



# Facial Keypoint Recognition

Final Presentation

w207 03, Spring 2021

**The Black Boxes**  
(Jackie, Joanie, Rakesh, Sandip)

# Project Goals

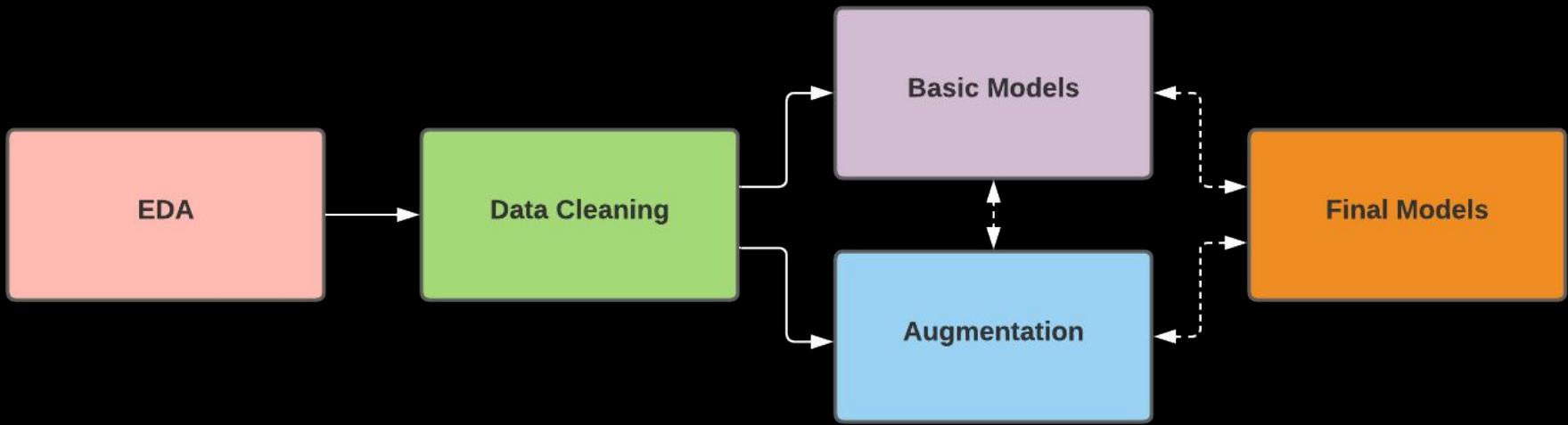


- Given an image, predict the location of (x,y) key points like nose\_tip, right\_eye\_center, etc
- Achieve best possible performance (RMSE):

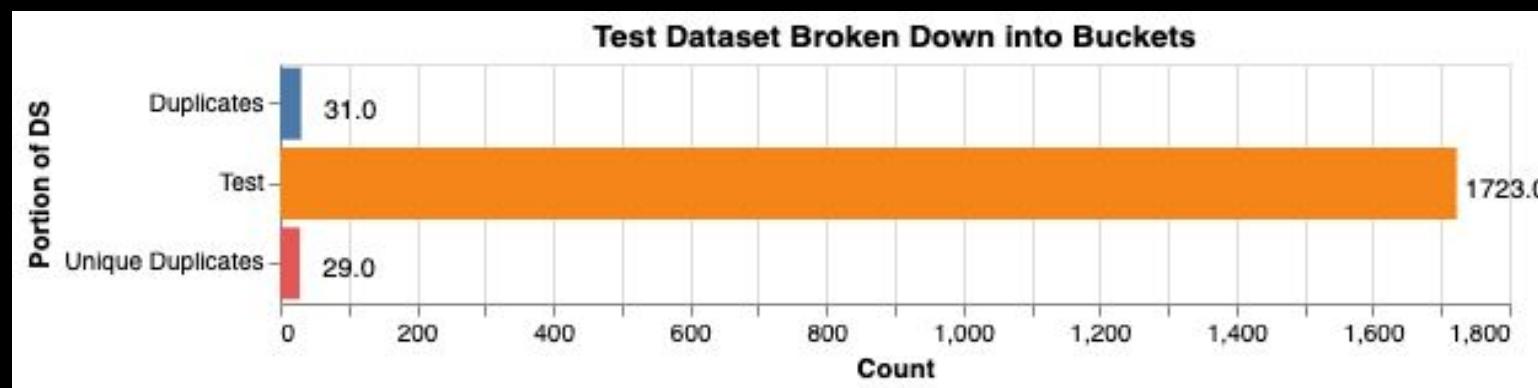
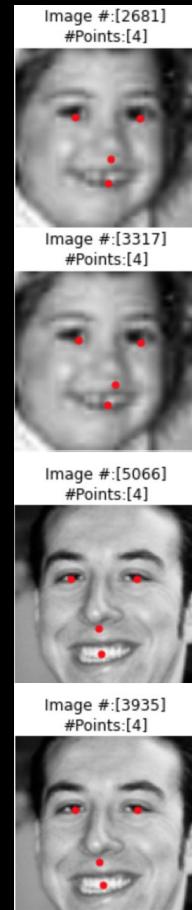
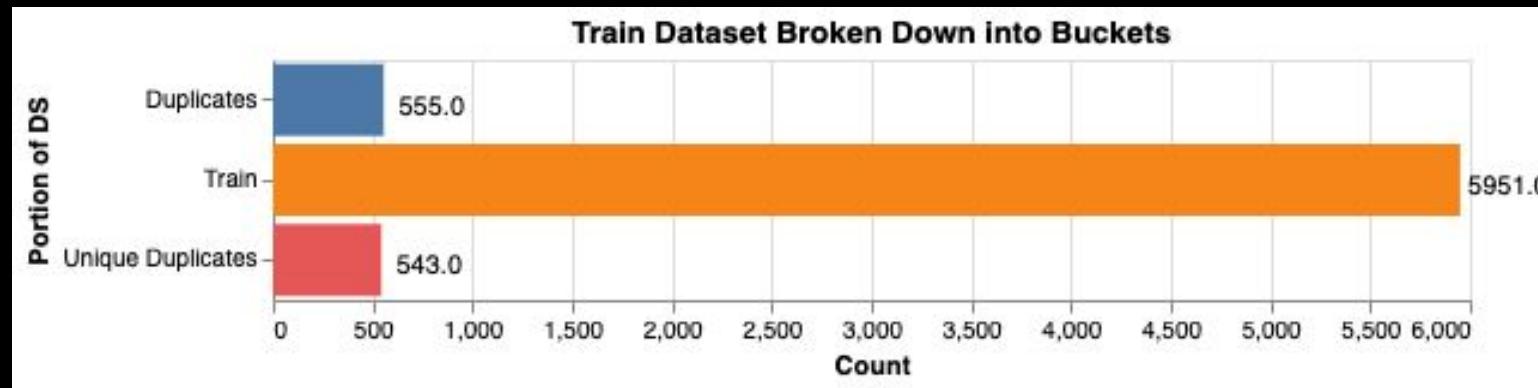
$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

- **Training dataset:** 7049 images,  $\leq 30$  X,Y co-ords.
- **Testing dataset:** 1783 images.

