AGE ESTIMATION FROM FRONTAL IMAGES OF FACES

ROHIT NAIR (RONAIR), SRIVATSAN IYER (SRRIYER) & SHASHANT DEVADIGA (SSDEVADI)

ABSTRACT

It is pretty easy for a human to look at a face and estimate the age of the person approximately. For a computer to automatically do that is quite a challenge. This would require using techniques from different areas such as feature detection, machine learning, and anthropometrics. This project report summarizes the result of using anthropometric models with varying parameters.

INTRODUCTION

We set out to explore the relation between anthropometrics measurements and the age. Specifically, we wanted to use such measurements and build a model that helps us predict ages.

In our project, we have used the dlib library [4] that helps us detect faces in images and locate various points on the face. These points correspond to boundaries and centre points of eyes, nose, eyebrows, lips and jaw. The points on the face help us calculate various feature vectors that we use for training and classification.

PROCESS

DATA EXTRACTION AND PRE-PROCESSING

Almost all datasets we have used so far come with an annotation for each of the image. Each image had metadata that had information such as age of the subject and gender. We use a common data structure to represent images from all datasets, and dataset specific extractor routine that populates the data structure. All the samples without these annotations are discarded.

FEATURE DESCRIPTION

We have used two models to predict ages. One model is a standard anthropometric model involving ratios between various parts of the face. [1] Second model is the one we proposed -- the jaw boundary. We present the algorithm for each of the two models below.

FACE LANDMARK (FACIAL RATIOS)

We use the following 7 facial ratios as feature vectors [1]:

- **Facial Index**: Ratio of the vertical length of the face without the forehead to the width of the face (distance between cheek bones)
- **Mandibular Index:** Ratio of distance between mouth and lowest point of the chin to the width of lower jaw.
- **Intercanthal Index:** Ratio of distance between rightmost point of left eye and left most point of right eye to the distance between leftmost point of left eye and rightmost point of right eye.
- **Orbital Width Index:** Ratio of width of one eye to the distance between the rightmost point of left eye and leftmost point of right eye.
- **Nasal Index:** Ratio of width of nose to the height of the nose.
- **Vermilion Height Index:** Ratio of height of upper lip to the height of lower lip.
- **Mouth-Face Width Index:** Ratio of width of lips to the distance between cheek bones.

We had also used, Eye Fissure Index, but we dropped it because we were unable to extract the exact points within the eye accurately.

