# Scene Recognition with Limited Data

**Neehar Peri, Kush Jain, Sunyu Wang, Uksang Yoo**
{nperi, kdjain, sunyuw, uyoo}@andrew.cmu.edu

## Abstract

In recent years, deep learning techniques trained on increasingly large datasets have brought about significant improvements in scene recognition and image classification. However, the performance of these novel techniques have not been extensively studied on small datasets, with many common deep learning models requiring millions of images to converge. Many of these large models have been pre-trained on large datasets for image recognition and classification tasks. These models are likely to have richer features than shallower light weight models due to the volume of training data, despite their different training objectives. In this paper, we present a novel approach that combines existing pre-trained feature extractors with light weight classifiers. These models are evaluated on two novel datasets: Places100, a subset of the Places365 scene classification dataset and Open-Places100, a derivative of Places100 to study a model's ability to differentiate between in-domain data and open-set examples. We first establish a baseline using ResNet-18 trained and evaluated on both datasets, measuring the accuracy of our end-to-end trained baseline. Motivated by the poor performance baseline, we propose using pretrained feature-classifier pairs to improve upon the baseline. We study VIT, CLIP, and ResNet pretrained features and pair these with neural network, SVM, and XGBoost classifiers. Lastly, since each set of pretrained feature-classifier pairs has unique failure modes, we propose a self-training framework to use the majority vote of our nine feature-classifier pairs to weakly label a larger dataset. The results show that our self-trained network improves performance compared to the pretrained feature - lightweight classifier combinations trained on small datasets, showing promise for semi-supervised applications where large sets of unlabeled data are available. Our code is available on Github.

## 1 Introduction

Supported by the improvement of computer hardware resources, deep learning techniques developed in recent years have seen tremendous successes in scene recognition tasks with large datasets. Trained with millions of images, these convolutional neural networks have demonstrated accuracies above 50% [11].

However, not only do these techniques rely on extensive computational resources, their superior performance has not yet been verified with small datasets. In fact, many existing deep learning techniques tend not to perform well when the amount of training data is below a certain—usually high—threshold [2].

To evaluate this claim, we create a subset of the Places365 dataset with 100 classes, and 20 images per class. We train on 10 images per class and evaluate on the held out 10 images in each class. We train a ResNet-18 model end-to-end on a small subset of 1,000 images sampled from the Places100 dataset [12]. As shown in Fig. 1, after approximately 50 training epochs, the training accuracy plateaues at about 11%, significantly lower than the above 50% accuracy that similar models achieved after trained with millions of images. The ResNet-18 model is unable to learn a robust feature representation given the limited data.

On the other hand, light-weight classic classifiers, such as multilayer perception (MLP), support vector machine (SVM), and XGBoost have demonstrated reasonable classification accuracy for small dataset problems. For scenarios where training data and computational resources are limited, a complementary relationship appears to exist between the heavy-weight deep learning techniques and the light-weight classic classifiers.

In order to generate more robust features, we look to large scale pretrained models. Some exemplars among these include the Residual Neutral Network (ResNet) [6], the Vision Transoformer (ViT) [4], and the Contrastive Language-Image Pre-Training (CLIP) [9]. All these three techniques leverage web-scale datasets to pre-trained models that convert complex internal information of images—such as RGB colors—into machine-learned feature representations. Given these feature representations, we can train light weight classifiers to address complete the scene recognition tasks.

Given that pre-trained models associated with these deep learning techniques are available to the research community, we are interested in exploring this complementary relationship by cascading pre-trained models and light-weight classifiers, while regarding the pre-trained models as “fixed feature extractors". We expect these combinations will yield satisfactory classification accuracy, since the pre-trained models are trained with millions of examples, and act as the image-processing feature extractors for the light-weight classifiers given their generalization power. More importantly, these combinations could perform scene recognition well even with small datasets and limited computational resources.

