# Good Evening Sir, this is zeel and my team mates Joyeeta and Apoorv, we are here to present our j component rev 1. Our topic is A MapReduce-based distributed SVM algorithm for automatic image annotation

# In particular we are carrying out image annotation on arial images, which is part of POST-DISASTER DAMAGE ASSESSMENT

Support Vector Machine (SVM) is extremely powerful and widely accepted classifier in the field of machine learning due to its better generalization capability. However, SVM is not suiTable for large scale dataset due to its high computational complexity. The computation and storage requirement increases tremendously for large dataset. In this project , we have proposed a MapReduce based SVM for large scale data MapReduce distribution model works on several frame works like Hadoop . MapReduce is a distributed programming model which works on large scale dataset by dividing the huge datasets in smaller chunk. It partitions the training data set into smaller subsets and train SVM in parallel using a cluster of computing nodes.

Support Vector Machine (SVM) techniques have been used extensively in automatic image annotation

Automatic image annotation is a method in which it automatically generates one or more labels to describe the content of an image. It is a process which is commonly considered as a multi-class classification. In this, images are annotated with labels based on the extracted low-level features​

1.

​However their implementation of the algorithm using a synchronous communication model due to MPI’s lack of support for asynchronous communication which could affect the speed of training time​

 However the frequent reallocation of training data during the optimization process may cause a reduction in the training speed​

3.

However the performance of the parallel implementation is heavily depended on which caching strategy is used to avoid re-computation of previously used elements.

In contrast to the exiting solutions, our approach is simple, efficient and scalable

Future

As part of future research work, there can be planning made to introduce load balancing techniques in order to achieve optimal resource utilization. We believe that load balancing would further enhance the performance of our algorithm in terms of speed.

planning can also be done to consider heterogeneity in training datasets as well as available processors in a cluster environment, in which the computation tasks have to be distributed and scheduled among all of the available processors equally to improve the overall performance.

Conclusion

 The performance of the MRSMO algorithm is evaluated in an experimental environment. By partitioning the training dataset into smaller subsets and optimizing the partitioned subsets across a cluster of multiple computing nodes, the MRSMO algorithm reduces the training time significantly while maintaining a high level of accuracy in binary and multiclass classifications especially for large size training dataset.