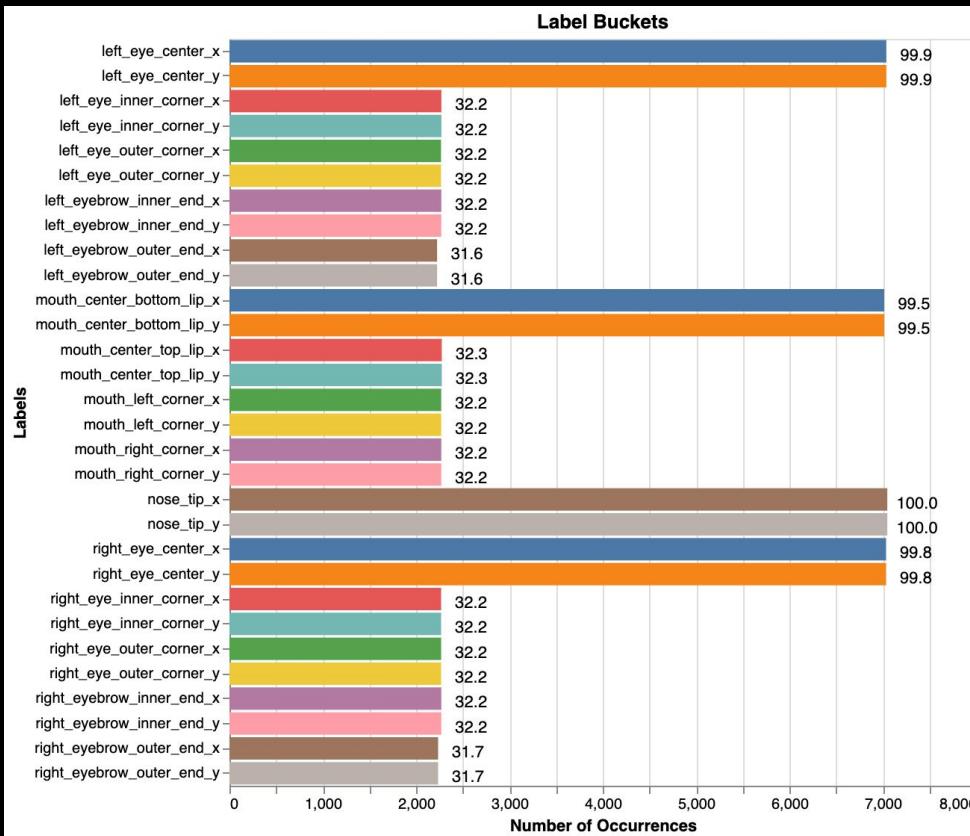
# Project Approach



# Exploratory Data Analysis: Duplicates



# Exploratory Data Analysis: Missing Data

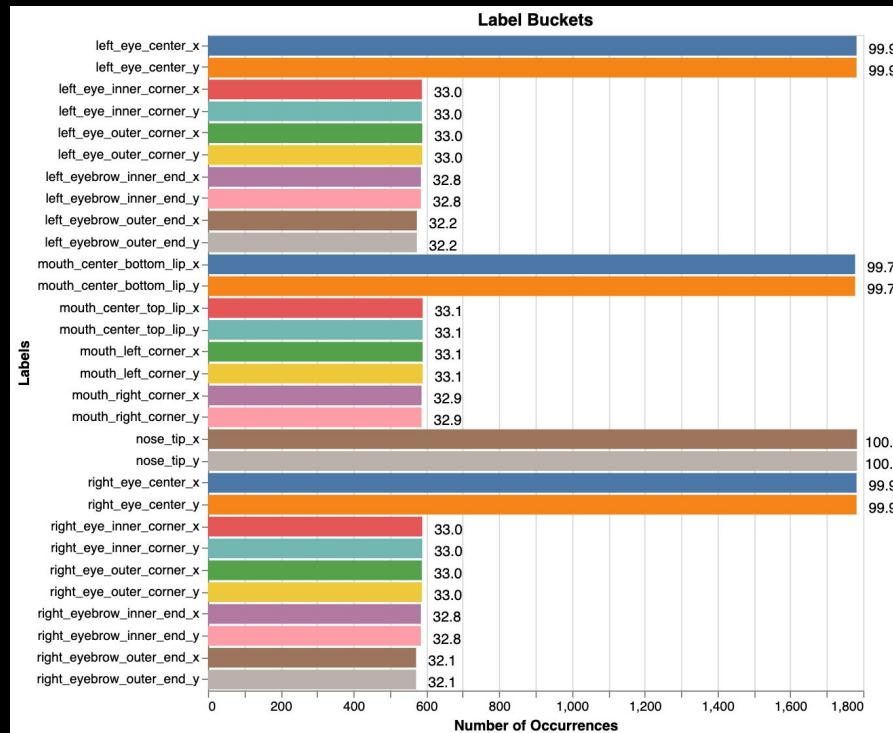


(x, y) Keypoints to predict:

- left\_eye\_center
- right\_eye\_center
- left\_eye\_inner\_corner
- left\_eye\_outer\_corner
- right\_eye\_inner\_corner
- right\_eye\_outer\_corner
- left\_eyebrow\_inner\_end
- left\_eyebrow\_outer\_end
- right\_eyebrow\_inner\_end
- right\_eyebrow\_outer\_end
- nose\_tip
- mouth\_left\_corner
- mouth\_right\_corner
- mouth\_center\_top\_lip
- mouth\_center\_bottom\_lip

~70% of data is missing for certain keypoints!

# Exploratory Data Analysis: Test Predictions

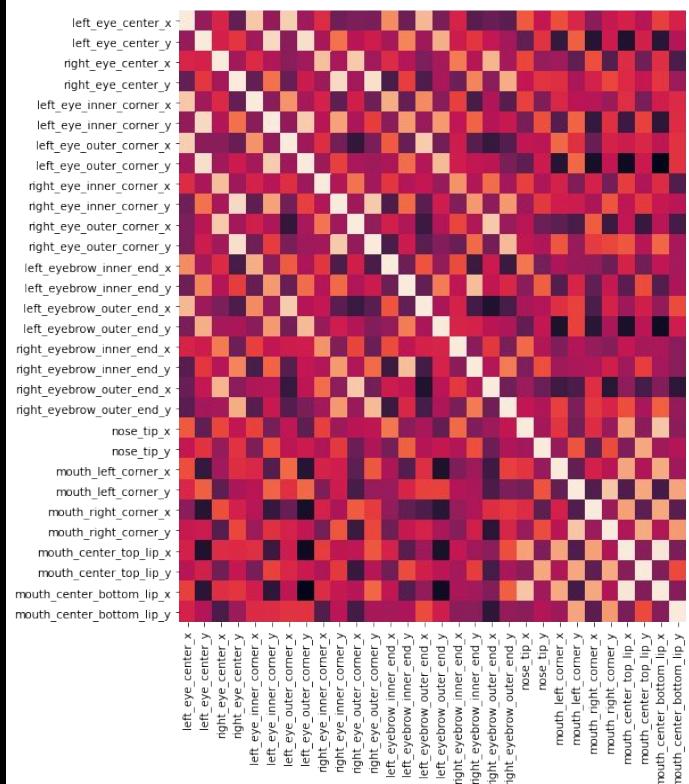


Most of the images we will need to predict only require 8 keypoints.

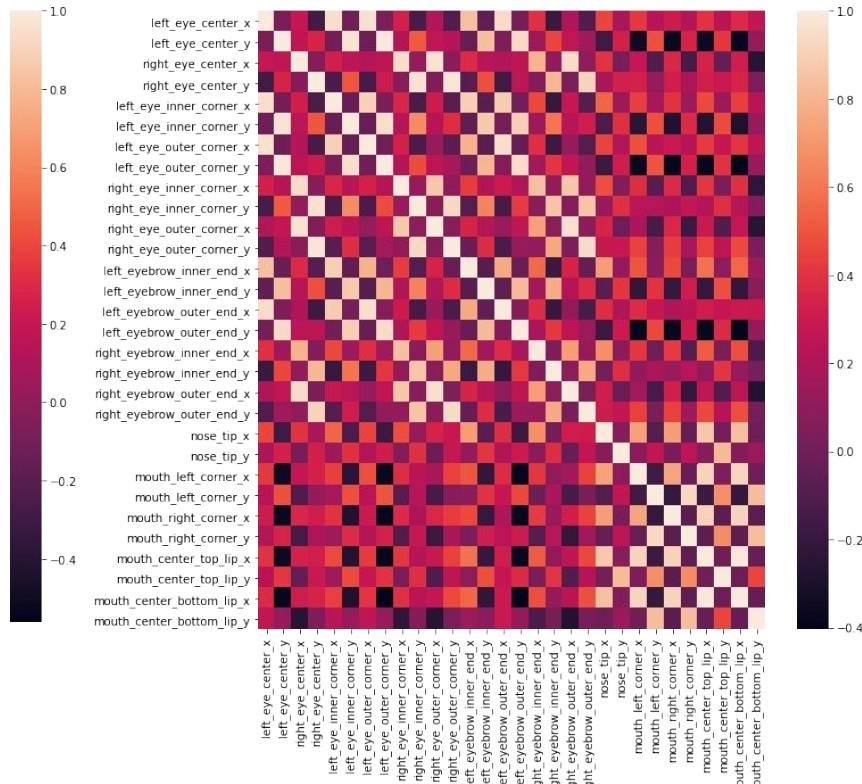
# Approach to Missing Data: Linear Model

