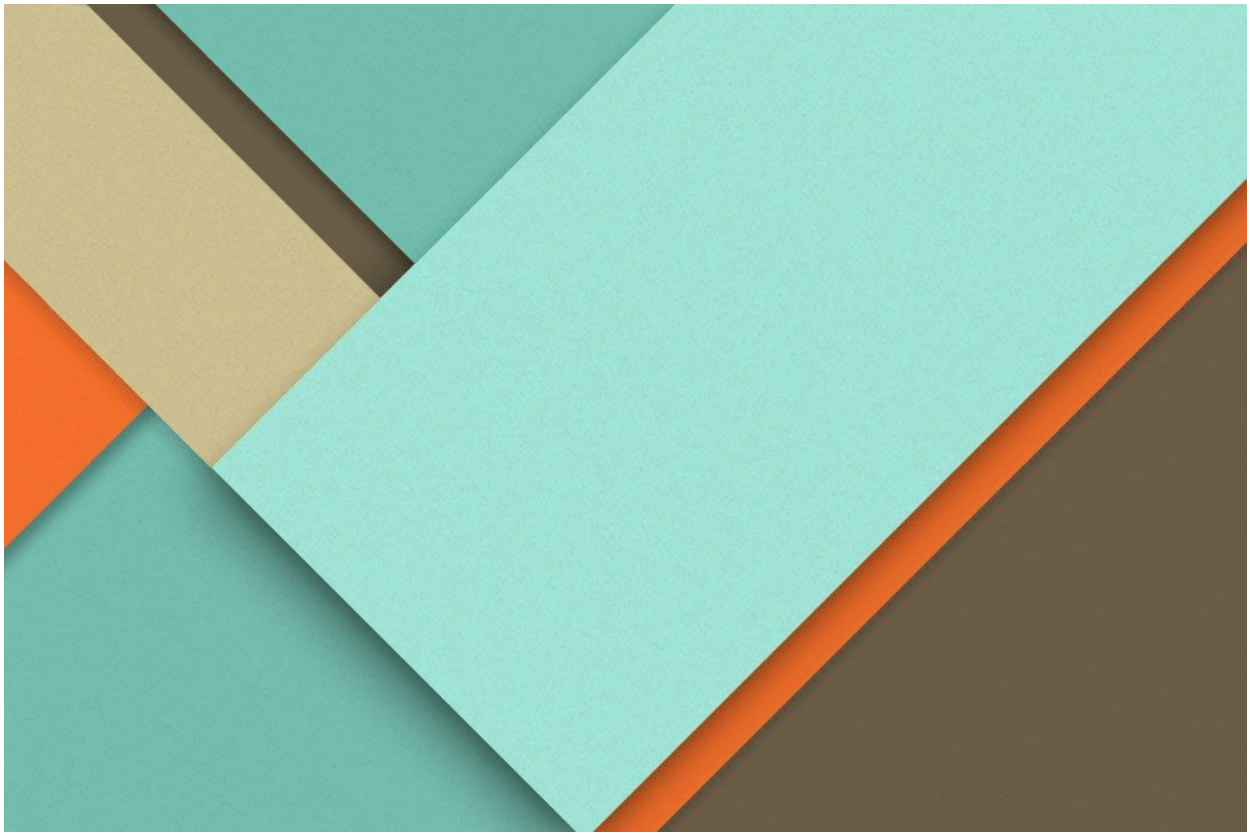


Unsupervised Symbolic Learning

Using Co-Occurrence Analytics and Spatially Localised Community Detection

Ruchi Gupte
Jio- Building AI for India
Summer Internship
18.05.20-31.07.20

Content:

Acknowledgement

I would like to express my gratitude and appreciation to all those who helped me throughout my term here at Jio. It was a great learning experience for me and has helped me to not only gain valuable knowledge and experience in the field but also helped deepen my understanding of data analytics and its real-world applications.

All this would have not been possible without the guidance and encouragement of the professionals here, who have helped me throughout the internship period. I would like to give my special thanks to Dr Abhinav Anand for assigning me such an amazing and challenging project and mentoring me throughout my term here and I would also like to thank him and Mr Anurag Sahoo, for guiding me and arranging all the facilities that helped me learn faster and more efficiently.

I would like to extend my deepest gratitude to Dr Shailesh Kumar, whose work and research on Co-occurrence analytics has really inspired me and his guidance and expertise have greatly contributed to the success of this project

My term here was a valuable learning experience and it would not have been possible without the guidance and support of the team here at Jio.

Sincerely,

Ruchi Gupte

Project Overview

Machine learning in data-analytics has advanced at a tremendous pace this past decade, from algorithms like scale-invariant feature transform (SIFT) to currently popular techniques such as neural networks and clustering. While both techniques use different approaches to analysing a dataset, they can be used in tandem to provide a more complex and perceptive algorithm. This paper explores an unsupervised learning algorithm that combines the intuitive nature of SIFT features and the speed and efficiency of unsupervised learning.

Various image datasets are analysed using co-occurrence based community detection using symbolic Identification (ID) assignment. A set of nxn images features are extracted and assigned certain identification values through loose unsupervised clustering techniques like K-Means. These ID's are then grouped into communities based on their spatial features to detect coherent groups of ID's which co-occur frequently. By analysing the point-wise mutual information(PMI) of each co-occurring pair of ID's, we were able to build a model which is able to detect communities of coherent ID's which represented important features for the given image dataset. The model is tested for its scalability and accuracy against conventional learning algorithms to test whether it can adequately detect useful features from a seemingly random and unlabelled dataset.

Motivation

Machine learning in the past decade has gone through tremendous changes, with new and improved methods and algorithms being constantly introduced to replace old ones. In this age of rapid progress, it is important to look back and extract the useful elements from these algorithms and try to incorporate them into our system to overcome the challenges faced by machine learning models today.

One such machine learning algorithm, Scale-invariant feature transform (SIFT), focuses on a more intuitive approach to data analytics. It obtains a feature list from an image dataset and compares each point in a new test dataset to find matching points. By considering the feature's location, scale, and orientation, important features are found and grouped into clusters denoting features which are good matches for each other. In contrast, modern techniques such as neural networks focus on finding undetected patterns in a data set without labels and using minimum human supervision. While both algorithms deal with pattern recognition, their fundamental approach is drastically different. While SIFT excels in more structured logical approach, unsupervised learning excels in efficiency and boasts better results. Thus a way to incorporate the intuitive nature of SIFT with the efficiency of modern unsupervised learning could help us get a process which focuses on both results and execution.

Co-Occurrence analytics is one such intuitive algorithm which combines these two concepts and is a pattern recognition technique which helps detect unknown patterns in unlabelled datasets. It deals with finding the possible co-occurrence of two given entities and finding the frequency of such entities with respect to other features. This paper deals with the process of incorporating Co-occurrence analytics with spatially localised community detection for a wide range of image datasets.