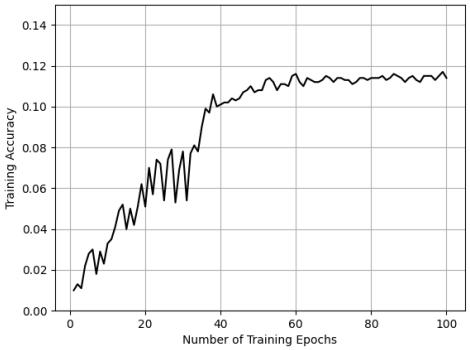


Figure 1: Baseline result: an empty ResNet-18 model trained on the Places100 dataset.

## 2 Data

We conduct our exploration of deep learning for small datasets with the Places365 dataset. The original dataset containing more than ten million images classified into 365 classes [12]. To simulate a small-data scenario, we constructed the Places100 dataset for our project. Places100 is a randomly sampled subset of Places365 under uniform distribution. It contains 100 classes, 20 images per class, hence 2,000 images in total. Note that this is only approximately 0.02% of the size of Places365. For each class, 10 images are used for training and 10 for testing.

Additionally, we construct the Open-Places100 dataset. The purpose of this dataset is to explore how our heavy-light cascaded approach performs with out-of-distribution classes. It is important to detect anomalous inputs when deploying machine learning systems. The use of larger and more complex inputs in deep learning magnifies the difficulty of distinguishing between anomalous and in-distribution examples [7].

Specifically, Open-Places100 consists of the same images contents as Places100, but different labels. Fifty classes are labeled as normal and are considered to be “known” classes. These classes are used in the same way as the classes in Places100. The remaining fifty classes are labelled as one class, called the “outlier” class. Within the outlier class, twenty-five classes or 250 images are used for training (i.e. outlier exposure), and the remaining twenty-five classes are completely witheld for testing. The known and unknown classes were chosen randomly from Places100.


Figure 2: Sample images in the Places365 dataset

These datasets are suitable for our project given that they are sufficiently small and are fully annotated, which is necessary for the supervising learning of the light-weight classifiers. Since these datasets are randomly sampled under a uniform distribution, their biases should be negligible.

## 3 Related Work

In this section, we briefly introduce the deep learning models that we have used as image feature extractors, namely, ResNet, ViT, and CLIP.

**Residual Neural Network (ResNet)**. He et. al. proposed using identity mapping by shortcuts to train deeper neural networks [6]. They find that "plain" deep networks have higher train and test accuracy on image benchmarking datasets. He et. al proposed a solution by construction: if added layers simply overparameterize the model, the added layers should learn identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. In order to facilitate such a learned model, Residual Networks explicitly encode identity mappings. If the input is optimal, the learned weights are 0 and the value of the input is passed onto the output. If the input is near optimal, the learned weights are small, allowing for small updates (which are easily trainable). Further, these identity connections are parameter free, and serve as gradient highways that preserve gradient in deeper models. Given the success of this simple model architecture, we use variants of ResNet in our experimental evaluation. We use the ResNet-18 model in our baseline experiments for learning from small datasets end-to-end and use ResNet-101 for extracting fixed features for training shallow models.

**Vision Transformer (ViT)**. Despite the success of transformers for natural language processing tasks, transformers have not been widely used for computer vision applications. Dosovitskiy et. al. suggests ways of preparing image-data for use in transformer encoders. Each image is tokenized into $n \times n$ patches [4]. Each patch is flattened into an $n \times n$ vector. This vector is embedded using a linear projection. This embedding vector is used as input to the transformer encoder. The transformer encoder applies a combination of layer normalization, self-attention, and attention layers to learn a non-linear combination of the input embeddings. Importantly, the key, query, and value weight matrices used for the attention operators can learn convolutions but are not restricted to just this function class. Ablation studies suggest that pretraining on very large datasets, including JFT-300M, is critical for learning robust fixed features. We leverage the feature from the z0 embedding at the $L^{th}$ layer of the transformer encoder as our fixed feature for building lightweight classifiers.

