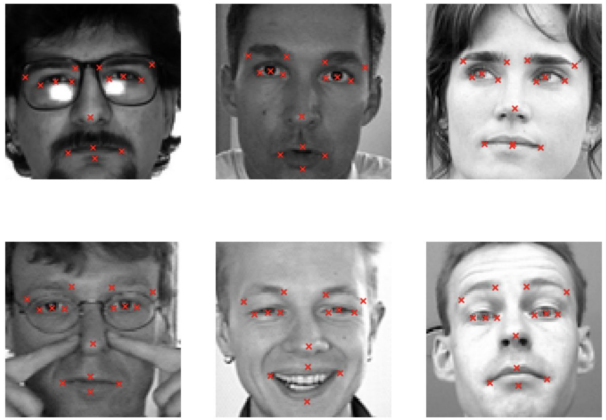
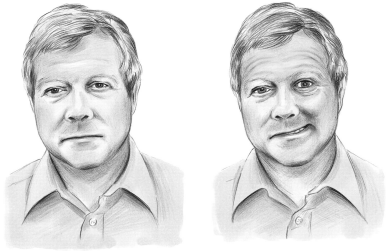
Facial Keypoints Recognition

a W207 Final Project by Harry Xu, Noah Randolph, and Chet Gutwein

Introduction

- Facial keypoints recognition can be used as a building block in several areas, such as information security and medical diagnosis
- Large amount of variation due to 3D pose, size, position, viewing angle, and illumination conditions
- Training and tuning models for keypoint recognition is computationally demanding due to multiple points to classify, total features = all image pixels, continuous data in each feature, and large training set needed





Data, Labels, and Feature Engineering

left_eye_center_x	0.999
left_eye_center_y	0.999
right_eye_center_x	0.998
right_eye_center_y	0.998
left_eye_inner_corner_x	0.322
left_eye_inner_corner_y	0.322
left_eye_outer_corner_x	0.322
left_eye_outer_corner_y	0.322
right_eye_inner_corner_x	0.322
right_eye_inner_corner_y	0.322
right_eye_outer_corner_x	0.322
right_eye_outer_corner_y	0.322
left_eyebrow_inner_end_x	0.322
left_eyebrow_inner_end_y	0.322
left_eyebrow_outer_end_x	0.316
left_eyebrow_outer_end_y	0.316
right_eyebrow_inner_end_x	0.322
right_eyebrow_inner_end_y	0.322
right_eyebrow_outer_end_x	0.317
right_eyebrow_outer_end_y	0.317
nose_tip_x	1.000
nose_tip_y	1.000
mouth_left_corner_x	0.322
mouth_left_corner_y	0.322
mouth_right_corner_x	0.322
mouth_right_corner_y	0.322
mouth_center_top_lip_x	0.323
mouth_center_top_lip_y	0.323
mouth_center_bottom_lip_x	0.995
mouth_center_bottom_lip_y	0.995

Training data contained in .csv file, each row containing 30 data labels and a string containing image data

Labels

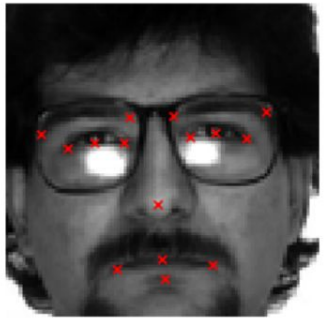
- Used 1,000 samples for dev set with 6,049 training samples left
- Inconsistencies in training labels - only 4 of 15 keypoint labels present in more than 50% of samples

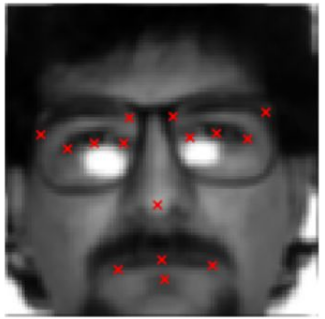
Training Data

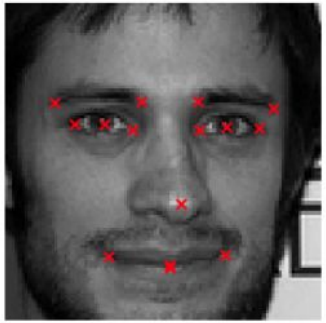
- Grayscale image with size of 96 X 96 pixels for a total of 9,216 features
- Pixel values normalized initially by dividing each value by 255.0
- Later used `from sklearn.preprocessing import StandardScaler`