- Features which have  $\leq 50$ -missing data-points are considered Dense.
- Features which have no missing-data-points are considered full.
- $\geq 2$  significantly correlated reference points used to triangulate.
- Correlations of more than 0.5 are considered significant.
- Data-points which are full in ('X' and 'y') are used to train the corresponding linear-model.
- Data-points which are full in 'X' but empty in 'y' are augmented by this model.
- This augmentation was possible by setting the acceptable accuracy ( $R^2$ ) to 0.45.
- $>0.45$  the augmentation does not converge;
- for a few features, the models that came up were less than 50% accurate.

# Approach to Missing Data: Linear Model

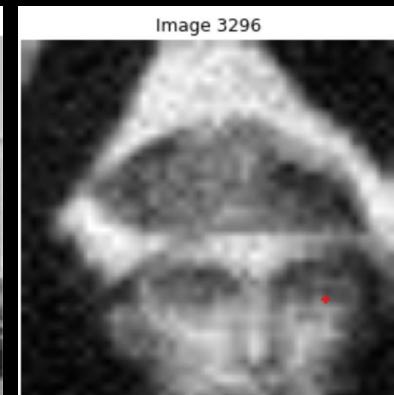
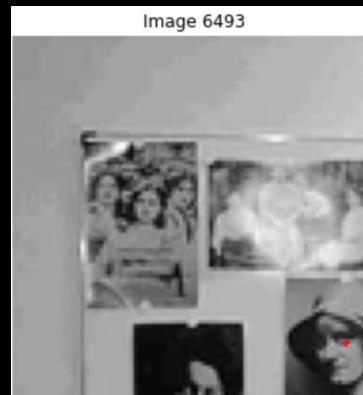
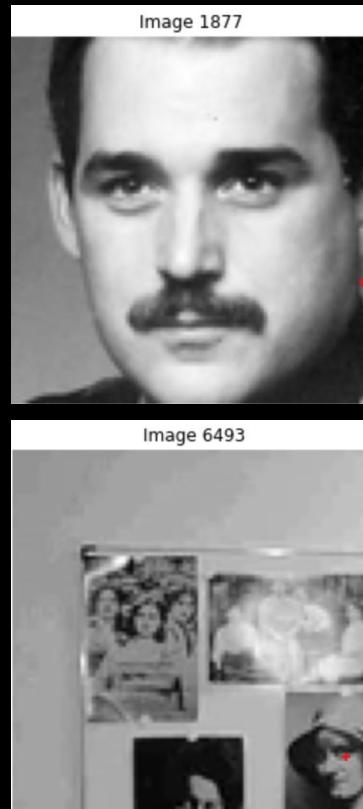
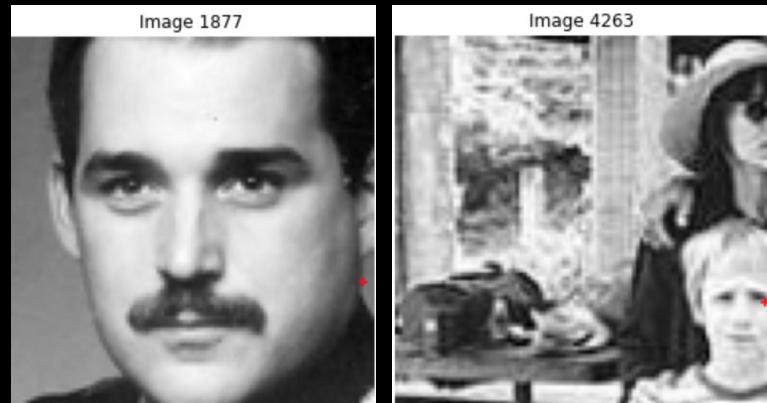
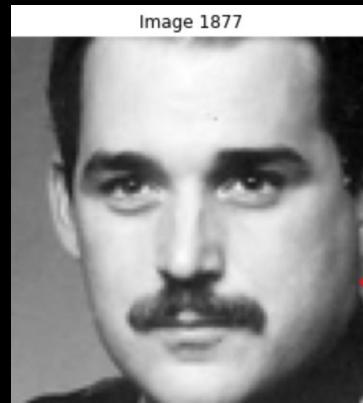
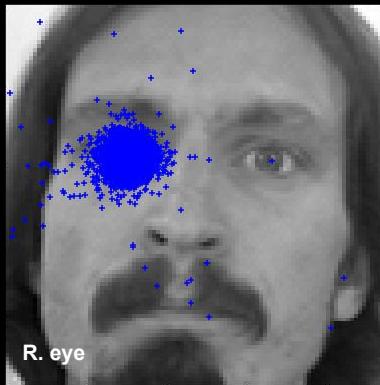
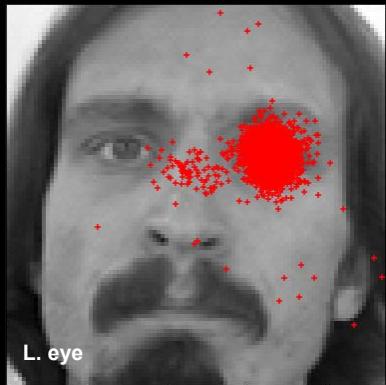


