A Combined Deep Learning and Semantic Embedding Model for Image Classification

Yohei Nishimura
University of Wisconsin - Madison
ynishimura@wisc.edu

Xin Yu
University of Wisconsin - Madison
xyu273@wisc.edu

Jackson Hellmers
University of Wisconsin - Madison
jhellmers@wisc.edu

Neal Desai
University of Wisconsin - Madison
nbdesai2@wisc.edu

Abstract

One of the challenges encountered with computer vision classification problems is that models generally fail to generalize well to a large number of class categories. To achieve improved generalization, we created a model that combines both visual and language components together. We leverage semantic information to improve the performance of the base image classification model so that it is able to make more reasonable predictions in a "zero-shot" context where the model is evaluated using images with object labels not found in the original training dataset. Though we find that the Softmax CNN visual model achieves almost a threefold improvement compared to the control baseline, the combined visual and semantic model achieves superior results providing a sixfold improvement compared to the control and almost double the precision of the Softmax CNN model.

1 Introduction and Motivation

Image classification methods are often built on the assumption that each image can be assigned to one of a relatively small number of mutually-exclusive classes. While standard, this assumption entails a significant reduction in modeling power: real-life visual environments are often ambiguous, may contain many classes outside those used for a given model, and may have varying degrees of semantic similarity between classes. Models employing this assumption cannot capture such subtleties and tend to generalize poorly in more complex contexts.

One such context is that of zero-shot learning, in which the model is evaluated on images outside the list of training classes. This arises naturally from the problem of image classification with a large number of classes, which can require an impractical amount of labeled training data for accurate classification. A model that can generalize to the zero-shot context will allow classification over a large number of classes with a relatively small dataset, and it is this generalization that is the primary motivation of the experiments conducted in this paper.

1.1 Background and related work

The problem of zero-shot learning was first introduced by Larochelle et al. [1] as an extension of previous work on one-shot learning [4] and on zero-shot learning in specific contexts (such as the "cold start" problem for recommender systems [8]). The core motivation behind zero-shot learning is training with insufficiently labeled data, which is related to the broader field of semi-supervised learning. Since its introduction, there have been several attempts that showed good performance on the problem, as surveyed in [12][11]. These attempts largely rely on adding semantic information to

the model to guide zero-shot decision making, with most using an external source of semantic data such as WordNet [5] hierarchies.

Socher et al. [9] were the first to approach the zero-shot problem by combining visual and language models. By using a language model to learn semantic structure directly from unlabeled text, their approach forgoes the need for labeled semantic data in addition to image data. Following this, Frome et al. [2] introduced DeViSE, which uses a unified visual-semantic model to provide better performance and greater scalability. The DeViSE method is the main focus of our project.

DeViSE models obtain semantic information by combining a CNN visual model with a word2vec-based [3] language model using word embeddings. The two models are first trained separately to generate embeddings and to provide a warm start for the combined model. The visual model is then altered to predict embeddings rather than discrete labels, with the language model providing input for the loss function, effectively converting the image classification task into an image regression task. Finally, the predicted word embeddings are mapped back to labels via a ranked similarity approach. Figure 1 (taken from the original paper) provides an overview of the structure of DeViSE models.

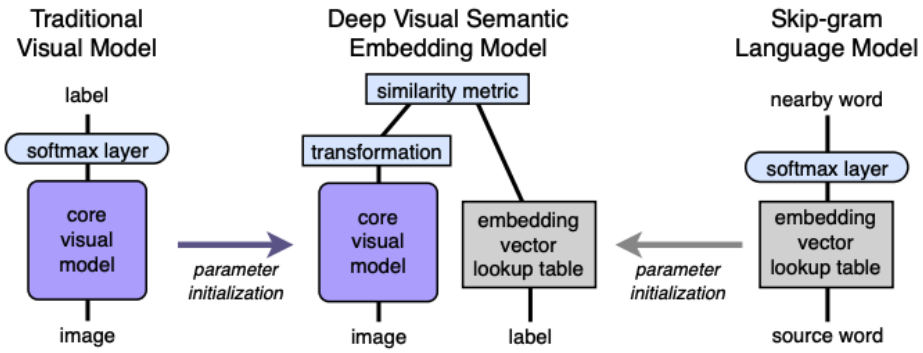

Figure 1: Visualization of DeViSE network structure (from Frome et al.)