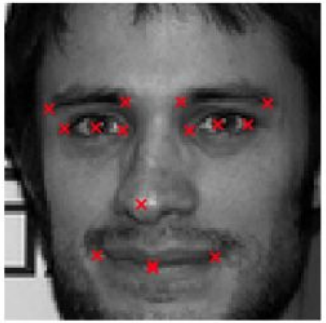
Blurring & Generating Training Samples

- Removing incomplete training samples: problematic because we significantly reduce the size of training data!!!!
- Using Average Replacement: rather than discard, use the average value of each training sample as a replacement
- Blurring: applying a gaussian blur to an image can help model performance
- Generate artificial training samples: with a scarcity of training data, we attempted to generate additional training samples
  - Flipped image on y axis
  - Adapted keypoint labels
  - Doubled the size of our training data
  - Slight decrease in model performance, introduced unwanted bias









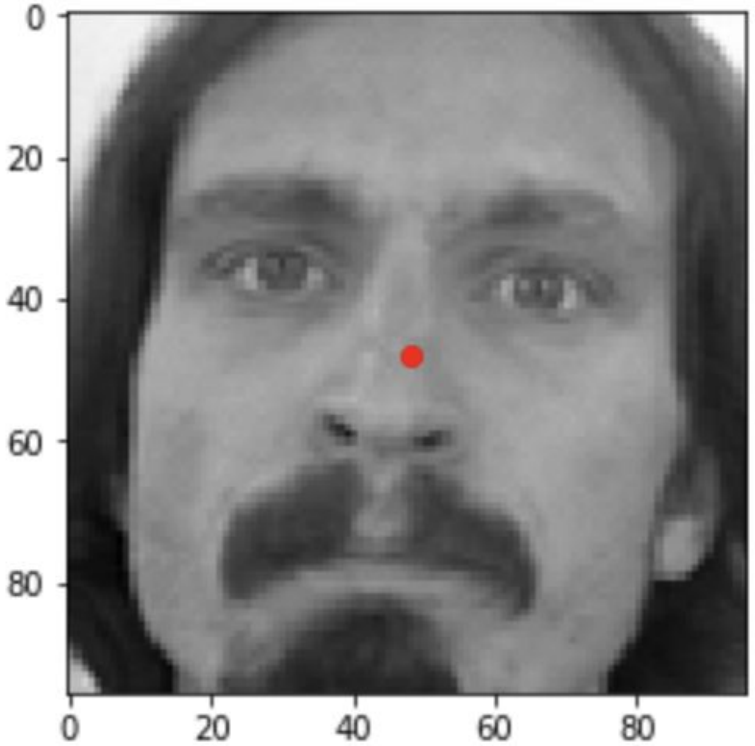
Baseline Submission

- Predicted each facial keypoint location, x and y, to be the average of 96 pixel positions, or pixel (x = 48, y = 48)

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

Score
17.79095

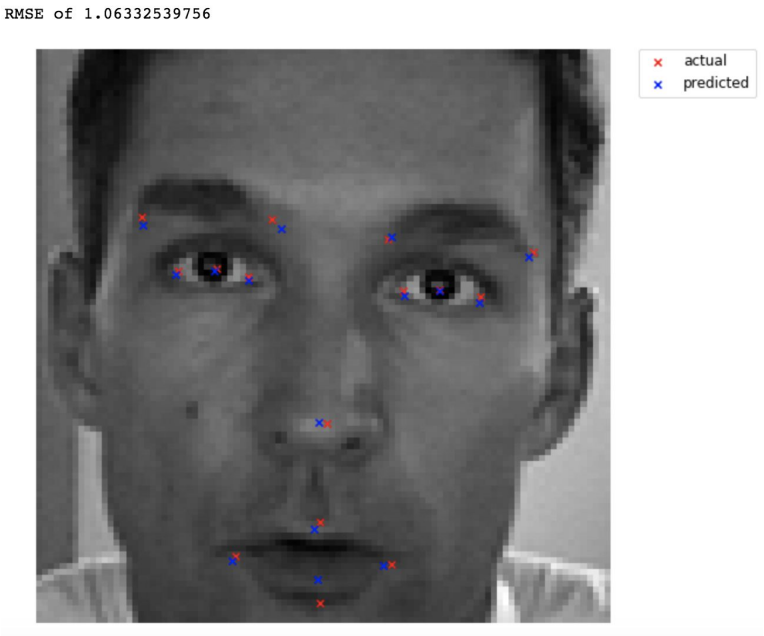
158	▲1	Francis Mitra		12.12504
159	▼89	axon		17.92907
160	▲4	iulian		24.95756



kNN Regression Model

Model	RMSE on Dev	RMSE on Test
Only complete labels	2.57	3.55
Missing labels filled by average	1.86	3.47
Missing labels filled by average and image blurred	1.82	3.45

