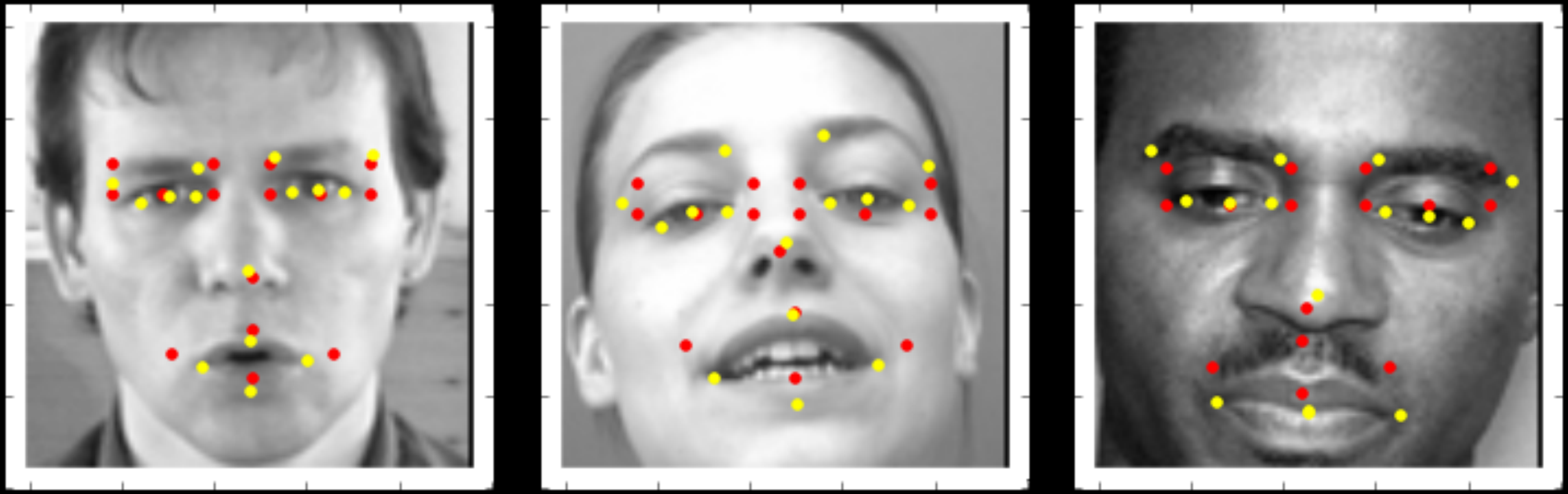


Facial Keypoint Detection

Kevin J. Stephano

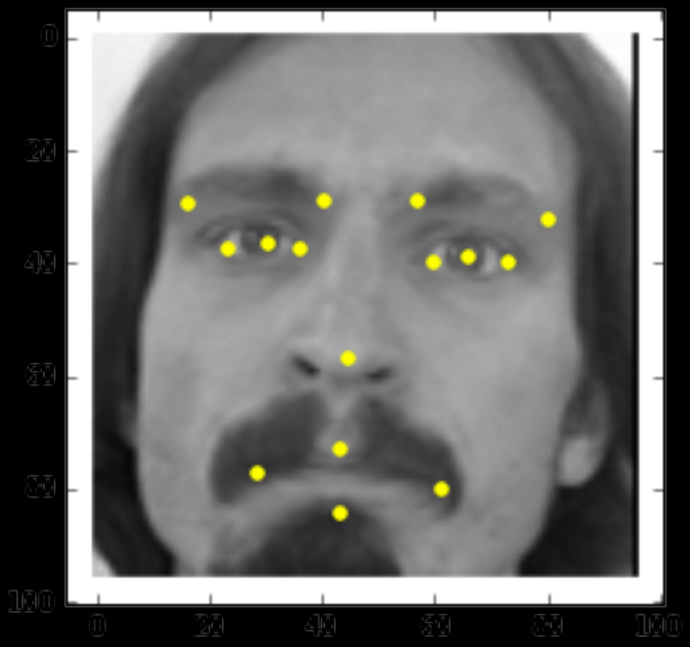
Background

- Kaggle Competition
- Coding was supposed to be in R
  - Slow
  - Lacking standard packages like Scikit-learn and opencv

The Data

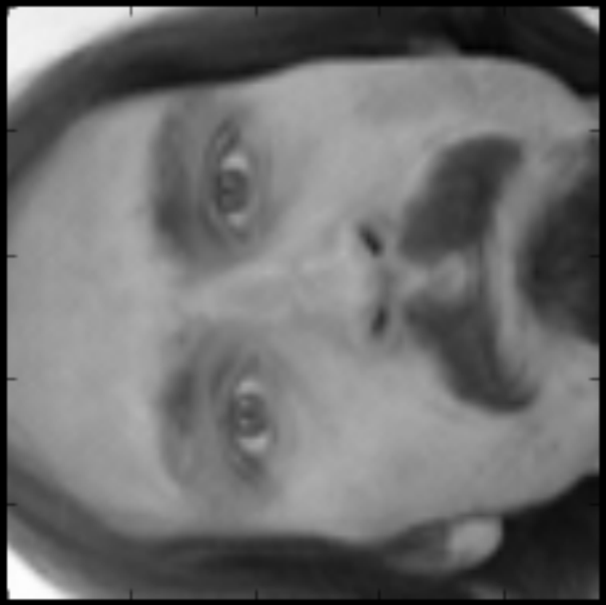
- 7049 96x96 Grayscale Images
- 15 Known Points Per image
- Images and Data were packaged in a CSV File
- Images are encapsulated in a string per line of a CSV File in a 1-D sequence