Techniques such as Viola-Jones Face Detection Algorithm, Haar Cascading and Linear discriminant analysis were also analysed to derive useful insights that would help develop and fine-tune the system to work efficiently and effectively.

Objectives

1. To develop an agnostic unsupervised learning pipeline which is able to extract relevant features from an unlabelled image dataset.
2. The system should then be able to extract useful strongly coherent features from an image dataset using a set of low-level feature extractors and would assign unique identification (ID) numbers to each set of coherent features.
3. Using these ID's as representations of the image dataset, we should be able to build communities of highly coherent features using Co-Occurrence analytics.
4. Test the algorithm on various datasets and feature sizes, allowing us to get a better picture of the limitations and advantages of such feature description methods.
5. Find the accuracy and effectiveness of the system as compared to known machine learning techniques.

Milestones

I. Research and Early Development (18.05.20- 24.05.20)

Studied various feature extraction techniques and developed a system pipeline to follow. Initial stages of the algorithm were developed and tested.

II. Rough Pipeline Implementation (24.05.20- 08.06.20)

Finalized algorithm for the system and implemented it using python. Testing was done on MNIST dataset with positive results during initial testing.

III. Final System Design and Testing (08.06.20- 28.06.20)

The system was finalised and structured properly using function files and a flexible and detailed variable coding. Detailed testing was done on MNIST and Fashion MNIST for various feature sizes and sets. The system was made data agnostic and robust.

IV. Additional Testing and Initial Documentation (28.06.20- 12.07.20)

System tested with a more complex dataset (Chest Xray) and analysed. System fine-tuned to store and retrieve data more efficiently. Compilation time reduced to facilitate faster analytics. Initial stages of project findings compiled and posted on GitHub.

V. Final Documentation (12.07.20-31.07.20)

Findings and results were organized and converted to an IEEE paper format. Project methodology, findings and implementation were detailed along with results and analysis.

Concepts and Theory Used

K-Means Clustering

K-means Clustering is a simple yet very effective unsupervised learning algorithm which forms 'clusters' of data points which are similar to each other. It is an unsupervised machine learning technique as it makes inferences about the dataset without any additional labels or classifications and divides it solely based o the dataset given to it. Each cluster is represented by a centroid which acts as an appropriate representation of all the points in that cluster.

K-means clustering algorithm starts by making an initial guess for the cluster centroids and then goes through each data point in the dataset and then finds the closest cluster for each data point. It then replaces each centroid by the average of data points in its given cluster. This step is repeated until convergence.

Considering xi = (xi1,...xip), if cluster centroids are m1,m2,...mk, and clusters are c1,c2,...ck, then one can show that K-means converges to a local minimum given by WithinCluster Sum of Squares WSS(C) shown in Equation 1 where ||xi −mk||2 is the Euclidean Distance between the data point and the cluster.

$$WSS(C) = \sum_{k=1}^{K} \sum_{i \in c_k} ||x_i - m_k||^2 \quad (1)$$

K-means clustering is a good fit for the community detection algorithm as it divides and loosely labels the dataset into clusters and assigns a unique ID to similar data points. This data can be used to extract the mutual information between the two given points which is an important step to obtain communities of data points \cite{b4}. A few advantages of K-means over other unsupervised learning algorithm are its high speed, scalability, a guarantee of convergence and adaptability. It does not judge the dataset beforehand and only has a cluster number as a parameter which allows it to be integrated and used with various datasets and models.

Co-Occurrence Analytics

Co-occurrence analytics deals with analysing the pairwise relationships between two given entities. Pairwise relationships deal with not only how often two entities occur together but also considers how much they occur together as compared to how much they occur randomly. This is measured by finding out the 'consistency' between these two entities. Consistency is the measure of how frequently and coherently an entity occurs with another. Thus by extracting the consistencies between all the data points in a given dataset, we can detect communities of data points which form coherent features for the dataset. [6]

The consistency between an entity 'A' and another entity 'B' is found by calculating the mutual information between the two entities. For this, we need to first find out how often A occurs, how often B occurs, and how often A and B occur together. Equation(1) is the standard formula for Pointwise Mutual Information(PMI) which gives us the measure of consistency for two given entities A and B.

$$PMI(AB) = \frac{P(AB)}{P(A) * P(B)} \qquad (2)$$

Here P(AB) is the probability of A and B occurring at the same time in the dataset, P(A) is the probability of A occurring in the dataset, P(B) is the probability of B occurring in the dataset. These values can have a varied range depending on the dataset and data points considered, therefore to effectively use the consistency values in the dataset we need to consider the Normalised Point-Wise Mutual Information (NPMI) given by Equation(2). Here the values of consistency in the range of [-1,+1] resulting in -1 indicating that the points never occur together, 0 for the points being independent, and +1 for complete co-occurrence.

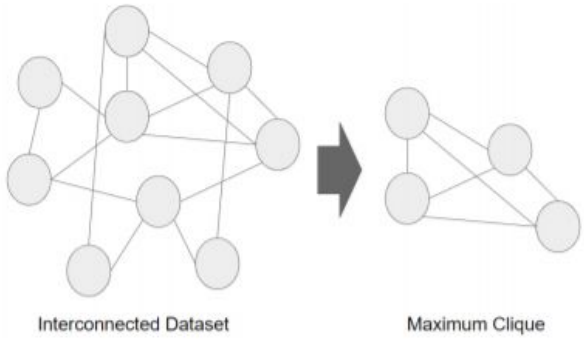
$$NPMI(AB) = \frac{PMI(AB)}{-\log_2 p(x = A, y = B)} \qquad (3)$$

Here the PMI of A and B is divided by the joint self-information of A and B. After calculating the consistencies of all possible entities in the dataset, we are able to use this data to find communities of frequently occurring data points using a community detection algorithm.