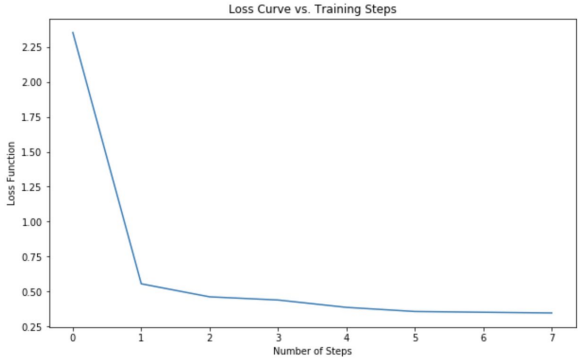
- We used GridSearch to find k=3 to be the optimal hyperparameter for our model
- The model is using uniform weights and standard Euclidean distance
- Produces decent results on our Dev data set, but does not produce as satisfactory results on Test data



Neural Net (MLPRegressor) Model

- We are using a Feed Forward Neural Network with two hidden layers sized (1000, 500)
- Input layer of 9216 (number of pixels) and output layer of 30 (number of x,y keypoints)
- Our activation function is the Rectified Linear Unit (ReLu)
- Our learning rate is held constant
- We are using the ‘Adam’s’ Optimizer which is a form of Stochastic Gradient Descent

Model	**RMSE on Dev**	**RMSE on Test**
Only complete labels	2.59	3.33
Missing labels filled by average	2.55	3.28
Missing labels filled by average and image blurred	2.27	3.6



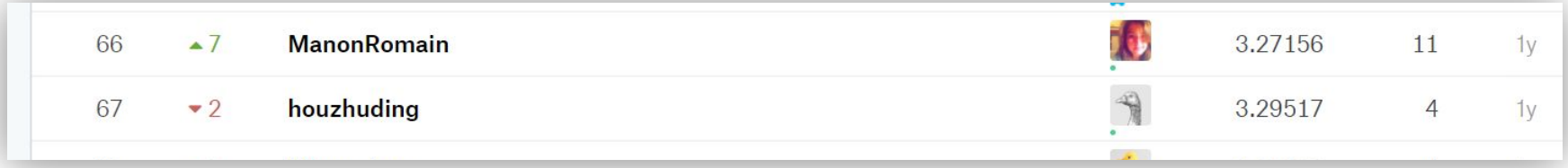
Final Outcome

Results:

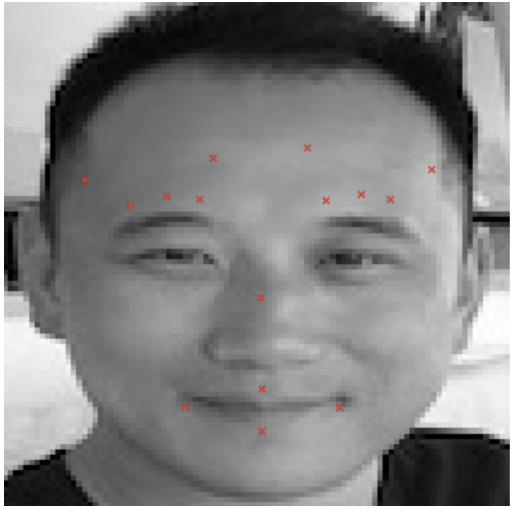
- RMSE better on dev set for k NN (1.82 vs. 2.27)
- RMSE better on test set for MLPRegressor (3.28 vs. 3.45)

If we had more time:

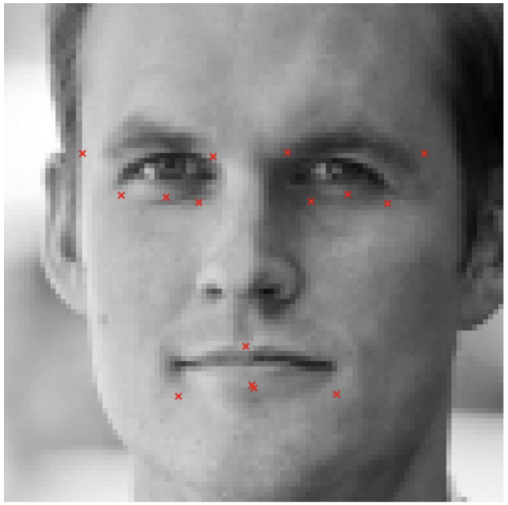
- Build a more complex neural net with convolutional hidden layers
- Feature engineering: more complex model to predict missing labels based on full sets of labels than just taking the average



Thank you - Any Questions?



Harry Xu
Slack: **@harryxu**
E-mail: **harryxu@berkeley.edu**



Noah Randolph
Slack: **@noahrandolph**
E-mail: **noah_randolph@berkeley.edu**

Chet Gutwein
Slack: **@cgutwein**
E-mail: **chet_gutwein@berkeley.edu**