Novelty Detection with Zero-shot Learning

Zachary Maurer (zmaurer)

***Abstract*— This project explores how zero-shot learning methods could be applied to novel class detection on edge devices such as mobile phones. In particular, we examine how learned image embeddings produced by a neural network could be used as inputs to an anomaly detection model, and how these embeddings compare to common alternatives such as ImageNet activations and softmax probabilities from discriminative classifiers. We evaluate the quality of these embeddings via common zero-shot learning metrics such as Normalized Mututal Information and Recall@k and then compare their performance on a novelty detection scenario. We evaluate these methods on two separate datasets: the Stanford Online Products Database and a proprietary dataset of consumer goods.**

I. Introduction

Neural networks have demonstrated great accuracy in a wide range of difficult, but controlled, image classification tasks such as ImageNet.[1,2] With the development of compute-efficient architectures and frameworks like MobileNet and TensorFlow Lite [3, 4], it is now easier than ever to deploy neural networks directly on phones and other edge devices, running a pre-trained classification network locally and with fast inference runtimes.

However, by introducing these models into the edge of the network, and into peoples hands, we are exposing these models to a much greater variety of possible inputs than explored during training. Many of these inputs are of unknown classes that the classifier has never seen during training. When exposed to such an unknown class, many standard classification networks will still predict this input as belonging to one of the known classes, possibly with high confidence. This is a problem because essentially the neural network has produced an incorrect meta-classification; instead of correctly declaring this example as an anomalous or novel class, the model has characterized it as a known entity.

This project explores novel class detection strategies that could be run locally on a smartphone using image embeddings produced by an on-device, pre-trained neural network. We propose using recent research in zero-shot learning and metric learning to create the image-level vector embeddings that will perform well in an anomaly detection modeling scenario.

II. Related Work

In this paper, we define the problem of novelty detection as correctly identifying images of a class that was previously unseen during training. It is important to note that this is different from the normal task of a classifier, which is to make correct predictions for unseen training examples of some known class. In other words, our a novelty detection classifier

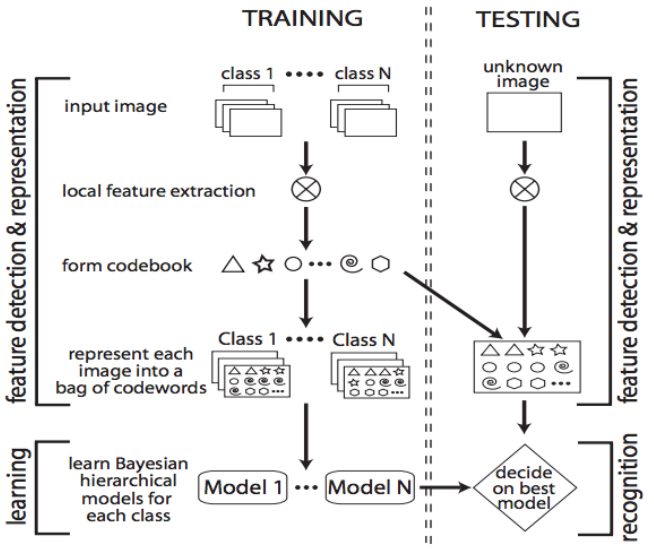

Fig. 1. A diagram of the codebook classification algorithm. Reproduced from Li Fei-Fei and Pietro Perona.

can be thought of as making predictions on whether each example belongs to the set of known classes or some greater set of novel classes. The goal of this procedure overall is to develop some sort of discriminative (or generative) bound around what the classifier ”knows” and what it ”doesn't know”.

Previously, this type of task has been approached in multiple different ways.

Content-Based Image Retrieval (CBIR) is a field of research that phrases the problem of class recognition as a search problem. Given a database of images, query that database using some representation of the input image. If no results are returned, this image could be classified as a novel class. Early work on this topic was done by Berman and Li [5,6] who demonstrated various distance based metrics and image processing techniques could be used to featurize images in a databases. An observation from these papers of key relevance to this paper is that images could be effectively summarized by some numerical vector (an embedding) that could be used as input to a trained classifier. Other work on this topic by Fei-Fei [7] demonstrated that these features could be extracted in an unsupervised manner by randomly sampling processed image patches from a group of images and then clustering them to create a ”codebook” of feature templates.

