Stool Classification Using Deep Metric Learning

Eric Nguyen
Cornell Tech
M.Eng. Computer Science
en274@cornell.edu

Chris Kruger
Cornell Tech
M.Eng. Computer Science
crk78@cornell.edu

Skyler Erickson
Cornell Tech
M.Eng. Computer Science
mse53@cornell.edu

Abstract

Chronic gut disorders in the U.S. are commonplace, however, they lack convenient and accurate measurement tools. We propose a system that allows a person to take a picture of their stool and extract clinical data from the image to track their symptoms. To train this system, we crowdsourced a stool dataset in the wild through http://train.auggi.ai and had physicians annotate them with the Bristol Stool Scale (BSS). The dataset gathered is highly imbalanced, with type 1 having just 6 images. We trained a ResNet18 classifier to predict the BSS, and then used deep metric embeddings (triplet and contrastive loss) with the goal of improving the performance of the low-shot classes. Triplet and contrastive loss are evaluated as potential improvements on our baseline ResNet18 models due to their additional information beyond purely vision. Triplet loss uses an anchor, positive, and negative to push similar images closer and dissimilar images further apart. Contrastive applies the same concept with pairs instead of triplets. Our ResNet18 trained model showed promising results with a mean per class absolute deviation of 0.82 BSS, which means on average, the model was within one BSS value from the true label, reasonable for clinical applications and a large improvement over what patients are able to do on their own. Compared to GI physicians, our deep learning models were not as accurate. Amongst the deep learning techniques, the ResNet18 performed the best, while the metric learning models were not able to improve performance over the ResNet18. This may be due to the lack of unlabeled data that help the metric models create embeddings that best separate the classes. With a modest increase in the data in these few labeled classes, we expect the performance to significantly improve and be more in-line with what doctors are able to predict from stool images.

1. Introduction

One in five Americans suffer from some form of chronic gastrointestinal (GI) disorder [1]. Managing treatment for these disorders, such as irritable bowel syndrome, often requires tracking symptoms and lifestyle patterns over time. These patient-reported inputs are often subjective and make it difficult to control their symptoms. We propose a model that can allow patients to take a picture of their stool and automatically extract clinical data from the image. This allows patients to track the physical manifestation of their disorder over time in a convenient way. This system also allows doctors to view objective and standardized metrics about the state of their patient's digestive system.

One standardized metric doctors use for GI health tracking is the Bristol Stool Scale (BSS). Based on a visual assessment, doctors assign a score on a 1 to 7 point scale to quantify the consistency of a patient's stool. The BSS scale is known to correlate well with transit time of food in the digestive system and identifies the presence of constipation and diarrhea, and precursors to colon cancer[2].

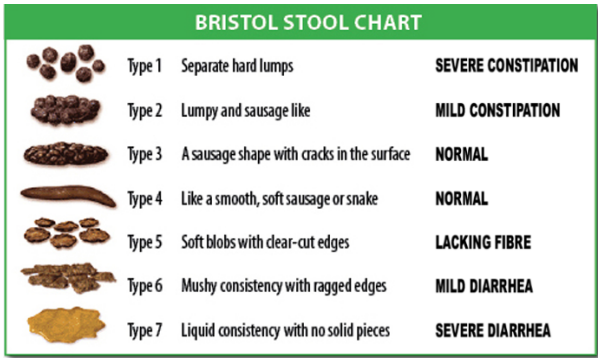

Figure 1. Bristol Stool Scale (BSS)

By leveraging deep learning, and specifically convolutional neural networks (CNNs), we evaluated computer vision approaches that can augment physicians' efforts to classify stool images based on their BSS value. As part of this effort, we have obtained 886 images of stool in the wild. These images were collected from both internet searches and crowdsourced images from our website (http://train.auggi.ai). The majority of the images were taken above a toilet with the camera aimed straight down

at the stool. Each image in our created dataset includes a BSS score, annotated by three physicians who specialize in gastrointestinal disorders. The images vary greatly in terms of sizes, view points, image quality and other characteristics.

The research outlined in this paper is an extension of work done for a project and startup called Auggi. Previous work focused primarily on semantic segmentation of stool (vs not stool) from an image. The current, in-place solution reaches a satisfactory intersection over union (IoU) score of 82%. However, the classification task of predicting the BSS proved challenging.