**Contrastive Language-Image Pre-Training (CLIP)**. Current approaches for using deep learning to train computer vision models is highly dependent on the availability of curated large-scale datasets. However, such datasets are often only created (and have labels) for one task. This makes it difficult to learn generic algorithms for image understanding. Often, methods trained on these single purpose datasets are brittle (to adversarial attacks), and don't generalize well in the real world. Radford et. al suggest that, to create generalized image representations, we should train on a wide variety of images with a wide variety of natural supervision that's abundantly available on the internet [9]. Importantly, Radford et. al. suggest not directly optimizing for any benchmark, but rather optimizing for an image-text classification task. Given an image, CLIP predicts which out of a set of 32,768 randomly sampled text snippets, was paired with the image in the dataset. In order to solve this task, CLIP models will need to learn to recognize a wide variety of visual concepts in images and associate them with their names. As a result, CLIP models can be applied to nearly arbitrary visual classification tasks. For instance, if the task of a dataset is classifying photos of "dogs" vs. "cats", CLIP can score which of the two descriptions is more likely to be paired with. CLIP closes the "robustness" gap by up to 75%, while matching the performance of the original ResNet-50 on ImageNet zero-sot without using any of the original 1.28 million labeled examples.

**Using Pre-trained Models as Fixed Feature Extractors**. The use of pretrained models is widely adopted by the deep learning community. The approach of combining features extracted from pre-trained deep learning models and other machine learning models have appeared in the context of deep convolutional neural networks (CNN). [5] view features extracted from different pre-trained CNNs as different "views" of the same training images. This perspective converts the image clustering problem into a multi-view clustering problem. By combining these pre-trained features, Guerin et. al are able to build complementary relationships between sets of images better represent the original data, achieving improved results compared with methods using standard CNN architectures alone. Similarly, [8] combined lightweight CNNs with transfer learning models to perform the specific task of recognizing banknotes. Leveraging transfer learning to address the small data problem, Linkon et. al used pre-trained deep learning models to augment their dataset, showing that a combined approach produced similar recognition accuracy with the best previous records.

## 4 Methods

We first train a randomly initialized ResNet-18 model on the Places100 dataset as a baseline (Fig. 1). Consistent with our expectation, the ResNet-18 model trained with limited data performs considerably worse than a model trained with the original Places365 dataset [11]. This performance degradation motivates our heavy-light cascading approach. Hence, we evaluate the three pre-trained deep learning models—ResNet-101, ViT, and CLIP—as image feature extractors and the three light-weight classifiers—MLP, SVM, and XGBoost—for scene recognition with the Places100 and the Open-Places100 small datasets. In total, we formed nine such heavy-light combinations, as shown in Fig. 3. Specifically, to provide a viable solution that works with limited compute, we do not perform backpropogation on the pre-trained models themselves. Rather, we use their penultimate layer for feature extraction, and only train the light-weight classifiers with our small datasets. Moreover, we train a second baseline, using a ResNet-18 model initialized with ImageNet weights. This baseline result is intended to provide a comparison between our proposed heavy-light cascaded approach and merely performing backpropagation within the entire deep learning model on a small dataset. Although we find that this approach performs better than random initialization, it is still sigificantly lower than training on the original Places365 dataset [11].

We evaluate the performance of each heavy-light combination using their respective training accuracy for both the Places100 dataset and the Open-Places100 dataset. We believe that using each combination's test accuracy as the evaluation metric sufficiently reflects these combinations' performances on scene recognition in our simulated small-data scenario.

Additionally, we evaluate how these combinations perform in outlier detection task, specifically whether they are able to differentiate outlier classes that were not seen during training from the classes that these models were trained on. Our assumption is that a traditional deep learning model will not perform well on small datasets as these models lack the priors that are encoded in lighter models, which are learned as the size of the data grows. However, pre-trained features could significantly

improve the performance of our lightweight classifiers, due to the generalization power that the large amounts of training data embed into the pre-train deep learning models.

