Bag of bones – Labeling forearm X-RAY images containing bone abnormalities using visual BOW.

By: Ilan Geffen, Eran Tal. Date: 23/3/2021

Introduction-In this project we sought to train model classifying healthy and abnormal forearm X-RAY images by abnormal we include: fractures, hardware, degenerative joint diseases, and other miscellaneous abnormalities, including lesions and subluxations. Initially we chose to delve into this idea as we both think that one of the most important and positive uses of computer algorithms in general is improving the ability to provide essential services to people, in this case healthcare. The study of medicine and medical technologies is deeply involved with imaging. With uses such as diagnosis, treatment or even educational purposes (which is how the idea came to mind after hearing numerous complaints from people that are close to us regarding how annoying studying anatomy is because all bones look the same). In this case we chose to focus on the diagnostic aspect of medical imaging and after reading into it we directed our attention towards radiology and X-RAY imaging. In order to create such model we had to "translate" the images into observations containing discrete features and labels within those features that accurately represent the image and contain relevant information in order to classify the image as healthy or abnormal. The method we chose to use is the visual BOW due to its ability to create visual features in the form of "words" representing image features and labels in the form of the number of occurrences each word appears within a given image. These together create the word histogram of a given image which we can use as an observation vector along with a binary label in order to train a model that is aimed at learning which combination of visual words is likely to represent an image containing abnormal bones and which do not.

Related work-We will now discuss work related to our project. The topic of X-RAY image processing is highly popular. In fact there is a whole dataset dedicated to bone abnormality recognition with X-RAY imaging called MURA. MURA even has its own prediction accuracy submissions as a part of the challenge and professional radiologist sit at a mere 0.778 Accuracy. This signifies just how relevant of a topic this is and the number of papers that touch upon it support this claim. To name a select few which helped us in formulating our algorithms flow:

- Enhanced Computer Aided Bone Fracture Detection Employing X-Ray Images by Harris Corner Technique
  C. Z. Basha, M. R. K. Reddy, K. H. S. Nikhil, P. S. M. Venkatesh and A. V. Asish.
  https://ieeexplore.ieee.org/document/9076436
- Long-bone fracture detection in digital X-ray images based on digital-geometric techniques
  Oishila Bandyopadhyay, Arindam Biswas, Bhargab B.Bhattacharya.
  https://www.sciencedirect.com/science/article/abs/pii/S0169260715002400
- Detection and prediction of osteoarthritis in knee and hand joints based on the X-ray image analysis
  G.W. Stachowiak, M.Wolski, T.Woloszynski, P.Podsiadlo.
  https://www.sciencedirect.com/science/article/pii/S2405451816300393

Technical description - The different steps and methods used by our algorithm are as follows:

Creating and training the model – Inputs: Image dataset – consisting of over 1800 images of labeled healthy and abnormal x-ray images which we splint into a training set and a testing set with a 70%-30% ratio.

1. Preprocess:

- For each image in the training set we apply a Contrast limited adaptive histogram equalization. in this process we create an intensity histogram for each 8x8 patch in the image. We then construct a CDF normalizing the number of occurrences of each pixel so that when reaching the maximal intensity within that patch the CDF value for that intensity will be 1 (as we have necessarily went over all the pixels within the image). that is for $a\ patch\ S\ and\ a\ pixel\ p\ with\ an\ intensity\ of\ 0 \le i \le 255\ the\ value\ of\ the\ CDF\ f_s\ given\ by\ the\ patch\ S\ for\ pixel\ p\ will\ be: f_s(p) = \frac{1}{|S|}\Sigma_{j \le i} \#_S(J)$. Where $\#_S(J)$ is the number of occurrences of intensity j in patch S. We then determine the new value of the pixel p by $f_s(p) * 255$. CLAHE also limits the amplification done by thresholding the CDF and redistributing all values exceeding the range