More recently, metric learning and zero-shot learning has extended many of the same concepts presented by this earlier

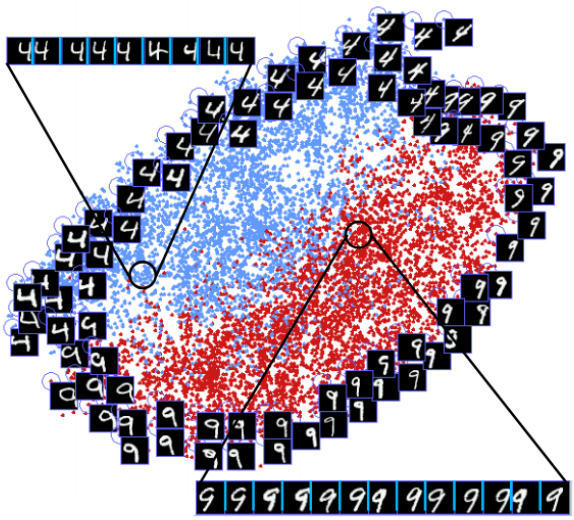

Fig. 2. A t-SNE visualization of two MNIST class embeddings produced by a contrastive loss. Reproduced from Raia Hadsell, Sumit Chopra & Yann LeCun.

work. In general, work on this topic has been focused on trying to determine methods of finding "description" vectors for examples for each of the known classes.[7, 8] Researchers in this field have recast this goal as the primary output of a neural network. [4, 5, 6]

In this situation, a convolutional neural network is trained to produce embeddings of an input that are similar to members of the same class and dissimilar to other classes. In the computer vision domain, early work on this was done by Hadsell et al.[8] in developing a contrastive loss function which a neural network could optimize over. They demonstrated that by specially creating a training set composed of pairs of images of same and non-same classes, and optimizing a loss function such that same class pairs we located closer on the manifold to non-same class images, the neural network could output structured latent embedding spaces. The visualization above is a low dimensional projection of the resulting manifold.

Later, this work was extended to the task of facial recognition where Schroff et al. developed a new training paradigm which consisted of 3-tuples of images (always containing 2 images of the same class and one of a negative class), that resulted in the triplet loss.[9] Most recently, work by Song et al. and Wang et al. respectively demonstrate alternative formulations of the distance-optimizing loss that leverage the structure of batched training instead of pre-sampling tuples from a set of images or propose alternative distance measures such as the angle between to vectors as a metric of similarity. Of particular note, we will expand on the loss constructed by Song as it forms the basis for the embeddings created in this project.

As summary, in contrast to a normal classification loss like binary cross-entropy, the zero-shot and metric learning cost functions explicitly encourage the network to produce embeddings that are distinguishable in some latent space.

These embeddings are then applied to novelty detection using them as inputs to a discriminative or generative modeling process. That is, by collecting a sufficient set of known description vectors, one can build a model of each known class, and when presented with a new example, estimate the likelihood that this new example belongs to any of the known distributions.

A comprehensive overivew of techniques in novelty detection can be found in reviews produced by Markou and Pimentel [12, 13]. Of particular revelance to this project is the idea of locality based anomaly detection, which characterizes anomalies as a function of local structure (e.g. density of nearest neighbors in some radius) and the concept of a One-Class Support Vector Machine which detects anomalies by computing a discriminative boundary around points in a known distribution by maximizing the margin between each example and the origin[14].

The research that is most similar to this project is found in Socher et al. "Zero-Shot Learning Through Cross-Modal Transfer" [15] who developed a cross modal embedding by minimizing the distance between the embedding produced by a neural network and the word2vec embedding of the class label. Then, they used these embeddings and the embeddings produced by a set of known classes to test an anomaly or novelty detection framework using density and distance based methods.

III. Problem Definition and Dataset

The problem of novelty detection can be broken down into two key sub-problems: (1) producing low-dimensional encodings of an input image, i.e. the embeddings and (2) given a set of embeddings with known classes, detect whether a new embedding from an unseen example fits the known joint distributions of embeddings and classes.

Concretely, given a collection of training images X and corresponding labels Y, where y_i is the classification label for x_i and each y_i belongs to the set K of known classes, can we accurately determine whether an unseen test example w_i is a member of one of the known classes K or if it belongs to some novel set of unknown classes N?

Our inital attempt at solving the first subproblem consists of training a neural network to produce semantic embeddings z_i for each training example x_i. These z_i are real dimensional vectors that are designed (by way of the loss function) to exist close together if they are members of the same class or exist far apart if they are members of a different class. Thus given all z_i for each y_i, we can construct a distribution of the embeddings for each known class and using this to determine whether a w_i is likely to have come from one of these distributions.

To be clear, for the purposes of this paper the datasets used were split into three subsets (1) training, (2) validation and (3) testing.

The training and validation splits contained members of the same "known" set of classes, but had no overlapping

Fig. 3. Examples from the Stanford Online Products Database.

