

Siamese Networks for Chromosome Classification

BioImage Computing Workshop, ICCV 2017

Authors - **Swati**, Gaurav Gupta, Mohit Yadav, Monika Sharma, Lovekesh Vig

Outline

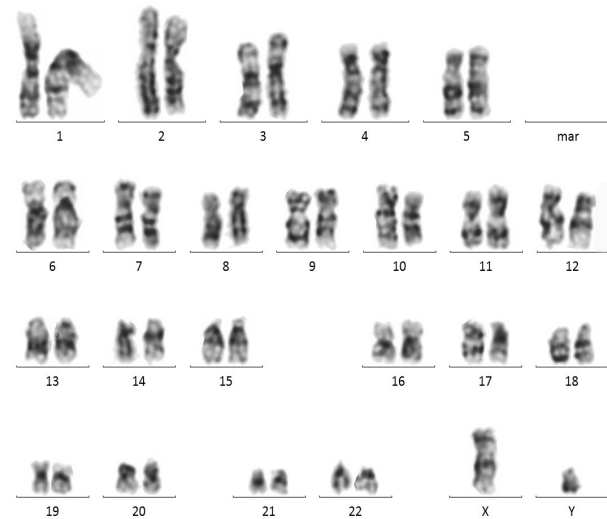
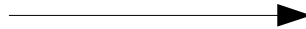
- Introduction
- Motivation
- Proposed Methodology
- Experiments
- Results
- Conclusion

Introduction

- During the metaphase stage of cell division, the condensed chromosome images are Giemsa stained under a light microscope.
- Karyotyping consists of the identification and classification of the 23 pairs of the chromosomes, obtained from a single cell.



Cell spread during metaphase stage

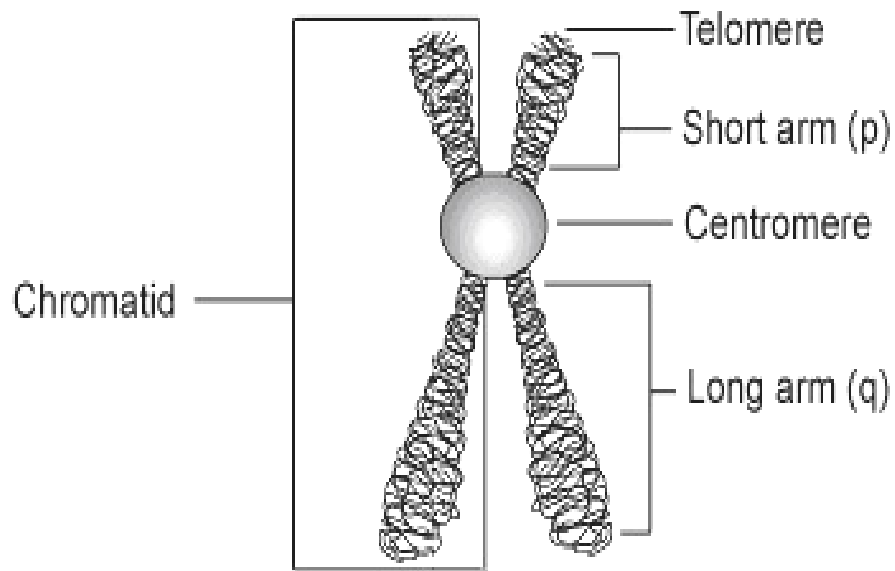


Desired karyotyped image

Characteristics of Chromosomes

These distinct features help in chromosome classification.

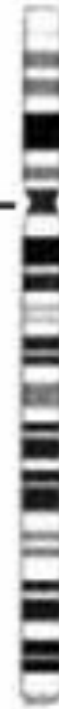
- **Length**
- **Centromere Position**
- **Banding Pattern**



chromosome 4
(G-band pattern)

centromere

human



Basic Process

- Images were obtained from Tata Memorial Hospital (TMH)
- Segmentation is done either using automated process or through Crowdsourcing.
- For our experiments, we assumed that we have segmented chromosome images.