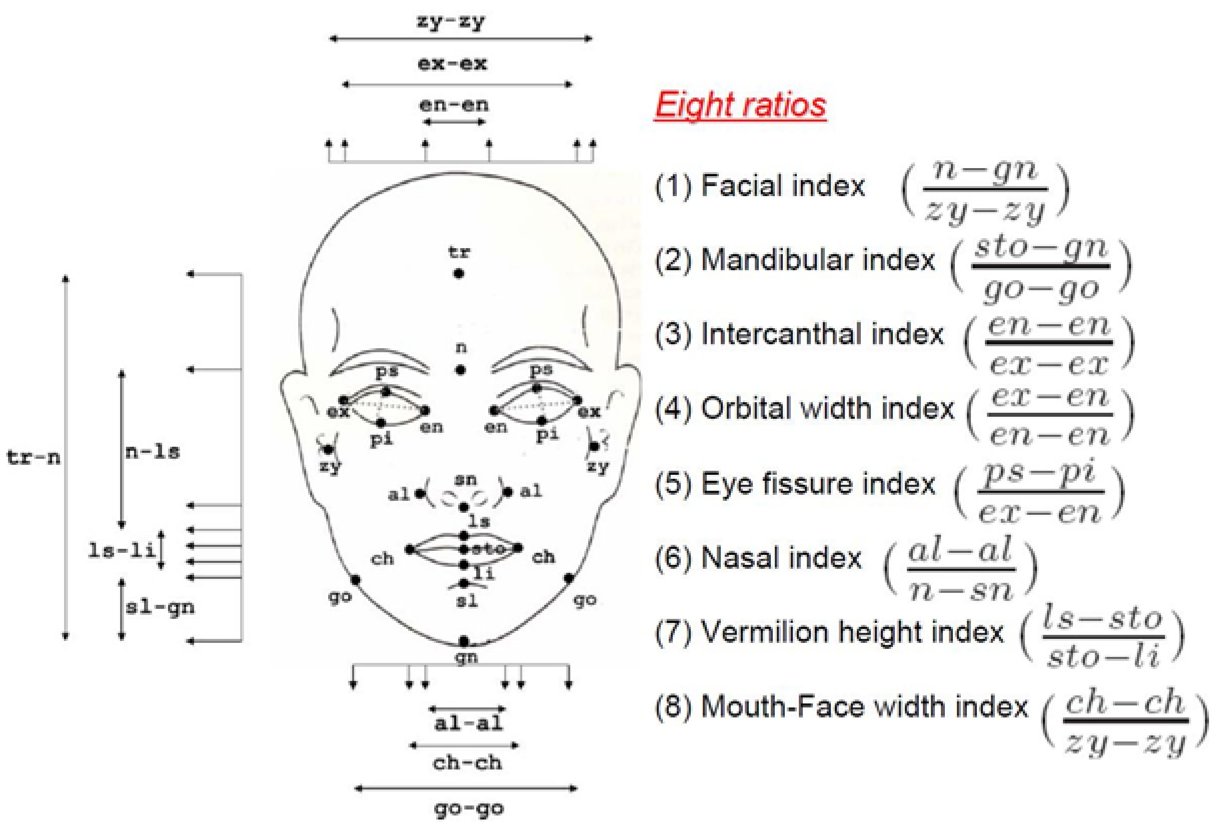


FIGURE 1: ANTHROPOMETRIC RATIOS [1]

FACE BOUNDARY

We proposed another metric and tried to see if we could use to aid the classification. We hypothesized that the boundary of the face also changes with age. We approximated the boundary along the face to be a set of 14 lines. We calculate the angle, in radians, that one line makes with the next. These angles give us a translation invariant measurement that can be extracted from images without worrying about where in the image the face is located.

FACE LANDMARK & BOUNDARY

Instead of relying on only one of the two models, we combined them to get 14 + 7 = 21 features.

FEATURE EXTRACTION

We use the dlib library [4] to recognize the faces and extract facial features in the form of points that represent the boundaries of eyes, nose, eyebrows, lips, jaws. If a face is detected, the library returns a 68-length vector elements of which correspond to precise locations on the face. We use these 2-dimensional points to calculate anthropometric ratios of the facial features and face boundary as detailed above. These ratios form the feature vector for the purpose of classification.

FIGURE 2: DETECTION POINTS RETURNED BY DLIB

FACE LANDMARK (FACIAL RATIOS)

We calculate one additional point that is not present in the set of points returned by the library. This point is the central point between the eyebrows. Mathematically, this point can be expressed as the centroid of 4 points -- two extreme points of the 2 eyebrows. This point represents the lowest point of the central vertical line running down through forehead. These points (along with the point that we just calculated) are in 2D Euclidean space, we can use the below distance formula to calculate the distance any two of them and hence obtain the ratio defined in the “Feature Description” section above.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

FACE BOUNDARY (FACIAL RATIOS)

In this approach, we start with the 15 points that dlib extracts as the boundaries of the face. We use the first three points and make two lines out of them. The angle between the two lines thus formed can be represented as:

$$Angle = \tan^{-1}\frac{y_3 - y_2}{x_3 - x_2} - \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1}$$

Each of the 14 angles thus calculated forms a feature as shown in Figure 3.

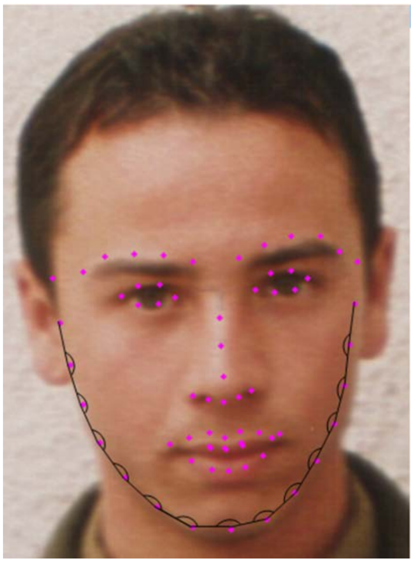


FIGURE 3: ANGLES IN JAW BOUNDARY

DATASET

We tried to download from various face recognition repositories where each of the faces are annotated with an age. Some of the dataset had homographic transformations applied to photos. Since we rely on accurate calculation of ratios calculation of ratios of facial features, we could not use such repositories.