The dataset of stool images is highly imbalanced, having mostly BSS scores of 3-5, and very few instances for the other classes (1, 2, 6, 7) of the BSS. For example, there were only 6 instances of type 1, indicating severe constipation. The current model used at Auggi counteracts this by classifying on what we called a consolidated Bristol Stool Scale, which aggregates the BSS to 3 classes instead. Even with this simpler classification task, the current model was only able to achieve an overall accuracy of 62%. Note: this classification was done on the whole image, and the segmentation mask was not used. The main deficiency in our model were the few instances of severe constipation and diarrhea.

In this paper, we aim to improve on the current Auggi classification model using dataset manipulation (artificially rebalancing) and deep metric learning. Our experiments consist of two baseline models using transfer learning with a ResNet18 for comparison, and two techniques to improve upon these baselines.

The next two experiments will use representation learning with triplet constraints. In our proposed system we will strip the final softmax layer off our baseline network and treat our CNN as a feature extractor instead of a self contained classifier. We will evaluate triplet and contrastive loss in their ability to optimize our feature extraction process to create optimal embeddings. These feature embeddings will then be passed into a nearest neighbors classifier and return a majority vote on expected class.

2. Related Work

2.1. Domain Review

In recent years, CNNs have been applied to many domains within the medical imaging field. Among the more widely researched and relevant areas is cancer detection and diagnosis. Though not specific to GI disorders and gastroenterology, the tasks involved in cancer detection are similar to our use case for stool classification: localizing the region of interest (or suspicious areas) and then specific classification of this region.

For breast cancer detection, a DenseNet model was used on 9,109 histopathological slide images and showed an accuracy of 95.4% for multi-classification of breast tumors [3]. Similarly, CNN and Long Short Term Memory (LSTM) models were able to achieve an accuracy of 91% for breast cancer classification on the same dataset [4].

To detect lung cancer, a semantic segmentation model U-Net was used to identify pulmonary nodules, followed by a CNN to classify the malignancy [5], [6]. The data was gathered from a Kaggle challenge which included low dose computed tomography scans. The same approach for segmentation was used by another group, but instead used a GoogleNet for classification.

For both breast and lung cancers, standardized medical images were used as the training set. For skin cancer detection, RGB images in the wild have been shown to be highly accurate as well. A CNN model trained at Stanford was able to match dermatologist level accuracy for the detection of various skin cancers [7]. This model was pretrained on general everyday objects and then fine-tuned on approximately 130,000 training images. Their goal was to one day enable a mobile application to increase the accessibility of health care technology in an affordable way.

This subset of research in cancer detection shows that given enough data, CNNs can be highly useful, and perhaps outperform human capabilities in detection and classification. However, for small or imbalanced datasets, CNNs struggle to learn enough from the limited dataset. This is particularly relevant for more rare diseases or data that is not often collected.

Given the difficulty in stool collection and highly sensitive nature of its images, there exists no large database of stool images. In addition to an our ongoing effort to build such a database, the scope of this research proposal consists of two goals. The first is to continue driving the application of deep learning in an understudied area of gastroenterology: stool classification. The second goal is to leverage and apply the latest techniques in deep learning that aim to remedy this small and imbalanced dataset challenge, which can be applied to other important medical specialties as well.

2.2. Methodology Review

Using a CNN as a feature extractor to drive classification is well charted territory. Past work has aimed on learning convolutional nonlinear features that can fit directly into a kNN classifier in a representation learning framework [8]. The concept of re-thinking the training of convolutional feature embeddings has proven successful for implementations beyond classification as well. One novel algorithm, SNE and Crowd Kernel (SNaCK) Embeddings has seen promising results by combining expert triplet hints with convolutional learning processes [9]. We hope to expand upon the value generated by learning better feature embeddings and apply them to our BSS classification task.

To learn better feature embeddings we will be doing a deeper exploration of the loss functions that drive improvements seen in aforementioned work. One of the most foundational loss functions in this area of work comes in the form of triplet loss. In this framework, expert triplets are passed in for each training mini-batch to group similar data points together and push differing ones apart - proven to be successful in the application of facial recognition [10].

While we see triplet loss implementation as an important first step in our evaluation of metric learning, a variety of shortcomings have been noted in its basic implementation [20][12]. One proposed alternative to vanilla triplet loss is a lifted loss deep feature embedding learning regimen that optimizes the objective of lifting a dense pairwise matrix within each mini-batch [20]. As reported, these efforts typically see a 3-4 percent increase in performance on standardized data sets.

Additional alternative loss metrics have been seen in the contrastive loss function. Contrastive loss has been effective at improving expert triplets by imposing pair based training instead of triplets [17].

For our proposed areas of work we focused on the core triplet loss function and then evaluated contrastive loss as an extension of potential performance-improving modifications.