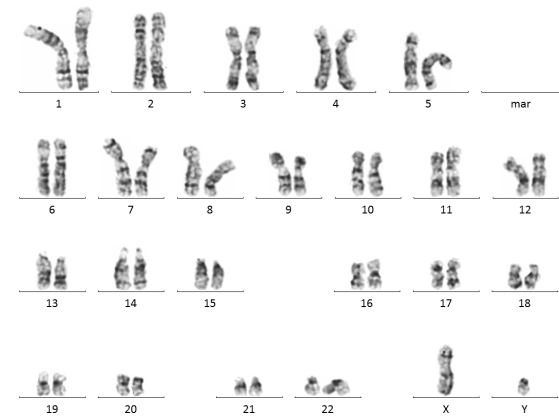
Metaphase image obtained
from TMH



Segmentation stage



Classifier



Karyotyped image

Motivation

- Many genetic disorders arise due to various translocations, deletions or inversions of chromosomes.
- Doctors analyze chromosome images for the diagnosis of these disorders.
- Considerable manual effort and time is required to produce a karyotyped image.
- This motivates to **automate the task of karyotyping** and assist expert doctors.

Prior work

- Previously, chromosome classification has been carried out based upon-
 - Geometrical features – Pixel distribution, Length, Centromere position [1]
 - Band pattern based features – Intensity of Band profiles [2]
 - Wavelet transform based features [3]
 - 2 stage Hierarchical ANN [4]
- To the best of our knowledge, Deep Learning was not used for chromosome classification earlier.
- Recently, the use of Deep CNN for chromosome classification was proposed in CVPR 2017 paper by TCS Delhi and Pune labs.
 - Demo available in Samuhaa platform (image.samuhaa.com/chromosome/)

[1] Cho, J., S. Y. Ryu, and S. H. Woo. "A study for the hierarchical artificial neural network model for giemsa-stained human chromosome classification." Engineering in Medicine and Biology Society, 2004.

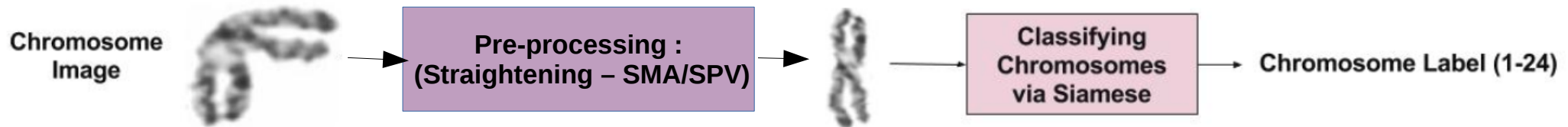
[2] Madian, Nirmala, and K. B. Jayanthi. "Analysis of human chromosome classification using centromere position." (2014)

[3] M. Javan-Roshtkhari and S. K. Setarehdan. "A new approach to automatic classification of the curved chromosomes," ISPA 2007.

[4] Wang, Xingwei, et al. "Automated classification of metaphase chromosomes: optimization of an adaptive computerized scheme." Journal of biomedical informatics (2009).

Proposed Methodology

- In this work, we intend to improve the classification accuracy when labeled data is scarce.
- Pre-processing is done to straighten the bent chromosomes to improve the classification accuracy.
 - SPV (Straightening Via Projection Vectors)
 - SMA (Straightening Via Medial Axis)
- Straightened images are fed to Siamese networks for classification.