In this experiment, we are using the Datatang Aging database to train and test our set of classifiers.The database consists of approximately front facing images of around 82 individuals. There are approximately 12 images at different ages for each individual. Almost all of the images have people looking into the camera except for a few.

SAMPLE IMAGES OF ONE PERSON FROM THE DATASET

EXAMPLE #1

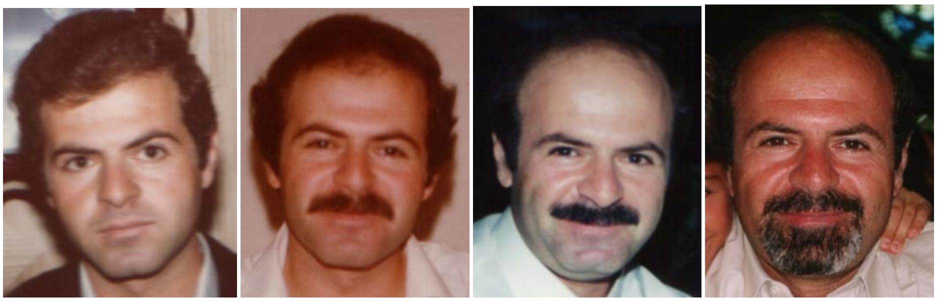


FIGURE 4: THE 4 IMAGES ABOVE ARE OF A PERSON AT THE AGE OF 22, 24, 40, 44

EXAMPLE #2

FIGURE 5: THE IMAGES ABOVE OF A DIFFERENT PERSON AT THE AGES OF 8, 11, 12, 15

COMMENTS ON DATASET

We have been using this dataset to train our classifier and test it. We have the below comments to make about it:

- This is probably the best dataset we have been able to find. There are tons of face recognition datasets, but only some have annotations of ages. Far fewer datasets have very clean front-facing images.
- This dataset has many samples in lower age group than in the higher age group. As would be noticed in the section about experimentation, this severely restricts us from being able to classify the ages in the higher age group.

TRAINING AND TESTING

We were not able to predict the age as a continuous variable. Instead we grouped them into different age groups and tried to predict the age group. For this classification purpose, we have tried 3 classifiers: Multinomial Naive Bayes Classifier, K-Nearest-Neighbors, and Fully Connection Neural Nets. For each of the classifier, we have put up the results in the section named “RESULTS”. To see how to run the program, please consult the README in the project repository.

SAMPLE OUTPUT #1

```
Classifier: <class 'core.classifier.ScikitNeuralNetClassifier'>
Classifier param: [[0, 10, 0]]
Feature: <class
'core.featureconverter.FaceLandmarkFeatureConverter'>
Age Group: (12, 23, 100)
Trial # 1
Confusion Matrix:
  97   17    0
  19   17    0
  10    7    0
Overall Accuracy: 0.682634730539 (114 out of 167)
Trial # 2
Confusion Matrix:
  88   25    1
  17   17    2
  10    7    0
Overall Accuracy: 0.62874251497 (105 out of 167)
Trial # 3
Confusion Matrix:
  99   14    1
  21   13    2
  11    5    1
Overall Accuracy: 0.676646706587 (113 out of 167)
Trial # 4
Confusion Matrix:
  71   36    7
  13   18    5
   7    9    1
Overall Accuracy: 0.538922155689 (90 out of 167)
Trial # 5
Confusion Matrix:
  50   50   14
   9   16   11
   5    4    8
Overall Accuracy: 0.443113772455 (74 out of 167)
Avg. Correct: 99.2
Avg. Accuracy: 0.594011976048
BTRG: 1.78203592814
```

