Classification of musculoskeletal radiographs using unsupervised classification techniques

Joseph Vele, Nikhar Gaurav, Prasanna Kumar Challa, Prathwish Shetty

Abstract

The aim of this project is to enable an easy unsupervised classification model for musculoskeletal radiographs by body parts; thus, reducing the manual time spent classifying large amounts of data. For this task we take the MURA dataset released by researchers at Stanford. This dataset has over 40,000 images from over 14,000 studies and has been manually classified by radiologists. We leveraged on various dimensionality reduction techniques like PCA and tSNE to help reduce the high dimensional data into a lower dimensional subspace, which then helped us run clustering techniques like kMeans, GMM and agglomerative to classify the data. The models developed were able to generate quality clusters with respect to the Silhouette Coefficients; albeit, the Normalized Mutual Information scores for the clusters were really low. Upon further investigation of the data, we concluded the data is even difficult to classify for humans, as the difference between two classes like like fingers and hand is not apparent. Although the dataset is challenging, we believe with improved image processing better results can be achieved.

Introduction

Large high scale musculoskeletal radiographs have played a pivotal part in the intersection of machine learning and the field of medicine. Previously, radiographs were used as the input data for various machine learning frameworks in order to help medical practitioners detect disease at early stages (ex. Cancer, etc.).[1] The researchers are trying to explore more into the domain of using radiographs to determine if there are any abnormalities in the body part or even early signs of diseases. But as the data of radiographs increase exponentially over time it will be difficult to manually classify and hand label all the radiographs for each specific body part.

Determining and labeling the body part the label belongs to has been challenging and time consuming. Hence, there is a need to develop an architecture that will sort the radiographs into a specific category without much human interference. We have taken the approach of unsupervised machine learning, particularly the techniques in clustering to enable us to classify the images into its respective clusters.

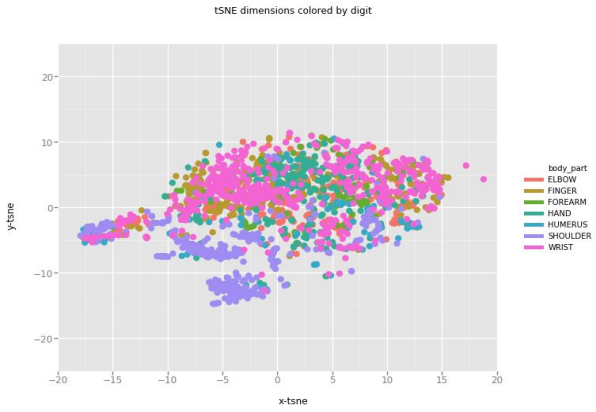
Related Work

In regards to radiography imaging, MURA has become one of the most widely used classification datasets to study and evaluate machine learning methods to detect abnormalities. This dataset has been around for six months and there are fifty submissions on the leaderboard with accuracy ranging from 0.518 to 0.834. The Cohen's kappa statistic for the best radiologists was 0.778 and the baseline model used by Stanford team had a Cohen's kappa of 0.705. The problem statement for our project is redefined to employ unsupervised machine learning techniques to cluster the images based on body parts instead of classifying them to detect the abnormalities.

Methods Used

1. **T-distributed Stochastic Neighbor Embedding (t-SNE)**

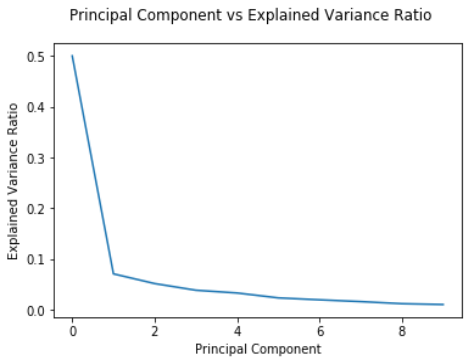
   t-SNE is a visualization technique that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding that is much easier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map.[2] The image was adjusted multiple times using different perplexities and learning rates.