Frome et al. found a general improvement in precision of 3% compared to the softmax baseline. The most important result, however, was observed in the zero-shot context, which produced an improvement of 6%-10% over the baseline.

1.2 Hypothesis

Based on prior research, we would expect the following: First, the combined model approach will produce better prediction results than a plain convolutional neural network model (CNN) using the Tiny ImageNet dataset. Second, we will be able to achieve more semantically reasonable predictions when evaluating zero-shot learning results.

2 Experimental Approach

We use a similar approach to that described in Frome et al. [2], but drafted the model from scratch using PyTorch [6] to implement the visual model and Gensim [7] for acquiring word embeddings. Unlike Frome et al., we use a using a different dataset, namely CIFAR100, to evaluate zero-shot learning for our visual semantic model. [1]

2.1 Visual Classification Model

The visual model used in DeViSE [2] takes advantage of the AlexNet architecture, training on the entirety of the ImageNet dataset which contains full-resolution images with 1000 unique labels. We found the magnitude of the ImageNet dataset to be daunting so chose to work with a subset of ImageNet titled Tiny-ImageNet. The images in this dataset are narrowed to only include 200 unique

[1] The CIFAR100 dataset has 100 labels, while Tiny ImageNet has 200 labels. Also, there are ten labels that are exactly the same in the two sets of labels.

labels and all images are cropped to a uniform 64x64 pixels. Due to the reduced image sizes, we found AlexNet to be too deep of a model for useful classification. Instead we turned to prior works using shallower CNN models for Tiny-ImageNet classification [13].

2.2 Language Model

To obtain word embeddings for image class labels, we downloaded a word2vec model [3] trained over the Wikipedia corpus. The model encodes words into a length 50 vector where semantically similar words will be classified by vectors that are close together in euclidean space.

It is important to note that the 50 parameter word2vec model we chose was the smallest model we could find, with other available models using anywhere from 100 to 1000 parameters. Since training the proposed DeViSE model requires transforming data back-and-forth between the label space and semantic, increasing the dimensionality of the language model greatly increases the cost and complexity of training.

2.3 Combined Visual-Semantic Model

The first step in creating the combined visual-semantic model was to translate Tiny-ImageNet class labels into word embeddings from the language model. This was not a one-to-one mapping, as each label comprises a set of (possibly multi-word) synonyms whereas each embedding corresponds to a specific word. For multi-word labels, we averaged the embeddings for each word to create a hybrid embedding. For example, the vector for the label 'fire ant' would equal the mean of the vectors for 'fire' and 'ant'.

To allow the CNN to predict embeddings, we replaced the softmax output layer with a single fully-connected layer in the size of embedding vectors (50 for the pre-trained model). We evaluated two candidate loss functions: the first

$$L(\hat{x}, x) = 1 - \cos(\hat{x}, x)$$

attempted to only minimize the cosine distance between the predicted and true label embeddings, while the second

$$L(\hat{x}, x) = 1 - \cos(\hat{x}, x) + \max(0, \cos(\hat{x}, x') - margin)$$

used a contrasting approach to both minimize distance with true embeddings and maximize distance with unrelated ones. (In the above, $\hat{x}$ is the predicted embedding, x is the given embedding of the true label, and x' is the embedding for a random incorrect class label.) We found the second candidate matched closer to the recommendation in the original paper and produced significantly better training performance than the first.

Finally, we used ranked lookup based on similarity to map predictions back to class labels. For each prediction, we checked if the correct label can be found in the five closest embeddings based on cosine distance. This is the same as the "hit @ k" metric in the original paper with k = 5.

3 Results and Discussion

3.1 Standalone Visual Model

While initially training the CNN we observed an excessive degree of overfitting (shown in Figure 2). We were able to reach accuracies of nearly 40% on training data but our testing accuracy leveled out at only 20%. To combat the overfitting we first introduced dropout layers between the convolution and fully connected layers with probability $p = 0.5$. This addition to the model lessened the severity of overfitting but did not completely eliminate it (see Figure 3). In a final effort, we implemented a data augmentation pipeline to apply a series of random transformations to each training image before it was fed into the model. These augmentations combined with the dropout layers finally provided us with a model that displayed similar learning curves (shown in Figure 4) for the entirety of training. When combined with data augmentations we found using dropout with probability $p = 0.4$ to perform better, as $p = 0.5$ was too large to allow meaningful learning.