3. Proposed Method

3.1. Image preprocessing and baseline classification

The focus of this paper is on classification, and therefore given a stool image from the wild, we assumed accurate localization and bounding boxes. We used these bounding boxes to crop and classify images without much of the background noise.

In our first baseline model, we used transfer learning on a ResNet-18 model pretrained on ImageNet. In our second baseline model, we rebalanced the dataset artificially by oversampling, with duplication, of the least common classes, and undersampling the most common class. This addresses the potential bias the dataset forces a model to learn for predicting the most common class. These will serve as baseline accuracies for the BSS predictions. The figure below shows the distribution of the original and rebalanced dataset:

We evaluated our ResNet-18 model on varying amounts of duplications for classes 1, 2, and 7, while class 4 was reduced by half (at random). The best rebalanced dataset performance achieved on ResNet-18 were class 1 : 5x duplication (created 5 copies of the images in this class), class 2 : 2x, and class 7 : 2x.

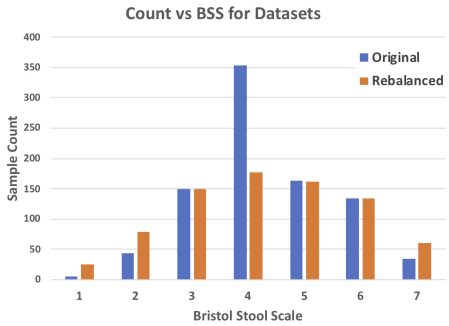

Figure 2. Bristol Stool Dataset Class Distribution

3.2. Classification with Metric Learning

To improve on these baseline models, we built upon prior work in the domain of metric learning for the purposes of classification. From a high level, our approach captures the notion of similarity through calculating the Euclidean distance between generated stool feature embeddings. Prior work has shown that this approach can be beneficial for imbalanced classes [14], which is very much present in our BSS classification task. We pursued our goal of optimal feature embeddings by training CNN feature extractors through a variety of loss functions. We evaluated the improvements via expert triplets by fine-tuning pretrained ResNet-18 models with triplet and angular loss.

3.2.1 Embedding Optimization via Triplet Loss

Triplet loss was originally developed in applications for facial recognition [10] and is used to learn optimal embeddings (or encodings) for a given class in a low shot learning application. In the original implementation, faces from the same person should be close together and form well separated clusters in the embedding space. The goal of the triplet loss is to make sure that a). two examples with the same label have their embeddings close together in the embedding space and b). two examples with different labels have their embeddings far away [15]. A loss is then defined over triplets of embeddings that consist of an anchor image, a positive image (same class as the anchor), and a negative image (different class than anchor). For some distance on the embedding space d, the loss of triplet (a, p, n) is:

$$l_{tri}(\mathcal{T}) = \left[\|\mathbf{x}_a - \mathbf{x}_p\|^2 - \|\mathbf{x}_a - \mathbf{x}_n\|^2 + m\right]_+,$$

Figure 3. Triplet Loss Formula

Along with its corresponding gradient calculations:

Where d(a,p) and d(a,n) are the distance between the anchor and the positive, and the anchor and the negative respectively. As we minimize the loss, d(a,p) is pushed to 0 and d(a,n) approaches a value greater than d(a,p) + margin

$$\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_n} = 2(\mathbf{x}_a - \mathbf{x}_n),$$
$$\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_p} = 2(\mathbf{x}_p - \mathbf{x}_a),$$
$$\frac{\partial l_{tri}(\mathcal{T})}{\partial \mathbf{x}_a} = 2(\mathbf{x}_n - \mathbf{x}_p),$$

Figure 4. Triplet Loss Gradients

[15]. As soon as the negative image becomes a large negative, the loss becomes zero.

To mine useful triplets for our training process, we will use an online mining approach. For each batch B that we train, we will compute the embeddings which will provide possible triplets. Of these possible triplets, we know that only only a subset containing two positive and one negative image is valid. We will choose the hard triplets where the distance between positive and anchor is the largest and use these P*K embeddings (where P is the number of classes and K is the number of samples) to calculate a loss [15].

We anticipate that training a classifier with this loss function will produce better vector embeddings than a classifier based on cross-entropy loss. These vector embeddings can then be used as inputs to a softmax classification layer. This approach should yield better results given the unbalanced nature of our training data. The preferred alternative that we will explore in our methodology will be to use the optimized embeddings in a fast k nearest neighbor implementation run against the vector embeddings of all of the training data.