SAMPLE OUTPUT #2

```
Classifier: <class 'core.classifier.ScikitNaiveBayesClassifier'>
Classifier param: []
Feature: <class
'core.featureconverter.FaceBoundaryFeatureConverter'>
Age Group: (7, 15, 22, 100)
Trial # 1
Confusion Matrix:
  60    0    0   12
```

```
  51    0    0   11
   8    0    0    6
  11    0    0    8
Overall Accuracy: 0.407185628743 (68 out of 167)
Trial # 2
Confusion Matrix:
  60    0    0   12
  51    0    0   11
   8    0    0    6
  11    0    0    8
Overall Accuracy: 0.407185628743 (68 out of 167)
Trial # 3
Confusion Matrix:
  60    0    0   12
  51    0    0   11
   8    0    0    6
  11    0    0    8
Overall Accuracy: 0.407185628743 (68 out of 167)
Trial # 4
Confusion Matrix:
  60    0    0   12
  51    0    0   11
   8    0    0    6
  11    0    0    8
Overall Accuracy: 0.407185628743 (68 out of 167)
Trial # 5
Confusion Matrix:
  60    0    0   12
  51    0    0   11
   8    0    0    6
  11    0    0    8
Overall Accuracy: 0.407185628743 (68 out of 167)
Avg. Correct: 68.0
Avg. Accuracy: 0.407185628743
BTRG: 1.62874251497
```

SAMPLE OUTPUT #3

```
Classifier: <class 'core.classifier.NearestNeighborsClassifier'>
Classifier param: [5]
Feature: <class
'core.featureconverter.FaceLandmarkBoundaryFeatureConverter'>
Age Group: (12, 23, 100)
Trial # 1
Confusion Matrix:
  85   28    1
  13   19    4
   7    8    2
Overall Accuracy: 0.634730538922 (106 out of 167)
Trial # 2
Confusion Matrix:
  85   28    1
```

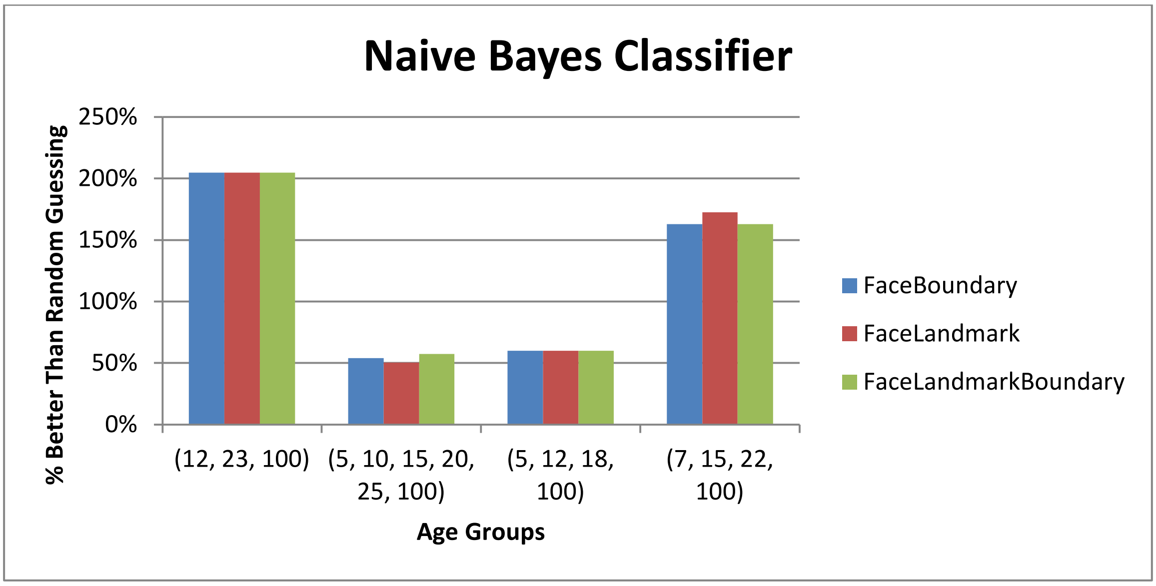
```
  13   19    4
   7    8    2
Overall Accuracy: 0.634730538922 (106 out of 167)
Trial # 3
Confusion Matrix:
  85   28    1
  13   19    4
   7    8    2
Overall Accuracy: 0.634730538922 (106 out of 167)
Trial # 4
Confusion Matrix:
  85   28    1
  13   19    4
   7    8    2
Overall Accuracy: 0.634730538922 (106 out of 167)
Trial # 5
Confusion Matrix:
  85   28    1
  13   19    4
   7    8    2
Overall Accuracy: 0.634730538922 (106 out of 167)
Avg. Correct: 106.0
Avg. Accuracy: 0.634730538922
BTRG: 1.90419161677
```