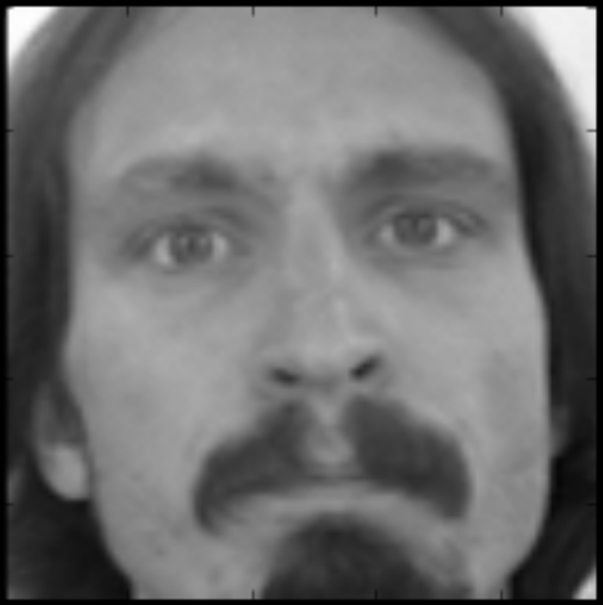
```
66.0335639098,39.0022736842,
30.2270075188,36.4216781955,
59.582075188, 39.6474225564,
73.1303458647,39.9699969925,
36.3565714286,37.3894015038,
23.4528721805,37.3894015038,
56.9532631579,29.0336481203,
80.2271278195,32.2281383459,
40.2276090226,29.0023218045,
16.3563789474,29.6474706767,
44.4205714286,57.0668030075,
61.1953082707,79.9701654135,
28.6144962406,77.3889924812,
43.3126015038,72.9354586466,
43.1307067669,84.4857744361,
“238 236 237 238 240 240   “
```

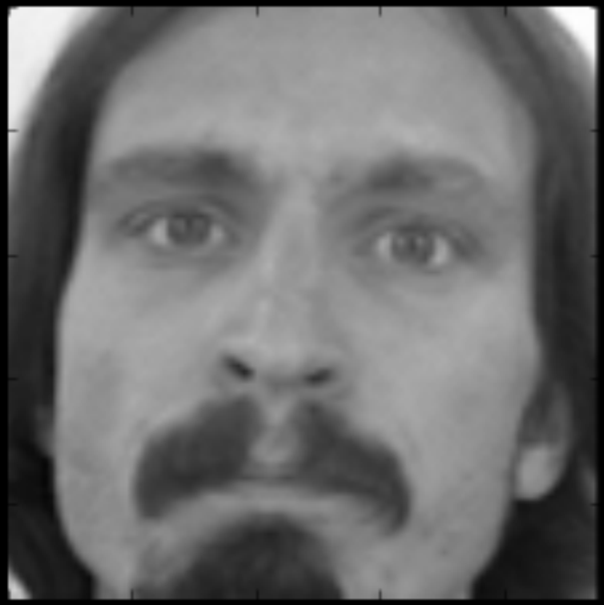


Scrubbing The Data

- Rotated Images 270 degress for classifiers
- Known points required flipping the image
- NaNs in known points since some points are not detectable


Original

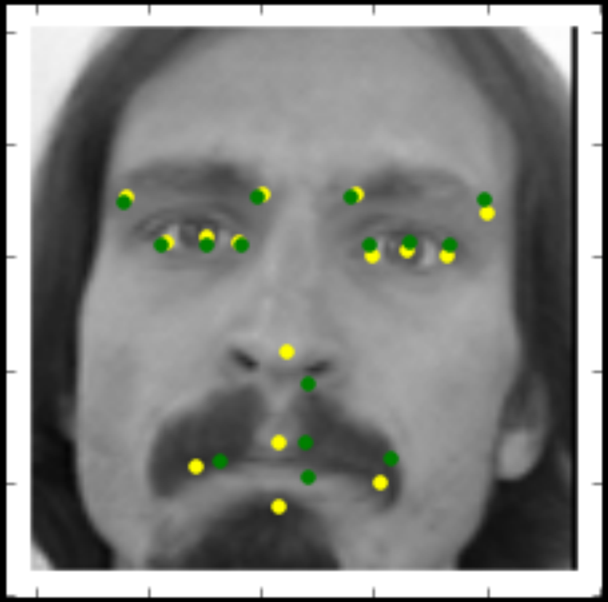

Rotate 270


Flip on Y-axis

Experiment:
Mean of Known Facial Points

- Doesn't use actual image data
- Root Mean Square Error for mean: 3.17
- Didn't use actual Kaggle competition evaluation
- Best Kaggle result is about ~2.0