3.2.2 Embedding Optimization via Contrastive Loss

Building upon our initial BSS kNN classification approach using embeddings trained with triplet loss, we will also explore the value of optimizing our image embeddings with contrastive loss. Triplet loss and metric learning in general have been at the forefront of many advances in the computer vision space, but they come with quite a few limitations. Some shortcomings of note include: poor SGD convergence due to needing to explore all possible triplets in cubic space (n3), likely incorrect application of a one size fits all global distance margin m, and triplet gradient calculations that only take in pairwise relationships between points instead of evaluating all points together. [12] See prior section for triplet gradient formulas.

We believe that there is a good chance the aforementioned limitations may have affected our performance on BSS image classification. To explore improvements on these potential shortcomings we not only implemented triplet loss during training, but contrastive loss as well.

Contrastive loss is a distance-based loss function that takes a slightly different approach towards optimization via pairs instead of triplets. Contrastive and triplet loss are similar in that they both focus on distance learning instead of classification error learning as seen in more standard loss functions. However, by focusing on pairs instead of triplets, contrastive loss minimizes the exploration space needed for each step and potentially reach convergence in faster time.

To map high dimensional features to low dimensional space with contrastive loss we move away from conventional learning systems such as summing errors across samples and instead do it by individual pairs. The pairs in this case are comprised of 2 input feature vectors in a high dimensional space. These two feature vectors will be compared via a distance metric to result in a binary variable Y being set to Y = 1 if the two vectors are similar or Y = 0 if the two vectors are dissimilar [17].

As in triplet loss, contrastive loss will have the notion of a margin that is used to threshold how vectors are grouped together or pushed apart. This margin will be represented by 'm' and interacts with a set of learned parameters 'W' that work to minimize the distance function 'D' for similar vectors and maximize 'D' for dissimilar vectors. The exact formula for contrastive loss is given as:

$$L(W, Y, \vec{X_1}, \vec{X_2}) =$$
$$(1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Figure 5. Contrastive Loss For Optimal Embeddings

The margin in the above formula is used to control the contribution that dissimilar pairs have to the overall loss function. It works by creating an envelope around the anchor point to determine what distance qualifies as dissimilar vs similar. This is helpful to control the performance of the model depending on dataset. An example of how contrastive loss performs with similar versus dissimilar vectors can be seen below:

By taking the contrastive loss via pairs approach we hope to improve upon our findings seen in our baseline experiment as well as optimization through triplet loss.

4. Datasets

Our experiments used two datasets to both evaluate our methodology with an established baseline, as well as explore incremental benefits in the BSS classification task.

4.1. CUB-200-2011

The CUB-200-2011 (CUB-200) dataset will serve as a baseline to evaluate our model performance using both triplet and contrastive loss metric optimization. CUB-200 is comprised of 200 species of birds with 11,788 individual images, along with robust annotations of each image [18]. For the purpose of this exercise we will be focusing solely

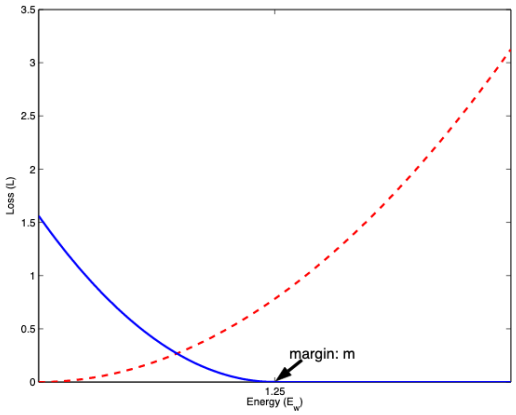

Figure 6. Graph of contrastive loss function against energy (distance). The dashed (red) line is the loss function for the similar pairs and the solid (blue) line is for the dissimilar pairs.[17]

on images and their classifications without use of supplemental annotation material.

Figure 7. CUB-200 Example Images

CUB-200 will be used to evaluate the performance of our triplet and contrastive loss models. These performance metrics will serve as a baseline relative to our performance on the stool dataset.