Spatially Localised Community Detection

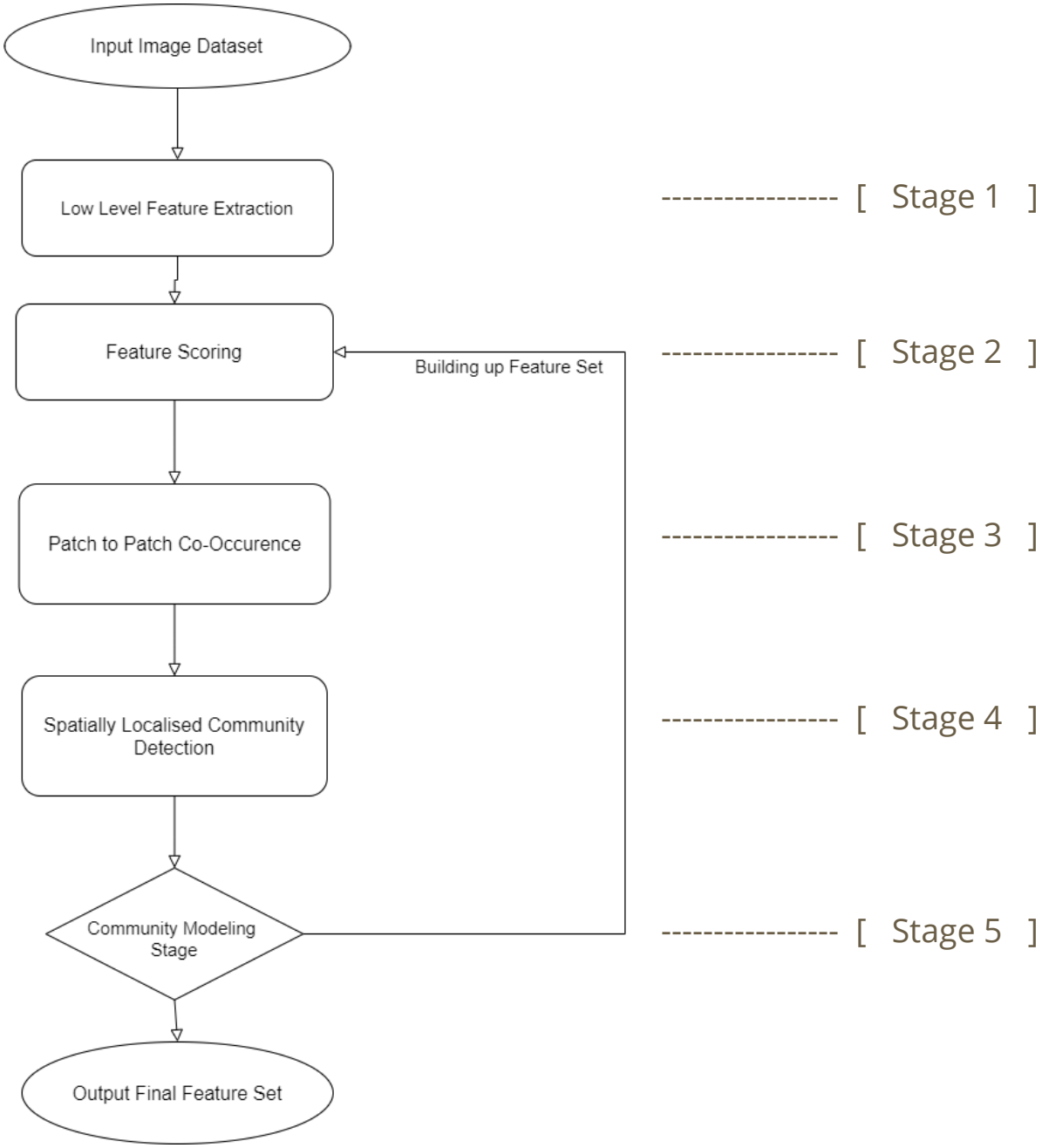
Community detection algorithms deal with taking interconnected data points in a dataset and detecting valid tightknit communities of data points. This can be done using various algorithms which are selected according to the inputs and structure of the dataset [8]. For this paper's community detection, the input pairs of consistencies for the dataset are taken and communities of such data points are detected by finding the Soft Maximum Cliques. This is a greedy algorithm which keeps adding and subtracting data points until it gets the maximum value of consistency within the community. Thus by finding Soft Maximal Cliques in a dataset, we are able to find communities of data points which are highly correlated to each other and form a cohesive group.

The algorithm starts off with a data point and continues to add data points to the community in order to increase the value of its defining performance metric (in our case, consistency), till the point that adding any additional element would result in a drop of in performance. The algorithm then proceeds to remove data points from the formed community to further increase the performance. The final resulting community would be a tight-knit group of data points, which are not only strongly related to each other but also strongly coherent as a group. Fig 1 shows us a basic example of graph theory where a set of interconnected data points are given and the Soft Maximal Clique is found from them.



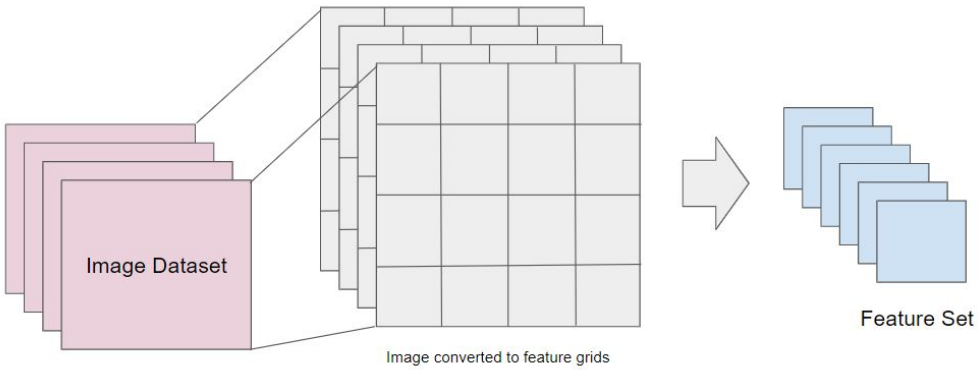
For image datasets, just detecting whether two given low-level features occur together would not provide sufficient information for the algorithm to detect data points and form coherent communities, as an image dataset is highly location-sensitive \cite{b5}. Thus it is also important to figure out and consider the relative placement of the frequently occurring data points as well\cite{b7}. It is also important to fit this data into the community detection algorithm which does not accept image location data. This is done by arranging and sorting the data points into directional tables, where each direction is allotted to its respective table along with the degree with which a certain point occurs with respect to another\cite{b11}. Thus using the data from all the tables, communities can be formed using the soft maximum clique algorithm.

Step Wise Methodology



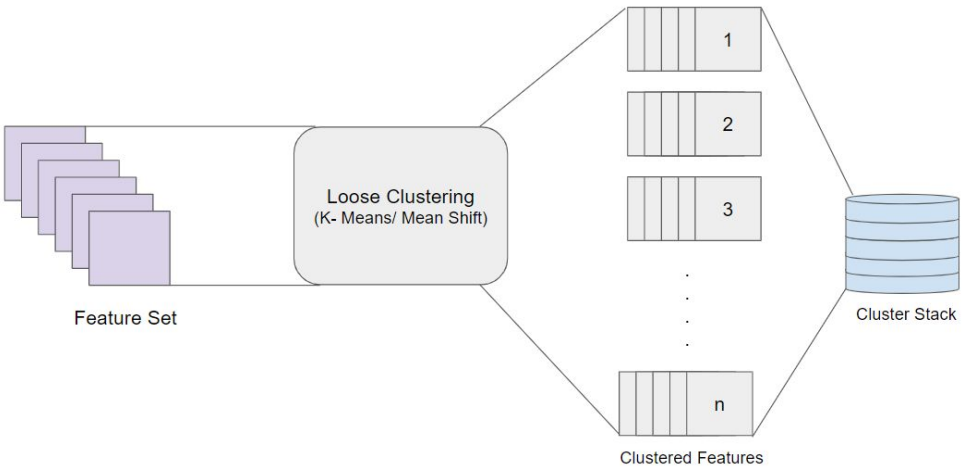
Stage 1-Low-level Feature Extraction

An unlabelled image dataset is obtained and put into the system for low-level feature extraction. Low-level feature extraction deals with obtaining relevant information from the images, which contain even the slightest amount of detail pertaining to the given dataset. This process will be done through various algorithms like Haar Features, convolutional neural networks, or feature discriminant analysis. For more basic datasets such as MNIST which is extremely polished and compact, features can be extracted simply from taking equal size grids from the image as individual features via a sliding window method. The more complex the dataset the greater the need to deploy complex low is then fed into a feature scorer.