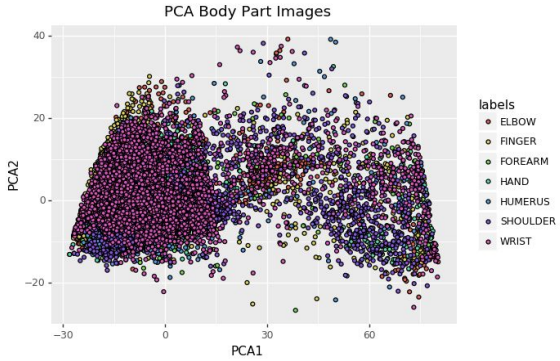


   It was evident by the illustration that a clear pattern did not exist and clustering the data by body part may not be feasible, regardless if the cluster was properly defined. In an attempt to form better clusters, we utilized PCA and Kernel PCA.

2. **Principal component analysis (PCA)**

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. The transformation is defined in such a way that the first principle component accounts for as much variability in the data as possible and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.[3] The number of components were chosen using the explained variance ratio, which ultimately led to 4 components, as the marginal explained variance for other components was negligible.





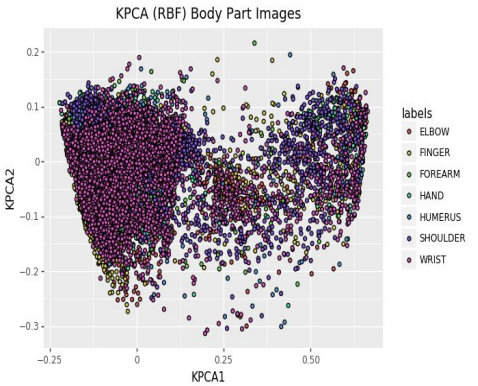
3. Kernel Principal component analysis (KPCA)

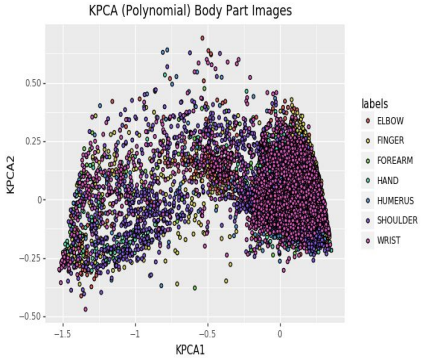
KPCA is an expansion of principal component analysis. Utilizing a predefined kernel, this method performs the same linear operations from PCA in a reproducing kernel Hilbert space. This results is a low-dimensional non-linear subspace. The kernel methods considered for this study were the following:

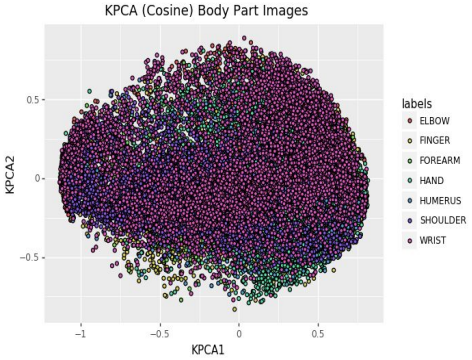
$$Radial\ Basis\ Function:\ K(x,x') = e^{-\frac{1}{2\sigma^2}||x-x'||^2}$$