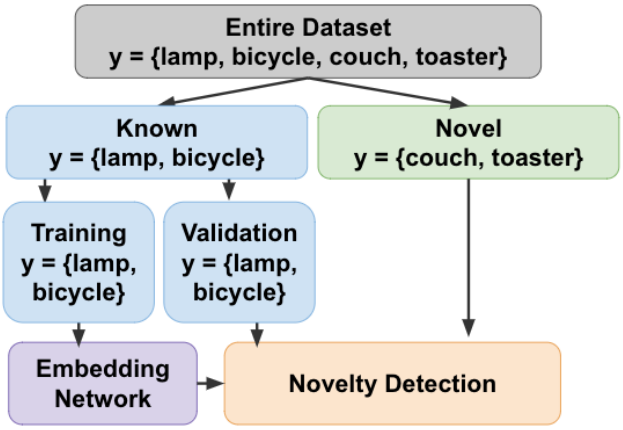

Fig. 4. A diagram of the dataset splitting process and overall modeling procedure.

examples. However, the test set contained a disjoint subset of classes, none of which appeared in the training or validation splits. All image data was converted to 224x224 RGB images.

IV. Methods: Embedding Creation

The three types of embeddings evaluated are described below:

A. ImageNet Bottleneck Activations (512-dimensional)

These embeddings are simply the final output of the 1x1 convolutional layer in a MobileNet that was pretrained on the ImageNet corpus. No fine tuning has been done on this network for our dataset. Similar to transfer learning, these features are sometimes used as features for a more specific classification task and can be used with reasonable success, relative to absence of effort required to train the network.[16, 18]

B. Softmax Logits (29-dimensional)

These embeddings are produced from a MobileNet that has only had the final prediction layer fine-tuned to classify a set of 29 image classes. No other layers have been trained during this process. This network was trained with the Adam optimizer and a fixed exponential decay on the learning rate every second epoch. In a novelty detection scenario, sometimes these probabilities are thresholded as a form of indicating the classifier's confidence in a particular prediction. Low confidence across all known classes indicates an anomaly. However, practically speaking, neural networks tend not to have well-defined ranges for the final activations and confidences, thus this method can be rather unreliable at detecting novelty.

C. Zero-shot Embeddings (100 and 256-dimensional)

These embeddings are outputted by a MobileNet that has been modified such that the final softmax layer is removed and the loss is computed directly on the final activations themselves, which now are set to output the z_i directly. The loss is derived from the work of Song et al. 2015 [5], which proposed an alternative metric learning loss that is designed to improve on the triplet and contrastive losses.

Briefly, the contrastive loss depends on training examples being prepared as 2-tuples of images which either belong to the same class or a different class. The network then produces the embeddings that minimize the distance for matching class tuples and maximizes the distance for mismatched tuples. The triplet loss extends this concept so that every tuple contains two examples with the same class and a third example from a different class. Thus, each tuple contains a positive match and negative match, which would be two separate examples in the constrastive loss scenario.

The key issues with both of these scenarios is that the dataset must be pre-formatted such that these tuples exist and the network architecture must be modified to take two or three images as input. Both of these requirements are somewhat unusual modifications for the normal architecture and training scemes, which consist of single image inputs and randomly sampled batched training examples.

Song et al. propose a different loss which minimizes the a similar objective to the others – that is, images of the same class should exist close together (defined by Euclidean distance) and images of a different class should exist far apart (given some margin). But, the positive and negative pairs are drawn from a pairwise comparisons of each input image x_i in a given batch to all other x_j.

Following Song, the loss function is expressed as follows.

Given a batch of c-dimensional embedded features X, compute the pairwise, squared Euclidean distance between each x_i and all other x_j, call this matrix D^2.

$$\tilde{J}_{ij} = \log\left(\sum_{(i,j)\in\mathcal{N}} \exp\{\alpha - D_{i,k}\} + \sum_{(j,l)\in\mathcal{N}} \exp\{\alpha - D_{j,l}\}\right) + D_{i,j} \quad (1)$$

$$\tilde{J} = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max(0, \tilde{J}_{ij})^2 \quad (2)$$

where $\mathcal{N}$ are all negative pairs, $\mathcal{P}$ are the positive pairs and α is the margin

As the original work was done in Caffe, we implemented this loss function in TensorFlow and integrated it into the MobileNet architecture.

V. Methods: Anomaly Detection

Using the embeddings we've created, we evaluated five novelty detection strategies.