In addition, we constructed the confusion matrices for the samples that the combinations misclassified on the Places100 dataset. These confusion matrices reveal the source of mistakes for the nine combinations, potentially enable a better understanding of the error modes. As shown in Figure r6, the errors of the nine models are disjoint, indicating that we may be able to train a better model by ensembling.

This motivates our final approach of self-training [10], which consisted of two steps. First, we ensemble all of our nine models together, and predict class labels on an extended dataset of 1000 images/class for the 100 known classes. This dataset simulates a larger set of images that have not been labeled, but that are still useful for training a larger neural network. To predict the class label of one of these images, we take the majority vote of our models, breaking ties by taking the vote of the ViT-NN model pair (our overall best performing model). Using this new dataset of 10,000 images, we train a ResNet-18 model both from scratch and using pretrained weights, which we have shown performs poorly with the smaller dataset.

Simply training a randomly initialized ResNet-18 model with these noisily labeled students achieves 30% accuracy, more than doubling the original result when training on the Places100 dataset. However, we further modify the model to add CLIP features to the ResNet-18 latent features, further improving performance by 30%.

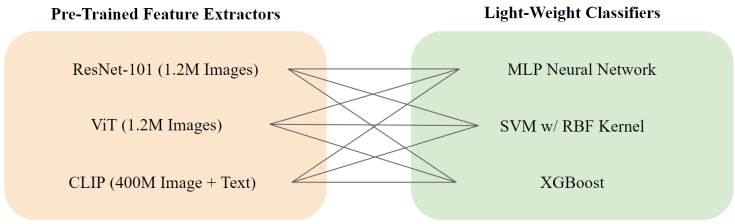


Figure 3: Combinations of the models evaluated in this study

## 5 Results

**Exploratory Data Analysis** To visualize the extracted features from the images through ResNet, ViT and CLIP, we randomly selected ten classes from the dataset and performed Principal Component Analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) visualization in 2D. Fig. 4 shows the results. Evidently, some classes such as bus-interior are better separated than others as observed in PCA and t-SNE. Based on this preliminary exploration, it seems CLIP features best cluster the images based on their labels. It is worth noting that both t-SNE and PCA are unsupervised approaches, and we add in labels after clustering.

**Results on Places100 Dataset** Fig. 5 shows the test accuracy of the nine total combinations of the pre-trained deep learning models and the light-weight classifiers on the Places100 dataset. We also overlay the baseline result of a randomly initialized ResNet-18 model trained on the Places100 dataset(ResNet-18) as well as the test accuracy of a pre-trained ResNet-18 model trained on the Places100 dataset (ResNet-18 w/ Pretraining) for comparison. The accuracy of the 3-layer MLP (NN) as a light-weight classifier varies with different numbers of training epochs. This is because we allow backpropagation of the MLP weights, while locking the weights of the pretrained deep learning models. The other light-weight classifiers' outputs (i.e. SVM [1] and XGBoost [3]) do not vary with different numbers of training epochs. Hence, their accuracy is plotted as horizontal lines.

For all the three deep learning models we used as image feature extractors, unsurprisingly, the 3-layer MLP behaves as the best light-weight classifiers. Within at most 40 training epochs, all the three combinations of pre-trained deep learning model and the MLP neural network achieved above 50% testing accuracy. This result matches the accuracy records on the Places365 dataset for end-to-end trained deep learning models. The next two best-performing light-weight classifiers are SVM and XGBoost. The relative performances of these two light-weight classifiers alternate. With pre-trained ViT, XGBoost's testing accuracy is higher than that of SVM approximately 20% accuracy,