$$Polynomial:\ K(x,x') = (x^T x' + c)^d$$

$$Cosine:\ K(x,x') = cos(\tfrac{\pi}{2}\ ||x-x'||^2)$$



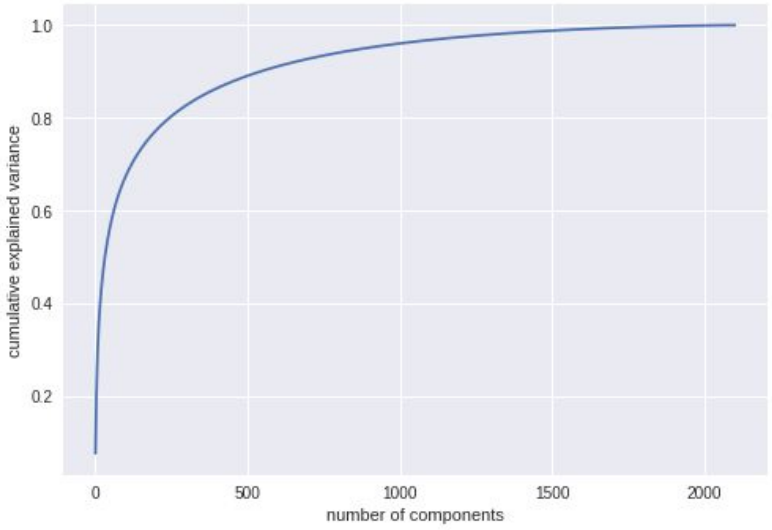




4. CNN

Instead of directly applying Principal Component Analysis, pre-trained ConvNets were used to reduce the dimension of data initially. VGG16, VGG19 - trained on imagenet data - were used by removing their top dense layers, since we just want the encoded features from the images and won't need to perform classification. The input for all the ConvNet models used was 224 x 224 x 3. The output of VGG16 and VGG19 contains 25,088 features, which is a large reduction in dimensions from the original 150,528 input features and possibly represent more meaningful features in the image structure.

PCA was used on top of VGG16 and VGG19 outputs to reduce the dimensionality further. The top 2000 components of PCA explained most of the variance and were then fed as input to the K-means clustering algorithm. Different models were compared by changing the number of clusters in K-means using NMI and SC as evaluation metrics and best results were produced by the model with 90 clusters.



5. K-Means Clustering

K-Means clustering aims to partition data points into k clusters in which each observations belongs to the cluster with the nearest mean. Hence, minimizing the within-cluster sum of squared error. It is similar to Gaussian Mixture model, as it utilizes the expected-maximization algorithm; however, with an all-equal diagonal covariance matrix.

$$SSE_i = \sum_{\mathbf{x} \in C_i} \| \mathbf{x} - \mu_i \|^2$$

$$SSE = \sum_{j=1}^{K} SSE_j$$

K-means is limited, especially for this dataset, as it assumes the clusters are isotropic. Also, in order to successfully generate clusters using K-means, the initial centroids need to be well-defined. For this study, we leveraged in existing algorithm which initializes the k-centroids far apart and then ran cross validation, maintaining the best model.

6. Agglomerative Clustering

Agglomerative clustering is a bottom up approach of hierarchical clustering, which aims to build a hierarchy of clusters. In agglomerative clustering, each observation starts in its own cluster, and is grouped based on the shortest predefined measurement of distance. Both the measurement and distance are hyper parameters, which were adjusted throughout the study. Distance metrics considered were l1, l2, and cosine.

1. L1 - Norm (Manhattan): $(\sum_{i}^{k} |x_i - y_i|)$
2. L2 - Norm (Euclidean) : $(\sum_{i} (x_i - y_i)^2)^{\frac{1}{2}}$
3. Cosine: $\frac{X \bullet Y}{\|X\| \, \|Y\|}$

Measurements considered were ward, complete, average, and single. Ward minimizes the variance of clusters being merged. Complete measures the farthest distance between two clusters. Average measures the average distance, and single measures the minimum distance between two clusters. We defined the optimal measurement and distance as the metrics that minimized the silhouette coefficient, which were l2 and average.

7. Gaussian Mixture Model

Gaussian Mixture Model is a probabilistic clustering model that assumes all the data points are generated from a finite number of Gaussian distributions with unknown parameters. It generalizes k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

$$G(X|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)|\Sigma|}} exp\left(-\frac{1}{2}(X - \mu)^T \Sigma^{-1} (X - \mu) \right)$$

Gaussian Mixture model implements the expectation-maximization algorithm for fitting the mixture of Gaussian model which contains two important steps:

- Expectation step (E step): This step estimates a probability distribution using the current parameters of the model and assigns each data point X_i to cluster C_i with the following probability. This probability assignment is referred to as soft assignment as compared to hard assignment {0,1} in other clustering methods.