equally among the other histogram values. We use this method in our preprocessing in order to intensify edges in the image. this is due to the vast majority of bone abnormalities having a feature indicating their exitance "an x-ray machine produces a small burst of radiation that passes through the body, recording an image on photographic film or a special detector. Different parts of the body absorb the x-rays in varying degrees. Dense bone absorbs much of the radiation while soft tissue, such as muscle, fat and organs, allow more of the x-rays to pass through them. As a result, bones appear white on the x-ray, soft tissue shows up in shades of gray and air appears black." (- radiologyinfo.org). as an example of why this specific method works well on bone abnormalities here is a quate that might give a better idea as to what features might result from bone abnormalities and in this case fractures: "A **bone fracture** is a medical condition in which there is a partial or complete break in the continuity of the bone." (– Wikipedia). So at any place where a fracture occurs, the discontinuity will be expressed in the image as a "less white" part due to the change in bone density caused in that particular area.

- Our second stage of preprocessing is using edge preserving smoothing. For this purpose We chose to use L0-gradient minimization. We used the implementation by Tsuzuk (reference to the article and the git below). This method uses linear regression in order to reduce the number of non zero gradients in the image while maintaining a relatively close intensities and levels of prominent details within the image. this is done by an iterative process of repeated linear regretting minimizing the equation:
  $\Sigma(\mathrm{S}-\mathrm{I})^2+\lambda C(h,v)+\beta\left((\partial_{\mathrm{x}}S-h)^2+\left(\partial_y S-v\right)^2\right)$. where S the target image. h represents the image pixels gradients in the x direction which we seek to turn to 0, v is the same for y, λ is smoothing weight parameter indicating how strongly we wish to prioritize the removal o f non zero gradients over keeping image details and β is an adaptive weight both reducing the number of nonzero gradients included in h,v and the urgency of flattening said remaining gradients to 0. By doing so we smooth all non-prominent edges and keep the other edges. The implementation we chose uses Furrier transformations in order to speed up the linear regression and achieve the value of convergence quicker. We simply use it in order to speed up the computational and thus the training process significantly. We chose to use this algorithm in order to reduce the salt and pepper like noise created by the CLAHE algorithm while maintaining the edges defining the abnormal segment for later detection. In short the combination of the two algorithms enables the enhancement and detection of the features defining a healthy and abnormal bone while keeping the number of features resulting from noise to a manageable degree.

2. In this step we creating a list of extracted descriptors of key points, from the processed images, that defines the images. We chose to implement this part using SIFT (later changed to BRISK). First, we are creating a scale-space of the image which will emphasize potential interest points, by using DoG (Difference of Gaussians) on different octaves (scales) of the image gaussian pyramid. DoG technique approximates LoG (Laplacian of Gaussian), which work faster, and takes advantage of the fact that LoG is useful for detecting edges.
   While in the LoG we use a kernel of gaussian and second level derivate, DoG to subtracts gaussian which has different std parameters, in order to get a good approximation, but in a less costly procedure. We do so on different scales which we will later compare in order to find the scale which has the most unique point relative to its neighbors. once we finish calculating the DoG, we compare each pixel at each octave to its 8 neighbors on the same scale level, and to the 9 neighbors at the above and below levels. if the pixel is a local extreme point, we refer to it as a potential interest point. Next, for each point we check the relevant patch by calculating the Harris matrix C for each potential pixel, and computing the smaller eigenvalues using the approximation:
   $\Sigma(\mathrm{S}-\mathrm{I})^2+\lambda C(h,v)+\beta\left((\partial_{\mathrm{x}}S-h)^2+\left(\partial_y S-v\right)^2\right)$ And choose the eigenvalues which are higher from some threshold t. At this stage CV2 SIFT algorithm implementation, computes orientation and magnitude of each key points, which isn't relevant for the descriptors calculation. After finding the key points, we take a patch of 16x16 pixels around the key points, and divide it to 16 semi-patches. For each semi-patch we calculate the derivative to calculate each pixel's orientation, which is quantified into 8 different directions. Next, we count the amount of pixel in each direction that surpass the given threshold. After which we get an 8 dimensions histogram for each

sub-patch, overall a 128 dimensions descriptor for each key point. In order to speed up the computation, we use Brisk algorithm. Brisk is using AGAST an algorithm for key points detection that uses the FAST algorithm's principles, which are faster than Harris corner detection. "Brisk description is based on identifying the characteristic direction of each feature for achieving rotation invariance. To cater illumination invariance results of simple brightness tests are also concatenated and the descriptor is constructed as a binary string. BRISK features are invariant to scale, rotation, and limited affine changes." [5] Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart Autonomous Systems Lab, ETH Zurich From each image we extract between 4 to 8 thousand descriptors, we do so for 2000 images, giving a list of around 10 million descriptors.