A. Normalized Distance from Mean Class Embedding

In this classifier scheme, we calculated the mean embedding for each of the known classes. Then, we fit a normal distribution to the Euclidean distances between all examples of a class and its mean class embedding. To classify an

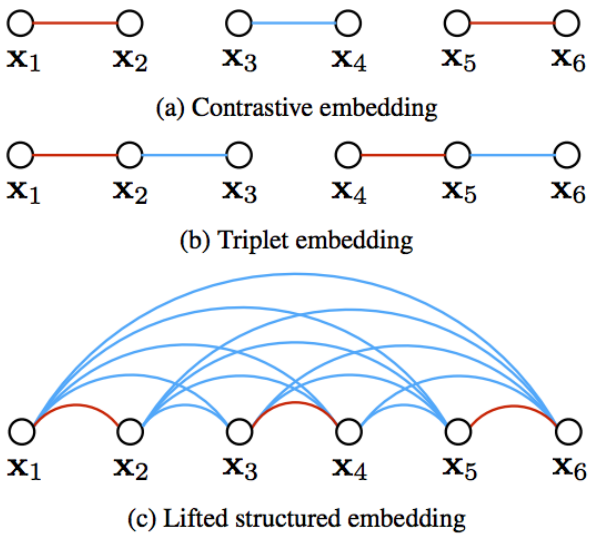

Fig. 5. A reproduction of a visual comparison of the Lifted Structured Feature Loss and other metric learning losses from Song et al. Original caption is, "Illustration for a training batch with six examples. Red edges and blue edges represent similar and dissimilar examples respectively. In contrast, our method explicitly takes into account all pair wise edges within the batch."

unseen example w_i, we calculated the mean Z-score from all class-distance distributions, if that average Z-Score was greater than some threshold, the example w_i seems to be distant enough to be qualified as an anomaly across all distributions.

We extended this classifier metric to used other distance metrics such as cosine similarity and Mahalanobis distance ($\sqrt{(x_i - \mu_i)^T Cov^{-1}(x_i - \mu_i^T)}$). However, the consine distance method demonstrated no benefits over the Euclidean distance and the computing the covariance matrix for the Mahalnobis distance yielded a singular matrix in for some classes. As a result, in future sections we only focus on the Euclidean distance based classifier.

B. One-Class SVM

In this model, a one class SVM was trained on all the embeddings of all known classes. Isolation Forests were also explored as a substitute to this model, but they seemed to perform much worse in initial tests, and we thus restrict our focus to the SVM.

C. Distance to Cluster Centers

This classifier extends the density based outlier detection strategies such as Local Outlier Factor and Local Outlier Probability [19,20] (LOF and LoOP, respectively). In those modeling scenarios, anomalies are hypothesized to exist in local neighborhoods of low density. Local Outlier Factor and Local Outlier Probability can be roughly summarized as the average distance between a point and its nearest neighbors, scaled by the average distances between each neighbors' distance to its nearest neighbors.

Unfortunately, for this project, as we are hoping to have in-situ anomaly detection on edge devices, this model doesn't work within our constraints as it requires storing a tens to hundreds of thousands of embeddings to compute the $k+1$ nearest neighbor queries and the nearest neighbor queries themselves are too slow to be used is a user-application that requires inference runtimes on the order of hundreds of milliseconds. (In many user interfaces, 100-200ms response times is considered "interactive".)

Thus, we used Affinity Propagation clustering as a way of summarizing the known embedding spaces of high connectedness into a fixed number of cluster centers. Interestingly, DBSCAN, an explcit density based clustering algorithm, often failed to produce a significant number of clusters and labelled many points as noise.

To train this classifier, we first computed the cluster centers, and then trained a One-class SVM on the distances between each embedding and all cluster centers. Thus to classify a new example, we would first compute the distance between the embedding of the example and all pre-computed cluster centroids and then classify that vector of distances using the trained SVM.

D. Distance to Nearest Neighbors

While inspired by LOF and LoOP, the previous method is really computing some metric of distance to areas of global density, instead of local density as in LOF and LoOP. In this anomaly detection scheme, the overall classifier is the same, except the One-Class SVM is trained on the distances between each example and its k nearest neighbors.

E. Distance to Nearest Cluster Centers

Again, the problem with the nearest neighbors approach is that it requires our algorithm to store some population of embeddings in order to compute the local distances. So, we borrow the idea of summarizing areas of density via clustering from (3) and modify (4) such that the One-class SVM is now trained on each embedding's distances to the k nearest cluster centers.

VI. Results: Network Training

The MobileNet with lifted feature loss was trained to produce zero-shot embeddings for 5-20 epochs on training splits from the proprietary consumer products dataset and a subset of classes from the Stanford Online Products Database. The training split for the consumer products dataset was roughly 33,000 images with 29 classes. The training split for the Stanford Online Products Database consisted of 6 classes with approximately 50,000 examples.