$$P\left(X_i \varepsilon C_k\right) = p\left(C_k | X_i\right) = \frac{p(C_k) p(X_i | C_k)}{p(X_i)}$$

- Maximization step (M step): This step estimates parameters to update the model for the current data i.e Maximize the expected log likelihood found on the E-step.

$$m_k = \frac{1}{N} \Sigma_{i=1}^{N} \frac{X_i P(X_i \varepsilon C_k)}{\Sigma_j P\left(X_i \varepsilon C_j\right)}$$

Evaluation Metrics Used

- **Silhouette Coefficient (SC)**

  The Silhouette Coefficient is calculated using the mean intra-cluster distance `[a]` (i.e. calculate the average distance for a point with every other points in the same cluster) and mean nearest-cluster distance for each point `[b]` (i.e. find distance between a point and the distance from the points in the nearest cluster). Then the Silhouette Coefficient for a the point is given by `(b - a)/max(a,b).`[4]

- **Normalized Mutual Information (NMI)**

  Normalized Mutual Information (NMI) is a normalization of the Mutual Information (a quantity that measures how much one random variables tells us about another) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).[5]

  But NMI is not a perfect measure for clustering comparison. As models that creates more clusters receive a higher NMI when compared with the ground truth division. Hence using NMI scores to determine the models can be misleading.[6]

Throughout this project we have used both the above stated measures in our evaluation of clusters, but have given relatively higher importance to Silhouette Coefficient over Normalized Mutual Information, because as stated above NMI can give higher scores just by increasing the number of clusters.

Data and Processing

The MURA Dataset[7] used in this project is obtained from Stanford ML Group. This is a large dataset of musculoskeletal radiographs containing 40,561 images from 14,863 studies, where each study is manually labeled by radiologists as either normal or abnormal. The body parts being radiographed in the study are elbow, finger, forearm, hand, humerus shoulder and wrist. Also the dataset is labeled if a radiograph is normal or abnormal. The breakdown of number of images per body part is as follows:

Study	Normal	Abnormal
Elbow	2925	2006
Finger	3138	1968
Hand	4059	1484
Humerus	673	599
Forearm	1164	661
Shoulder	4211	4168
Wrist	5765	3987
TOTAL	**21953**	**14873**

For the initial analysis we considered the entire dataset of 36,826 images and after further analysis of the image dataset using dimensionality reduction techniques like t-SNE and PCA we reduced our scope of interest to only “Normal body parts”, which reduced the number of radiographs in the study to 21,935.

The next step of the data processing was focussed on extracting the information from the radiographs and process them in a way to be easily comprehensible for processing further along the process. We tried using multiple feature extraction on the images to obtain the best possible information from the images. Few of the methods employed by us include:

- *Reshape the images:* In order to go further into the process of dimensionality reduction and clustering, we needed our images to be of standard sizes and of appropriate size such that we do not lose much information and at the same time keep the size of image small enough to be computationally efficient; hence, we decided on a pixel size of 150 x 150

- *Color scale:* The radiograph images look grayscale to the human eye, when in fact these images are of the RGB format. So for our initial assessment of dimensionality reduction and clustering we converted the images from RGB format to grayscale. But for the technique of dimensionality reduction using Convolutional Neural Network, we took the images in its RGB format.

- *Image Masking:* The idea behind image masking was to amplify the outline of bone images by making non-black pixels (≠[0,0,0] in terms of RBG) to [255,255, 255] . In the example below, we initially naively believed that "black" section of figure a. would equate to [0,0,0]; however this was erroneous, as illustrated by our figure b. Upon further inspection we realized the "black" section was in fact values less than 34. By adjusting our algorithm we were able to achieve figure c; albeit, each image varied in terms of pixels. Ultimately, it was decided that image masking was not an avenue we could take for this study.