RESULTS

NAÏVE BAYES

Feature	Age Group	Average Accuracy	BTRG
FaceLandmark	(12, 23, 100)	0.6826	2.0479
FaceLandmark	(7, 15, 22, 100)	0.4311	1.7246
FaceLandmark	(5, 12, 18, 100)	0.1497	0.5988
FaceLandmark	(5, 10, 15, 20, 25, 100)	0.0838	0.5030
FaceBoundary	(12, 23, 100)	0.6826	2.0479
FaceBoundary	(7, 15, 22, 100)	0.4072	1.6287
FaceBoundary	(5, 12, 18, 100)	0.1497	0.5988
FaceBoundary	(5, 10, 15, 20, 25, 100)	0.0898	0.5389
FaceLandmarkBoundary	(12, 23, 100)	0.6826	2.0479
FaceLandmarkBoundary	(7, 15, 22, 100)	0.4072	1.6287
FaceLandmarkBoundary	(5, 12, 18, 100)	0.1497	0.5988
FaceLandmarkBoundary	(5, 10, 15, 20, 25, 100)	0.0958	0.5749

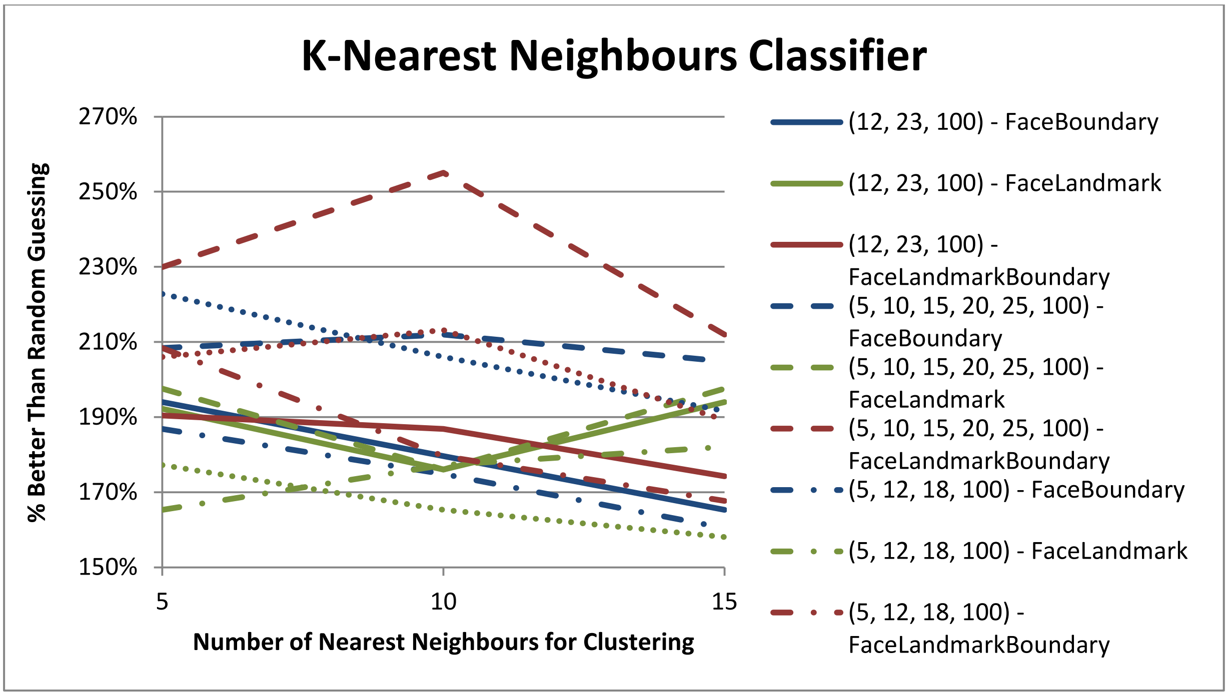
BTRG: Times better than random guessing



K-NEAREST NEIGHBOURS

Feature	Age Group	Number of Nearest Neighbours								
		5			10			15		
		Average Correct	Average Accuracy	BTRG	Avg. Correct	Average Accuracy	BTRG	Avg Correct	Average Accuracy	BTRG