3. Principal component analysis: after the previous stage we now have a dataset where each row has 64 features representing a histogram of a feature extracted from all of the images within our image dataset. We use PCA to reduce its dimensionality to 40. Principal component analysis revolves around reducing dimensionality while keeping most of the information from the data. This is done by finding the linear transformation from a n dimension feature space onto a k<n dimension feature space such that the variance of the data is maximized in relations to the original data. The reason we use PCA as a data preprocessing method is to simply reduce the runtime of our algorithm as we are dealing with a large dataset. This way our k-means algorithm has to consider 40 features instead of the original 60 features while maintaining a sufficient accuracy.
4. K-means: we use k-means to cluster our data by image features such that the center of each cluster is a centroid representing a feature who's histogram is the mean of all the others features within that cluster (the point closest to all other points in within the cluster). Depending on the chosen number of clusters these centroids are able to represent a group of visual features that are relatively similar to each other and by doing so reducing the number of words in the visual BOW algorithm by using these centroids as our words.
5. In the final processing stage, we create a word histogram representation for each picture. This gives us the advantages of both keeping the descriptors properties and k-means associated advantages.
   By creating a histogram of the labeled features in the picture we get a homogenous representation for the pictures, which take advantage of the descriptors properties: robust to affine transformation and illumination. The k-means lets us find features clusters centroids for regular and abnormal bones features, while regular bones would have many features in common, since the general structure of the forearm bone is similar, abnormal bones have a higher verity of non-similar features. For example, a bone hardware would have a unique interest points, which won't appear in regular bones (such as a metal plate). Therefore, we expect that an abnormal bone would be identifiable by certain unique certain features, which do not appear in regular bones, so we can train the model to recognize these expressions to classify the pictures. We are preforming the same stages: preprocessing and extracting feature descriptors from the images. Next we use scipy.spatial.cKDTree to classify the image descriptors to their most similar centroids and we count for each centroids, how many features were assigned to it in order to get a vector of K dimensions, that represent the image. We are doing so for each image.
6. Support vector machine: now that each of the images in our image dataset is represented as a vector of occurrences of visual words and a label we can start training our model. The model trains to classify a vector of occurrences of visual words as either healthy (1) or abnormal (-1). To do so we simply use a standard supervised, machine learning algorithm for prediction and labeling and for that purpose we chose to use SVM. The SVM finds a separator in regards to a transformation kernel mapping our data into a separable form and uses the separator in order to predict the label of a new sample. In our case we used the radial basis function as a kernel as we are dealing with a dataset that due to high dimensionality is not likely to be linearly separable.

Implemented methods from class - Our project is built upon key algorithms and concepts seen in class throughout the semester. We used these tools in order to structure the skeleton of our algorithm and it's flow. The methods from class demonstrated in our project are mainly the visual bag of words, serving as the key implementation behind the idea of our project, along with feature detecting tools such as the Harris corner detector using SIFT (later replaced with BRISK), general usage of descriptors and intensity histograms for image classification, representation and enhancement, other image preprocessing tools such as smoothing patches and clustering pixels. Also note that while using a tool for object detection and recognition such as the visual BOW, in this project we have used it in a

more delicate manner for finding bone structure abnormalities (a non-consistent structure within a broader structure, being the bone itself) and classification.

Data used – As stated in the references our main data source for both training, testing and model validation was the reliable MURA dataset provided by the Stanford ML group. This vast dataset aimed towards a bone X-RAY deep learning competition features a variety of labeled X-RAY images of different body parts. It features conditions such as fractures, hardware, degenerative joint diseases, and other miscellaneous abnormalities, including lesions and subluxations. MURA is one of the largest radiographic image datasets containing 40,561 images of musculoskeletal radiographs from 14,863 studies.