4.2. Auggi Stool

Our Auggi stool dataset is comprised of 886 total images in jpg and png formats. The annotations include segmentation and bounding box coordinates and BSS annotated by three physicians. The stool dataset is broken out with the following distribution, along with an example binary mask of one sample image following:

Bristol Stool Scale	Data Count
1	6
2	44
3	150
4	354
5	163
6	134
7	35
Total Samples	886

Figure 8. Sample Binary Mask of Stool

5. Experiments Evaluation Metrics

5.1. Metric Learning Baseline - CUB-200

For our metric learning techniques baseline, we applied triplet and contrastive loss to the CUB-200 dataset. We followed the lead of prior papers to determine the optimal evaluation metrics on our baseline with cluster F1 scores [20] and recall at different k values [19]. These evaluation metrics fit our nearest neighbors use case better than standard accuracy as we have the additional hyperparameter of neighbor selection. Additionally, we evaluated against these metrics as they allowed a more condensed view of model performance. The below chart outlines our results with embeddings optimized through triplet and contrastive loss, with recall at varying k values:

Method	**F1 Score (%)**	**Recall @ k (%)**			
		k=1	**k=2**	**k=4**	**k=8**
Triplet Loss	21.3	43.4	49.4	58.6	63.9
Contrastive Loss	21.9	44.1	50.9	59.0	64.5

Table 1. Clustering F1 Score and Classification Recall for CUB-200-2011 dataset.

We will use recall for each K (nearest neighbor values) and F1 score to evaluate this CUB-200 baseline dataset. In general, CUB-200 recall improves with a greater number of neighbors. This makes sense and is possible due to the size of the dataset - 11,788 images across 200 classes means that there are up to 8 valid neighbors to be matching to. Unfortunately since our stool dataset is a considerably lower

shot problem, we primarily used the CUB-200 baseline to ensure our model was working correctly. In our experiments we were able to achieve roughly the same results as seen in prior works [12] and therefore feel confident in our code design.

5.2. Stool Classification - Evaluation Metrics

For stool classification, we evaluated each of the 4 techniques (fine-tuned ResNet-18, fine-tuned ResNet-18 with rebalanced dataset, triplet, and constrastive losses) using mean per per class accuracy, mean average precision (mAP) and mean per class absolute deviation. The mean per class absolute deviation was used because we care not only if the BSS prediction is correct or not, but also by how much.

We also included results from a stool classification study by physicians who provided their visual prediction for 34 BSS images. There are several ways to evaluate performance for physicians, but we chose to calculate accuracy by using the majority vote class as the absolute ground truth, and predictions by doctors outside the majority vote were considered "incorrect". It's worth noting that the dataset in this study was entirely different from the images "in the wild" that is used in our study, however, it allows us to have some level of comparison between clinical measurements.

6. Results Performance Discussion

6.1. Stool Classification - Aggregate Results

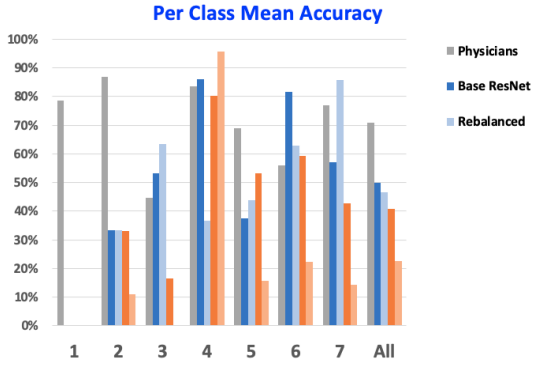

Figure 9. BSS Mean Per Class Accuracy

The physicians performed fairly similar across each of the 7 classes, but this was due to the roughly balanced nature of their dataset. For our stool data in the wild dataset, the trend was clear that for classes with fewer samples, the performanced dropped significantly on mean per class accuracy and mean per class absolute deviation.

All four deep learning approaches achieved less accurate results than what physician GI specialists were able to predict. Amongst the four techniques, the best performing model was the ResNet18 model using original (unbalanced) dataset. Using the (artificially) rebalanced dataset did not improve the mean per class accuracy, but it did slightly improve the mean per class absolute deviation. The metric learning approaches performed significantly worse from the ResNet18 model on original unbalanced dataset.