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

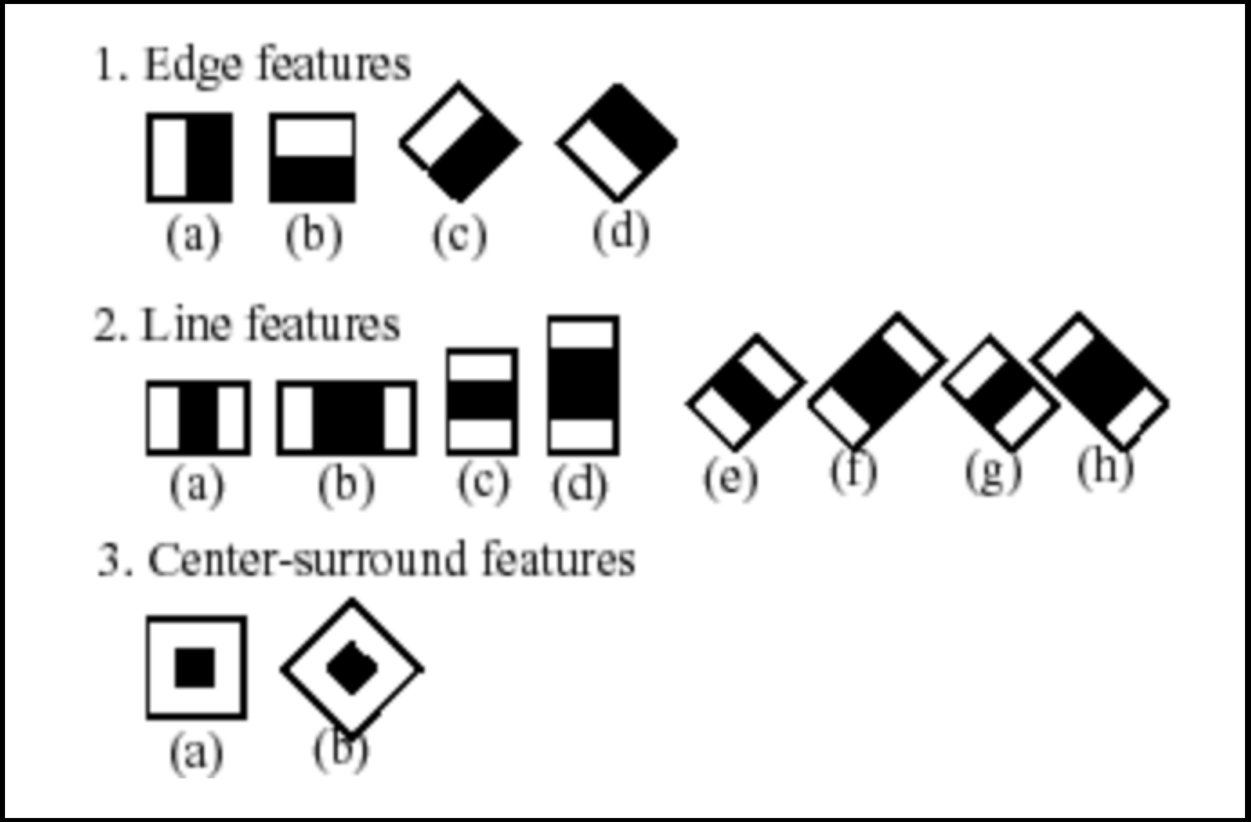


Experiment: PCA and SVM

- Attempted to classify a key point via SVM
  - left center of eye
- SLOW!
- Explained variance ratio was low
  - First point was ~30%
  - Took 100 points to get to a cumulative ~90%
- Cross Validated RMSE of: 9.04

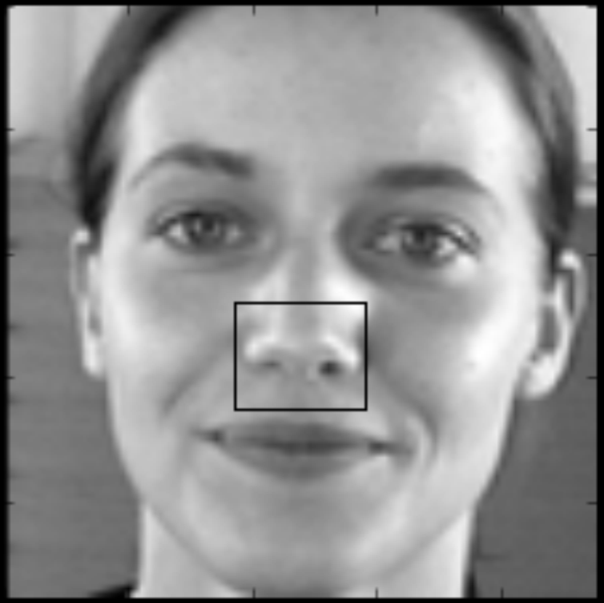
Python Opencv Objdect

- Pre-trained classifiers
  - Face Detect
  - Left Eye, Right Eye, Eye Pair
  - Nose
  - Mouth