Before Augmentation

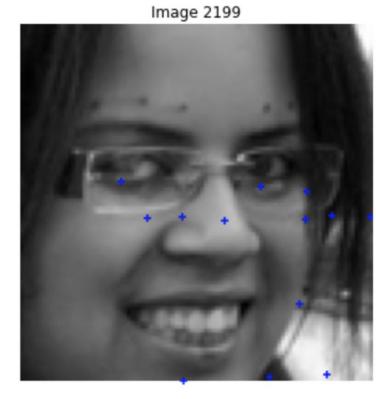
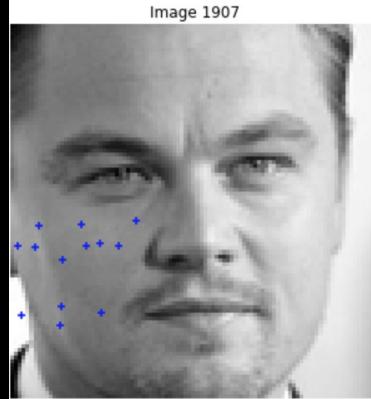
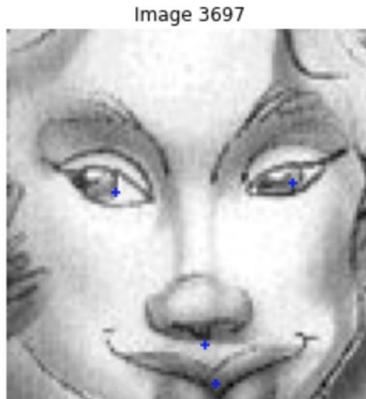
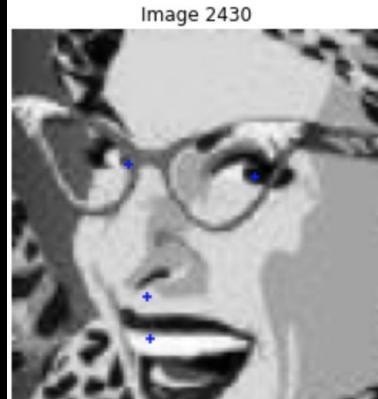
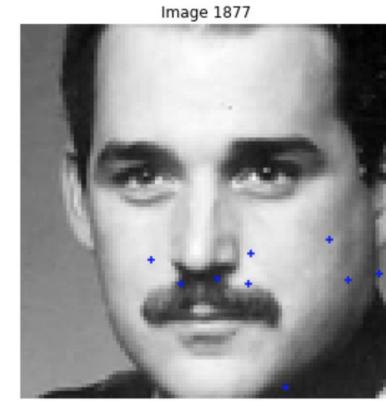
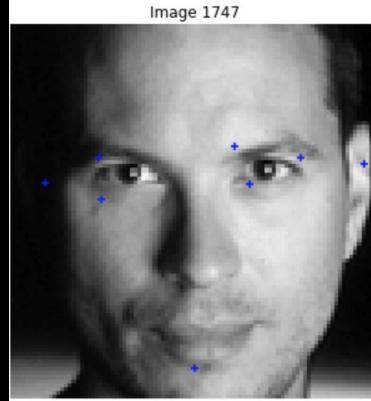
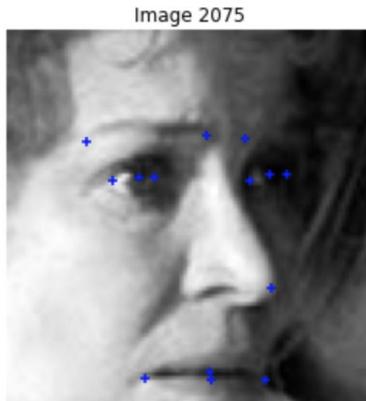
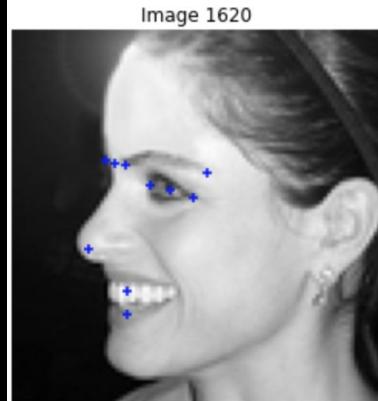


After Augmentation

# Exploratory Data Analysis: Outliers

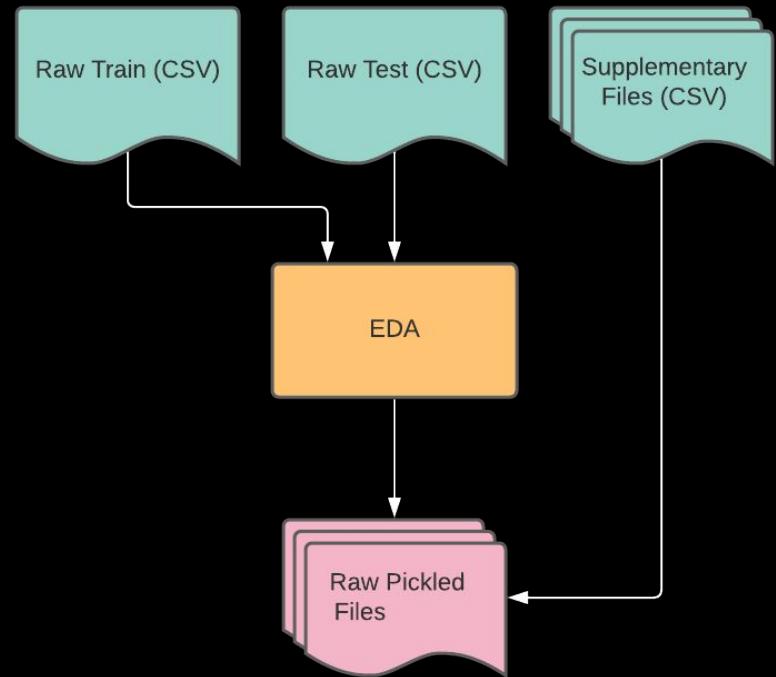


# Exploratory Data Analysis: Interesting & Mislabelled



# Exploratory Data Analysis: Conclusions

- **Duplicates:**
  - Same image repeated.
  - Same person, different angles/expressions.
- **Distractions:**
  - Eyewear, hats, moustaches.
  - Blurry, grainy photos.
  - Multiple people in same.
  - Collages.
  - Angled.
- **Unconventional Images:**
  - Cartoons.
  - Babies.
  - “Pictures” of people.
- **Mislabelled:**
  - Labels in wrong place.
- **Outliers:**
  - Labels very far off their intended position.
- **Miscellaneous:**
  - Extreme angles.
  - Cut off pictures.



# Exploratory Data Analysis: Conclusions

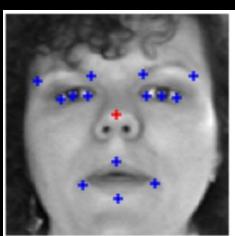


Image 143

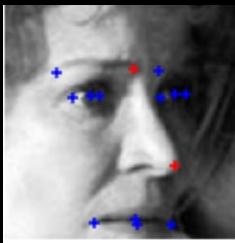
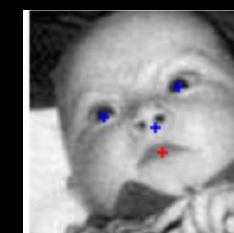
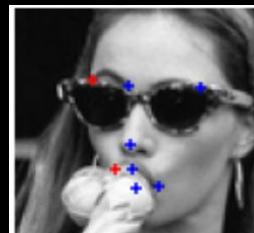
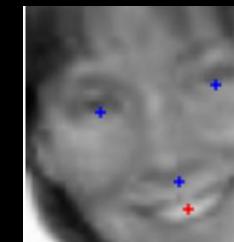
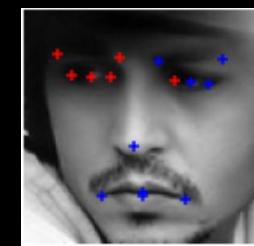
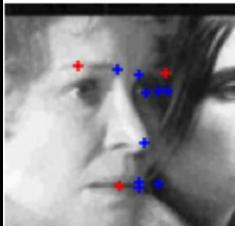
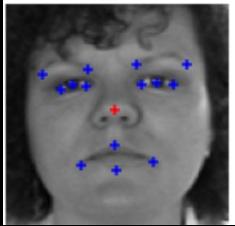
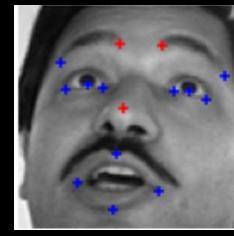
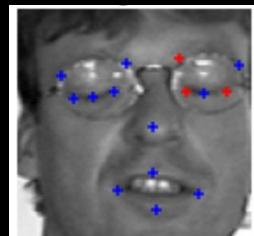
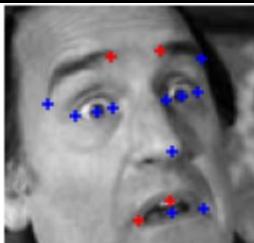