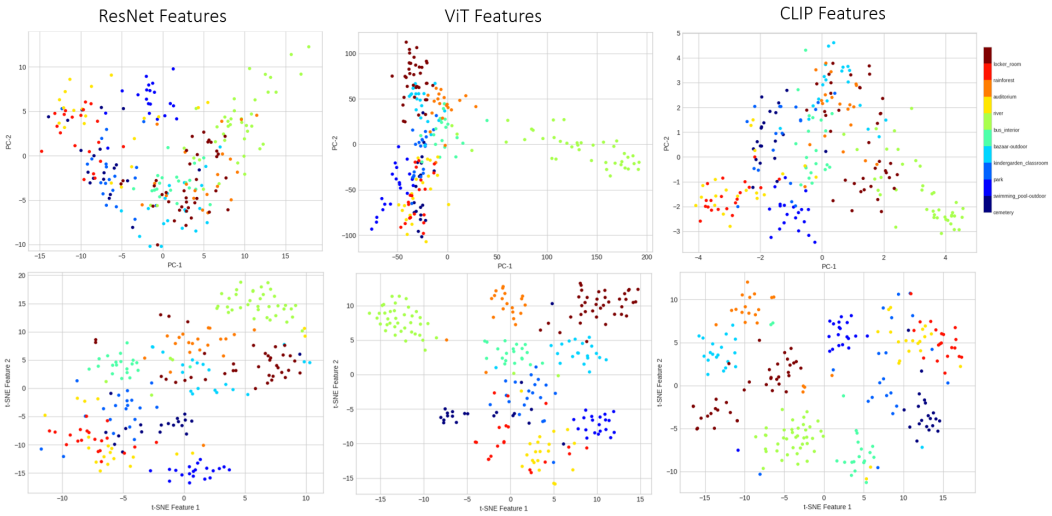


Figure 4: Top row: Principal Component Analysis (PCA) of selected classes of pretrained model features. For both ResNet and ViT, the first two PCs accounted for 18% of variances while for CLIP features, first two PCs acounted for 15% of variances. Bottom row: t-distributed stochastic neighbor embedding (t-SNE) visualization of the features.

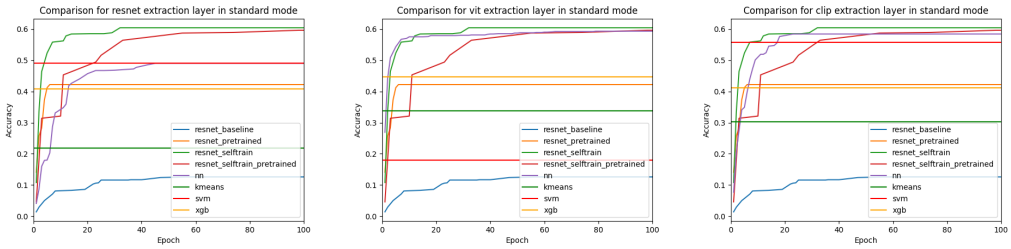


Figure 5: Testing accuracies of all the nine heavy-light combinations trained on the Places100 dataset along with the baseline results. The left, middle, and right images correspond to combinations with pre-trained ResNet-101, ViT, and CLIP, respectively.

whereas with pretrained ResNet-101 and pre-trained CLIP, SVM outperforms XGBoost. Surprisingly, SVM's testing accuracy beats that of the MLP neural network with a pre-trained ResNet-101. The performance of SVMs and XGBoost are more sensitive to the type of input features than MLPs.

All the nine heavy-light combinations yielded higher testing accuracy than our baseline result of the randomly initialized ResNet-18 model trained on the Places100 dataset. Moreover, seven out of the nine combinations—namely, ResNet-101 + MLP, ResNet-101 + SVM, ResNet-101 + XGBoost, ViT + MLP, ViT + XGBoost, CLIP + MLP, CLIP + SVM, CLIP + XGBoost—produced higher testing accuracies than our baseline result of pre-trained ResNet-18 trained on the Places100 dataset. These results align with our expectation that the pre-trained features significantly improve the performance of the lightweight classifiers due to the large quantity of data that provide them with robust image feature representations.