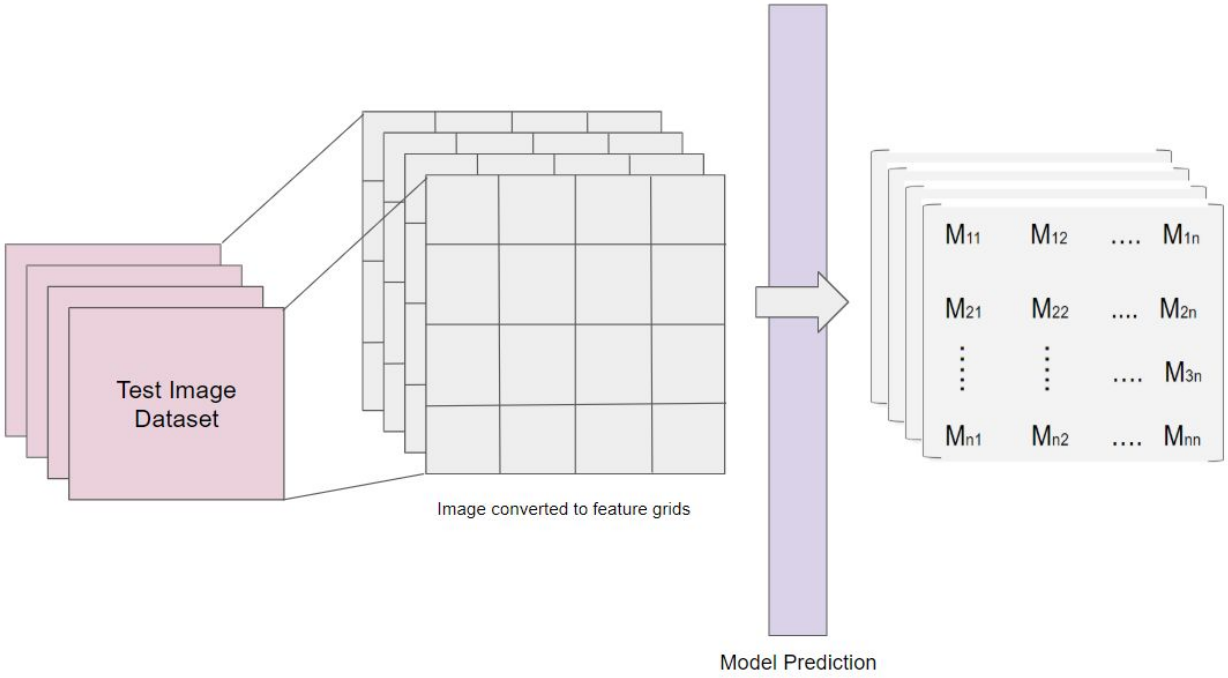


Stage 2- Feature Scoring

After low-level feature extraction, the feature scorer accepts the various low-level features and performs loose clustering on them to form clusters of similar features. For this, clustering techniques like K-Means, Mean-shift or Spectral Clustering can be used as they all club various features based on their location to the nearest cluster centroid. After clustering, each cluster is given its own unique identification (ID) and each feature under that cluster is labelled as the unique ID which symbolised that cluster, hence "symbolic ID".



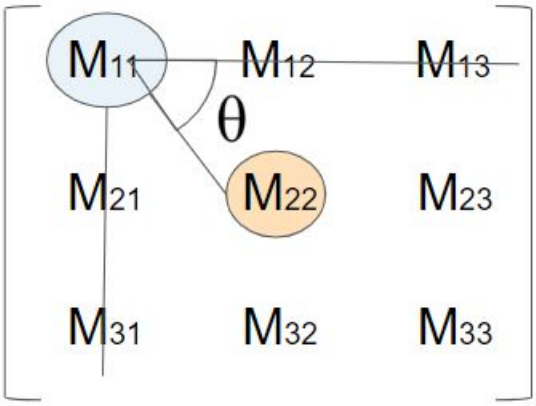
Now a new test dataset is analysed and labelled according to the predictions from the prior training model. The test images are again made into equal size grids and are swapped with the predicted symbolic ID for that test feature. Thus we have converted the set of training images from the image domain to the symbolic ID domain, where each feature is represented by its respective ID. These matrices are stored in NumPy files and sent to the next block which determines co-occurrence between each feature in the test image dataset.



Stage 2.5- Hardcoded Directional Matrix

To determine the co-occurrence between two points or features in an image, the spatial location of each point is necessary. It is important to extract and determine if a point lies to the north, south, east or west to the other and by what degree. To do this, calculating the spatial location of each feature in every image is not only tedious but also unnecessary.

This is because, for a given nxn size of an image or ID matrix, the relative position between points within the matrix remains the same, irrespective of the position contents. Therefore it is possible to apply a hard-coded directional matrix to each test image which contains all the directional features of a certain point.



The above figure shows the method of calculating the directional component between two entities M11 and M22. The angle θ is obtained by calculating the slope of the two points. Since the indices of a matrix are spatially equivalent to the points on an axis M11 indicates (1,1) while M22 indicates (2,2). Thus by finding the slope of line M11 and M22 we are able to obtain the angle by finding its inverse tangent value in radians. The North and South values can be found by calculating $\sin^2\theta$ while the West and East component are found using $\cos^2\theta$. This process of finding the directional values is iterated for all possible combination of entities with taking one point as the pivot and then finding the directional value of that pivot point with the remaining points. All the directional data related to that pivot point is then stored in an array and a new pivot point is chosen and the process is repeated. All these arrays are stored in another 2-dimensional array with the same size as the symbolic matrices.