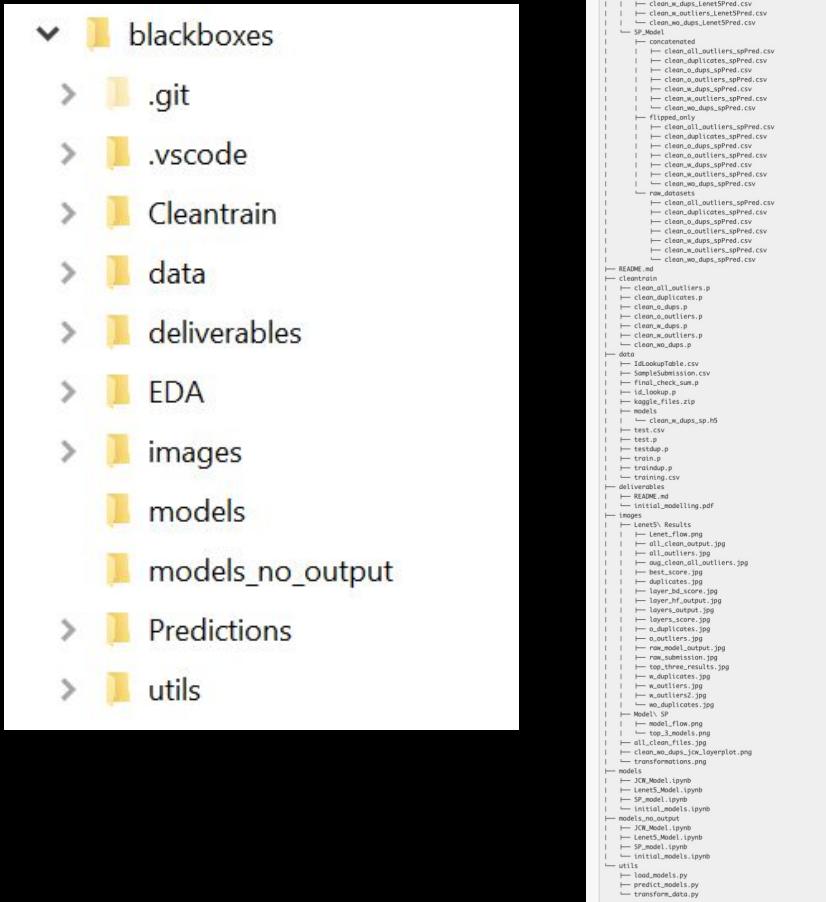
Image 2090



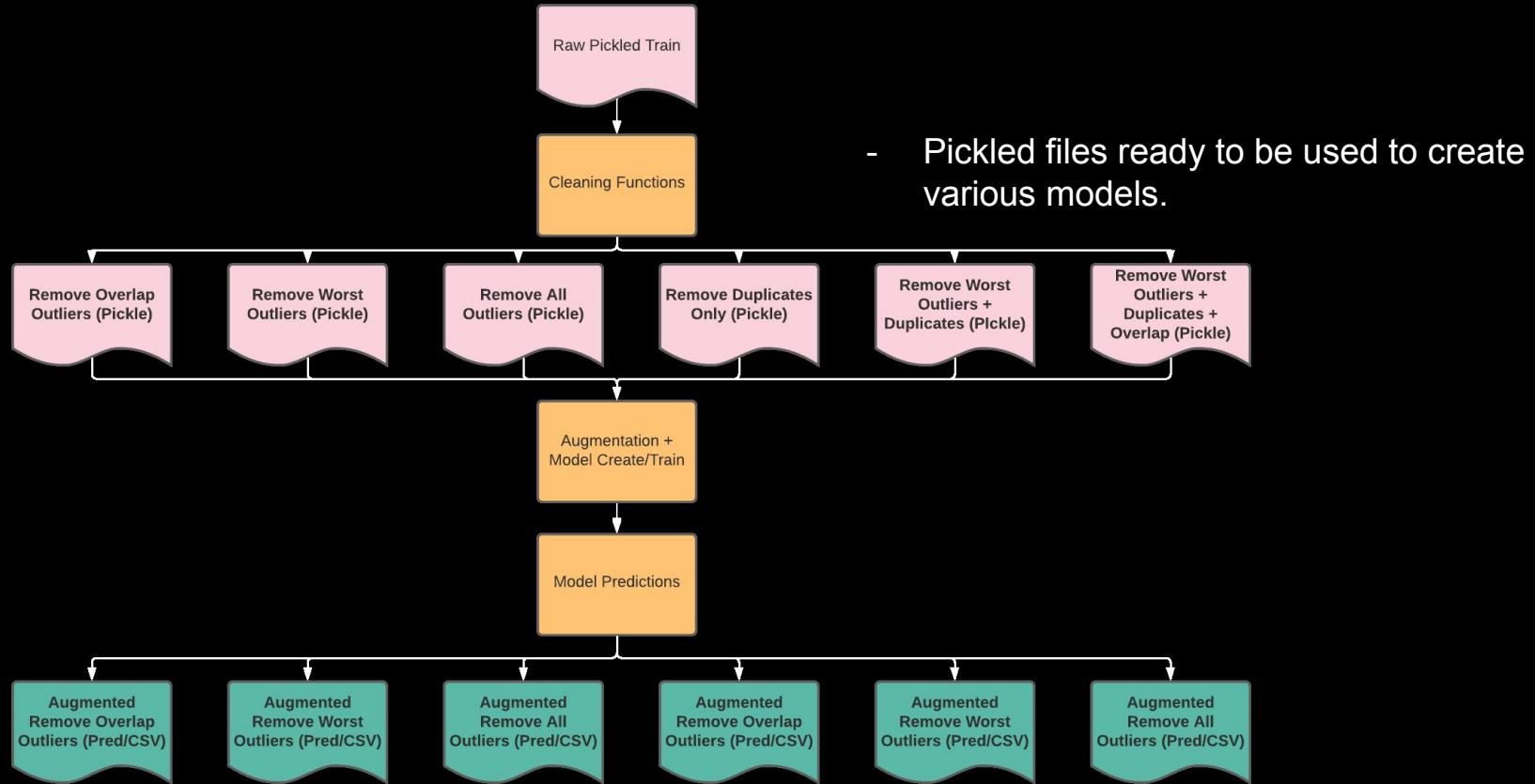
# Repository Structure

## Level 0 / Main Categories:

- EDA
- Predictions
- README
- CleanTrain
- Data
- Deliverables
- Images
- Models
- Models\_no\_output
- Utils