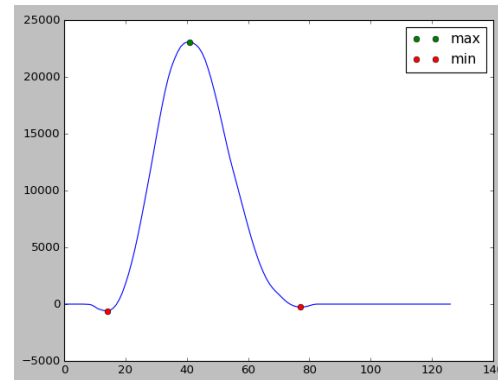
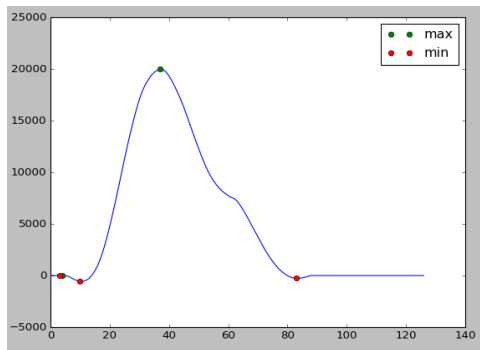
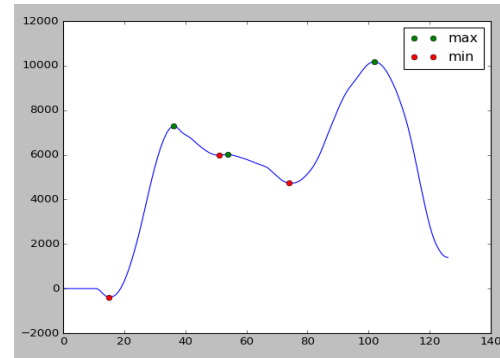
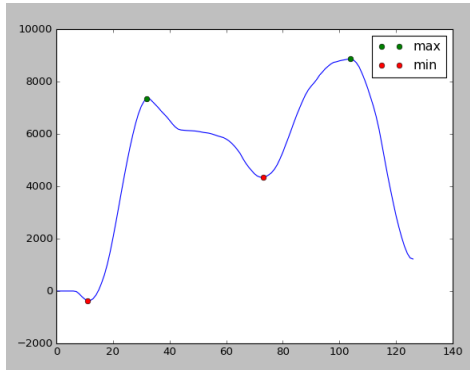


Straightening via Projection Vectors (SPV)

- Finding threshold and binarization --> B
- Find bending orientation --> C
- Find bending centre of curved chromosomes --> D
- Splitting and aligning individual arms --> E
- Stitching arms and reconstruction --> F and G



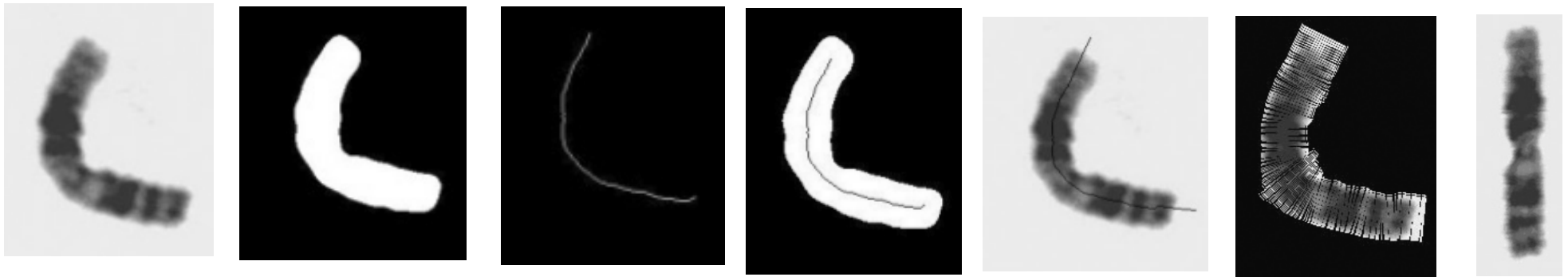
Locating Bending Axis



Horizontal projection distribution showing maxima and minima for different rotation angles

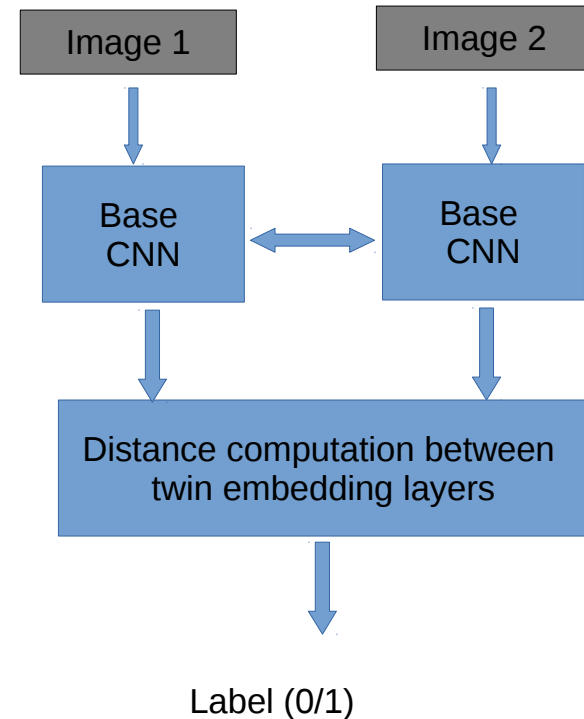
Straightening Via Medial Axis (SMA)