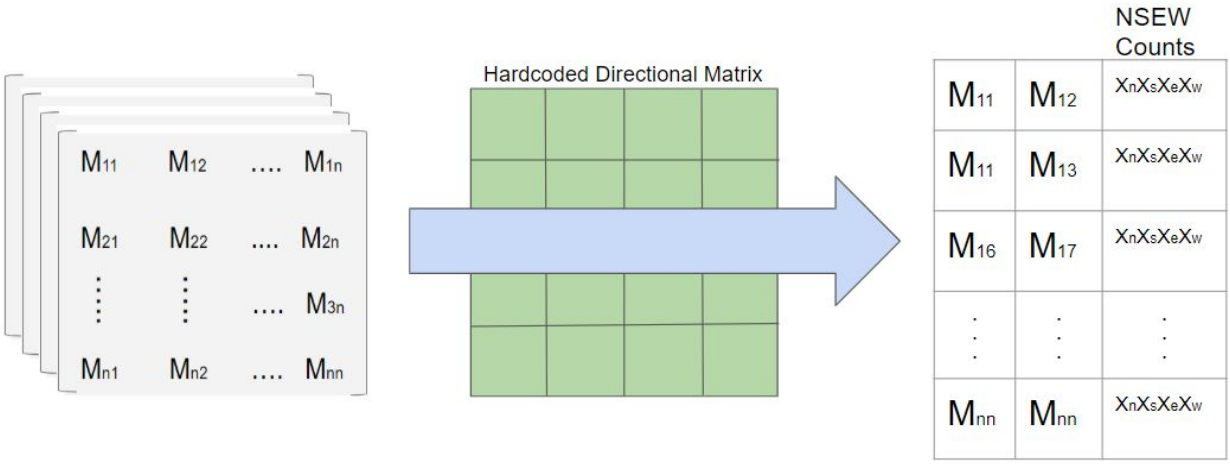
Thus what we have essentially is an nxn NumPy array where nxn is the size of the symbolic ID matrix created in the Feature Scoring Block. Each element at position (n,n) in the array is a matrix which contains the north, south, east or west quantities of every spatial point in the nxn array relative to the location (n,n). Thus the Hard-coded Directional Matrix formed and fed into the Patch wise co-occurrence block along with the Test dataset symbolic ID matrices formed during feature scoring.

Stage 3- Patch wise co-occurrence

Now at this stage using the symbolic ID matrices of the test dataset and the hardcoded directional matrix we are able to divide ever possible ID pair into 4 directional tables: North, South, East, West. This is done to analyse how each point/feature co-occurs with others with respect to position. The positional element is considered due to the fact that the dataset consists of images, which are (essentially highly spatially oriented matrices). Thus observing how each feature co-occurs with the others at a low level can be a very important characteristic in understanding how features occur at a higher level.

E.g. Let's say, that feature 'A' is frequently occurring to the north of feature 'B' and to the east of feature 'C'. This information would extremely important during the community detection stage later on.

Each symbolic ID matrix, representing an image, is superimposed on the hard-coded directional matrix and the symbolic ID pairs along with their directional values are appended onto their respective tables directional tables. After processing every symbolic ID matrix, each table is further compressed and refined by aggregating the respective directional component of each unique pair in the matrix. This can easily be done o python using pivot tables, where the Symbolic ID pair is taken as the pivot table index, and the directional value is aggregated. This aggregated directional value is referred to as count.



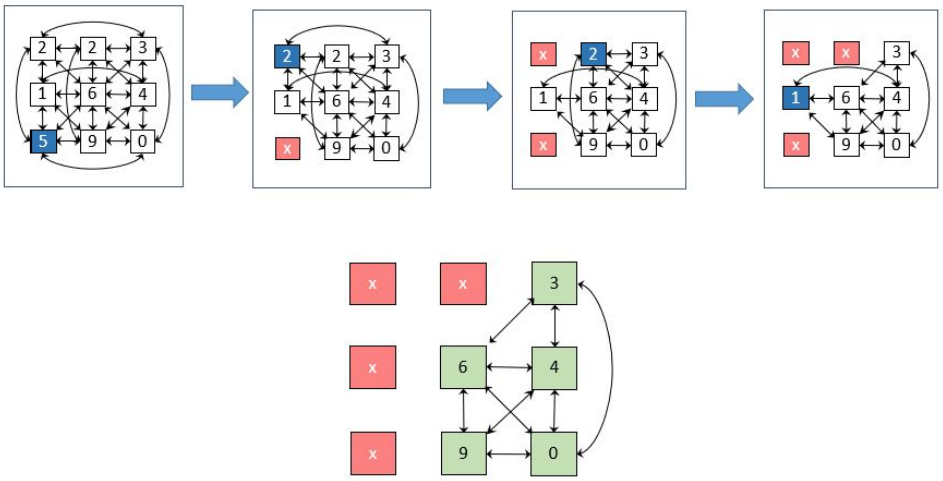
The directional count value is converted to consistency using a modified NPMI formula given in Equation 4. P(AB) is the joint count of A and B, While P(A) and P(B) is the count of all the instances of Entity A and B respectively. The resulting values are normalised and are in the range of -1 and 1 to help scale the data.

$$NPMI(AB) = \frac{\log \frac{P(AB)}{P(A)*P(B)}}{\log \frac{1}{P(AB)}} \qquad (4)$$

After obtaining the final directional North, South, East, West tables containing the symbolic ID pairs and their counts and consistencies, the tables are then fed into the Community detection algorithm in the "Community Modelling Stage".

Stage 4

For the process of Community Detection, the algorithm starts with the maximum number of features interconnected together. A placeholder value is kept indicating the consistency of this community as a whole. Then the soft maximal clique is found by dropping one feature at a time in an attempt to increase the consistency. Thus as the community becomes smaller, the consistency value of the community increases to a point, after which the consistency cannot be increased by removing a node.



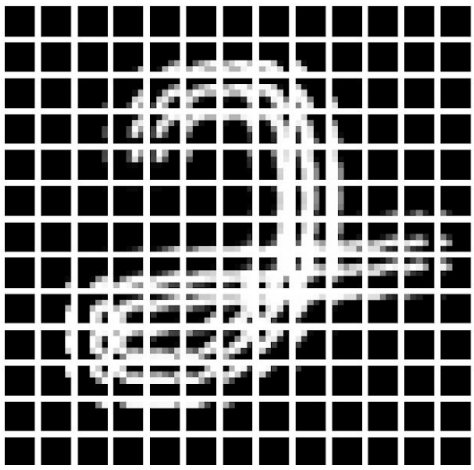
Stage 5 Community Modelling Stage

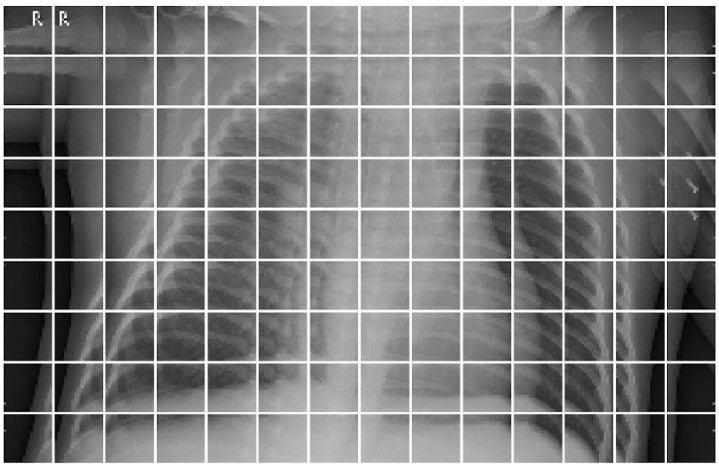
At this stage, the given features detected in the community detection stage are analysed and complied to detect the effectiveness and accuracy of the given system. The communities are structured and stored and the final list of features and communities are presented as an output to the system.