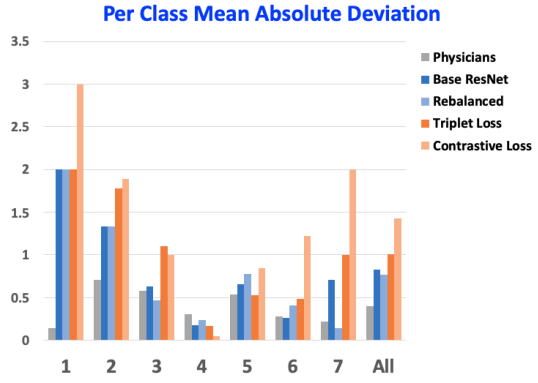

Figure 10. BSS Mean Per Class Absolute Deviation

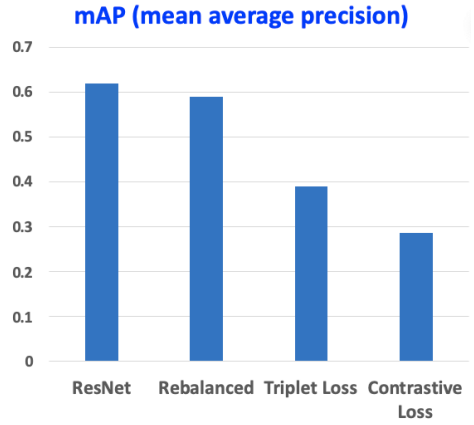

Figure 11. mAP Score

In particular, the metric learning approaches did not perform as we had hoped on these few labeled classes. Some possible explainations for this shortfall may include the lack of more unlabeled examples. Although metric learning can work well in low-shot learning situations of few labeled examples, it works better when it has a number of unlabeled examples to help the model learn embeddings that best separate the classes. We simply did not have more unlabeled images to help the metric models learn separable embeddings.

6.2. ResNet18 Performance

The initial ResNet18 model achieved a mean per class accuracy of 49.8%. This model had difficulty predicting BSS Classes 1 and 2, which had scores of 0% and 33.3%, respectively, and brought the performance down significantly. This is not surprising given the low count for these classes in the test set. Class 1 had only 1 sample in the test set, while class 2 had 9 samples. It's worth noting that Class

7 had 7 samples but was able to achieve 57.1%. The most common class is class 4, and it achieved 85.9%, the best amongst all classes. Using mean average precision (mAP), the Rebalanced technique scored 0.59.

When evaluating performance using the mean absolute deviation per class, the results showed more promise. Across all classes, the mean absolute deviation was 0.76 (BSS value), which means the predicted BSS score was off by less than one BSS score. Given there is some visual subjectivity between physicians who annotate the ground truth to images (and real samples), is within reason to be within 1 BSS value between doctors, and not have complete conformance of the BSS value for a sample. A study by Blake et. al. [2] showed physician GI specialists have a mean absolute deviation of 0.39 BSS value.

To achieve these results, freezing 70% of the network's layers balanced the transfer learning from ImageNet data, while still allowing the network to fine-tune on the stool dataset. Experiments varying the degree of layers frozen were tried, along with an exhaustive hyperparameter search. We used Stochastic Gradient Descent with mini-batch, and cross-entropy to evaluate the loss. We also used data augmentation (random horizontal and vertical flip, random crop and color jitter).

6.3. ResNet18 with Rebalanced Dataset Performance

The ResNet18 with rebalanced dataset (Rebalanced) was able achieve a mean per class accuracy of 46.5%, which dropped from the non-balanced dataset. For most classes the accuracy dropped, although notably, for class 7 it improved. The per class absolute deviation did have an improvement that dropped to 0.77 BSS vs. 0.82 BSS (lower deviation is better). This suggests that although the rebalanced set was correct less often, on the whole, its predictions were closer to the ground truth than the non-balanced ResNet18 model. The mAP score also dropped to 0.59.

For the ResNet18 rebalanced experiment, the same hyperparameters for the unbalanced dataset proved the best performing. This includes the same number of layers frozen (70%), as well as stochastic gradient descent, cross-entropy loss and data augmentation (random horizontal and vertical flip, random crop and color jitter).

6.4. Triplet Loss

The original unbalanced dataset (with horizontal and vertical flips, random crops, and color jitter augmentations) was fed through the triplet loss implementation [21] and optimized with a thorough grid search. The best results were found using a ResNet18 backbone in which the first four layers were frozen. In addition, the model was trained for 30 epochs, a learning rate of 0.0004, and a batch size of 12. A margin of 1.5 was found to be optimal to train the triplet loss classifier. The results are detailed as below:

Method	**F1 Score (%)**	**Recall @ k (%)**			
		k=1	**k=2**	**k=4**	**k=8**
Triplet Loss	45.2	45.2	41.8	55.4	55.4

Table 2. Clustering F1 Score and Classification Recall for Triplet Loss Embedding Optimization on Stool dataset.

As compared to the Cub-200 dataset, the recall scores look similar. The k=1 recall is almost exactly the same, however the k=8 recall on the Stool Dataset does not achieve the same results, peaking at 55.4. It's worth noting that F1 scores are nearly double for the Stool Dataset over the Cub-200 dataset.