Miss Step: Cropping

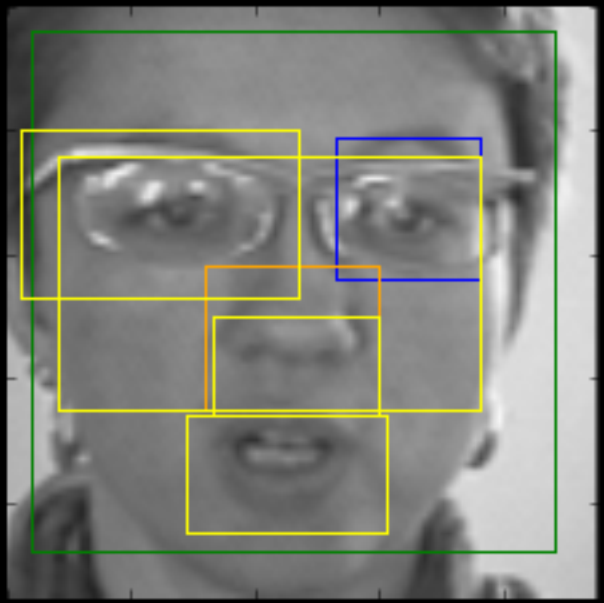
- Crop down images to apply detectors at smaller chunks of the image
- Make passes to cut down the sections of the face starting with the detected face
- Removes information

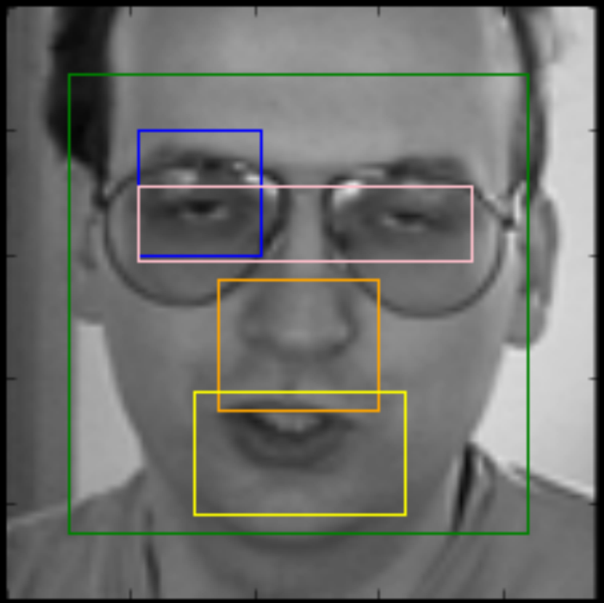


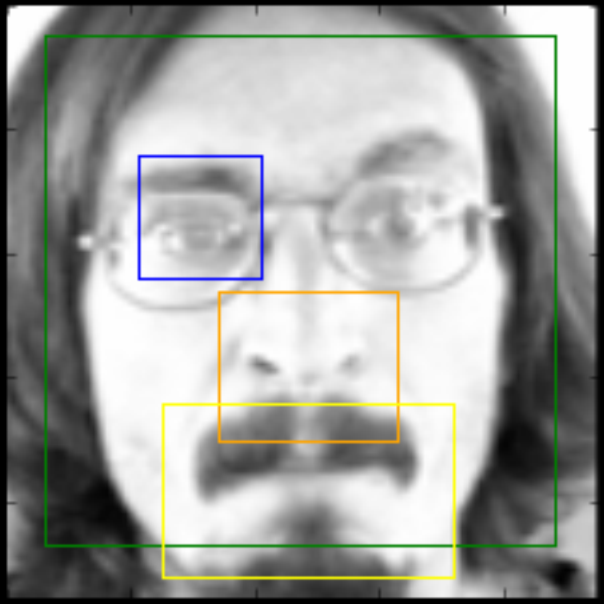
Binning the Results

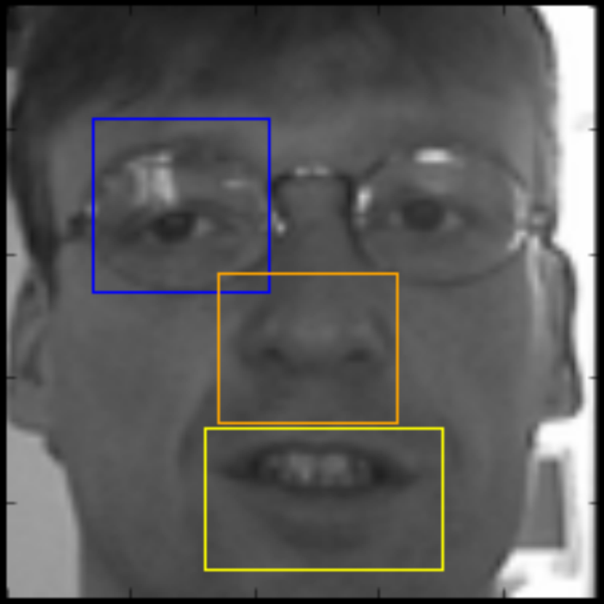
Total	Left Eye	Right Eye	Nose	Mouth
3	TRUE	FALSE	FALSE	FALSE
91	TRUE	FALSE	FALSE	TRUE
6	TRUE	FALSE	TRUE	FALSE
697	TRUE	FALSE	TRUE	TRUE
1	TRUE	TRUE	FALSE	FALSE
156	TRUE	TRUE	FALSE	TRUE
14	TRUE	TRUE	TRUE	FALSE
4065	TRUE	TRUE	TRUE	TRUE