Implementation

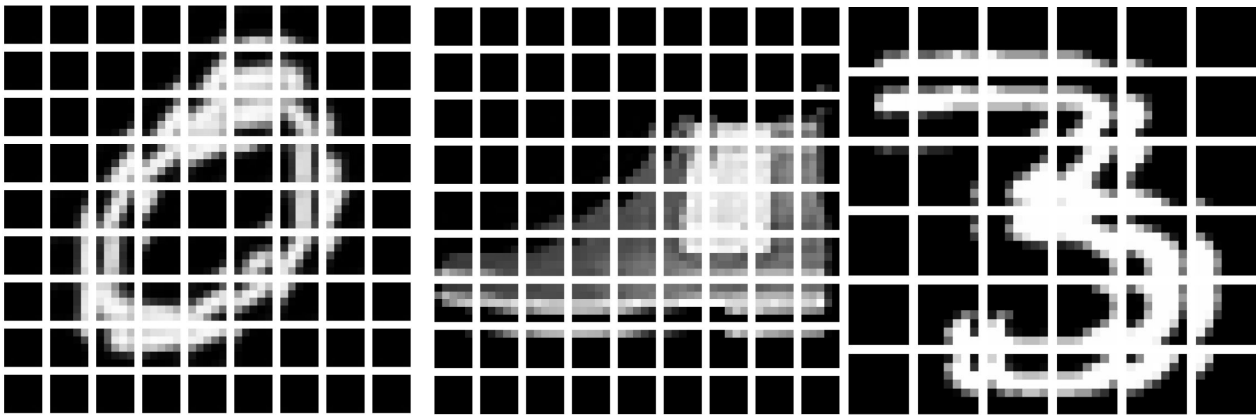
Dataset and Feature Set

For the initial implementation, a set of sample images were taken from the MNIST Data set which contains 28x28 grayscale images of handwritten numbers ranging from 0 to 9. Due to its simplicity and compact nature, it is the perfect test dataset. Since the images in the dataset are already grayscale and of an appropriate size, it does not require much data pre-processing and formatting. Testing was also conducted on a Chest X-ray dataset to estimate the efficiency of the system on more complex datasets.

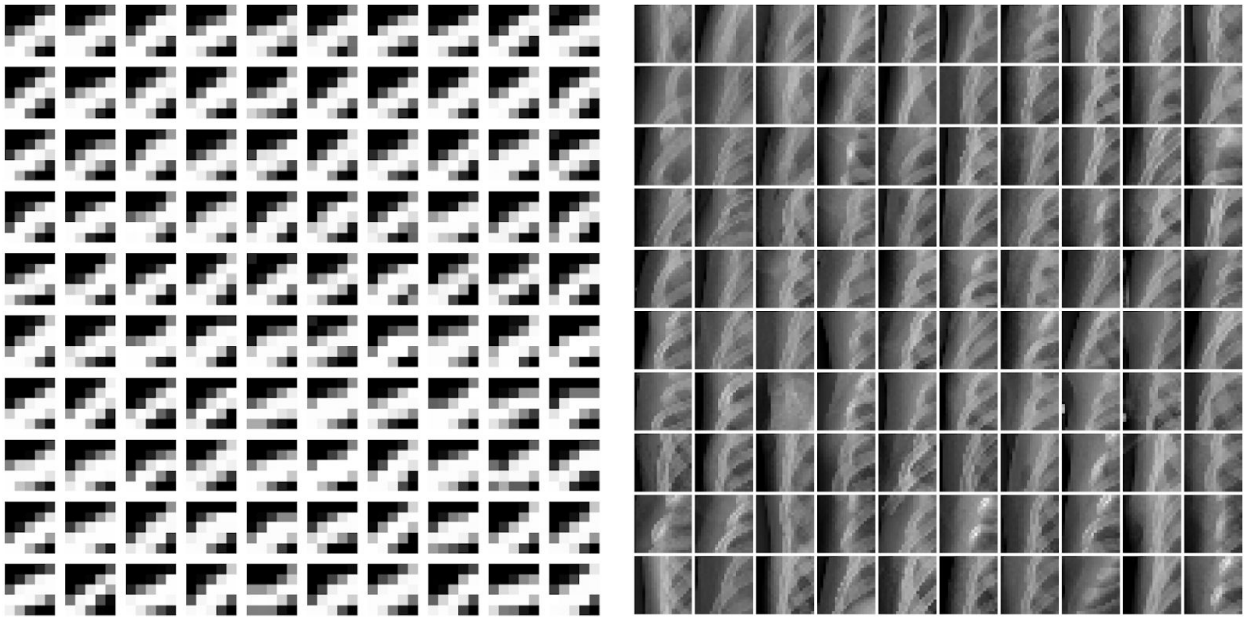




For the feature set, the image was divided into multiple nxn overlapping boxes with a stride of (n+1)/2. These boxes act as the low-level features that are fed into later stages of the system. The above figure shows the several overlapping 5x5 feature boxes formed with a stride of 3 from an image in the MNIST dataset for the handwritten number '2' along with a 20x20 feature set of an image in the Chest X-Ray dataset with a stride of 10. Such features are extracted from all the given images in the training dataset and are compiled into a single array of feature files and are called the feature set. Other features sets were also analysed and compiled:

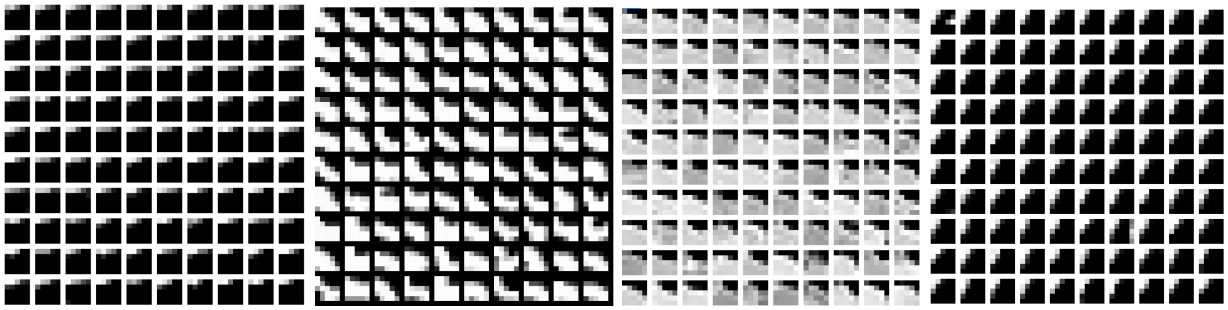


Loose Clustering and Parameters



The above figure shows the result of K-means clustering on the 5x5 feature set and the Chest X-Ray feature set obtained during feature extraction. It has adequately segregated the data points into clusters of similar features. Each cluster is assigned a unique identification number according to its cluster number.