- Thresholding and Binarization
- Find MA by thinning procedure
- Find end points of MA
- Overlap MA on original chromosome
- Extend MA to cover whole chromosome
- Find perpendicular lines to MA
- Copy the pixel values to new image



Siamese Networks

- Siamese networks have twin neural networks (CNN in our case) with shared weights.
- Learn to predict the similarity or dissimilarity between images.
- Twin networks are joined by a distance function computed between representations of the highest layer – Euclidean Distance.
- Nearest neighbour is used for final prediction of label.
- Loss function used is contrastive loss.



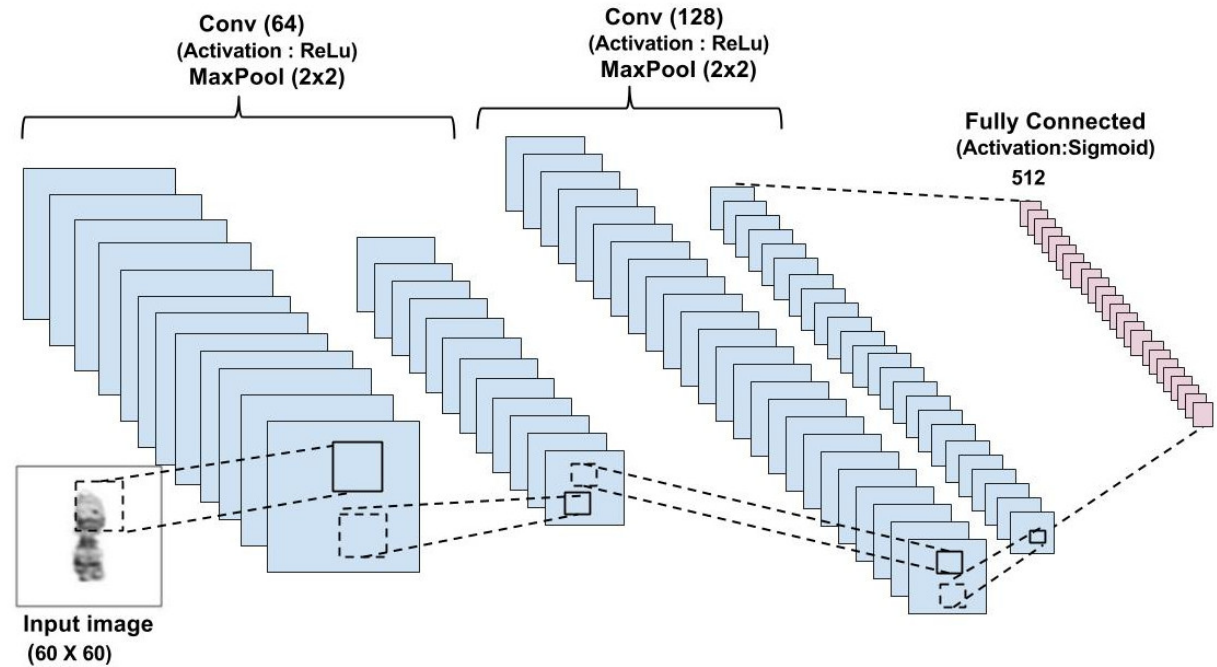
$$L(S, W) = \sum_{i=1}^N \sum_{j=i}^N E_W(y_{ij}, X_i, X_j)$$

$$E_W(y_{ij}, X_i, X_j) = \frac{1}{2} \times (1 - y_{ij}) \times (D_W(X_i, X_j))^2 \\ + \frac{1}{2} \times y_{ij} \times (\max\{m - D_W(X_i, X_j), 0\})^2$$

where, X_i = Image1, X_j = Image2,
 y_{ij} = Label (0/1)
 m = margin
 D_w = Euclidean Distance

Architecture of Base CNN

- Conv2D (64,3,3)
 - Activation = 'relu'
 - MaxPool2D (2,2)
 - Dropout (0.25)
 - Conv2D (128,3,3)
 - Activation = 'relu'
 - MaxPool2D (2,2)
 - Dropout (0.25)
 - Dense (512)
 - Activation = 'sigmoid'
-
- The last layer 'Dense(512)' is used as twin embedding layer.



