Oatgurt

(g) Orange (h) Decoded Orange (i) Onion (j) Decoded Onion (k) Yogurt (l) Decoded Yogurt

Figure 6. Examples of natural images in the test set that have been decoded into product iconic images by the iconic image decoder. This result is obtained with the fine-tuned DenseNet-169 features, which corresponds to VAE-CCA+SVM-ft in Table 1. Subfigures (a), (c), (e), (g), (i) and (k) show the example input image from the test set, and Subfigures (b), (d), (f), (h), (j) and (l) show the decoded iconic image from the decoder $p_{\boldsymbol{\theta}^{(2)}}(\mathbf{y} \mid \mathbf{z})$ using VAE-CCA model as in Figure 5(c).

the test set into iconic images with VAE-CCA and a fine-tuned DenseNet-169 as feature extractor. Such visualization can demonstrate the quality of the representation using the model, as well as enhancing the interpretability of the method. Using VAE-CCA in the proposed manner, we see that with challenging natural images, the model is still able to learn an effective representation which can be decoded to the correct iconic image. For example, some pears have been misplaced in the bin for Royal Gala apples in Figure 6(a), but still the image decoder manages to decode a blurry red apple seen in Figure 6(b). In Figure 6(h), a mix of an orange and an apple are decoded from a bin of oranges in Figure 6(g), which indicates these fruits are encoded close to each other in the learned latent space. Even if Figure 6(e) includes much of the background, the iconic image decoder is still able to reconstruct the iconic images accurately in Figure 6(f), which illustrates that the latent representation is able to explain away irrelevant information in the natural image and preserved the features of the oatgurt package. Thus, using VAE-CCA with iconic images as the second view not only advances the classification accuracy but also provides us with the means to understand the model.

6. Conclusions

This paper presents a dataset of images of various raw and packaged grocery items, such as fruits, vegetables, and dairy and juice products. We have used a structured labeling of the items, such that grocery items can be grouped into more general (coarse-grained) classes and also divided into fine-grained classes. For each class, we have a clean iconic image and a text description of the item, which can be used for adding visual and semantic information about the items in the modeling. The intended use of this dataset is to train and benchmark assistive systems for visually impaired people when they shop in a grocery store. Such a system would complement existing visual assistive technology, which is confined to grocery items with barcodes. We also present preliminary benchmark results for the dataset on the task of image classification.

We make the dataset publicly available for research purposes at `https://github.com/marcusklasson/GroceryStoreDataset`. Additionally, we will both continue collecting natural images, as well as ask for public contributions of natural images in shopping scenarios to enlarge our dataset.

For future research, we will advance our model design to utilize the structured nature of our labels. Additionally, we will design a model that use the product description of the objects in addition to the iconic images. One immediate next step is to extend the current VAE-CCA model to three views, where the third view is the description of the product.

References

[1] Aipoly Vision app. https://www.aipoly.com/. Accessed on 2018-02-28.

[2] Microsoft Seeing AI app. https://www.microsoft.com/en-us/seeing-ai/. Accessed on 2018-02-22.

[3] Orcam. https://www.orcam.com/en/. Accessed on 2018-02-28.

[4] S. Bargoti and J. P. Underwood. Deep fruit detection in orchards. In IEEE International Conference on Robotics and Automation, 2017.

[5] J. Butepage, J. He, C. Zhang, L. Sigal, and S. Mandt. Informed priors for deep representation learning. In Symposium on Advances in Approximate Bayesian Inference, 2018.

[6] S. Caraiman, A. Morar, M. Owczarek, A. Burlacu, D. Rzeszotarski, N. Botezatu, P. Herghelegiu, F. Moldoveanu, P. Strumillo, and A. Moldoveanu. Computer vision for the visually impaired: the sound of vision system. In IEEE International Conference on Computer Vision Workshops, 2017.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[8] C. Doersch. Tutorial on variational autoencoders. CoRR, abs/1606.05908, 2016.

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International Conference on Machine Learning, 2014.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, Jun 2010.

[11] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In Advances in Neural Information Processing Systems, 2013.

[12] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, and L. Fei-Fei. Fine-grained car detection for visual census estimation. In AAAI Conference on Artificial Intelligence, 2017.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, June 2014.

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015.

[18] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, 2012.

[20] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In European Conference on Computer Vision, 2014.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In International Conference on Computer Vision, 2015.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, 2013.

[23] H. Muresan and M. Oltean. Fruit recognition from images using deep learning. Technical report, Babes-Bolyai University, 2017.

[24] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In International Conference on Machine Learning, 2011.

[25] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. In International Conference on Learning Representations, 2018.

[26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In International Conference on Learning Representations, 2014.

[28] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, Oct 2010.

[29] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In IEEE International Conference on Computer Vision, 2015.

[30] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[31] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014.

[32] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In International Conference on Machine Learning, 2014.

[33] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool. Deepfruits: A fruit detection system using deep neural networks. Sensors, 16(8):1222, 2016.

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[35] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In Advances in Neural Information Processing Systems. 2015.

[36] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In IEEE

Conference on Computer Vision and Pattern Recognition, 2016.

[37] R. Vedantam, I. Fischer, J. Huang, and K. Murphy. Generative models of visually grounded imagination. In International Conference on Learning Representations, 2018.

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In IEEE Conference on Computer Vision and Pattern Recognition, pages 3156–3164, 2015.

[39] Škrjanec Marko. Automatic fruit recognition using computer vision. (Mentor: Matej Kristan), Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2013. FIDS30 dataset was accessed 2018-02-24 at `http://www.vicos.si/Downloads/FIDS30`.

[40] G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, P. Luley, H. Bischof, M. Lindschinger, I. Schmid, and L. Paletta. Mango - mobile augmented reality with functional eating guidance and food awareness. In International Workshop on Multimedia Assisted Dietary Management, 2015.

[41] W. Wang, H. Lee, and K. Livescu. Deep variational canonical correlation analysis. CoRR, abs/1610.03454, 2016.

[42] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[43] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems, 2014.

[45] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt. Advances in variational inference. arXiv preprint arXiv:1711.05597, 2017.

[46] C. Zhang, H. Kjellström, and C. H. Ek. Inter-battery topic representation learning. In European Conference on Computer Vision, pages 210–226. Springer, 2016.

[47] N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In European Conference on Computer Vision, 2014.