Newton ML Tests

Testing Methodology

- https://docs.google.com/spreadsheets/d/1vkRBkQoHqImQviCFe3ebjCd3kJSi7wd49k0zTZgO3wA/edit#gid=0
- For detection, evaluation metrics would be number of True Positives(TP), FP & FN. A predicted bounding box will be considered TP if its similarity score with a "true" bounding box is above a threshold. The similarity score can be calculated using
    - Normalised intersection area of predicted bounding box with the true bounding box
    - Euclidean distance between their centroids. Suggest better ways for calculating similarity scores if any.
- For tracking, an evaluation metric "tracking loss" is proposed. Can use some feedback for improvement. Similarity score will be calculated in the same manner as described above.

Testing Results

- Detection and tracking (visual) testing results: https://docs.google.com/spreadsheets/d/15IaavXYARHRYFYcUvveVwSovZ5UODAurMXbrFF0kE7w/edit#gid=0
- Age/Gender/Face recognition results: https://docs.google.com/spreadsheets/d/1LIL-SZAFH6CAfLHgbvlFaQfnNuQziOOwBGAHZ2OWHwM/edit#gid=0

Analysis of Results

Motion Analytics

1. General observations for each motion detection algorithm:
    - **MOG[Rank-3]**: *Pros* - Best localisation for positive regions (silhouettes) among other algorithms. It has high precision with positive regions having sharp. boundaries. *Cons* - Sensitive to light variations, shadows and slow moving objects. It is slow in discarding moving entities when they come to a halt, leading to a trailing false positive region (ghosts). Also, it superimposes background patterns over true positives and does not fill up closed positive regions.
    - **MOG2[Rank-5]**: *Pros* - only pro that MOG2 seems to have over MOG is that it does not superimpose background patterns over true positives and fills up closed positive regions. *Cons* - overall worse performance that MOG (more sensitive to light variations, shadows and slower update rate for fast moving objects - leading to long trailing false positive regions)
    - **GMG [Rank-2]**: *Pros* - good update time for fast moving objects, quickly detects stationary objects that start to move and quickly discards moving objects when they come to a stop. Thus very short trailing false positive regions or ghosts (much better than MOG and MOG2). Moreover it has very

low sensitivity to light variations, shadows and slow moving objects. *Cons* - very noisy, it detects many small regions as false positives, yields poor localisation for true positives and takes longest time (among other algorithms) to initialise a background model.

- **VIBEBG[Rank-3 to 5 (varies)]:** *Pros* - faster update time than MOG and MOG2, but not faster than GMG. Thus yields shorter trailing ghosts than MOG and MOG2. Better localisation of true positive regions (silhouettes) than GMG and MOG2, but not as good as MOG. Not noisy, does not give many small false positive regions unlike GMG. *Cons* - Highly dependent on initial frames for background modelling. Thus entire performance can go haywire due to some initial bad frames (in such cases, it detects even background regions as positives and is very slow to discard them). Also, its sensitive to light variations and shadows.
- **MLBGS [Rank-1]:** *Pros* - fastest update time - quickly discards moving objects that come to a halt, quickly updates stationary objects that start to move, no trailing ghosts, invariant to light, shadows and slow moving objects. *Cons* - not so accurate localisations for true positives, especially if moving entities are close together (only MOG does neat localisation in such situations), computationally very costly.
- **PBAS [Rank-2]**: *Pros* - better localisation for positive regions than MLBGS. Considerably lighter than MLBGS in computation load. *Cons* - Slower in discarding moving objects when they come to a halt. Slow in discarding false positives after a bad frame. Not invariant to shadows. (I believe these shortcomings can be overcome by tuning the hyperparameters of the algorithm)

2. Specific observations for each motion detection algorithm on tested videos: Refer spreadsheet https://docs.google.com/spreadsheets/d/1OMQfEcSAjyeIr6CsJl1m9T7siJ8A9TmeGU-yiFaREls/edit#gid=0

Face Detection and Recognition

❏ For Detection: Performance of face detection (bounding box accuracy) using NPD is best among other face detector models (DLIB, HAAR)

❏ For Recognition:

- Dataset used: colorFeret Dataset details:
    - ➔ Feret dvd1
      -Number of classes: 275
      -Images per class: 11-30 (frontal as well as non-frontal faces)
      -Image dim: 512x768
      -Train-Test split: 3:2 ratio for each class
    - ➔ Feret dvd2
      -Number of classes: 739
      -Images per class: 5-10 (frontal as well as non-frontal faces)
      -Image dim: 512x768
      -Train-Test split: 3:2 ratio for each class
- Methodology: Testing was performed in multiple settings:

1. using aisaac custom face recognition. NPD was used as face detector, faceVGG caffemodel used as feature extractor, LRL2 classification algo used for training DATK weights.
2. using OpenFace. ‘Shape_predictor_68_face_landmarks.dat’ was used as the landmarks model with ‘outerEyesAndNose’ as keypoints. ‘nn4.small2.v1.t7’ was used as the torch DNN model. Various classification algorithms were tried out - SVM, RadialSVM, GMM, LinearSVM, DecisionTree, LogisticRegression and NearestNeighbour-Mean.