Challenges

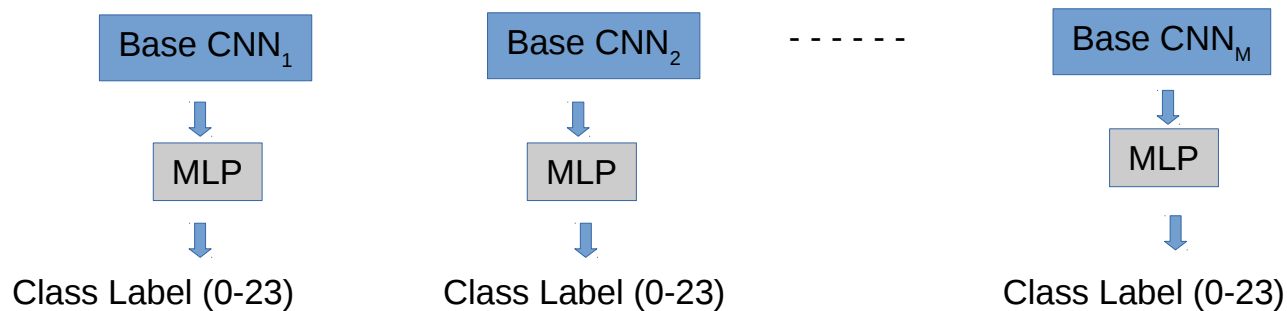
- Skewed data i.e., large ratio of dissimilar to similar pairs (R).
- Prediction requires all the embeddings for training and testing data to be in memory for finding nearest neighbour.

Data creation

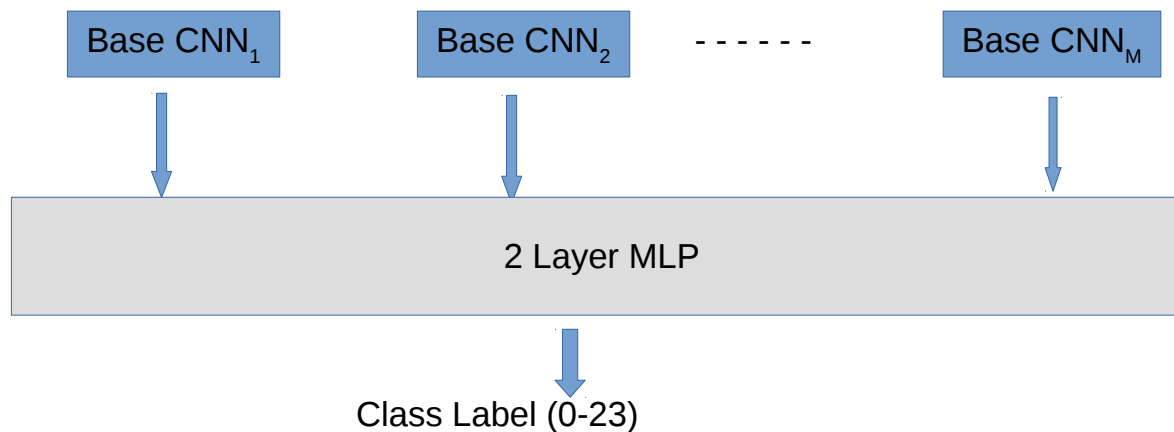
- To solve the problem of skewness, we fixed the ratio of dissimilar to similar pairs $(R) = 2$
 - a) *Train multiple siamese models(M) with random selection of dissimilar pairs and including all similar pairs.*
 - e.g., if P is total number of similar image pairs, then 2P dissimilar image pairs are selected randomly (for every model).
 - b) *Train multiple siamese models(M) with same number of dissimilar and similar pairs of images (balanced) from all the possible different class label pairs.*
 - We have fixed the number of similar pairs to be chosen from every class.
 - For 24 classes, we have $^{24}C_2$ class pairs.
 - From each class pair, choose equal number of samples such that total number of dissimilar pairs is twice the number of similar pairs.