As Figure 9 shows, the mean per class accuracy of the triplet loss implementation yielded subomtimal results. This accuracy across all classes for the best KNN implementation (where K=8), yielded only an accuracy of 0.35. The mean per class absolute deviation also appears to be worse (higher) than the Base ResNet18 implementation, achieving a 1.01 and .82 deviation respectively. The mean average precision for the triplet loss implementation was 0.40, which is also below the base ResNet18 implementation 0.62.

As noted above, the relatively small number of classes for this dataset resulted in less than optimal results for the triplet loss implementation. Specifically, the model seemed to do well for classes in the middle of the Bristol Stool Scale (3-5), but it performed particularly poorly against the extremes of the scale (1-2, 6-7). These extremes are also the classes where there is signficantly less training data, which demonstrates that the triplet loss did not seem to perform as well as expected for our low shot learning case.

It was hypothesized that perhaps an implementation where the negative anchor was made to be +- one from the positive anchor would help during training time, thus forcing the model to learn "hard" samples, and differentiate from similar classes. Unfortunately this resulted in worse results than the random sampling of negative anchors of the triplet loss function. The greatest handle on model accuracy seems to come from the margin that the loss function drives all positive - negative anchors to be below. Small changes in the margin appear to have a large impact, and at a value of 1.5 the loss seems to converge with stability until it plateaus after about 30 epochs.

6.5. Contrastive Loss

As a whole, the contrastive loss [21] performed the worst of all of the different loss function implementations we experimented with. The k=1 recall scores are again similar to the Cub-200 dataset, however the k=8 recall performs even worse than the triplet loss implementation. The F1 score does still however remain high (46.9) as opposed to the recall reported by the CUB-200 dataset (21.9). The best

model implementation was achieved by freezing the first 5 layers of the ResNET18 backbone, using 128 features for the vector embedding, training over 30 epochs with a batch size of 12. The margin was again set to 1.5 as this appeared to be the best for the contrastive loss as well.

Method	F1 Score (%)	Recall @ k (%)			
		k=1	k=2	k=4	k=8
Contrastive Loss	46.9	46.8	40.7	45.2	47.5

Table 3. Clustering F1 Score and Classification Recall for Contrastive Loss Embedding Optimization on Stool dataset.

The mean per class accuracy, and per class mean absolute deviation were also the lowest of all the implementations we experimented with, with scores of 0.29 and 1.05 respectively. The mean average precision of the best knn implementation (k=8) was 0.30.

One possible explanation for the poor performance of the contrastive loss implementation is that it only operates on pairs of images, rather than triplets. This means that for each training run we lose 1/3 of the information from the training batch as compared to the triplet loss implementation. Thus after running for 30 epochs (the same as the triplet loss implementation), we are pulling much less information from the training set. It is possible that training for longer could be helpful, however the training loss is already near 0 and it appears that we are already approaching a region where the model is beginning to overfit. Perhaps a different underlying model architecture with fewer parameters (other than Resnet18), could be beneficial to reduce overfitting.

6.6. Addendum: Support Vector Machine

In the course of gathering the results of the above experiments, we also decided to explore some additional classification methods beyond k-Nearest Neighbor. Of the alternative classifiers we saw the best performance with a support vector machine (SVM) classifier.

Results are as follows:

Method	F1 Score	Recall	Mean Accuracy
Triplet Loss	58.2	58.2	36.9
Contrastive Loss	45.8	45.8	22.7

Table 4. Test evaluation metrics of SVM trained on reference embeddings optimized on MTG cards.

We were surprised to see that the template matching approach taken with kNN may not be the ideal solution to our BSS classification task. While we did not set out to prove alternatives to nearest neighbor approaches, the significantly better SVM performance has shown that there may be additional avenues to explore for future improvements.

7. Conclusion

The results of our study showed a promising system that could predict the BSS score on images in the wild. Using a ResNet18 proved to be the best performing model and provided reasonably accurate results for potential use in the clinical or at-home setting. However, the potential for deep metric learning on stool classification has yet to proven its ability to learn in a low-shot learning environment.

Overall, the results thus far are encouraging and suggest that neural network based classifiers are already approaching human levels of classification, but have not yet surpassed GI specialist performance. Given the small data set that was used for this implementation, it can be hypothesized that these very same pipelines may perform significantly better when supplemented with additional data.