The network was trained with the Adam optimizer with an initial learning rate of 0.005 and a 0.9 decay after every 2 epochs.

After around 10 epochs, the loss plateaued when training over both datasets. Restarting the training process from that checkpoint with a more agressive or smaller learning rate seemed to make no difference. All models were trained on a 2014 MacBook Pro i7 CPU.

VII. Results: Embedding Quality

After training the model, it was crucial to evaluate the quality of these embeddings, before using them as input to the novelty detection model. We used the same metric as Song et al. used to evaluate their embeddings in the original lifted feature loss paper which is the Normalized Mututal Information (NMI) score based off the Affinity Propagation clustering of these embeddings. Essentially, this score measures how informative the clustered label is to identifying the true label of all examples and vice versa. The defintion of NMI is provided below, but a full defintion of NMI can be found in [21].

$$NMI(\Omega, \mathcal{C}) = \frac{Mutual\ Information(\Omega, \mathcal{C})}{H(\Omega) + H(\mathcal{C})/2}$$

where Ω is the set of clusters and $\mathcal{C}$ is the set of classes.

We calculated these scores by clustering the embeddings for both the train and test datasets (29 classes and 73 classes, respectively) using the embeddings produced by the networks above. We used two clustering approaches. The first was an "oracle" based metric, where we used K-Means to cluster the embeddings, setting k to the true number of classes. The second clustering method used Affinity Propagation, which dynamically estimates the number of clusters. Since Affinity Propagation produced many more clusters than K-Means, we also computed the Adjusted Mutual Information which tries to adust for informative assignments that occur by chance in clusterings with many clusters.

Additionally, we evaluated Recall@k, which is a common metric computed in zero-shot learning research. Given a set of test embeddings W, the Recall@k is the percentage of times that for each embedding w_i with class y_i, there exists another example n_j in the k nearest neighbors of w_i, such that n_j had class y_i.

Embedding	Dimension	K-Means NMI	Affinity Prop AMI	Affinity Prop NMI
Softmax Logits	29	0.2679	0.1842	0.4362
ImageNet Bottleneck	512	0.3430	0.2815	0.5779
Zero-Shot	256	0.4008	0.3257	0.6366

Fig. 6. Clustering based embedding quality measurements for the test set of the Consumer Products dataset.

Embedding	Dimension	Recall@1	Recall@5	Recall@7
Softmax Logits	29	53.46	71.33	72.97
ImageNet Bottleneck	512	76.64	86.78	87.77
Zero-Shot	256	86.50	92.21	92.74

Fig. 7. Recall@k measurements for the test set of the Consumer Products dataset.

In the table above, it is clear that the embeddings produced by the Lifted Structured Loss tended to perform better by these metrics than any of the other embeddings on both datasets. In addition, increasing the embedding dimension seemed to be related to a positive increase in NMI and Recall@k.

Embedding	Dimension	K-Means NMI	Affinity Prop AMI	Affinity Prop NMI
ImageNet Bottleneck	512	0.2278	0.1288	0.2779
Zero-Shot	256	0.3250	0.1469	0.3060

Fig. 8. Clustering based embedding quality measurements for the test set of the Stanford Online Products Database dataset.

Embedding	Dimension	Recall@1	Recall@5	Recall@7
ImageNet Bottleneck	512	72.84	91.23	92.63
Zero-Shot	256	76.09	92.97	94.00

Fig. 9. Recall@k measurements for the test set of the Stanford Online Products Database dataset.

The performance of these embeddings is roughly consistent with those found in the Song et al. paper. However, in that paper, a much larger network than MobileNet is used as the base for training the embeddings and likely for that reason our NMI metrics on the Stanford Products Dataset are lower. Further evidence for this is that our ImageNet BottleNeck activations on the Stanford Products Dataset are also lower, despite being trained on the same dataset as the larger network in Song et al.

We also evaluated the anomaly detection qualitatively by visualizing the embeddings as a low-dimensional projection via t-SNE. Each visualization shows multiple perplexities which controls roughly the weighting between "local and global aspects of your data".[22]

In this figure we plot the least informative embeddings (softmax logits) versus the most informative embeddings (zero-shot) on the consumer products training set. Each color represents an example of a distinct class. We can see that due to the mixing of classes in both visualizations, it appears as if there is no larger global structure. However, as we increase the perplexity we see that the zero-shot embeddings at least starts to develop some sort of local cluster structure (fig 10) when the softmax embeddings remain unchanged (fig 11).