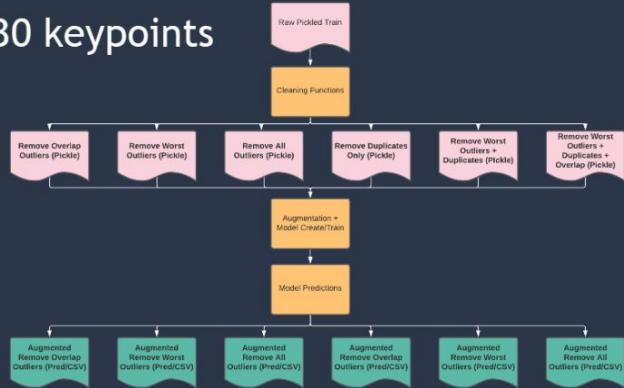


# Data Cleaning & Augmentation

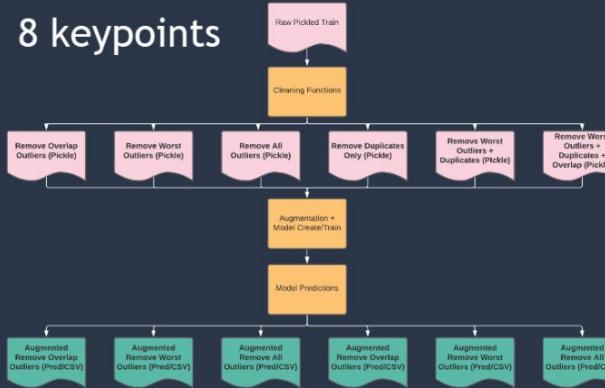


# Data Augmentation - Another approach

30 keypoints



8 keypoints

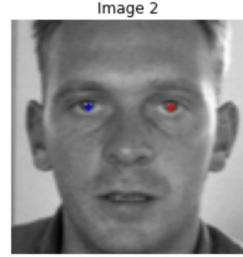
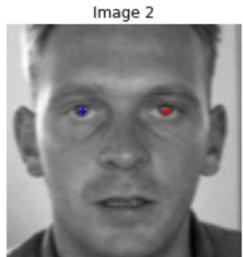
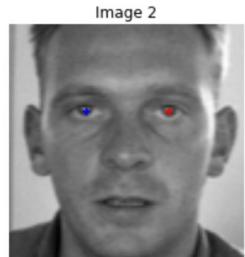
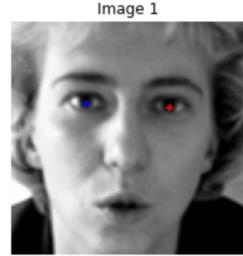
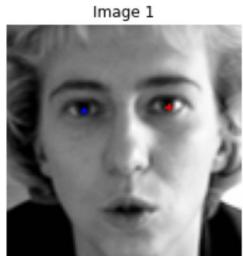
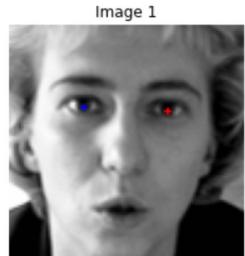
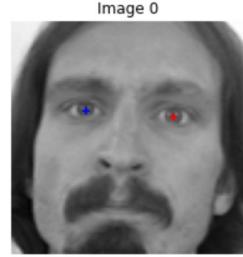
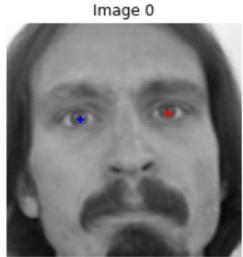
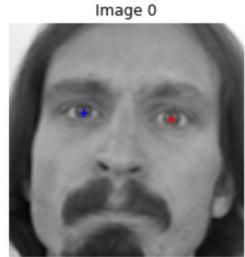


+

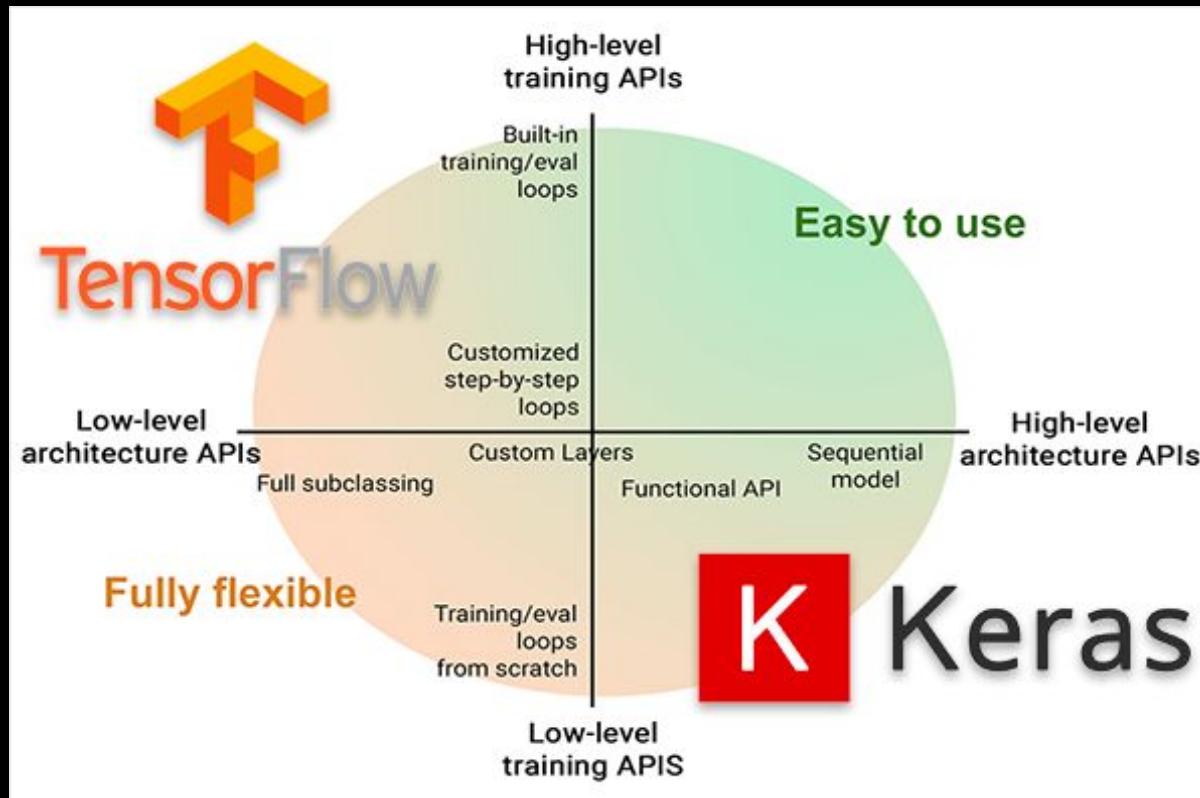
Combined Predictions



# Exploratory Data Analysis: Augmentations



# TensorFlow vs. Keras



- Keras is the user-friendly user interface to create CNN's using a TensorFlow backend.
- <https://www.pyimagesearch.com/2019/10/21/keras-vs-tf-keras-whats-the-difference-in-tensorflow-2-0/>

# Some modelling terminology

## Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

Convolved  
Feature

## Pooling

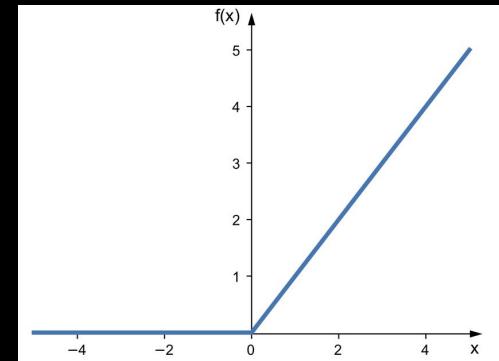
Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters  
and stride 2

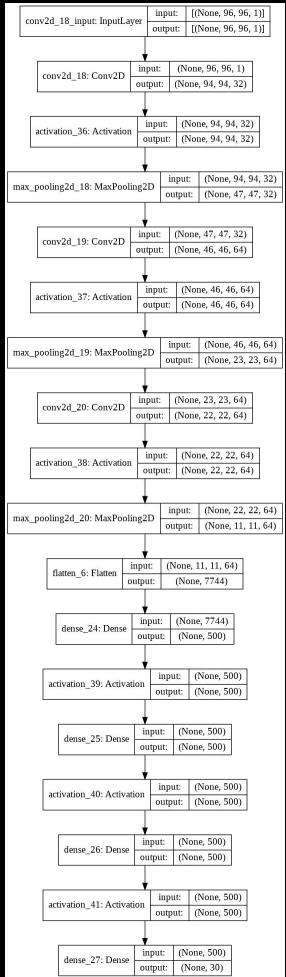
6	8
3	4

ReLU  
(Rectified  
Linear Unit)



# Model 1 (SP)

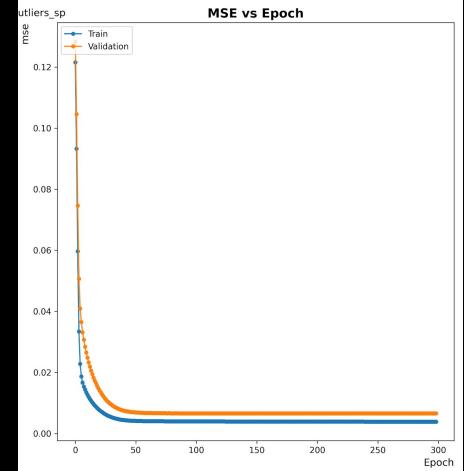
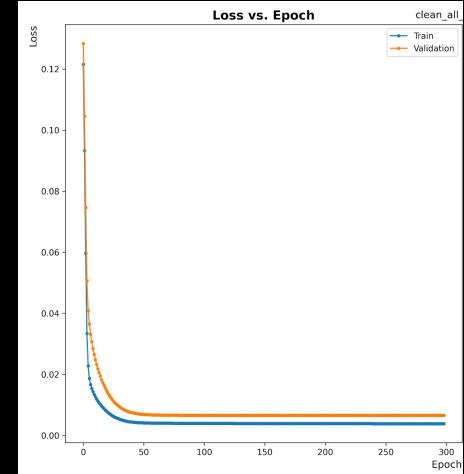
- Based on approaches by Shinya Yuki and Daniel Nouri.
- Input layer takes 4D array (Nsample, NRow, NCol, 1)/(1, 96, 96, 1)
- 3 x 2D convolution layers.
- 32-64-128 filters (doubles each layer).
- Each convolutional layer is followed by a 2x2 max-pooling layer.
- 2 x fully-connected layers.
- Densely-connected layers have 500 units.
- Rectified linear unit ('ReLU') activation in each layer.
- Optimizer: Stochastic gradient descent (SGD), learning rate = 0.01, momentum = 0.9, Nesterov = True.
- Batch size of 128.
- Trained for 300 epochs.