In future research, building out a dataset that closer balances all classes will likely be needed to train a high performing system, which we fully expect to occur soon. Other CNN architecture beyond ResNet18 can be explored as the backbone for stool images, especially with a small dataset. Perhaps a smaller and more lightweight model would help us to further improve the model accuracy. Researchers may also explore how more data might change the triplet loss and contrastive loss implementations. Code for the implementation can be found here: https://github.com/skyler1253/DL$_{Final Project}$

References

[1] Elsenbruch, Sigrid (2011), "Abdominal pain in Irritable Bowel Syndrome: A review of putative psychological, neural and neuro-immune mechanisms." *Brain, Behavior, and Immunity*, 25, 3: 386394. 1

[2] Blake, M. R., Raker, J. M., and Whelan, K. (2016), "Validity and reliability of the Bristol Stool Form Scale in healthy adults and patients with diarrhoea-predominant irritable bowel syndrome. " *Alimentary Pharmacology amp; Therapeutics*, 44, 7: 693703. 1, 7

[3] Nawaz, Majid, Sewissy, Adel A. and Hassan, Taysir A. Soliman (2018), "Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network. " *International Journal of Advanced Computer Science and Applications*, 9(6), 2018. http://dx.doi.org/10.14569/IJACSA.2018.090645 2

[4] Nahid, Abdullah-Al, Mehrabi, Mohamad Ali, and Kong, Yinan (2018), "Histopathological Breast Cancer Image Classification by Deep Neural Network Techniques Guided by Local Clustering. " *Biomed Res Int*, 2018: 2362108. Published online 2018 Mar 7. http://dx.doi.org/10.14569/IJACSA.2018.090645 2

[5] Sori, W.J., Feng, J. and Liu, S. (2018), "Multi-path convolutional neural network for lung cancer detection " *Multidimensional System Signals Processing* , https://doi.org/10.1007/s11045-018-0626-9 2

[6] Rossetto, Allison M., and Zhou, Wenjin (2017), "Deep Learning for Categorization of Lung Cancer CT Images " *2017 IEEE/ACM International Conference on Connected Health*, https://ieeexplore.ieee.org/document/8010653 2

[7] Esteva, Andre, Kuprel, Brett, Novoa, Roberto A., Ko, Justin, Swetter, Susan M., Blau, Helen M., and Thrun, Sebastian (2017), "Dermatologist-level classification of skin cancer with deep neural networks " *Nature 2017*, Volume 542, pages 115–118 2

[8] Ren, W., Yu, Y., Zhang, J., Huang, K. (2014). "Learning Convolutional Nonlinear Features for K Nearest Neighbor Image Classification. " *2014 22nd International Conference on Pattern Recognition*. doi:10.1109/icpr.2014.746 2

[9] Wilber, M. J., Kwak, I. S., Kriegman, D., Belongie, S. (2015). "Learning Concept Embeddings with Combined Human-Machine Expertise." *2015 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2015.118 2

[10] Schroff, F., Kalenichenko, D., Philbin, J. (2015). "FaceNet: A unified embedding for face recognition and clustering." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2015.7298682 3

[11] Song, H. O., Xiang, Y., Jegelka, S., Savarese, S. (2016). "Deep Metric Learning via Lifted Structured Feature Embedding." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.434 3, 5

[12] Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y. (2017). "Deep Metric Learning with Angular Loss." *2017 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2017.283 3, 4, 6

[13] Jamal, Sabri (2016), "Stool Detection and Classification in Colorectal Cancer. " *http://www.diva-portal.org/smash/get/diva2:957328/FULLTEXT01.pdf*

[14] Wang, N., Zhao, X., Jiang, Y., Gao, Y. (2018). "Iterative Metric Learning for Imbalance Data Classification." *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. doi:10.24963/ijcai.2018/389 3

[15] Moindrot, O. (2018, March 19). "Triplet Loss and Online Triplet Mining in TensorFlow. "Retrieved from https://omoindrot.github.io/triplet-loss 3, 4

[16] Marc-Olivier Arsenault (2018, February 15). Lossless Triplet loss Towards Data Science. Retrieved from https://towardsdatascience.com/lossless-triplet-loss-7e932f990b24

[17] Hadsell, R., Chopra, S., Lecun, Y. (2006). Dimensionality Reduction by Learning an Invariant Mapping. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR06). doi:10.1109/cvpr.2006.100 3, 4, 5

[18] Wah C., Branson S., Welinder P., Perona P., Belongie S. The Caltech-UCSD Birds-200-2011 Dataset. Computation Neural Systems Technical Report, CNS-TR-2011-001. 4

[19] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal.Mach. Intell., 33(1):117128, 2011. 5

[20] H. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. InCVPR,2016. 3, 5

[21] Adam Bielsky, Siamese-Triplet, (2019), GitHub repository, https://github.com/adambielski/siamese-triplet 7