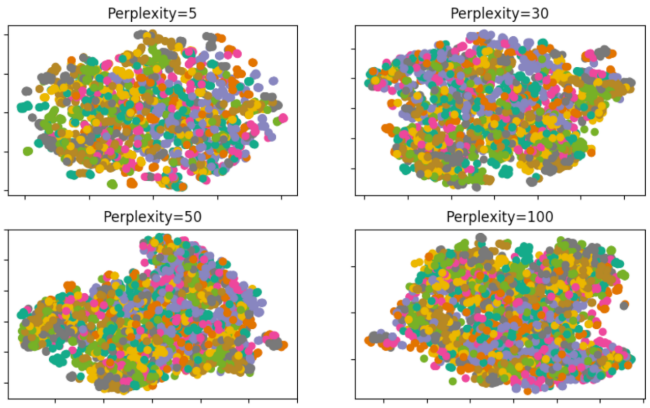

Fig. 10. A t-SNE visualization the softmax embeddings of the training set. Each color denotes a different class. Note how the structure remains essentially the same over different perplexities.

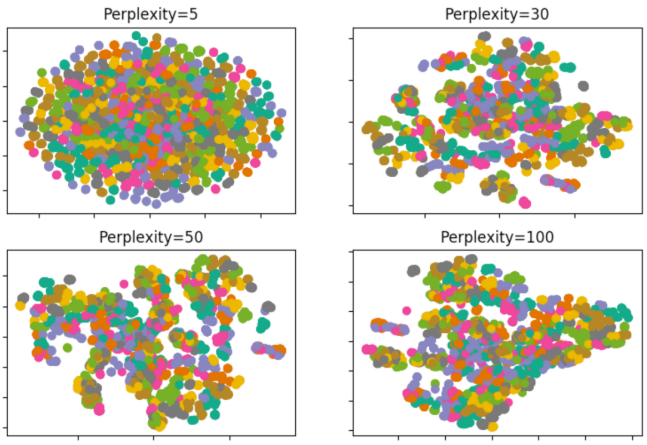

Fig. 11. A t-SNE visualization the trained zero-shot embeddings of the training set. Each color denotes a different class. Note the completely different structures of the projection over different perplexities.

This is somewhat similar to what Hadswell et al. observed before they augmented the dataset to specifically sample certain other examples in the dataset. Since our training procedure only randomly samples examples to compare with each other, it may be that if we had the ability to use more prior knowledge to sample hard-negatives from the entire dataset for each training example, then we could have a more structured t-SNE projection.

Furthermore, we see this same type of local structure emerge when we visualize the projection of the training and test set simultaneously (fig 12). Known classes (blue) tend to cluster together whereas novel classes (orange) are more diffuse.

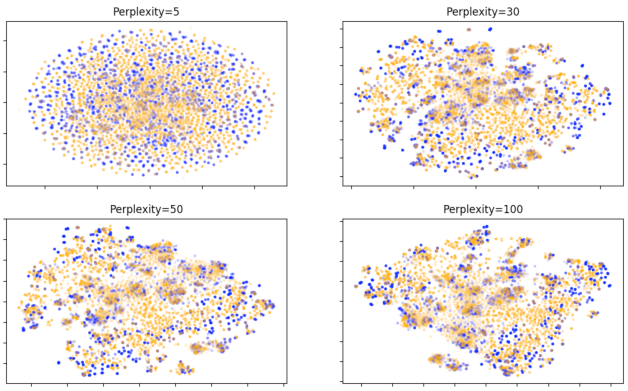

Fig. 12. A t-SNE visualization of novel classes from the test set (orange) and known classes from the validation set (blue). Note how the blue points form some set of local clusters, whereas orange classes are located in the interstices of the blue clusters. This may be evidence that novel examples are mapped to areas of the embedding space that are low local density.

VIII. Results: Novelty Detection

To evaluate the performance of our novelty detection systems, we set a rough cut-off of 15% false positive rate, as the maximal accepted value. For each classifier, we tuned the hyper parameters until roughly meeting this 15% false positive rate and at that point we calculated the true positive rate.

Model	True Positive Rate	False Positive Rate
Normalized Mean Distance	0.2474	0.1524
One-Class SVM	0.2101	0.1524
Affinity Prop. Cluster Distance	0.2182	0.1560
Nearest Neighbors Distance	0.4134	0.1585
K-Means Centroid Distance	0.5056	0.1524

Fig. 13. Novelty detection performance on the Consumer Products dataset.

Model	True Positive Rate	False Positive Rate
Normalized Mean Distance	0.2890	0.4838
One-Class SVM	0.1344	0.1561
Affinity Prop. Cluster Distance	0.0463	0.1559
Nearest Neighbors Distance	0.1533	0.1647
K-Means Centroid Distance	0.2863	0.1550

Fig. 14. Novelty detection performance on the Stanford Online Products Database dataset.