FL	(12, 23, 100)	107	0.6407	1.9222	98	0.5868	1.7605	108	0.6467	1.9401
FL	(7,15,22,100)	74	0.4431	1.7725	69	0.4132	1.6527	66	0.3952	1.5808
FL	(5,12,18,100)	69	0.4132	1.6527	74	0.4431	1.7725	76	0.4551	1.8204
FL	(5,10,15,20,25, 100)	55	0.3293	1.9760	49	0.2934	1.7605	55	0.3293	1.9760
FB	(12, 23, 100)	108	0.6467	1.9401	100	0.5988	1.7964	92	0.5509	1.6527
FB	(7,15,22,100)	93	0.5569	2.2275	86	0.5150	2.0599	80	0.4790	1.9162
FB	(5,12,18,100)	78	0.4671	1.8683	73	0.4371	1.7485	67	0.4012	1.6048
FB	(5,10,15,20,25, 100)	58	0.3473	2.0838	59	0.3533	2.1198	57	0.3413	2.0479
FLB	(12, 23, 100)	106	0.6347	1.9042	104	0.6228	1.8683	97	0.5808	1.7425
FLB	(7,15,22,100)	86	0.5150	2.0599	89	0.5329	2.1317	79	0.4731	1.8922
FLB	(5,12,18,100)	87	0.5210	2.0838	75	0.4491	1.7964	70	0.4192	1.6766
FLB	(5,10,15,20,25, 100)	64	0.3832	2.2994	71	0.4252	2.5509	59	0.3533	2.1198

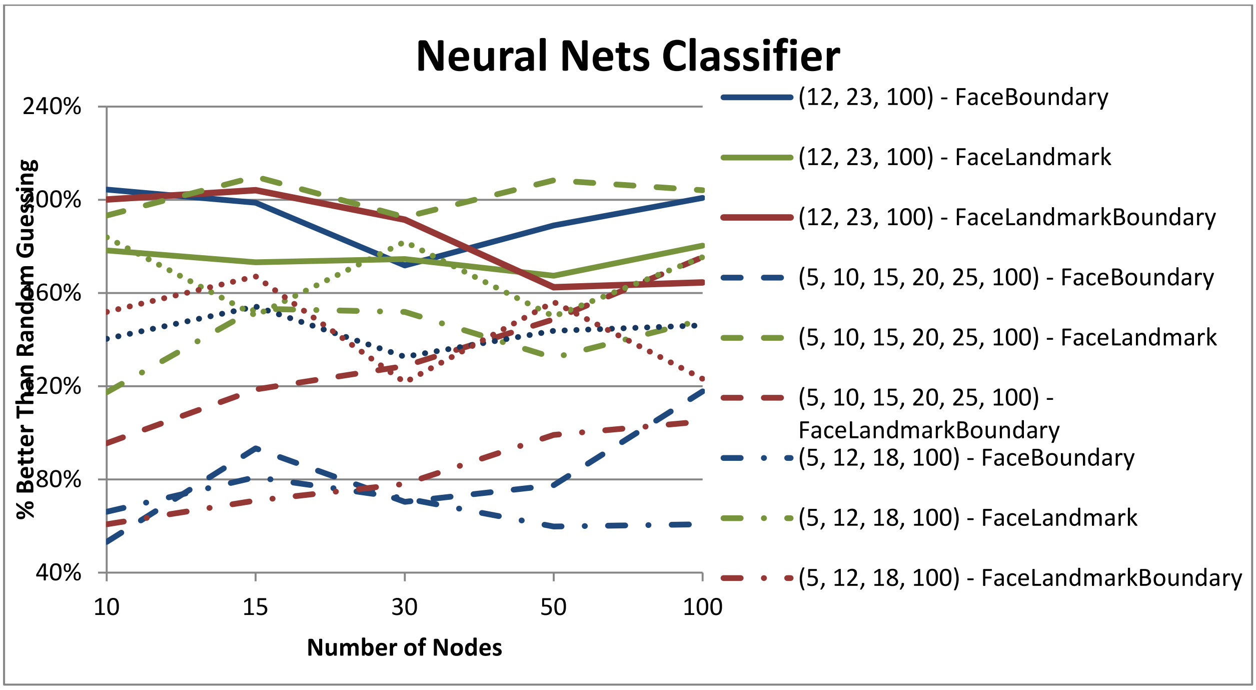
FaceLandmark - FL
FaceBoundary - FB
FaceLandmarkBoundary- FLB
BTRG: Times better than random guessing



