Remove as much noise as possible

1. zoomImage.py

Crop image to squared matrix with dimension fewer than initial keeping whole useful image in there. Modified stochastic hill climbing. Decision function – sum of nonzero points in frame. By visual inspection frame size is assigned to be 30x30. Algorithm starts by randomly initialized top left coordinates xi, yi. Frame moves in direction where decision function has max value. Step size is 1. When max is reached, new coordinates are stored in sorted list. Algorithm repeats until number of top scores reaches N=10. Coordinates that correspond to max of decision function is considered to be top left of new image with size 30x30. All 10000 images from training and test set are processed. Both training and test sets.

1. removeOutliers.py

Use outlier detection algorithms to remove more noise. Contamination fraction is 0.3 by inspection. IsolationForest, LocalOutlierFactor, EllipticEnvelope and OneClassSVM from scikit-learn are used for outlier’s detection. Decision is based on results from all 4 algorithms. Training and test arrays are processed and stores in new files. Both training and test sets.

1. centerImages.py

Center images. Both training and test sets.

1. imageTransform2.py

Create additional images from existing set using random transformations: shift, zoom and rotate. Best parameters are: 40 clones for each image

shiftBounds = [0, 6]

rotationBounds = [-40, 40]

zoomBounds = [0.7, 1.2]

Only training set.

1. svc.py

Fit original cropped centered set without noise with classifiers: Linear SVC, SVC with RBF kenel, Random forest, XGBoost, Logistic rergression, Naive Bayes.

Results are stored in summary.xlsx

1. cnn11.py

Use convolution neural network to fit enriched dataset. Save model and plot progress

1. cnnPredict.py
2. Load model saved by cnn11.py and make prediction on test set. Save response variable as .csv file.