Summary of experimental results- After testing several possible parameters, we choose to use : CLAHE window size: 20 pixels, l0 lmd: 0.15, l0 beta_rate: 2.0, number of words: 10,000 and PCA that reduce the descriptors dimensions to 40. For these parameters, the SVM model that was trained with 1400 samples, predicted correctly 74.5~76.8% of the 600 test samples. The average confusion matrixes scores are: Sensitivity: 86.842%, Specificity: 56.8%, Precision: 76.8%, Negative Predictive Value: 71.4%.

Results discussion- We can see that model has a better predicting chance for positive image, with an average sensitivity of 86.8%, while having 56.8% correction rate in the negative labeled ones. We believe that this is due to the fact that all unhealthy bones would usually share regular bones features, with some added unique features that make the decision almost absolute, while the other directions does not hold making healthy bones harder to distinguish. Increasing the number centroids, might improve the result since it would allow a representation of more descriptors, however this would increase the run time significantly. The advantages of the algorithm is that using CLAHE along with L0 we are able to detect weak edges that might indicate the presence of an abnormality in the bone structure, while keeping noise to a manageable degree, along with SIFT (BRISK) descriptors making for a relatively thorough feature detection process. One of the main disadvantages of the algorithm, is that using VBOW overlooks spatial relations within the descriptors. Another weakness resulting from k-means is that the number of words has to be decided by the user, there is no obvious way to choose the number of centroids and also the results can be heavily effected by outlier features. One of the main challenges was the preprocessing stage. We had to find a way to keep important details like weak edges but also eliminate noise. During the development of the project we tested smoothing with kernels (like gaussians), but these resulted in the disappearance of weak edges that were impactful on the prediction. From the same reason Canny edge detector did not suffice as well (although for a full fracture it did emphasized the unique patterns). Therefore, we had to use some method that would strengthen weak edges (CLAHE), and another method that would eliminate some of the noise (L0) while maintaining most of the details. Using the k-means on the descriptor list without reducing its dimensionality resulted in a significantly higher run time. Since many stages of the processing are computationally heavy, we divided the processing stages to 3 main stages, where at the end of each stage we save the results to a file. This helped divide algorithm's flow's load, to smaller controllable segments.

How would we have continued- If we were to continue working on this project one major improvement that we would have loved to make is detection of the abnormal area within the image and further classification. What I mean by that is that the algorithm receives the same input but returns an image with the areas containing the abnormal features are segmented and marked as well as a label describing the specific conditions within the image if there are any. The difficulties we would have to tackle for such advancements are the representation of the data in order to keep the key point and the descriptor so that we would know which area to mark as well as modifying the classification algorithm to return the abnormal words found and maybe look for similar looking words within the image. Another difficulty we would have to face in order to do so is to label each condition so that the prediction algorithm could learn to classify them properly. This not only implies the possibility of needing to model each conditions preprocess differently but also finding a properly labeled dataset. A challenge to say the least.

Table of references and tools used:

References:

[1] L0-gradient minimization: article{l0smoothing2011,
author = {Li Xu and Cewu Lu and Yi Xu and Jiaya Jia},
title = {Image Smoothing via L0 Gradient Minimization},
journal = {ACM Transactions on Graphics (SIGGRAPH Asia)},
year = {2011},

[2] Faster and better: a machine learning approach to corner detection Edward Rosten, Reid Porter, and Tom Drummond

[3] MURA –Large Dataset for Abnormality Detection in Musculoskeletal Radiographs arXiv:1712.06957 **[physics.med-ph]**

Code used:

Opencv library:

- Contrast limited adaptive histogram equalization
- FAST (originally SIFT)
- BRISK

Scikit-learn:

- PCA – sklearn.decomposition.PCA
- K-means – sklearn.cluster.KMeans
- SVM

L0-gradiend minimization https://github.com/t-suzuki/l0_gradient_minimization_test/blob/master/l0_gradient_minimization.py