# Model 1 (SP): Performance

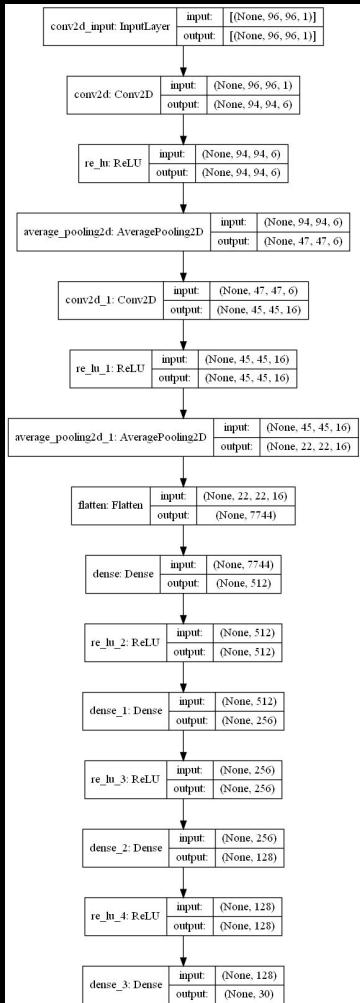
- Model achieved top score of ~3.53
- Used approach combining predictions using 8 and 30 keypoints.

29 submissions for span852		Sort by Private Score	
All	Successful	Selected	
Submission and Description		Private Score	Public Score
<a href="#">combined_clean_o_outliers_spPred.csv</a> a minute ago by span852		3.53171	3.76181
combined_4			
<a href="#">combined_clean_w_dups_spPred.csv</a> a few seconds ago by span852		4.01852	4.14986
combined_5			
<a href="#">combined_clean_duplicates_spPred.csv</a> 2 minutes ago by span852		4.05111	4.16324
combined_2			



# Model 2 (JN)

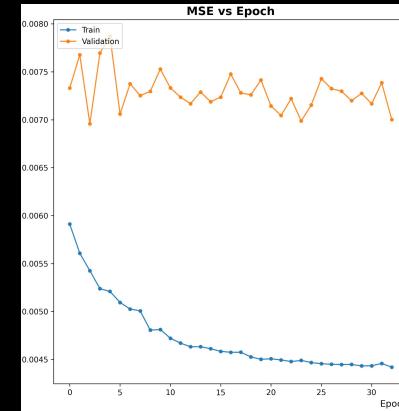
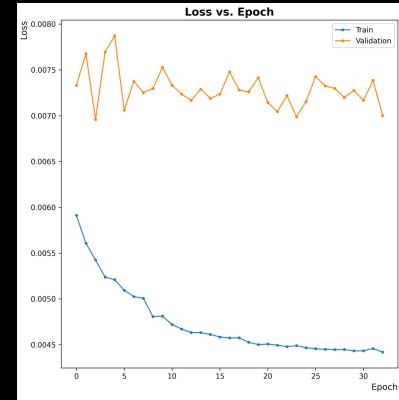
- Inspired by the famous LeNet-5 model
- Input layer takes 2D array ( 96, 96, 1)
- 2 sets of [2D convolution layers + rectified linear unit (RELU) + 2DAverage Pooling] x 3 Dense layers
- 128 batch size, 300 epochs with a patience set at 30 (with early stop set).
- Adam optimizer - stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments and recommended for CNN
- ``act = Adam(lr = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-8)``
- ``loss = 'mean_squared_error'``
- Various flavors of this model were created and used for predictions



# Model 2 (JN) - Approach and Performance

The following variations of models were created:

1. Baseline, no augmentation (3.29689 RMSE)
2. Approach 1: all versions of cleaned train data (clean section) set were used to create models and predictions (3.23581)
3. Approach 2: all versions of cleaned train data (clean section) set were used + varying the layers in the model used to create models and predictions (3.31447)
4. Approach 3: all versions of cleaned train data (clean section) set were used + varying layers + image augmentation (brightness and dim) to create models and predictions (3.31447)
5. Approach 4: Use different cleaned versions of train data set flip the images, add brigthness=0 dim = 0 for 30 keypoints and 8 keypoints. (2.8062)
6. **Approach 5:** Use only clean\_all\_outliers of train data set flip the images, add brigthness=1.4 dim = 0.3 for 30 keypoints and 8 keypoints - full layers (2.48797)

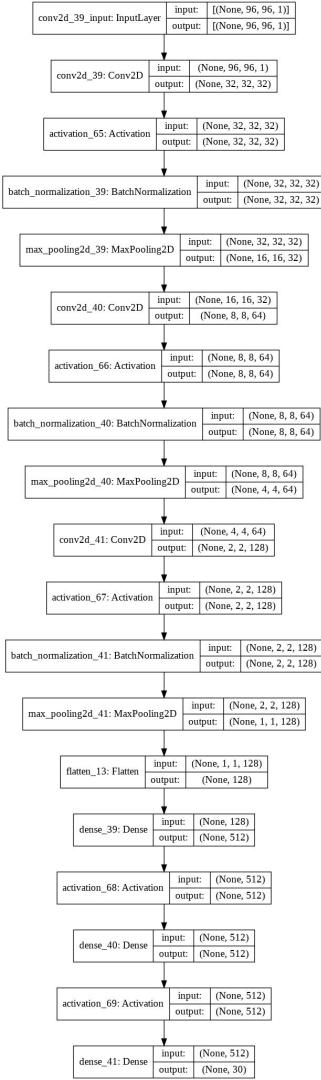


Thank goodness for GPUs!

Submission and Description	Private Score	Public Score
combined_clean_all_outliers_Lenet5Pred.csv a few seconds ago by JackieN 8+30+flip+bright=1.4+dim=0.3+extra layer	2.21913	2.48797
combined_clean_all_outliers_Lenet5Pred.csv 9 minutes ago by JackieN 8+30+flip+bright=1.4+dim=0.3	2.25203	2.52140
combined_clean_all_outliers_Lenet5Pred.csv 33 minutes ago by JackieN 8 + 30 + horizontal flip + bight = 0 + dim = 0	2.49707	2.80652

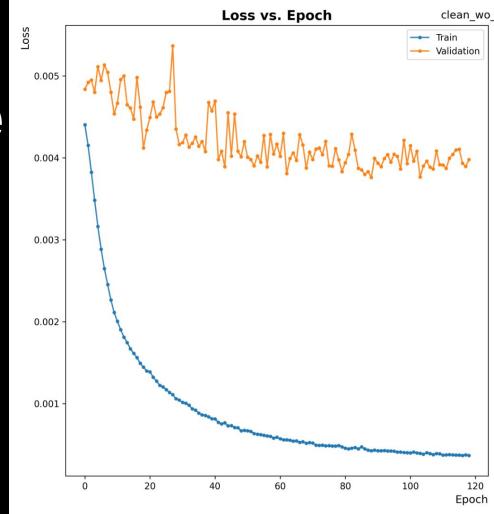
# Model 3 (JCW)

- Following [Shinya Yuki's tutorial](#)
- Convolutional NN with 3 convolutional layers and 2 fully connected layers
- Performed maximum pooling of 2x2 after each convolutional layer
- ‘Relu’ activation function
- Adam optimizer - stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments and recommended for CNN