Computing the AUC score was another viable alternative, however despite varying the hyperparameters over a large range for each classifer, some still did not produce sufficently diverse results, and the computation of the AUC score would require uneven interpolation across classifiers. For this reason, we decided on a hard threshold.

In figures 13 and 14, we see that classifiers which involved some element of locality and density peformed best. Training a classifier based on either the distances to global cluster centers and the nearest neighbor points or the nearest cluster centroids produced the best results.

IX. Discussion

Looking specifically at the embedding quality results, we see that the trained embeddings ("Zero-shot" in the tables above) learned via the Lifted Structure Feature Loss had the highest NMI, AMI and Recall@K scores. The worst embeddings were the softmax logits from a pre-trained classifer and the second best embeddings were the ImageNet Bottleneck Activations. As mentioned previously, this trend is consistent with the results presented in the paper by Song et al.

In addition, increasing the dimensionality of the trained embeddings also increased the measured quality of the embeddings. This trend is consistent with other learned embeddings such as word2vec and GloVe, where 200 dimensional or greater vectors demonstrate signifcant performance gains over their lower dimensional versions.

From the tables of each models performance in the previous section, it is clear that the Distance to Nearest Cluster Centroids anomaly detection model performed the best by a significant margin. Furthermore, the use of the zero-shot embeddings demonstrated significant performance gains over the ImageNet Bottleneck embeddings.

This trend was consistent across both datasets tested for this paper.

Less sophisticated classifiers such as the Normalized Mean Distance, One-Class SVM and Affinity Propagation Cluster Distance classifiers likely failed to perform well because the embedding space lacks a global structure. Since the loss is only optimized over pairs of images found in a training batch, the loss is technically not maximizing the distance between entire sets of examples of non-matching classes. In other words, examples that are sampled together are positioned in the latent space relative to each other, but this does not necessarily hold for all members of all classes since the minibatch generation is unlikely to cover all permutations.

In previous iterations of metric learning, this issue was addressed by a form of hard-negative mining where researchers would synthesize datasets such that each of the training tuples (using the contrastic or triplet loss) was paired with other examples that were semantically relevant positive or negative comparisons. For example, Hadswell et al. in their paper explicitly pairs examples from the MNIST dataset with examples from the same class that had been translated or rotated. By generating a training dataset of positive examples in this way and minimizing the embedding distance between them, the researchers demonstrated they could produce rotation and translation invariant embeddings. In our modeling scenario, we used the most common training procedure where examples are randomly sampled to form a minibatch during training.

Another important observation is our novelty detection system performed better on the consumer products dataset and worse on the Stanford Products dataset. We believe there are two possible reasons for this result. First, while the Stanford Product database test-split had significantly fewer classes than the consumer products test-split (6 vs. 73, respetively), the individual sample diversity within the Stanford Products dataset is much higher. The consumer products dataset is made up of rotations and translations of common consumer products on a uniform background with a fixed perspective. However, the Stanford Products dataset was scraped from eBay postings and demonstrates a lot of diversity in form, color, shape and persepctive.

For example, one class in the consumer products dataset is a branded bottle of shampoo of a specific size, shape and color. All images of that class consist of the same basic object, translated and rotated into different positions. However, in the Stanford products dataset, examples drawn from a single class like "bicycle" might include tandem bikes, mountain bikes, bikes of different colors, bikes missing wheels and bikes photographed in different perspectives (e.g. head-on, profile, occluded, etc.) Thus, despite having fewer classes the samples in the Stanford products dataset present a much more diverse set of individual images which may be difficult for the network to embed in a performant manner.

A second factor that might have influenced this difference in performance could be related to the relatively small size of the validation split used to test the Stanford Products dataset, which may have unfairly increased the false positive rate of all models.

Train/Val Set	Test Set	True Positive Rate	False Positive Rate
Stanford Products Database	Consumer Products	0.5873	0.1536
Consumer Products	Stanford Products Database	0.9955	0.1597

Fig. 15. Novelty detection performance on the hybrid dataset.

Building on this observation about sample diversity, as an experiment to test the systems robustness to significant variants in input, we also created a hybrid test set where the anomaly detection system was trained and validated on a set of examples from the consumer products dataset and tested on a set of examples drawn from the Stanford Products Dataset. We also evaluated the opposite formulation where train and validation examples came from the Stanford Products dataset and test examples were drawn from the consumer dataset. Thus, in both cases, the novel examples in this formulation differed significantly from the training distribution.