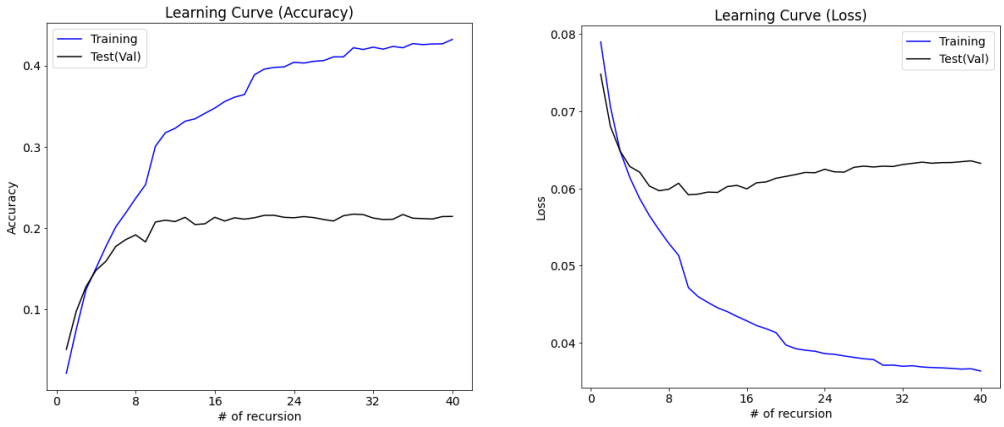

Figure 2: Learning Curves for baseline CNN model without overfitting prevention.

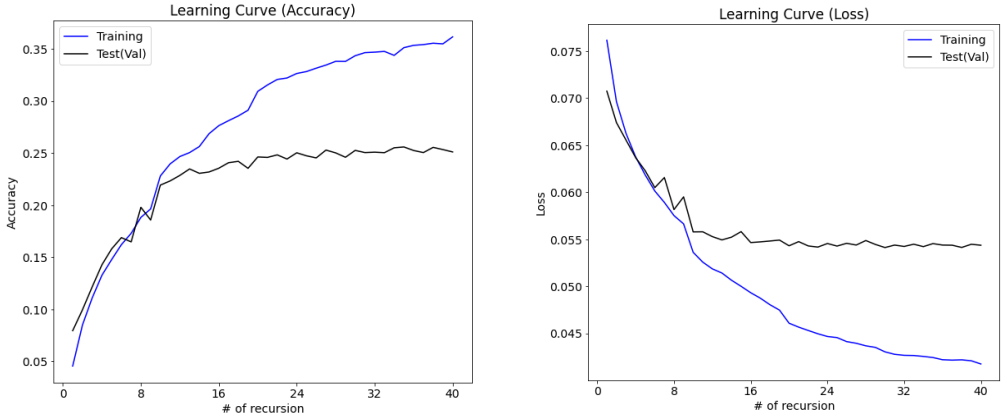

Figure 3: Learning Curves for baseline CNN model with dropout layers.

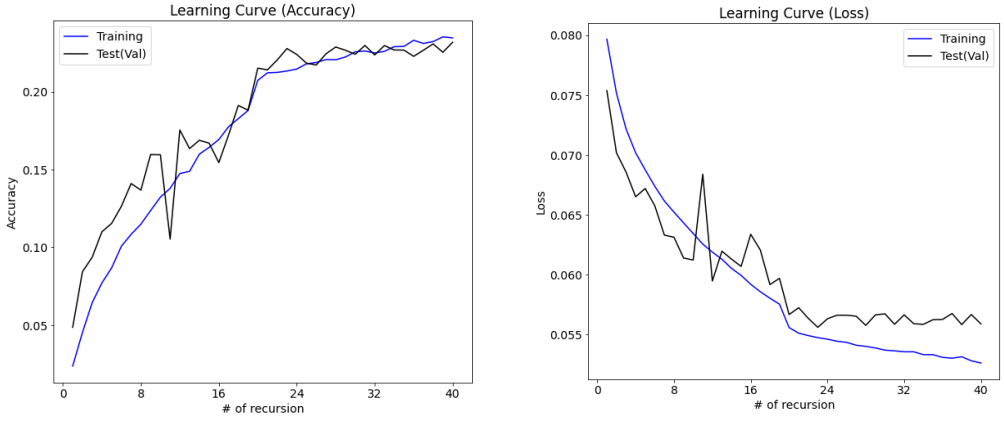

Figure 4: Learning Curves for baseline CNN model with dropout layers and data augmentation.