A new test image feature set generated from the test dataset, which is then loaded into the Feature Scoring module along with the trained K-means model. Each image is divided into blocks of features and each feature is replaced with the unique ID predicted by the Kmeans model. This results in an array of mxm matrices, where each matrix is the symbolic ID representation of the respective image. Thus we have successfully concerted the test dataset from the image domain to the symbolic ID domain. Some other clusters found were as follows:

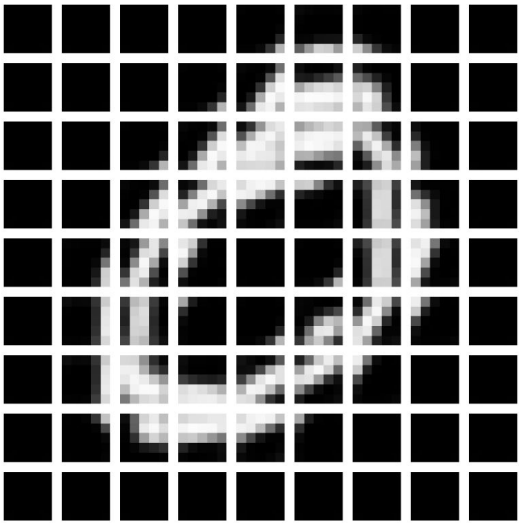


Parameters: 'algorithm': 'auto', 'copy_x': True, 'init': 'k-means++', 'max_iter': 300, 'n_clusters': 75, 'n_init': 10, 'n_jobs': 'deprecated', 'precompute_distances': 'deprecated', 'random_state': None, 'tol': 0.0001, 'verbose': 0

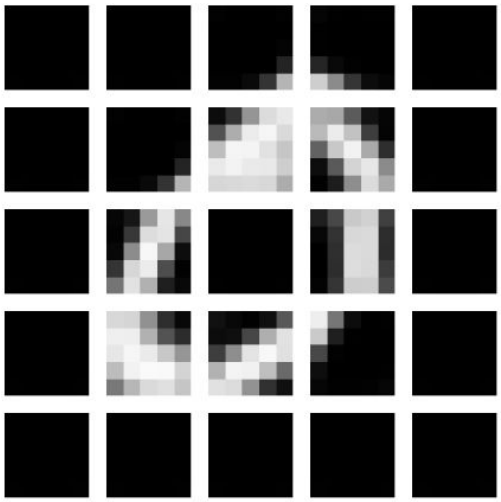
Midway Progress Checking

To confirm whether the loose clustering algorithm has correctly clustered the given feature set, the given test image is converted to the symbolic ID Domain and then each ID is replaced with the centroid of that given cluster-ID. This way if there has been an error in the clustering process, the output image would be distorted, giving an indication that the clustering was unsuccessful at correctly grouping the feature set

Test image with 5x5 centroid patches



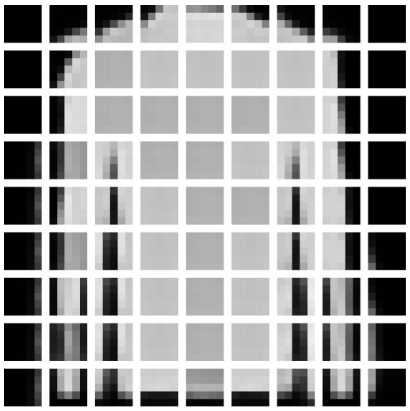
After Removing Strides

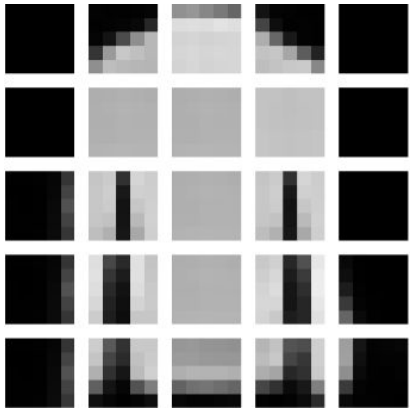


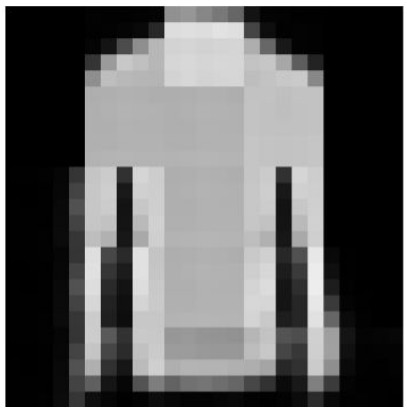
[2, 2, 2, 2, 16, 51, 34, 2, 2],
[2, 2, 2, 16, 60, 30, 27, 2, 2],
[2, 2, 16, 60, 75, 21, 37, 17, 2],
[2, 2, 60, 41, 5, 63, 11, 39, 2],
[2, 65, 15, 28, 2, 16, 11, 29, 2],
[2, 62, 26, 43, 16, 52, 61, 7, 2],
[2, 62, 71, 36, 12, 47, 7, 2, 2],
[2, 56, 66, 59, 5, 43, 2, 2, 2],
[2, 2, 2, 2, 2, 2, 2, 2, 2]

[2, 2, 16, 34, 2],
[2, 16, 75, 37, 2],
[2, 15, 2, 11, 2],
[2, 71, 12, 7, 2],
[2, 2, 2, 2, 2]

Other Examples :

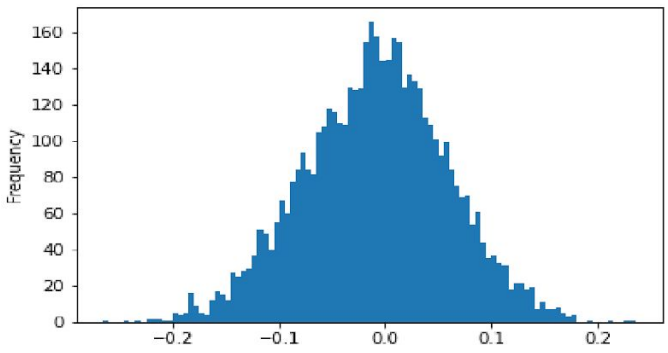


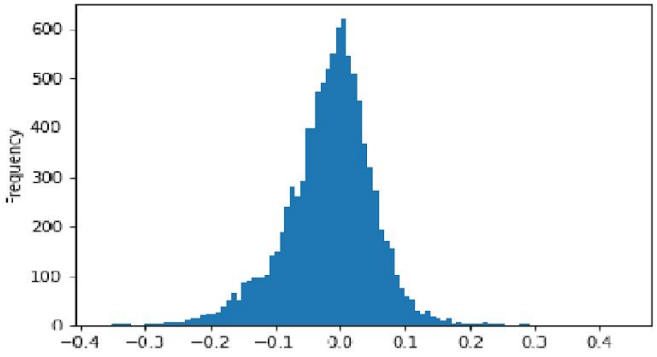




Patch to Patch Co-Occurrence

The given symbolic ID matrices are then run through the hardcoded directional matrix to obtain the North, South, East and West tables with each row containing two unique ID values and their combined count in that particular direction. This count table is converted into consistencies using the NPMI formula, giving consistencies in the range of -1 and 1, between each ID pair. The figure below shows the histogram plot of the consistency values for the North table for a 5x5 feature set of the MNIST dataset.