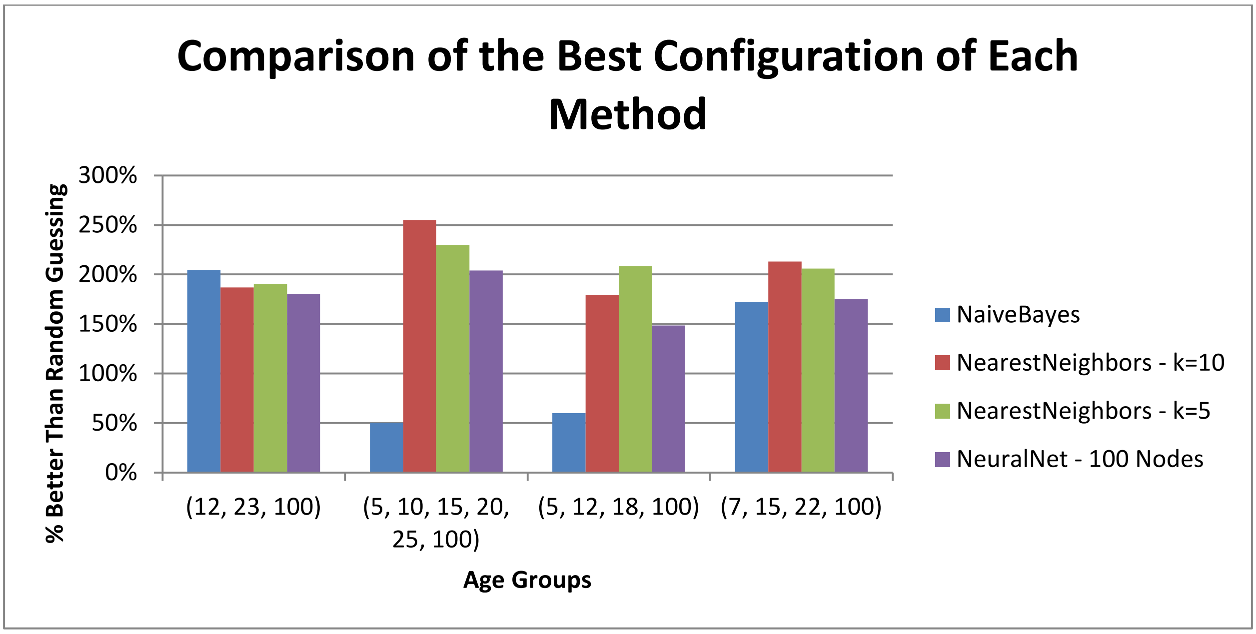
NEURAL NET

No. of Nodes	Age Group	Feature								
		FaceLandmark			FaceBoundary			FaceLandmarkBoundary		
		Average Correct	Avg. Accuracy	BTRG	Avg Correct	Avg Accuracy	BTRG	Avg. Correct	Average Accuracy	BTRG
10	(12, 23, 100)	99.2	0.5940	1.7820	113.8	0.6814	2.0443	111.4	0.6671	2.0012
10	(7, 15, 22, 100)	76.8	0.4599	1.8395	58.6	0.3509	1.4036	63.4	0.3796	1.5186
10	(5, 12, 18, 100)	49	0.2934	1.1737	27.6	0.1653	0.6611	25.4	0.1521	0.6084
10	(5, 10, 15, 20, 25, 100)	53.8	0.3222	1.9329	14.8	0.0886	0.5317	26.6	0.1593	0.9557
15	(12, 23, 100)	96.4	0.5772	1.7317	110.6	0.6623	1.9868	113.6	0.6802	2.0407
15	(7, 15, 22, 100)	62.8	0.3760	1.5042	64.4	0.3856	1.5425	69.8	0.4180	1.6719
15	(5, 12, 18, 100)	64	0.3832	1.5329	33.8	0.2024	0.8096	29.6	0.1772	0.7090
15	(5, 10, 15, 20, 25, 100)	58.4	0.3497	2.0982	26	0.1557	0.9341	33	0.1976	1.1856
30	(12, 23, 100)	97.2	0.5820	1.7461	95.6	0.5725	1.7174	106.6	0.6383	1.9150
30	(7, 15, 22, 100)	76	0.4551	1.8204	55.4	0.3317	1.3269	50.8	0.3042	1.2168
30	(5, 12, 18, 100)	63.4	0.3796	1.5186	30	0.1796	0.7186	32.6	0.1952	0.7808
30	(5, 10, 15, 20, 25, 100)	53.6	0.3210	1.9257	19.6	0.1174	0.7042	35.8	0.2144	1.2862
50	(12, 23, 100)	93.2	0.5581	1.6743	105.2	0.6299	1.8898	90.4	0.5413	1.6240
50	(7, 15, 22, 100)	62.6	0.3749	1.4994	60	0.3593	1.4371	65.2	0.3904	1.5617
50	(5, 12, 18, 100)	55.2	0.3305	1.3222	25	0.1497	0.5988	41.4	0.2479	0.9916
50	(5, 10, 15, 20, 25, 100)	58	0.3473	2.0838	21.6	0.1293	0.7760	41.4	0.2479	1.4874
100	(12, 23, 100)	100.4	0.6012	1.8036	111.8	0.6695	2.0084	91.6	0.5485	1.6455
100	(7, 15, 22, 100)	73.2	0.4383	1.7533	61	0.3653	1.4611	51.4	0.3078	1.2311
100	(5, 12, 18, 100)	62	0.3713	1.4850	25.4	0.1521	0.6084	43.8	0.2623	1.0491
100	(5, 10, 15, 20, 25, 100)	56.8	0.3401	2.0407	32.8	0.1964	1.1784	48.8	0.2922	1.7533

BTRG: Times better than random guessing



COMPARISON OF BEST METHODS



OBSERVATION

From the experiments that we performed, we observe:

- Although the classifiers give a relatively good accuracy for more granular age groups, we found that the output is often unstable for higher ages because the face structure stops changing significantly after a certain age. In other words, the classifiers are able to distinguish better between kids and adults. However, it does not perform well when tasked with predicting more accurate ages. This is most probably because the dataset we are using does not have enough samples in higher age groups.
- The curvature of the jaw can also be used to predict the age group of the person. However, it gives a lower accuracy as compared to using facial ratios. We have also implemented the three classifiers using a combination of facial ratios and jaw boundary angles and this gave better results for KNN.
- The classification is highly dependent on the pose of the face. The accuracy of proper prediction is higher when the face is not looking away from the camera. In Fig. 3, the first image does not work well while the other two images get predicted accurately.
- Among the 3 classifiers, KNN gives us the best results for the different age groups and feature set as compared to Neural Nets and Bayes' classifier.

CONCLUSION

- The models that we have built can efficiently distinguish between kids and adults, for relatively smaller number of age groups.
- The detection of facial features and classification of age is highly sensitive to the pose of the person and somewhat sensitive to the facial expression.
- The accuracy of classification reduces as we increase the number of age buckets.

REFERENCES

1. Xin Geng, Yun Fu, Kate Smith-Miles, "Automatic Facial AgeEstimation" www1.ece.neu.edu/~yunfu/papers/pricai10_t4.pdf
2. X. Geng, Z. H. Zhou and K. Smith-Miles, "Automatic Age Estimation Based on Facial Aging Patterns," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2234-2240, Dec. 2007.
3. API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.
4. Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009
5. Datatang database: www.datatang.com/data/43941

ACKNOWLEDGEMENT

We wish to thank Prof. David Crandall for his support and advice throughout the duration of the project. We also wish to thank Datatang aging database that we used to train and test our classifiers. Without the data, this experiment would not have been possible.