To better understand the source of mistakes for the nine combinations, we constructed confusion matrices for each of the models. Fig. 5 shows the confusion matrices for each of the models. Of note, we can observe that the misclassifications are different among different combinations. Hence, a promising improvement based on this analysis is to further combine these cascaded heavy-light models as ensemble models.

**Results on Open-Places100 Dataset** Fig. 7 shows the testing accuracy of the nine heavy-light combinations trained on the Open and the self-trained models. Baseline model performed poorly in this case, failing to predict outliers that did not exist in the training data. However, our proposed

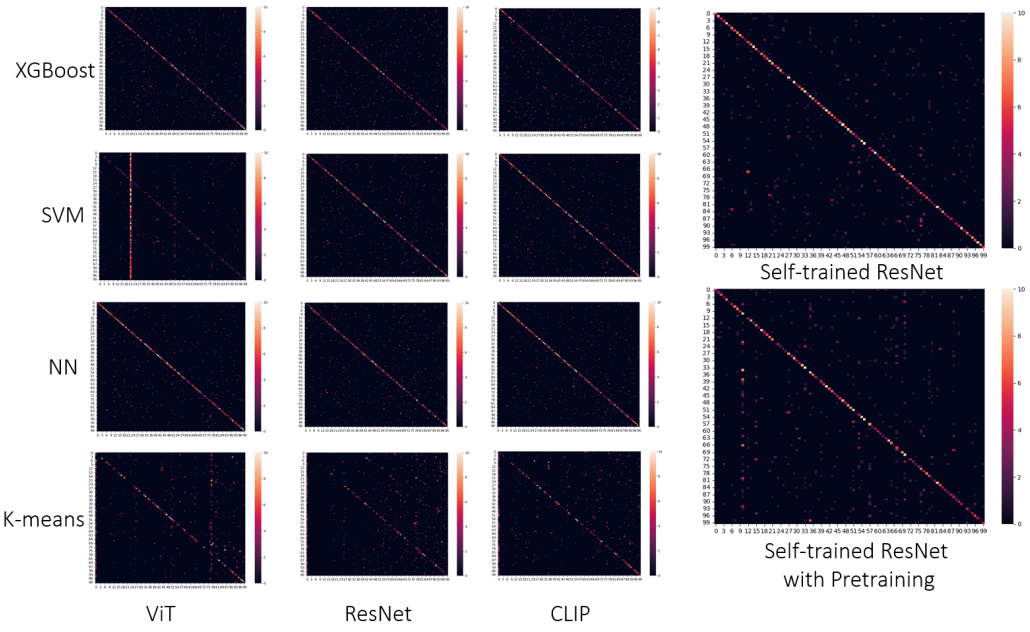


Figure 6: Confusion matrix for the nine heavy-light combinations and self-trained ResNet-18

models yielded surprisingly high accuracies within 10-20 epochs of training. We note that self-trained models outperformed both our other proposed models and baseline with ResNet features, indicating higher resilience to poor instance feature representation. Our heavy-light combinations performed relatively poorly at this task, as well, achieving between 10% and 30% accuracy, significantly lower than the standard setup numbers. We suspect this is due to the pre-training objective of the deep classifiers being significantly different than the OpenSet task, making them less suited to generalize to this task. Overall, the self-trained models performed the best out of all of our baselines and heavy-light combinations, even at early training epochs.

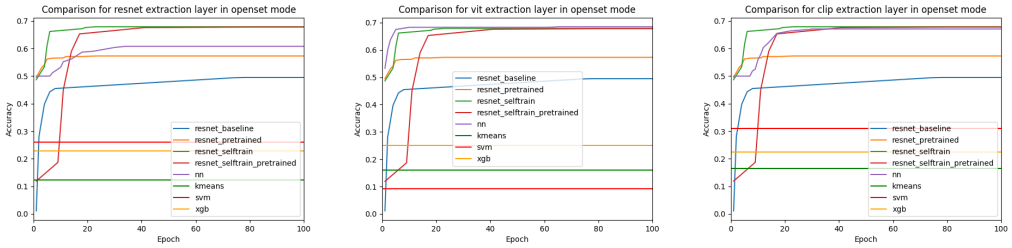


Figure 7: The open-set testing accuracies for all nine heavy-light combinations. From left to right, the pre-trained deep learning models are Resnet-101, ViT, and CLIP.