3.2 Combined Visual-Semantic Model

We were unable to experiment with a large variety of hyperparameter adjustments due to the excessive resource usage and training time of this project. The hyperparameters we mainly focused on were the loss margin and learning rate.

Though we initially set the margin value to 1, the value was found to be too large as all misclassified points fell within the margin and resulted in minimal update progress during back-propagation. When the margin was adjusted to 0, it was found to lead to sub-optimal performance, as the margin was too strict to converge. We eventually settled on a margin of 0.1 as was used in the DeViSE paper and found it to strike a proper balance for convergence.

Learning rate was also an important parameter for us to tune. Since training was a time consuming process, choosing too small of a value resulted in the model making very little progress during each epoch. In order to speed up convergence, we chose to use the Adam optimizer instead of using the proposed SGD optimization method. Additionally, learning rate decay was implemented to allow us the convenience of choosing a larger initial learning rate that slowly decreases to help with finer model tuning in later training epochs.

During training, accuracy was measured using the "hit @ k" approach outlined in section 2.3. We found that for various values of k, the model architecture with the highest accuracy was not a constant. For lower values of k such as 1 or 3, the softmax CNN ended training with a greater test accuracy. For larger values of k including 5 and 10 the combined model was superior. Shown below in Figure 5 are the learning curves for the final model with accuracy calculated using k=5.

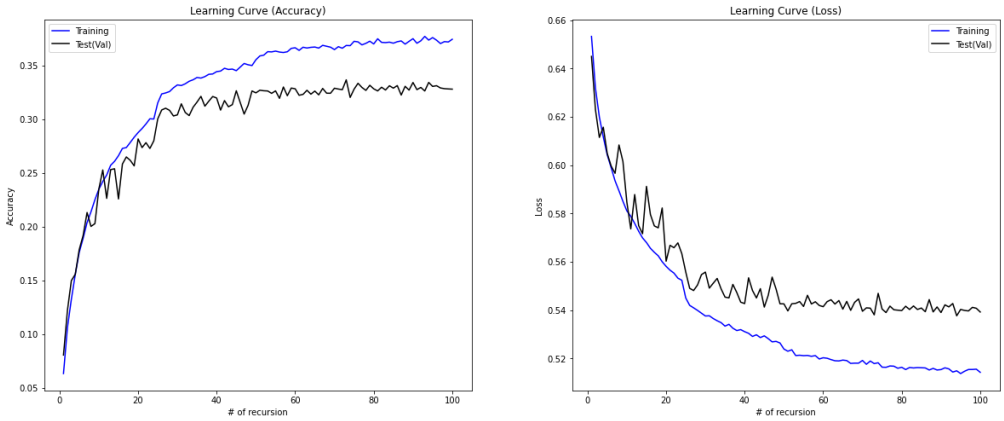

Figure 5: Accuracy (k=5) and Loss curves for the combined model.

3.3 Zero shot classification

The true advantage of this model is shown by its impressive ability to generalize to labels it was not explicitly trained on. Since our model outputs a word embedding vector, we are no longer constrained to the original set of labels and are instead able to search over an entire corpus of words. When directly comparing the outputs of the DeViSE model against a standalone CNN on never before seen image classes, we found that not only does the combined model stand a fighting chance at determining the true label (a task that is impossible for the standalone CNN), but the predicted labels are also semantically closer to the true label than the labels from the CNN.

Shown below in Figure 6 and Figure 7 are side-by-side comparisons of the model outputs when presented the same image. In Figure 6, it can be seen that while the combined model was not able to accurately predict the true label 'shark' it was able to produce a variety of oceanic related terms unlike the traditional softmax CNN where predictions like 'albatross' and 'snake' are in no way similar to the true label. Even more impressively, Figure 7 shows an example where the visual-semantic model was able to produce the true label 'tiger' as well as other large cat breeds such as 'lion' and 'puma'.