The directional tables along with their consistency values are sent into the community detection algorithm in the next stage. The community Detection Algorithm would then detect useful communities in the dataset based on the given NPMI values between various features. Thus the final output would be various communities of features which are highly coherent together and can be taken as important features of that dataset.

NORTH

a_id	b_id	count_north	north_consistency
2	2	9563.6	0.087990
2	18	761.8	-0.051386
18	2	662.4	-0.051842
2	50	637.6	-0.057249
2	1	603.6	-0.036108
...	...	...	...
68	95	0.2	-0.053325
139	71	0.2	-0.018385
139	79	0.2	-0.081458
68	106	0.2	-0.067261
126	23	0.2	-0.090473

SOUTH

a_id	b_id	count_south	south_consistency
2	2	9563.6	0.087990
18	2	761.8	-0.051386
2	18	662.4	-0.051842
50	2	637.6	-0.057249
1	2	603.6	-0.036108
...	...	...	...
42	26	0.2	-0.148043
42	42	0.2	-0.106636
42	55	0.2	-0.094912
126	56	0.2	-0.199285
74	71	0.2	-0.219101

WEST

a_id	b_id	count_west	west_consistency
2	2	7483.4	-0.005078
18	2	980.3	0.010353
2	18	882.5	-0.001891
50	2	878.5	0.016848
2	50	833.7	0.014526
...	...	...	...
128	73	0.2	-0.116126
42	105	0.2	-0.042035
128	40	0.2	-0.119251
43	46	0.2	-0.099535
124	134	0.2	-0.117219

EAST

a_id	b_id	count_east	east_consistency
2	2	7483.4	-0.005078
2	18	980.3	0.010353
18	2	882.5	-0.001891
2	50	878.5	0.016848
50	2	833.7	0.014526
...	...	...	...
101	13	0.2	-0.055396
128	137	0.2	-0.079107
134	139	0.2	-0.088910
122	37	0.2	-0.082641
82	108	0.2	-0.066080

Conclusion

The given system was tried and tested with MNIST, Fashion MNIST and a Chest X-ray Dataset, where the features were successfully extracted and clustered. The relative strength of these features was calculated and fed into the community detection algorithm to correctly extract important feature communities. The system was proved to be an effective agnostic data mining tool which was able to obtain intuitive and relevant features for the datasets without any labelling or supervised components. This algorithm successfully combined the concept of SIFT Features with unsupervised learning algorithms such as K-Means Clustering using Co-Occurrence Analytics in the image domain.

Future Work

The process could be incorporated in the feature recognition part of many popular machine learning models and could increase the efficiency of a system by allowing the model to focus on important features and by discarding outlets in a dataset. Another possible approach could be the revision to the concept of dropouts in neural networks, where instead of relying on a purely random selection of feature dropping, dropouts could be implemented in a more intuitive way. Thus a previously 'black-box' method of neural networks could have a logical intuitive structure that one could build upon and improve.

The project can also be extended to various other image and natural language processing datasets by substituting the directional tables with Bigrams i.e. distance data for 2 given words, allowing us to find trends in language and can also be integrated with a neural network algorithm by mining important features from a dataset to be used by the model.

References

[1] M. Wei and P. Xiwei, ”WLIB-SIFT: A Distinctive Local Image Feature Descriptor,” 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP), Weihai, China, 2019, pp. 379-383, doi: 10.1109/ICICSP48821.2019.8958587.

[2] H. Lv, X. Huang, L. Yang, T. Liu and P. Wang, ”A k-means clustering algorithm based on the distribution of SIFT,” 2013 IEEE Third International Conference on Information Science and Technology (ICIST), Yangzhou, 2013, pp. 1301-1304, doi: 10.1109/ICIST.2013.6747776.

[3] S. Kumar, C. V. and C. V. Jawahar, ”Logical Itemset Mining,” 2012 IEEE 12th International Conference on Data Mining Workshops, Brussels, 2012, pp. 603-610, doi: 10.1109/ICDMW.2012.85.

[4] M. Bandara, S. Weragoda, M. Piraveenan and D. Kasthurirthna, ”Overlay Community detection using Community Networks,” 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 680-687, doi: 10.1109/SSCI.2018.8628653.

[5] M. Aghagolzadeh, H. Soltanian-Zadeh, B. Araabi and A. Aghagolzadeh, ”A Hierarchical Clustering Based on Mutual Information Maximization,” 2007 IEEE International Conference on Image Processing, San Antonio, TX, 2007, pp. I - 277-I - 280, doi: 10.1109/ICIP.2007.4378945.

[6] Newman, M. & Cantwell, George & Young, Jean-Gabriel. (2020). Improved mutual information measure for clustering, classification, and community detection. Physical Review E. 101. 10.1103/PhysRevE.101.042304.

[7] A. Amelio and C. Pizzuti, ”Is normalized mutual information a fair measure for comparing community detection methods?,” 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Paris, 2015, pp. 1584-1585, doi: 10.1145/2808797.2809344.

[8] P. Chejara and W. W. Godfrey, ”Comparative analysis of community detection algorithms,” 2017 Conference on Information and Communication Technology (CICT), Gwalior, 2017, pp. 1-5, doi: 10.1109/INFOCOMTECH.2017.8340627.

[9] Jianying Hu, Haitao Lang, Wei Hu and Ling Zhou, ”Image classification with visual words co-occurrence matrix,” Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shengyang, 2013, pp. 1172-1176, doi: 10.1109/MEC.2013.6885242.

[10] Y. Benezeth, P. -. Jodoin, V. Saligrama and C. Rosenberger, ”Abnormal events detection based on spatio-temporal co-occurences,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 2458-2465, doi: 10.1109/CVPR.2009.5206686

[11] I. Infantino, F. Vella, G. Spoto and S. Gaglio, ”Image Representation with Bag of bi-SIFT,” 2009 Fifth International Conference on Signal Image Technology and Internet Based Systems, Marrakesh, 2009, pp. 287-293, doi: 10.1109/SITIS.2009.54.

[12] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, ”Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[13] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”, Mendeley Data, http://dx.doi.org/10.17632/rscbjbr9sj.2

Closing Remarks

My internship with Jio has given me a great opportunity to grow and learn. I have got a good grasp of how machine learning techniques work and learnt new and exciting concepts such as Co-occurrence analytics. I have realised the importance of making a given system data agnostic to ensure that it can be implemented easily and seamlessly in various fields.

While it was a remote internship, the ease of access to resources and guidance, coupled with constant mentoring ensured that I was learning and contributing every step of the way.

During my training, I enjoyed being challenged every day and have gained valuable practical experience and skills, which has helped me reach closer to my career objective.

Thank You.