## 6 Discussion and Analysis

The most representative results in this study are the testing accuracy of the heavy-light combinations on the Places100 dataset, as shown in Fig. 5. These results verify our assumption that the generalization power granted by the enormous amounts of data used to train the pre-trained deep learning models indeed complements the simplicity of the light-weight classifiers. Even without backpropagating through the entire deep learning model, these heavy-light combinations attained similar and superior performances compared with the baseline ResNet-18 model trained in an end-to-end manner. An interesting phenomenon with the end-to-end trained ResNet-18 is that the pre-trained ResNet-18 with ImageNet weights produced lower testing accuracies as the number of training epochs increases on the Places100 dataset. This could be caused by the small size of the Places100 dataset and the

| Feature-Extractor | Model | Accuracy |
|---|---|---|
| N/A | ResNet-18 | 12.60% |
| N/A | ResNet-18 (Pretrained) | 42.20% |
| N/A | ResNet-18 w/ Self-Training | **60.40%** |
| N/A | ResNet-18 w/ Self-Training (Pretrained) | 59.60% |
| CLIP | NN | 58.40% |
| CLIP | KMeans | 30.30% |
| CLIP | SVM | 55.70% |
| CLIP | XGBoost | 41.10% |
| ResNet-101 | NN | 49.00% |
| ResNet-101 | KMeans | 21.90% |
| ResNet-101 | SVM | 49.10% |
| ResNet-101 | XGBoost | 40.80% |
| VIT | NN | 59.30% |
| VIT | KMeans | 33.80% |
| VIT | SVM | 17.90% |
| VIT | XGBoost | 44.70% |

Table 1: Best Accuracies for All Models on Standard Setup

| Feature-Extractor | Model | Accuracy |
|---|---|---|
| N/A | ResNet-18 | 49.50% |
| N/A | ResNet-18 (Pretrained) | 57.30% |
| N/A | ResNet-18 w/ Self-Training | **67.90%** |
| N/A | ResNet-18 w/ Self-Training (Pretrained) | 67.80% |
| CLIP | NN | 67.10% |
| CLIP | KMeans | 16.40% |
| CLIP | SVM | 31.10% |
| CLIP | XGBoost | 22.50% |
| ResNet-101 | NN | 60.80% |
| ResNet-101 | KMeans | 12.40% |
| ResNet-101 | SVM | 26.10% |
| ResNet-101 | XGBoost | 22.80% |
| VIT | NN | 68.50% |
| VIT | KMeans | 16.00% |
| VIT | SVM | 9.30% |
| VIT | XGBoost | 25.00% |

Table 2: Best Accuracies for All Models on Open-Places100 Setup

property that deep learning models such as ResNet-18 tends to overfit, hence large amounts of data are required to compensate for the overfitting.

Our self-training model also outperformed our baseline models of ResNet-18 and ResNet-18 with pretrained weights when evaluated at Epoch 100. These results verify that ensembling our nine models together and using them to label simulated "unlabeled" data can enhance the overall quality of training a ResNet model. We suspect the reason this model did not outperform our lightweight classifiers is that 10,000 images is still far too small of a dataset. Typically, to train a deep model and obtain meaningful results, one would need hundreds of thousands to millions of images in the train set.

