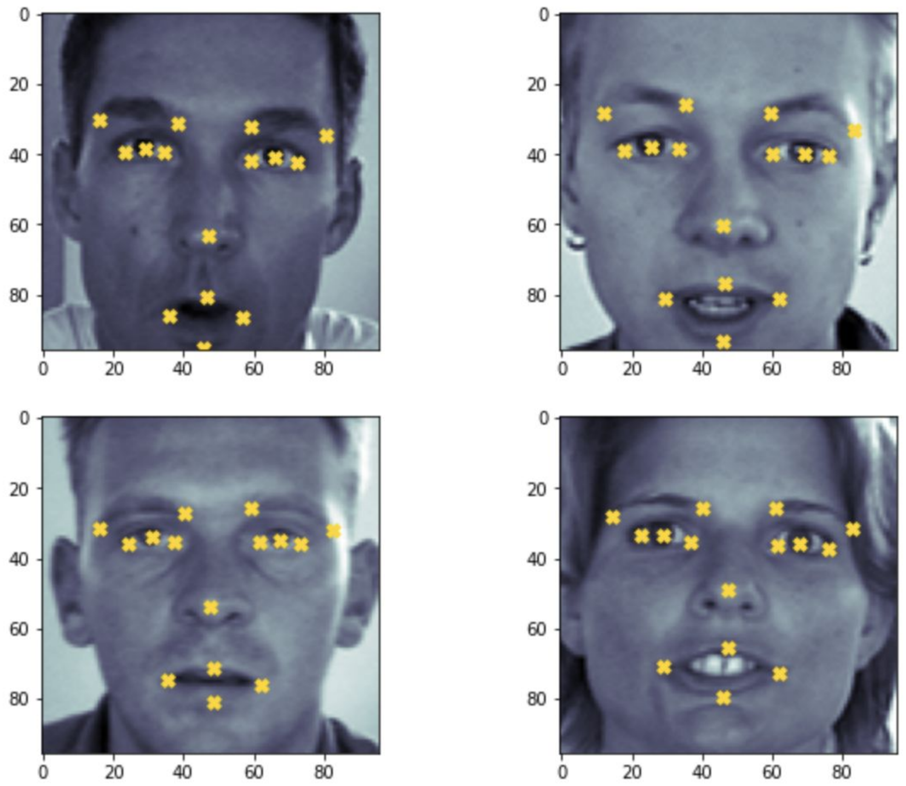
Facial keypoint detection

Jordan Meyer
Sweta Bhattacharya
Aswin Thiruvengadam
Leon Rafael Gutierrez Angulo

Objective

Detect the location of 15 keypoints on face images[1]



Practical applications:

- Tracking faces in images and video
- Analysing facial expressions
- Detecting dysmorphic facial signs for medical diagnosis
- Biometrics / face recognition

[1] https://www.kaggle.com/c/facial-keypoints-detection/

Data summary[2]

Each example contains

- Single channel 96 x 96 pixel grayscale image
- 15 keypoint features
  - 30 - X and Y locations
- 7,049 training samples
- 1,783 submission examples
- Coordinates are not available for all samples, the density of missing labels is shown on the right

Feature Pair (x , y)	Percent Missing	Dataset
right_eyebrow_inner_end	68%	A
right_eyebrow_outer_end	69%	A
left_eye_inner_corner	68%	A
left_eye_outer_corner	68%	A
right_eye_inner_corner	68%	A
right_eye_outer_corner	68%	A
left_eyebrow_inner_end	68%	A
left_eyebrow_outer_end	68%	A
mouth_left_corner	68%	A
mouth_right_corner	68%	A
mouth_center_top_lip	68%	A
mouth_center_bottom_lip	0.4%	B
right_eye_center	0.2%	B
left_eye_center	0.1%	B
nose_tip	0.0%	B

Dataset provided via Dr. Yoshua Bengio, University of Montreal

Characteristics

Search for unique images

Methodology:

- Fit & Transform PCA with 50 components
- Run DBSCAN clustering with minimum 100 samples per group
- Analyze ungroupable images → 3% of training data

Issues Addressed:

1. Angled faces
2. Cartoons
3. Portraits
4. Off-center

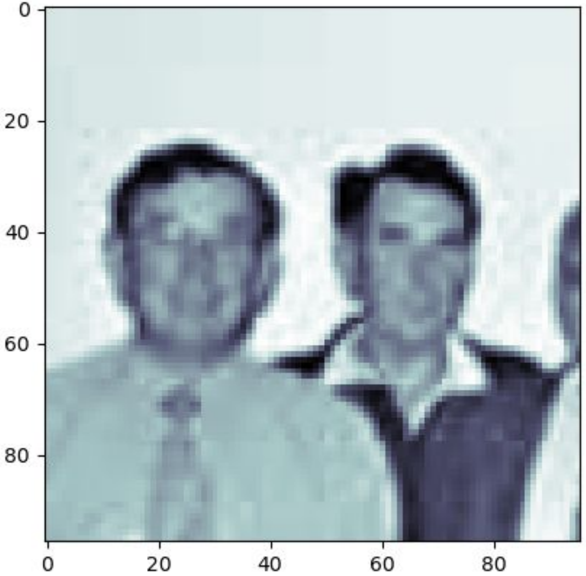
Future Opportunities:

5. Props
6. Mislabelled

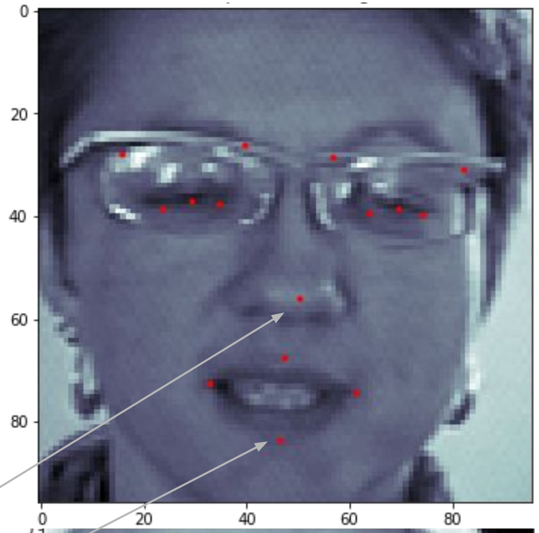
Characteristics

continued...

Which face to landmark?

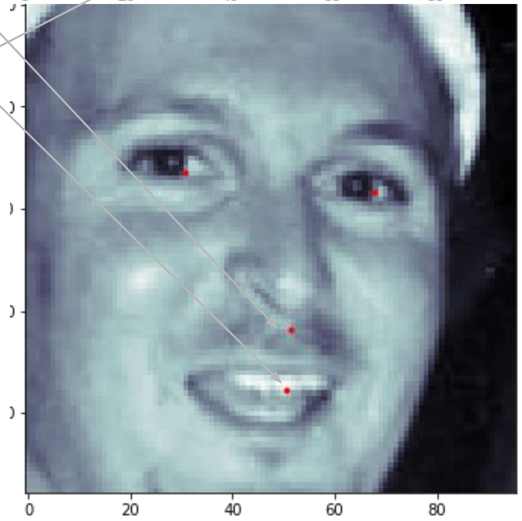


Definitions of some landmarks are different between datasets



“nose tip”

“mouth center bottom lip”