Though these specific cases are excellent examples of how our combined visual-semantic model is able to generalize, we sought a quantitative approach that we could use to measure the precision of the outputs generated by each of the models and classes. For each model, the word similarity metric was generated by taking the top 3 predictions for each labeled image in the CIFAR100 dataset (n = 50,000) and averaging the cosine similarities of each of these predictions. The control baseline was generated by taking the average of the cosine similarities between 10,000 combinations of random labels between CIFAR100 and Tiny Imagenet. The control is meant to be a measure of the similarity

between two random words and is a great benchmark by which to judge the semantic performance of each model. In this case, we see that the softmax CNN, though constrained to the 200 class labels found in Tiny ImageNet, provides a nearly threefold improvement compared to the control. Finally, the DeViSE model represents approximately a twofold improvement compared to the Softmax CNN and close to a sixfold improvement over the control, as shown in Figure 8.

DeViSE	Softmax CNN
1. Sea	1. Albatross
2. Dolphin	2. Bucket
3. Coral	3. Jellyfish
4. Snorkel	4. Snorkel
5. Ocean	5. Snake

Figure 6: Model comparison for 'shark'

DeViSE	Softmax CNN
1. Cat	1. Bison
2. Puma	2. Cardigan
3. Lion	3. African Elephant
4. Tiger	4. Baboon
5. Dog	5. Koala

Figure 7: Model comparison for 'tiger'

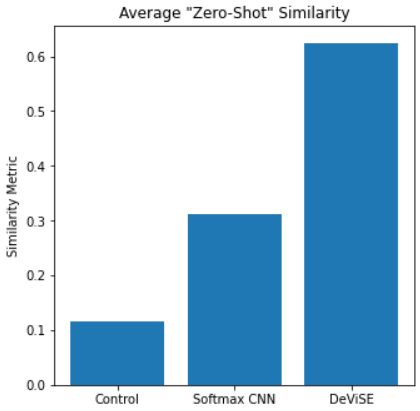

Figure 8: Zero-shot classification similarity for both Softmax CNN and DeViSE models (n=50,000). The control is a baseline that is meant to represent random relationships between words and was calculated using an average of 10,000 distinct label pairs

4 Conclusion and Future Opportunities

Based on our results, we find that the combined visual and language model has improved validation accuracy for $k = 5$ when compared against the standalone visual model. Though the validation accuracy for the visual model alone was around 0.25, the combined model, using the "hit @ k" approach with $k = 5$ generated an accuracy of approximately 0.31. However, we also see that in a zero-shot context, the combined model is able to produce more semantically appropriate predictions

even if it isn't able to guess the true label (with a similarity metric of 0.63 between true and predicted labels for the combined model compared to 0.31 for the visual model alone). This is consistent with the claims made in our initial hypothesis.

Though we were able to see true benefits of model generalization with our zero-shot experiments with the images used for training, we could certainly improve overall model performance by using a larger dataset. Though we're using the Tiny Imagenet dataset, our accuracies were about 25% compared to over 35% for the state-of-the-art CNN [10]. We would expect to see improved performance for both the visual and combined models if using the full ImageNet dataset with 14 million images instead of the $100,000$ we were working with. Similarly, we could also use a more comprehensive corpus for the language model. We used a pre-trained Wikipedia language model instead of creating our own because the size of the Wikipedia dump files exceeded 200GB. Future researchers could also explore using alternative language sources including text from Twitter and other social media.

References

[1] *Zero-data learning of new tasks*, volume 1, 2008.

[2] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[4] Erik Gundersen Miller. *Learning from one example in machine vision by sharing probability densities*. PhD thesis, Massachusetts Institute of Technology, 2002.

[5] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[7] Radim Řehůřek, Petr Sojka, et al. Gensim—statistical semantics in python. *gensim.org*, 2011.

[8] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.

[9] Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D Manning, and Andrew Y Ng. Zero-shot learning through cross-modal transfer. *arXiv preprint arXiv:1301.3666*, 2013.

[10] Jason Ting. Using convolutional neural network for the tiny imagenet challenge. 2016.

[11] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37, 2019.

[12] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

[13] Leon Yao, John A. Miller, and Stanford. Tiny imagenet classification with convolutional neural networks. 2015.