Proposed Architectures

- To reduce the time for prediction, we trained two types of augmented Siamese Networks :
 - Training Multi-layer perceptrons (MLPs) on top of all siamese base network embeddings.



- Training an MLP by concatenating the embedding of all siamese base networks.



Experiments

- Baselines were created using :
 - 2 Layer MLP trained on Flattened images (for both straightened and unstraightened images)
 - Deep Convolutional Neural Network (for both straightened and unstraightened images)
 - Vanilla Siamese Network (CNN) + Nearest neighbour
- Proposed methods :
 - Siamese Networks + MLP
 - Concatenated Siamese Networks + MLP (Con-Saimese Networks)
- All experiments were performed using both SMA and SPV straightening methods except for Vanilla Siamese network.

Experimental Setup

- Number of images:
 - Training = 1296
 - Validation = 235
 - Test = 209
- Size of images = 60 * 60 grayscale
- Number of dissimilar to similar pairs :
 - Vanilla Siamese networks = 817K : 36.6K
 - Data creation (a) = 73.2K : 36.6K (for each siamese model)
 - Data creation (b) = 18K : 9K (for each siamese model)
- Number of siamese models trained = 10
- Number of output labels = 24
- GPU Used : Quadro M3000M/PCIe/SSE2 (4GB) , RAM = 64 GB

Results - Baselines

| Network | Accuracy | Epochs × PET | Prediciton time |
|---|--------------|------------------------------------|-----------------|
| 2 Layer MLP (unstraightened) | 59.7% | 1921×0.3 sec (0.16 hrs) | 4.088 sec |
| 2 Layer MLP + SMA | 67.9% | 1871×0.3 sec (0.16 hrs) | 4.088 sec |
| 2 Layer MLP + SPV | 72.3% | 1791×0.3 sec (0.15 hrs) | 4.088 sec |
| Deep CNN (unstraightened) | 68.5% | 4700×4.1 sec (5.35 hrs) | 12.085 sec |
| Deep CNN + SMA | 78.4% | 1832×4.1 sec (2.09 hrs) | 12.085 sec |
| Deep CNN + SPV | 83.7% | 1957×4.1 sec (2.23 hrs) | 12.085 sec |
| Vanilla Siamese network + Nearest + SPV | 85.6% | 155×2844 sec (124.17 hrs) | 15.088 sec |

* SPV = Straightening Via Projection Vectors
 SMA = Straightening Via Medial Axis
 PET = Per Epoch Time
 Epochs = Best validation accuracy

Our Results

| Network | Accuracy | Epochs × PET | Prediciton time |
|-----------------------------------|--------------|--------------------------------|-----------------|
| Siamese network + MLP + SMA (Avg) | 78.5% | 18.2×365 sec (1.84 hrs) | 4.760 sec |
| Siamese network + MLP + SMA (Max) | 79.4% | 19×365 sec (1.92 hrs) | 4.760 sec |
| Siamese network + MLP + SPV (Avg) | 81.3% | 17.4×365 sec (1.76 hrs) | 4.760 sec |
| Siamese network + MLP + SPV (Max) | 83.8% | 18×365 sec (1.83 hrs) | 4.760 sec |

Results for Data creation (a)