- Best result was obtained by aisaac custom face recognition using faceVGG caffemodel with fc6 as feature extraction layer (test accuracy =**0.920281**(1443/1568)).
- Other hyperparameter settings were:
  -maximumFaceSize: [0.9 0.9]
  -minimumFaceSize: [0.3 0.3]
  -DATKPredictionProbThresh: 0.1
  -LRL2NumOfIterations: 1000
  -LRL2OptimizationAlgo: 0
  -LRL2RegularizationParam: 0
  -LRL2Tolerance: 0.01
  -mode: 0 (transfer learning)

Age Identification

- Dataset used: Temple Dataset (collected by us). Dataset details:
  -Number of classes: 4 { 0-15, 15-30, 30-50, 50-100 }
  -Image split:
  Train-
  [0-15]: 49
  [15-30]: 588
  [30-50]: 1227
  [50-100]: 628

  Test-
  [0-15]: 24
  [15-30]: 294
  [30-50]: 613
  [50-100]: 314
- Methodology: NPD was used as the face detection algorithm, while 2 different caffe models were compared for feature extraction: age_net, age_VGG. Different feature extraction layers were tried out (fc7 and fc6). LRL2 classification algo was used for training DATK weights.
- Best result was obtained on using **age_VGG** caffemodel with **fc6** as feature extraction layer (test accuracy =**0.792771**(987/1245)).
- Other hyperparameter settings were:
  -maximumFaceSize: [0.9 0.9]
  -minimumFaceSize: [0.3 0.3]
  -ageRecognitionThreshold: 0.7
  -DATKPredictionProbThresh: 0.1
  -LRL2NumOfIterations: 1000
  -LRL2OptimizationAlgo: 0
  -LRL2RegularizationParam: 0

-LRL2Tolerance: 0.01
-mode: 0 (transfer learning)

Gender Identification

- Dataset used: Temple Dataset (collected by us). Dataset details (image split):
  Train- Female: 1087; Male: 1113
  Test- Female: 249; Male: 299
- Methodology: NPD was used as the face detection algorithm, while 3 different caffe models were compared for feature extraction: gender_VGG, gender_GoogleNet, gender_deploy. Different feature extraction layers were tried out (fc7 and fc6). LRL2 classification algo was used for training DATK weights.
- Best result was obtained on using **gender_VGG** caffemodel with **fc6** as feature extraction layer (test accuracy = **0.427007**(234/548)).
- Other hyperparameter settings were:
  -maximumFaceSize: [0.9 0.9]
  -minimumFaceSize: [0.3 0.3]
  -genderRecognitionThreshold: 0.99
  -DATKPredictionProbThresh: 0.1
  -LRL2NumOfIterations: 1000
  -LRL2OptimizationAlgo: 0
  -LRL2RegularizationParam: 0
  -LRL2Tolerance: 0.01
  -mode: 0 (transfer learning)

Emotion Recognition

- Dataset used: KDEF Details of dataset:
  -Image dim: 562x762
  -Number of classes: 7 (afraid, angry, disgusted, happy, neutral, sad, surprised)
  -Images per class: 350
  -Train-Test split: 3:2 ratio for each class
  -Dataset consists frontal as well as non-frontal faces
- Methodology: Testing was performed in multiple settings:
  3. using aisaac emotion recognition out of the box, ie using just the standalone caffe model EmotiW_VGG_S end to end.
  4. custom analytics around out of box emotion recognition, ie using EmotiW_VGG_S as feature extractor + DATK LRL2 algorithm as classifier.
  5. using OpenFace. 'Shape_predictor_68_face_landmarks.dat' was used as the landmarks model with 'outerEyesAndNose' as keypoints. 'nn4.small2.v1.t7' was used as the torch DNN model. Various classification algorithms were tried out - SVM, RadialSVM, GMM, LinearSVM, DecisionTree, LogisticRegression and NearestNeighbour-Mean.
- Best Result was obtained using OpenFace with RadialSVM as classification algorithm (test accuracy: **0.664135021097**(787/1185))

Human Detection and Tracking

- Dataset used (Video used): EnterExitCrossingPaths1cor.mpg which is part of CAVIARdataset. Video details:

-Frame dimensions: 384 x 288
-Codec: MPEG-1 Video
-Framerate: 25 fps
-Length: 13sec

- Methodology: Testing was performed at 5FPS processing frame rate. Performance of various human detection algorithms was compared using a detection and tracking script, with different combinations of their hyperparameters. Following parameters were varied:
    - ➔ Detection algo : DPM
      { dpmInterval, dpmMinNeighbours, dpmFlags, dpmThreshold }
    - ➔ Detection algo : ICF
      { icfInterval, icfMinNeighbours, icfFlags, IcfThreshold, IcfStepThrough }
    - ➔ Detection algo : DARKNET
      { darknetCfg, darknetLabels, darknetWeights, darknetThreshold }
    - ➔ Detection algo : HOG
      { hogHitThreshold, hogStride, hogPadding, hogScaleFactor, hogGroupThreshold }
    - ➔ Detection algo : HAAR_FULLBODY
      { haarScaleFactor, haarGroupThreshold }
    - ➔ detect_iouAreaThreshold
    - ➔ track_iouAreaThreshold
- Best result (**F1Score = 0.858847**) was obtained for the following combination:
  -Detection Algo: DARKNET
  -darknetCfg: yolo-voc.cfg
  -darknetLabels: vocnames.txt
  -darknetWeights: yolo-voc.weights
  -darknetThreshold: 0.100000
  -detect_iouAreaThreshold: 0.200000
  -track_iouAreaThreshold: 0.400000

Head Detection

- Datasets used: MallDataset and HollywoodHeads
- Methodology: Visual testing and evaluation was performed in 2 different settings:
    1. Aisaac head detection out of the box using haarcascade_head.xml. Other settings included were:
       -headBlobDeleteTime: 60
       -maximumHeadSize: [0.9 0.9]
       -minimumHeadBlobFrames: 20
       -minimumHeadSize: [0.3 0.3]

  2. Custom trained DARKNET model for head detection. The model was trained on HollywoodHeads dataset
- Testing was performed on MallDataset and AirportCCTV_Videos. The custom trained DARKNET model outperformed the haarcascade model (visually evaluated).