To understand the source of misclassifications in our self-trained model, we looked at the most common label-prediction mismatch in Tab. 3 (ranked based on number of misclassifications). The misclassification rate is calculated as the percentage of specific label-prediction mismatch in the set of images of a label. We can note that all of the most common misclassifications seem to be tied to label assignment arbitrariness. For instance,labels “Bedroom” and “Bedchamber” are synonyms and the images in their respective classes do not show many distinguishing features from each other

| True Label | Predicted Label | Misclassification Rate |
|---|---|---|
| Food Court | Dining Hall | 60.0% |
| Pond | Swamp | 50.0% |
| Tree Farm | forest Broadleaf | 83.3% |
| Clean Room | Storage Room | 100.0% |
| Bar | Discotheque | 57.1% |
| Pier | Boardwalk | 66.7% |
| Bedroom | Bedchamber | 57.1% |
| Indoor Flea Market | Storage Room | 44.4% |
| Youth Hostel | Bedroom | 42.9% |
| Amusement Park | Carrousel | 33.3% |

Table 3: Self-trained ResNet-18 Most Common Misclassifications

despite their different labels. As noted earlier, these class distinctness can be visualized with PCA and t-SNE plots (Fig. 4). This explains a significant portion of the misclassifications from our models and may also be interpreted that the models are not overfitting to specific labels but rather effectively learning shared visual features in a labeled class.

Inconsistent with our assumption, the majority of heavy-light combinations performed poorly on the Open-Places100 dataset. A closer look reveals that this outlier detection task is completely different than the scene recognition task for which these deep learning models were pre-trained. In other words, the pre-trained deep learning models could be unsuited for outlier detection task despite their large training data because the class "outlier" never appeared during their training. Hence, when cascaded with the light-weight classifiers, the image features extracted from the pre-trained deep learning models are significantly less informative for the light-weight classifiers to identify the outlier class. Yet, we note that the self-trained "student" network performed remarkably well with the Open-Places100 setup, showcasing our self-training method's ability to distill useful information from the heavy-light combination models and achieve good performance even with this difficult task.

## 7 Conclusion

This study explored the scene classification problem with MIT Places365 dataset with two distinct and novel approaches: pretrained feature-classifier combinations and a self-training framework. We then evaluated these models on two types of datasets: standard 100 class subset of the Places365 and the 50 known classes and 1 outlier class setup. Based on pre-trained feature extractors, we studied the approach of training classification models that do not require large training datasets involved in transfer learning or training from scratch. We showed that these models vastly outperformed the baseline ResNet-18 model trained from scratch in the standard dataset. We also showed that the self-trained ResNet with generated labels performed better in the standard set than any of the baseline and feature-classifier combinations while performing similarly in the open-set setup, showing promise for semi-supervised applications where large sets of labeled data are not available.

## References


[1] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*. Ed. by David Haussler. ACM, 1992, pp. 144–152.

[2] Lorenzo Brigato and Luca Iocchi. "A Close Look at Deep Learning with Small Data". In: *2020 25th International Conference on Pattern Recognition (ICPR)* (2021), pp. 2490–2497.

[3] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *CoRR* abs/1603.02754 (2016).

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image

Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2021.

[5] Joris Guérin, Stéphane Thiery, Eric Nyiri, Olivier Gibaru, and Byron Boots. "Combining pretrained CNN feature extractors to enhance clustering of complex natural images". In: *Neurocomputing* 423 (2021), pp. 551–571. ISSN: 0925-2312.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[7] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[8] Ali Hasan Md. Linkon, Md. Mahir Labib, Faisal Haque Bappy, Soumik Sarker, Marium-E Jannat, and Md Saiful Islam. "Deep Learning Approach Combining Lightweight CNN Architecture with Transfer Learning: An Automatic Approach for the Detection and Recognition of Bangladeshi Banknotes". In: *2020 11th International Conference on Electrical and Computer Engineering (ICECE)*. 2020, pp. 214–217.

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763.

[10] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. "Self-Training With Noisy Student Improves ImageNet Classification". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 10684–10695.

[11] Bolei Zhou. *Performance of the Places365-CNNs*. `https://github.com/CSAILVision/places365`. Accessed: May 1 2022.

[12] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 Million Image Database for Scene Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.6 (2018), pp. 1452–1464.