We found our anomaly detection system performed extremely well in this situation.

From these results, we can see that our system is quite performant even when the test set is drawn from a completely disjoint set of classes with extremely dissimilar examples from the training data. Furthermore, this lends additional support to the notion that our anomaly detection system performed worse on the Stanford Products database due to sample diversity. In the results above, we see a massive increase in the true positive rate, when the novelty detection model is trained on embeddings from a dataset with low sample diversity and tested on one with high sample diversity.

Specific performance comparisons and benchmarks are hard to define for novelty detection, since as we have demonstrated above, novelty detection is highly dependent on the diversity of both the training and test sets. Song et al. do not evaluate the novelty detection capabilities of their embeddings and Socher et al. evaluated their novelty detection system on a different dataset.

X. Future Work

Future work on this project could evaluate the combined inference runtimes on a mobile device of the methods described previously. Furthermore, training a more sophisticated embedding network and experimenting with novel loss functions such as the angular loss would be welcome. Finally, instead of training a smaller MobileNet directly, one could investigate training an a larger network such as Xception and then transfer learn that knowledge to a MobileNet via distillation. Finally, evaluating this method on similar datasets as previous novelty detection papers would help contextualize whether or not this novelty modeling technique could be applied to a more general set of problems.

XI. Conclusion

In this paper, we demonstrated that image embeddings learned via a semantic loss function from the Zero-Shot Learning research community can be used to develop a

relatively robust novelty detection. We intrinsically evaluted these learned embeddings, compared them to common substitues such as softmax logits and ImageNet Bottleneck features and found that the Zero-shot embeddings had higher measures of quality and predicitveness as defined by Normalized Mutual Information and Recall@k.

Moreover, we extrinsically evaluated the quality of these embeddings by measuring their performance on a novelty detection task. In this case, the learned zero-shot embeddings significantly outperformed all other embedding types.

We further proposed a new novelty detection model, inspired by the ideas of LOF and LoOP that can run in a memory and compute constrained environment and demonstrates promising performance on the datasets we evaluated. This model summarizes the areas of density in the distribution of learned embeddings from the training classes via clustering and then each new example is classified as novel or not by training a classifer on the distances from the query embedding to each of the k nearest cluster centroids. By only restricting the predictions to distances to the k nearest areas of density (represented by clusters), we believe that this captures the local structure that is developed by the Lifted Structured Feature Loss from Song et al.

XII. Acknowledgements

Thank you very much to the CS221 course staff for organizing such an interesting class.

References

[1] He et al. "Deep Residual Learning for Image Recognition" (2015) https://arxiv.org/abs/1512.03385
[2] Szegedy et al. "Going Deeper with Convolutions" (2014)
[3] Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile vision Applications" (2017) https://arxiv.org/pdf/1704.04861.pdf
[4] https://www.tensorflow.org/mobile/tflite/
[5] https://homes.cs.washington.edu/ shapiro/andy-cviu.pdf
[6] https://homes.cs.washington.edu/ shapiro/iccv2005.pdf
[7] Fei-Fei and Perona, "A bayesian hierarchical model for learning natural scene categories", CVPR 2005
[8] https://cs.nyu.edu/ sumit/research/assets/cvpr06.pdf
[9] Schroff et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering" (2015) https://arxiv.org/pdf/1503.03832.pdf
[10] JSong et al. "Deep Metric Learning via Lifted Structured Feature Embedding" (2015) https://arxiv.org/abs/1511.06452
[11] Wang et al. "Deep Metric Learning with Angular Loss" https://arxiv.org/abs/1708.01682
[12] Pimentel et al. "A review of novelty detection" (2014) http://www.robots.ox.ac.uk/ davidc/pubs/NDreview2014.pdf
[13] Markou et al., "Novelty detection: a reviewpart 2:: neural network based approaches" (2003)
[14] Scholkopf, Williamson et al. "Support Vector Methods for Novelty Detection" (2000) http://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection.pdf
[15] Socher, Richard et al. "Zero-Shot Learning Through Cross-Modal Transfer" (2013)
[16] Norouzi et al. "Zero-shot Learning by Convex Combination of Semantic Embeddings" (2014) https://arxiv.org/pdf/1312.5650.pdf
[17] "MobileNets: Open-Source Models for Efficient On-Device Vision" https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html
[18] "Building powerful image classification models using very little data" https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
[19] "LOF: Identifying Density-Based Local Outliers"
[20] Kriegel et al. "LoOP: Local Outlier Probabilities" (2009)
[21] "Normalized Mututal Information" https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html
[22] "How to Use t-SNE" https://distill.pub/2016/misread-tsne/