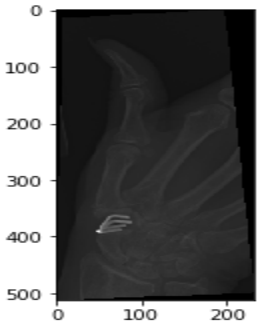


Figure a

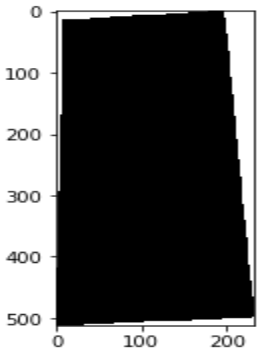


Figure b

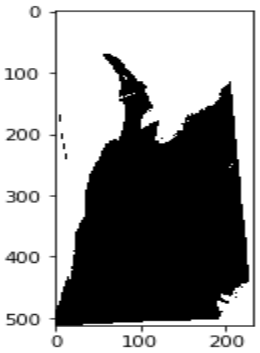
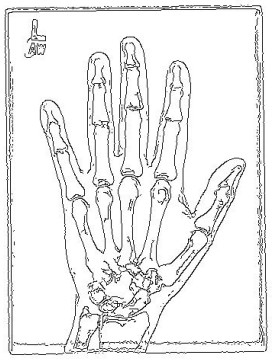


Figure c

- *Edge Detection:* The most important aspects of any radiographs are the edges of the bones or the body parts being examined. Hence, we thought extracting the edges from the radiographs will help us better understand the body part. Below we can see how a radiograph of an hand looks before (fig a) and after edge detection(fig b).

Normal Radiograph



Radiograph after edge detection

- *Image Normalization:* Normalization is a process that changes the range of pixel intensity values. This process aims to get a value between 0-1. Each channel (Red, Green, and Blue are each channels) is 8 bits, so they are each limited to 256, in this case

255 since 0 is included. Since 255 is the maximum value, dividing by 255 expresses a 0-1 representation.

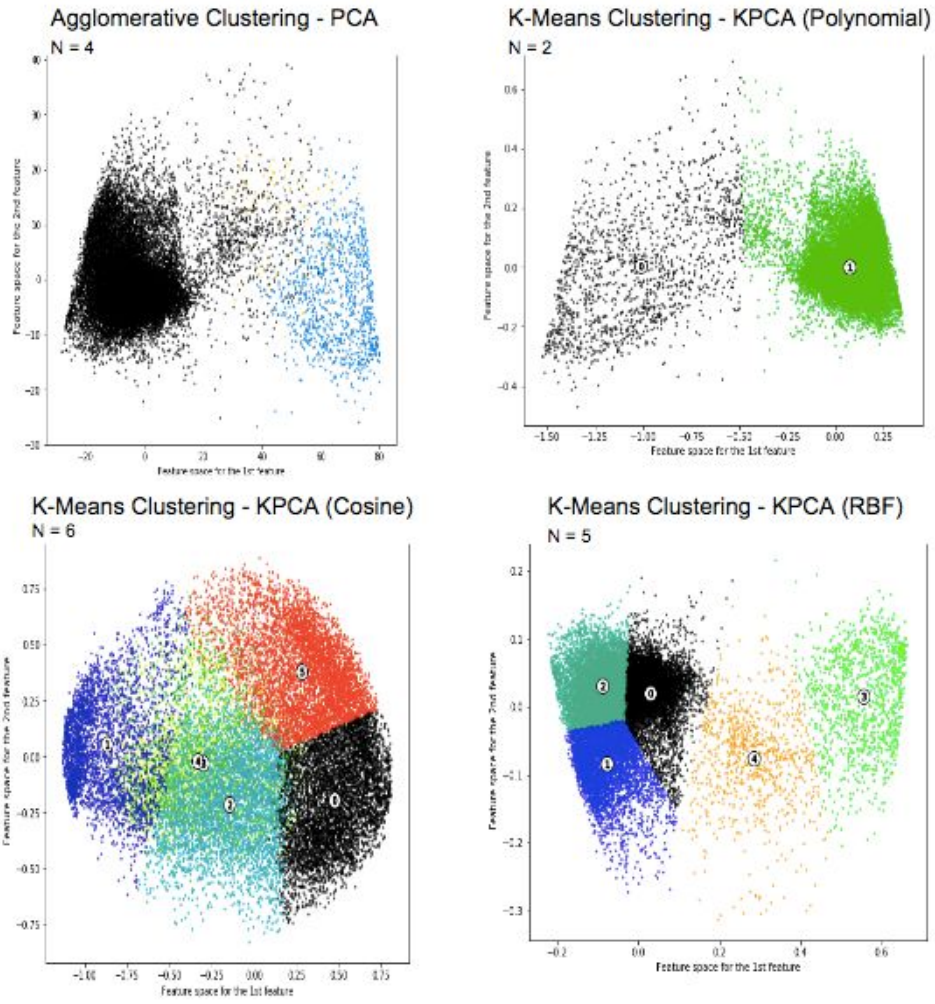
- *Centering of Data:* The centering of data aims to shift the scale of the data while keeping the units. So for centering here we subtract a constant from every value of a variable. For this project we have subtracted all the values by their mean, thus centering the data around 0. So after doing this the mean of the data will be exactly zero but is not affected otherwise: its standard deviation, skewness, distributional shape and everything else all stays the same.