One Eye Detected

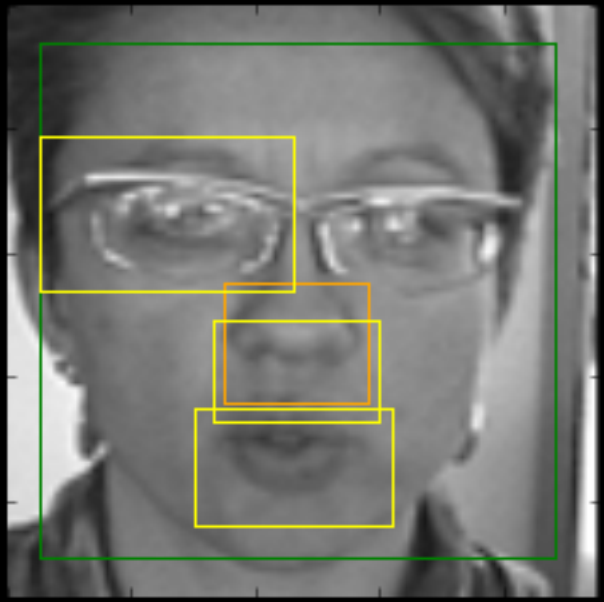


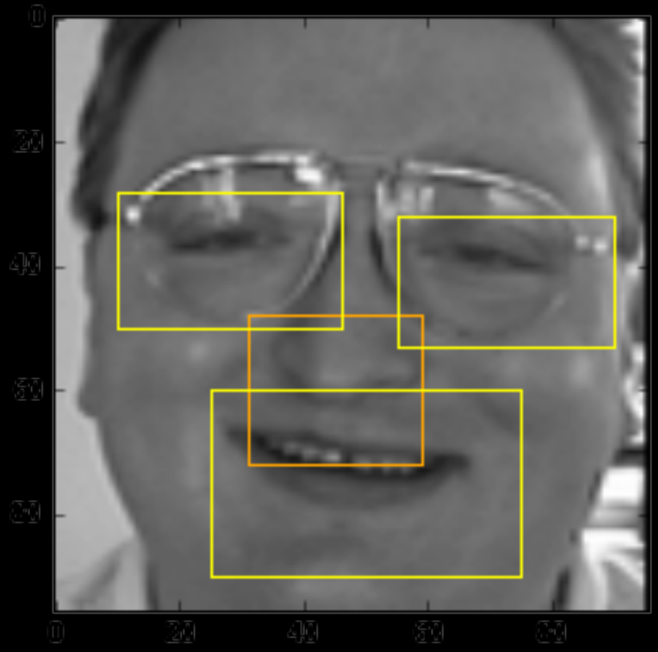


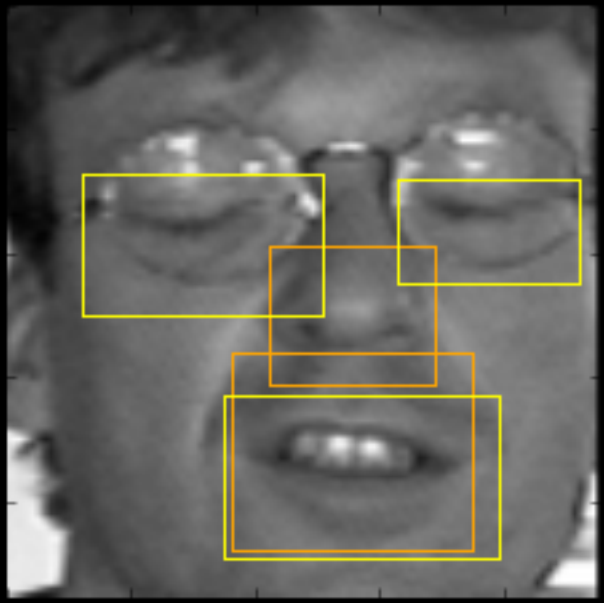


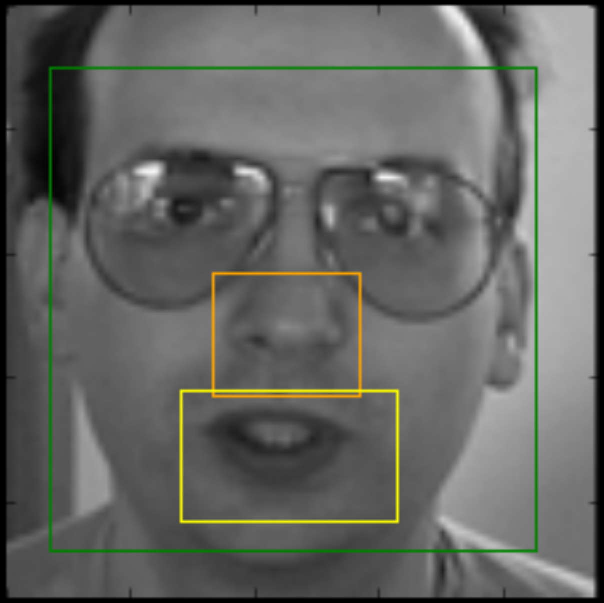


No Eyes Detected

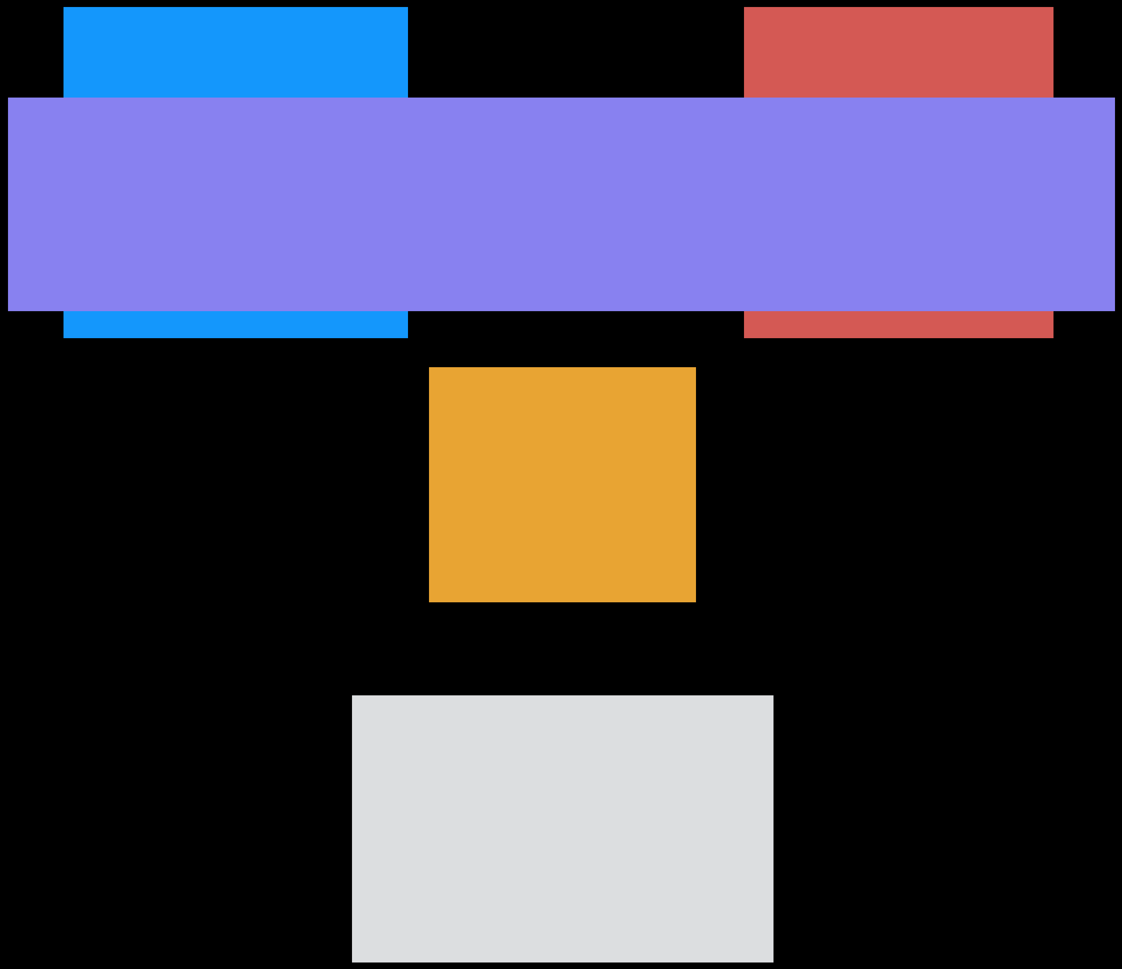


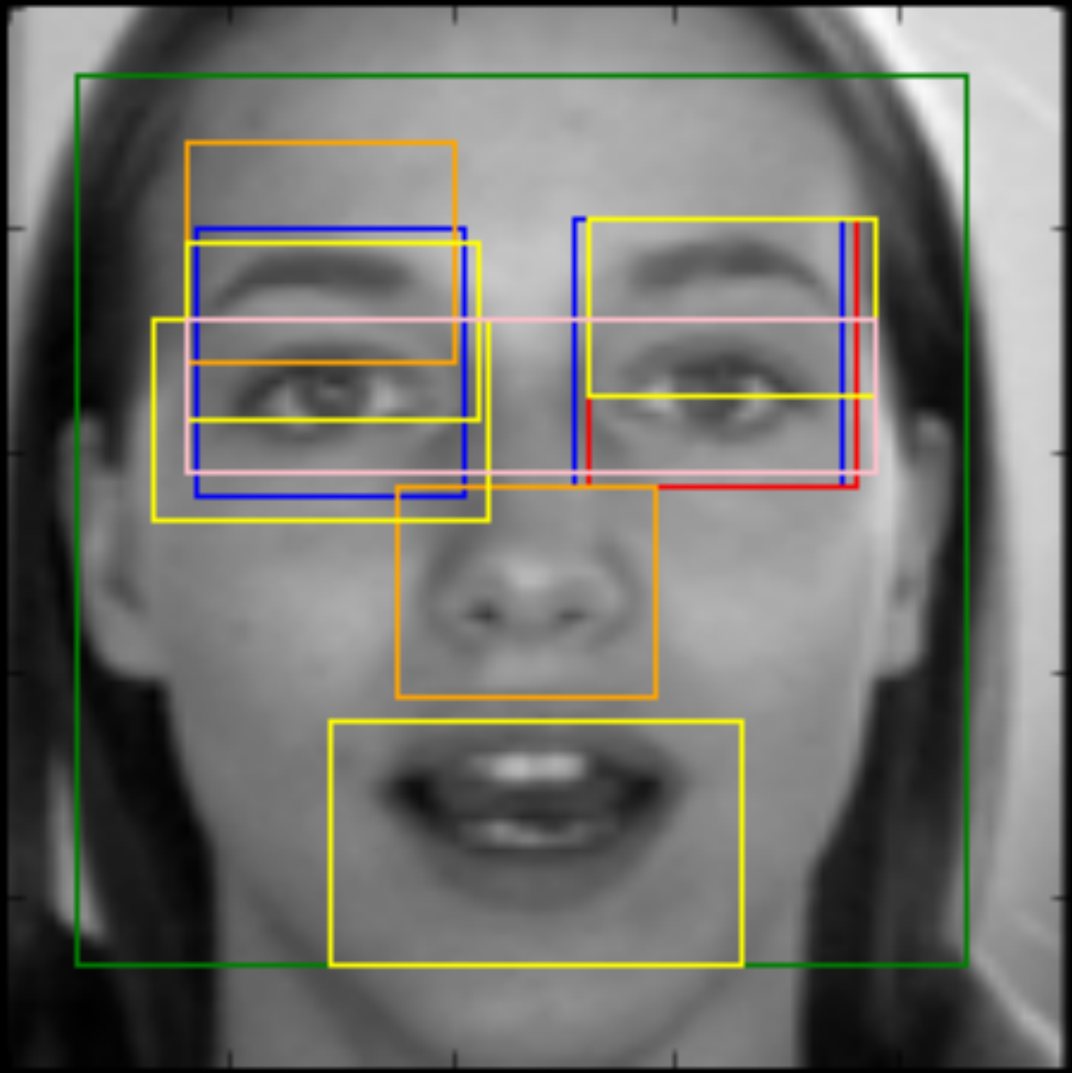