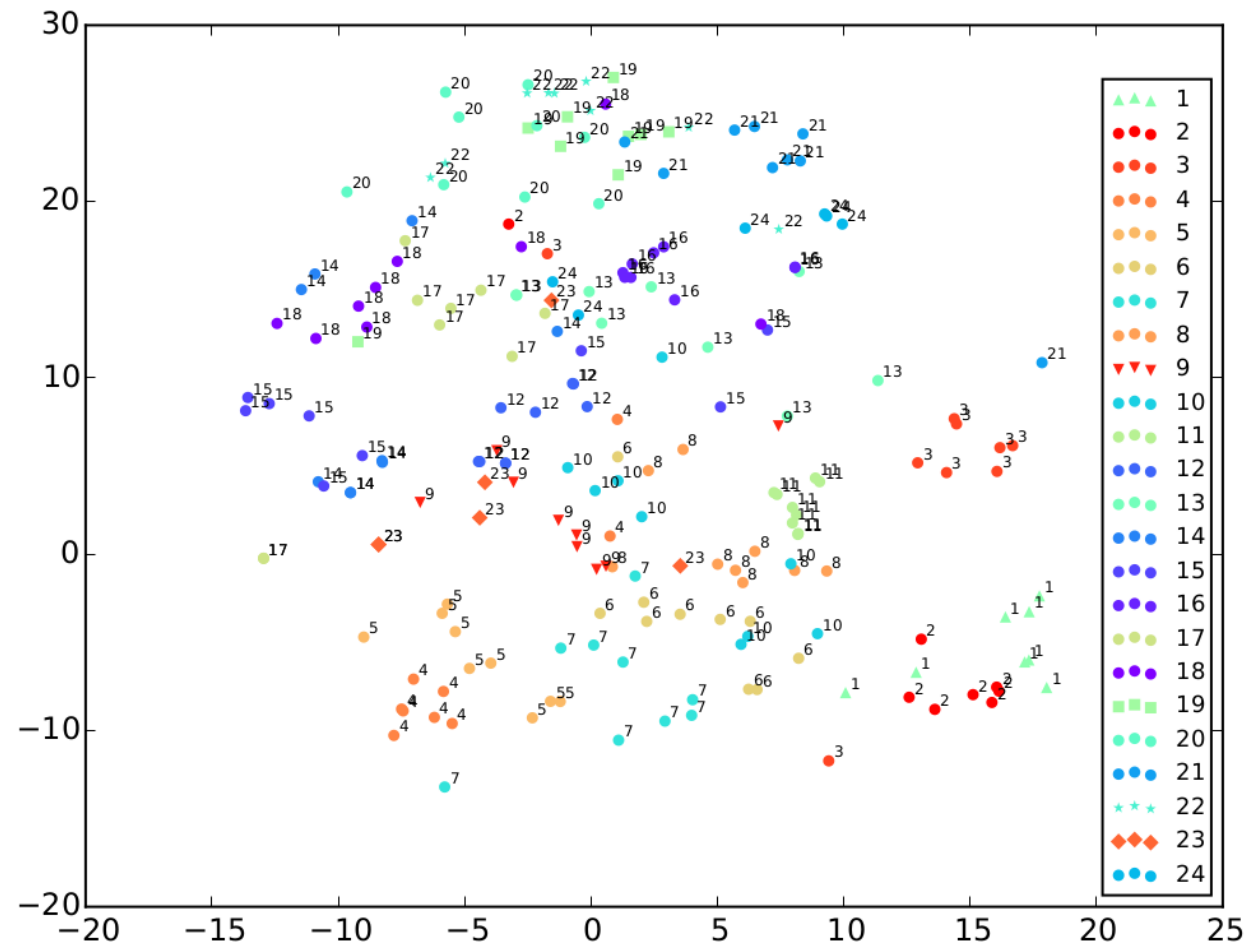
Our Results

| Network | Accuracy | Epochs × PET | Prediciton time |
|-----------------------------------|--------------|-------------------------------|-----------------|
| Siamese network + MLP + SMA (Avg) | 78.6% | 62.8×90 sec (1.57 hrs) | 4.760 sec |
| Siamese network + MLP + SMA (Max) | 80.4% | 63×90 sec (1.58 hrs) | 4.760 sec |
| Siamese network + MLP + SPV (Avg) | 83.3% | 61.2×90 sec (1.53 hrs) | 4.760 sec |
| Siamese network + MLP + SPV (Max) | 85.2% | 60×90 sec (1.50 hrs) | 4.760 sec |
| Con-Siamese networks + MLP + SMA | 79.8% | 61×90 sec (1.52 hrs) | 4.760 sec |
| Con-Siamese networks + MLP + SPV | 84.6% | 60×90 sec (1.50 hrs) | 4.760 sec |

Results for Data creation (b)

T-SNE plot

- Plotted t-SNE plot for Siamese network + MLP + SPV (Max) case for which we got best results.



Conclusions

- Vanilla siamese network surpasses the performance of Deep CNN but training time is too high.
- Proposed methods speed up both training and prediction by ***83 and 3 folds***, respectively.
- Augmented siamese networks maintains the superior performance of vanilla Siamese network.

THANK YOU!