Results

The following table compares the performance of all the dimensionality reduction and clustering algorithms implemented in this project.

Clustering Technique	**Dimensionality Reduction**	**Clusters**	**SC**	**NMI**
GMM	PCA	2	0.6200	0.0500
GMM	KPCA (cosine)	2	0.1500	0.0120
GMM	KPCA (polynomial)	4	0.2330	0.0410
GMM	KPCA (RBF)	4	0.2310	0.0570
KMeans	PCA	4	0.2860	0.0400
KMeans	KPCA (cosine)	6	0.2920	0.0130
KMeans	KPCA (polynomial)	2	0.7600	0.0100
KMeans	KPCA (RBF)	5	0.3000	0.0700
Agglomerative	PCA	4	0.6800	0.1200
Agglomerative	KPCA (cosine)	3	0.2800	0.0900
Agglomerative	KPCA (polynomial)	3	0.7800	0.0100
Agglomerative	KPCA (RBF)	3	0.6900	0.0100
KMeans	CNN + PCA	90	0.0028	0.3921

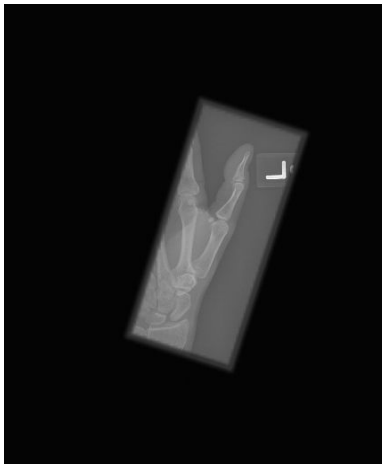
As we can see, there is no clear tradeoff between the values of SC and NMI in any of the methods. K-Means and Agglomerative clustering methods with polynomial kernel PCA produced better results than other algorithms in terms of SC. While, K-Means with CNN and PCA performed well in terms of NMI. Depending on the final objective of the problem any of these methods could be chosen to cluster MURA.



Problems Faced

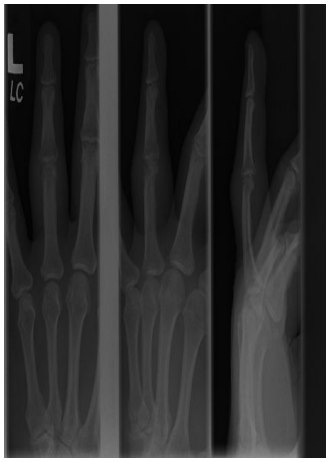
- **Variability of images within category**

  One of the biggest problems we faced when classifying the images was inconsistency of the images within the same category. From the images below, we can see that though all three images belong to the category fingers, they are all uniquely different.