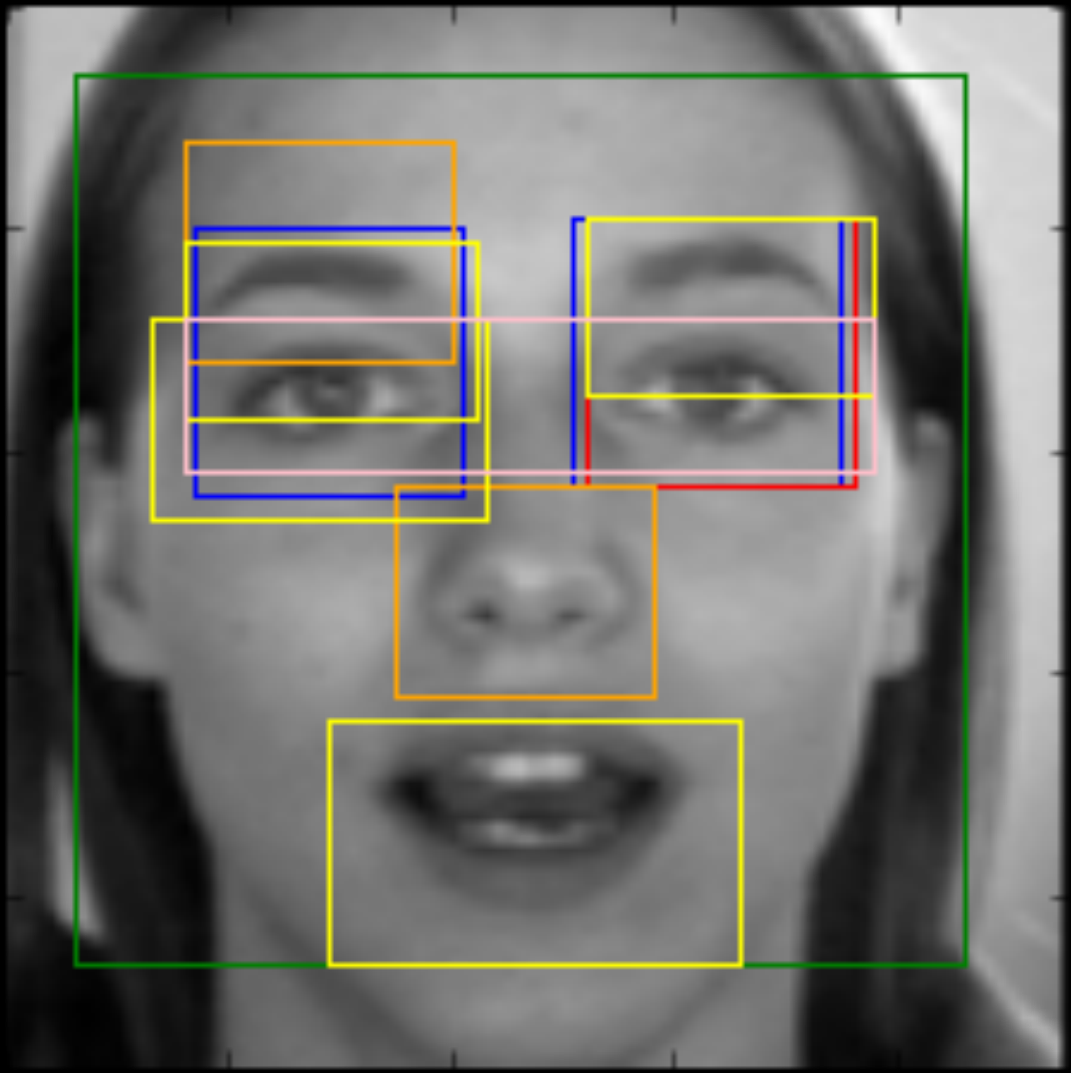
1 or More Boxes Detected





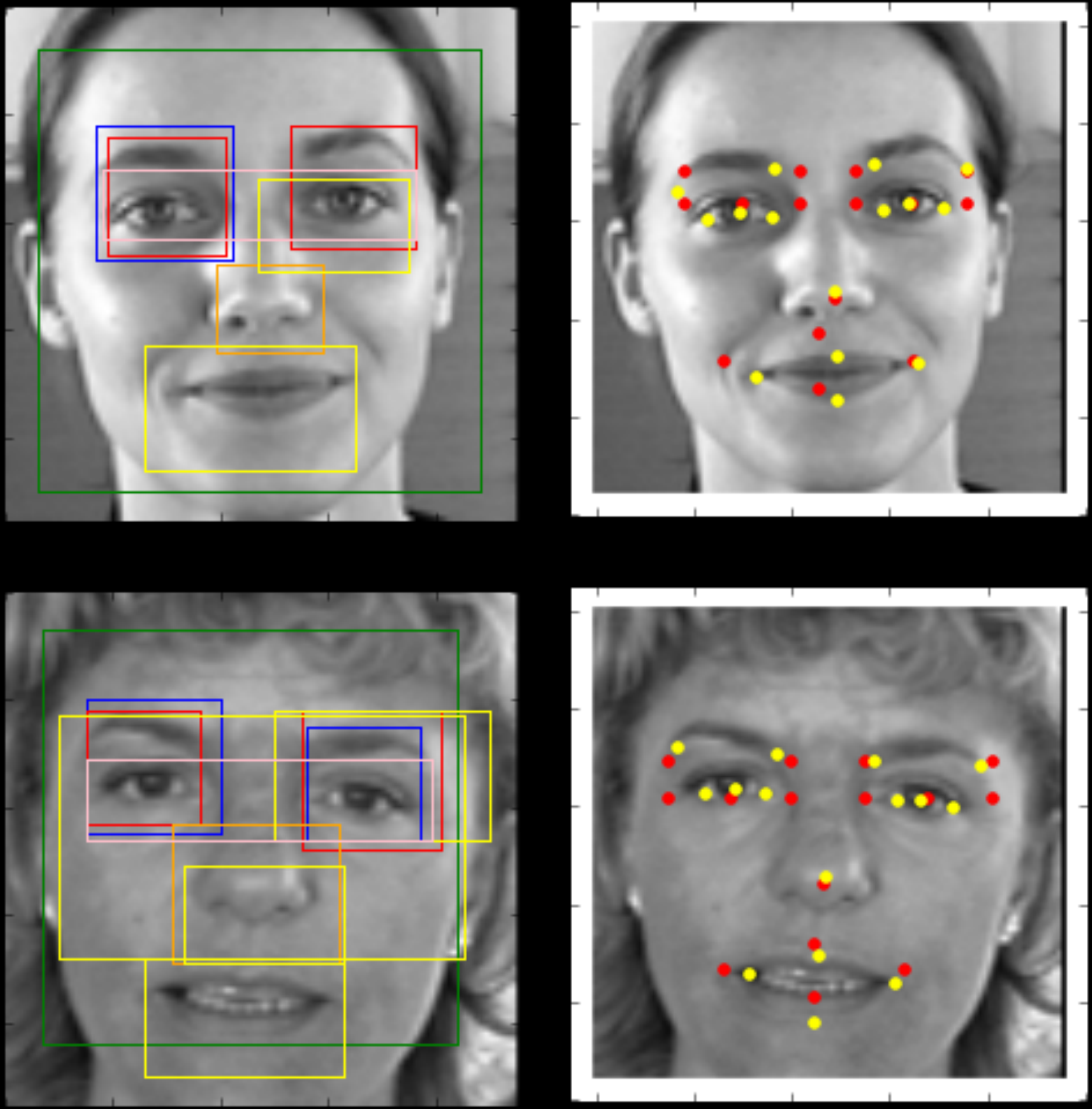
Picking out Boxes

- Left Eye
  - Edges closest to Left side of face detect box
- Right Eye
  - Edges closest to Right side of face detect box
- Nose
  - Box closest to center of face detect box
- Mouth
  - Box closest to lower boundary of face detect box



Predicting Key Points

- Eye Brows
    - Area between eye and eye pair boxes
- Eyes
    - Minimum of common edges between eye pair and individual eye boxes.
    - Inner line marked by eye box
- Nose
    - Center of box
- Mouth
    - Upper half of box, corners at upper quarter of box



Final Result

- RMSE: 5.36
- The better approach might be to use the object detected boxes and then other algorithms to edge detect the individual feature you want to predict