# Model 3 (JCW) - Approach and Performance

- Additional Approaches:
  - Setting use\_bias to True/False
  - Removing/adding a BatchNormalization layer
  - Appending flipped images to the dataset
  - Brightening all the images
  - Appending flipped images and that whole set brightening
  - Appending both flipped and brightened to train set
  - 8 + 30 approach



[clean\\_wo\\_dups\\_jcwPred\\_nobatch.csv](#) 3.73212 3.85070

2 days ago by [Joanie Weaver](#)

[add submission details](#)

[clean\\_wo\\_dups\\_jcwPred\\_flipped\\_append.csv](#) 3.64806 3.64608

2 days ago by [Joanie Weaver](#)

no batch, bias true, flipped appended to orig data

[clean\\_wo\\_dups\\_jcwPred\\_nobatch\\_nobias.csv](#) 3.83627 3.83681

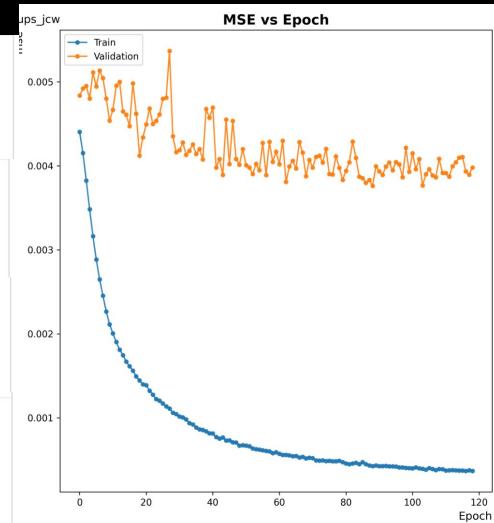
2 days ago by [Joanie Weaver](#)

[add submission details](#)

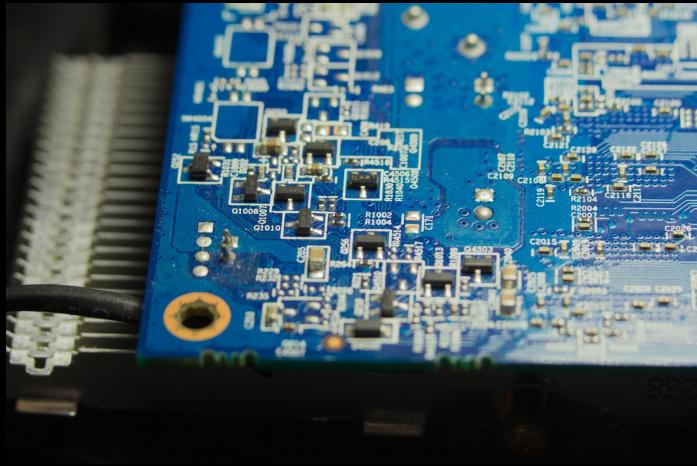
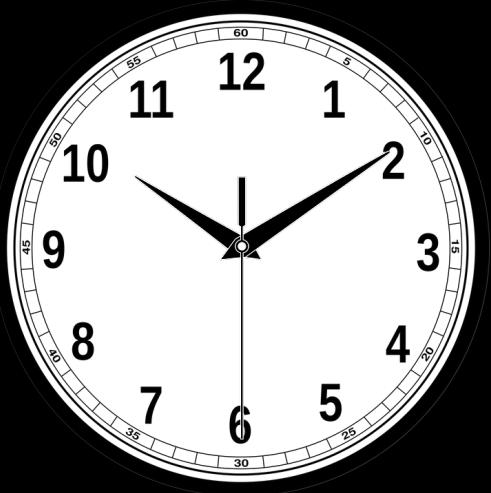
[clean\\_wo\\_dups\\_jcwPred.csv](#) 3.55585 3.56512

an hour ago by [Joanie Weaver](#)

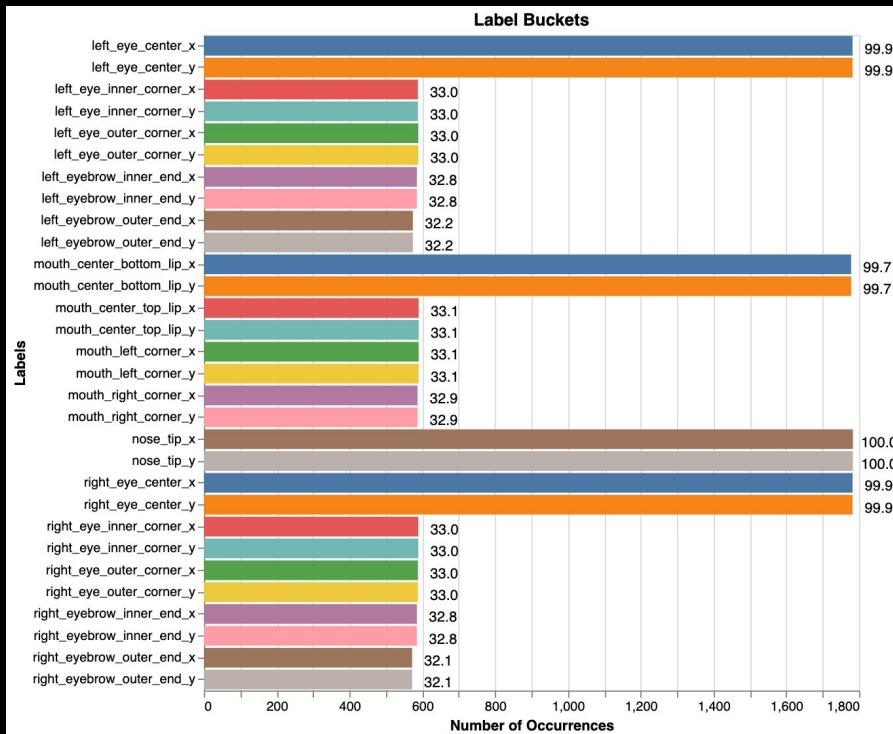
appended flipped and brightened images to train



# Challenges



# Conclusions: Test Predictions



IdLookup Table from Kaggle:

row_id	image_id	feature_name
0	1	left_eye_center_x
1	2	left_eye_center_y
2	3	right_eye_center_x
3	4	right_eye_center_y
4	5	left_eye_inner_corner_x
...	...	...
27119	27120	right_eye_center_y
27120	27121	nose_tip_x
27121	27122	nose_tip_y
27122	27123	mouth_center_bottom_lip_x
27123	27124	mouth_center_bottom_lip_y

Counts of the Number of Features Required for Each Test Images:

num_features	count
8	1190
30	556
28	18
26	9
24	3
6	2
20	2
22	2
18	1

Most of the images we will need to predict only require 8 keypoints.

# Conclusions

- Success of cleaning
- Data is two datasets combined and it makes sense to treat it that way (30 vs 8 key points)
- Success of having more accurate data (vs. augmenting with inaccurate values)
- Learning about CNNs
- Wish we had more time to explore:
  - Stacking models in a k-fold approach
  - Additional augmentation on missing data
  - Transforming new variations of the training images
- We had fun :)

# Conclusions and Challenges

- Cleaning and how we cleaned made an impact
- We could explore augmentation for weeks and there's a lot more here we didn't get to try
- Data is two datasets combined (30 vs 8 key points)
- Having more accurate data improved predictions vs. more inaccurate data
- The wonderful challenge of time and money!!
  - It was obvious that we could spend a lot more time on this challenge
  - More GPU's == More \$\$
  - Google Colab is free, pretty easy and provides a fair amount of GPU
- Collaborating in GitHub on Python notebooks
  - We managed but caution had to be taken
- CNN - New area outside of the scope of models we touched on in class. It's a big world!!
- Given time, we would have liked to have attempted stacking models in a k-fold approach, tried additional augmentation on missing keypoints as well as creating new variations of the training images.

# Thank you

Thanks for listening! Any comments or questions?



Getting Started Prediction Competition

## Facial Keypoints Detection

Detect the location of keypoints on face images



Kaggle · 175 teams · 4 years ago