Finger: Patient 1

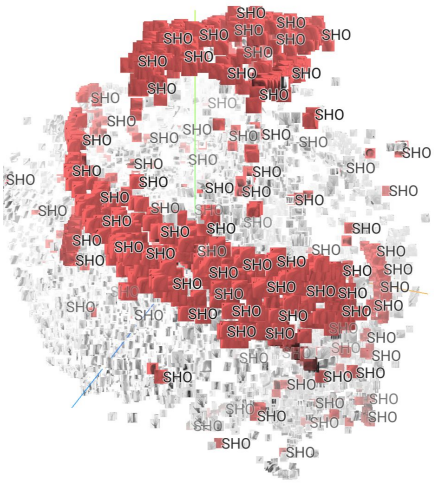
Finger: Patient 2



Finger: Patient 3

- **Mirror images with category**

  Another common misclassification observed by us is, left and right parts of the body being classified as two separate clusters. The example below shows the tSNE visualizations of the various body parts.



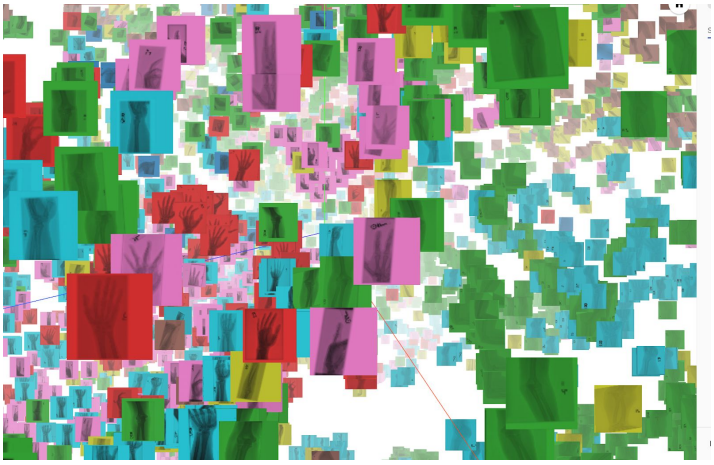
Two separate clusters formed for shoulders after tSNE

The points highlighted in red are data points for 'shoulders', and we can see that two distinct clusters are being formed for shoulders and this is due to the presence of "right" and "left" shoulders in the data.

- **Similarity among category**

  The similarity of images between categories is another problem we encountered during the project. As can be seen from the results of tSNE clustering below, the red, blue and pink represent the three categories hand, finger and wrist. However, from the images

we can clearly see that though they are distinct categories, they all look similar to some degree. Hence, making clustering harder.



Images from different categories appear closer to each other after tsne

References

[1] Google. An Augmented Reality Microscope for Cancer Detection [Internet]. Google AI Blog [cited 2018 Dec 4];Available from: http://ai.googleblog.com/2018/04/an-augmented-reality-microscope.html

[2] Maaten L van der, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res 2008;9(Nov):2579–605.

[3] Contributors to Wikimedia projects. Principal component analysis - Wikipedia [Internet]. Wikimedia Foundation, Inc.2002 [cited 2018 Dec 5];Available from: https://en.wikipedia.org/wiki/Principal_component_analysis

[4] sklearn.metrics.silhouette_score — scikit-learn 0.20.1 documentation [Internet]. [cited 2018 Dec 4];Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

[5] sklearn.metrics.normalized_mutual_info_score — scikit-learn 0.20.1 documentation [Internet]. [cited 2018 Dec 4];Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html

[6] Amelio A, Pizzuti C. Is Normalized Mutual Information a Fair Measure for Comparing Community Detection Methods? [Internet]. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15. 2015. Available from: http://dx.doi.org/10.1145/2808797.2809344

[7] MURA Dataset: Towards Radiologist-Level Abnormality Detection in Musculoskeletal Radiographs [Internet]. [cited 2018 Dec 4];Available from: https://stanfordmlgroup.github.io/competitions/mura/

[8] Gaussian mixture models [cited 2018 Dec 6]; Available from: https://scikit-learn.org/stable/modules/mixture.html