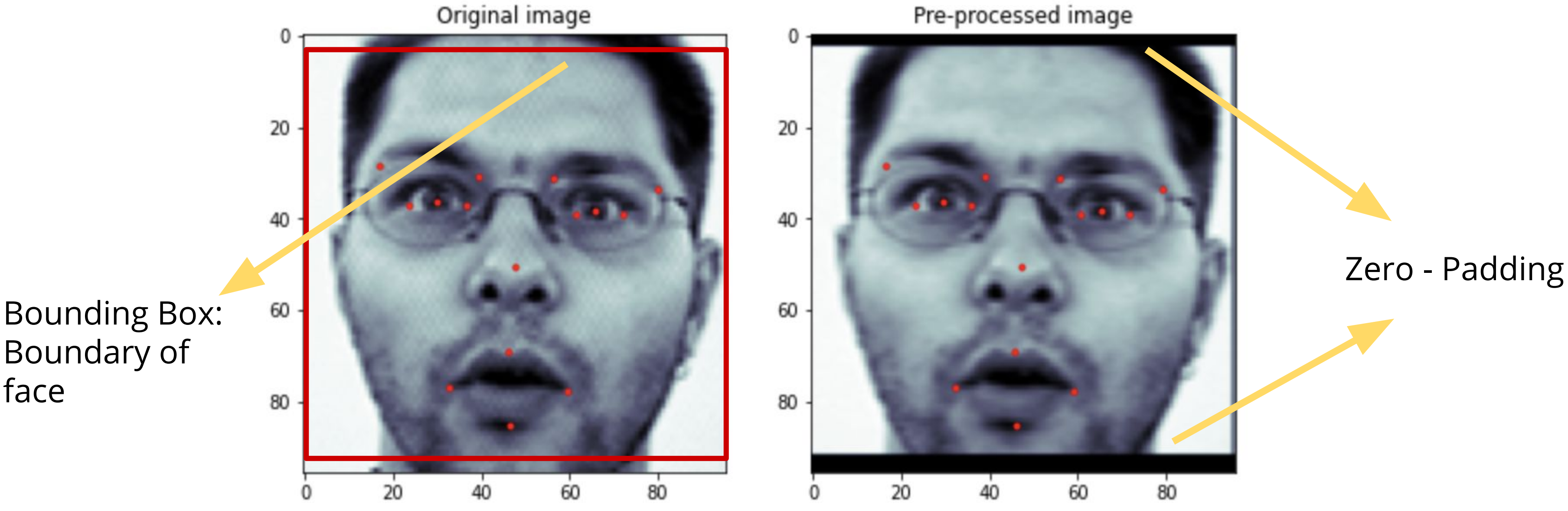


Is there a correct one?

Preprocessing

Face detection and centering

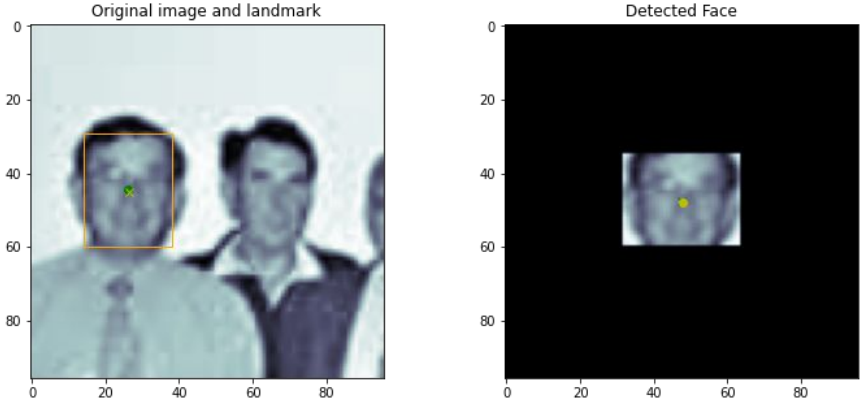
- Used MTCNN package to detect faces
- Generate and Crop bounding box
- Center image & pad



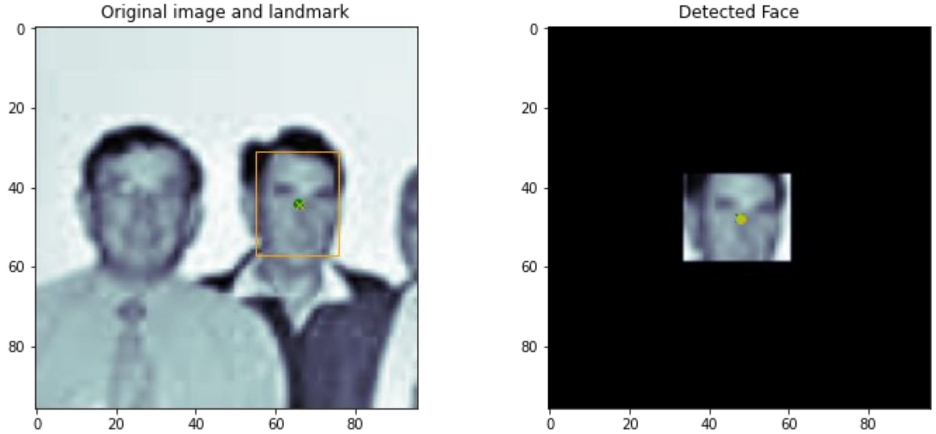
Preprocessing Challenges

- Parameters:
  - Resize cropped image?
  - Which image?
    - Largest? Closest to center?

Example 2: Largest area, unscaled



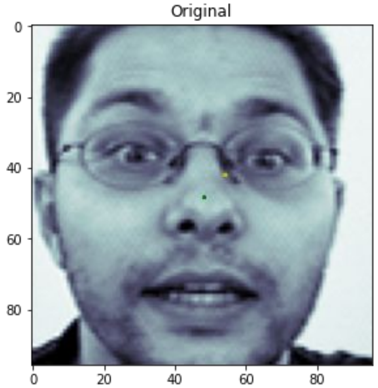
Example 1: Closest to center, unscaled



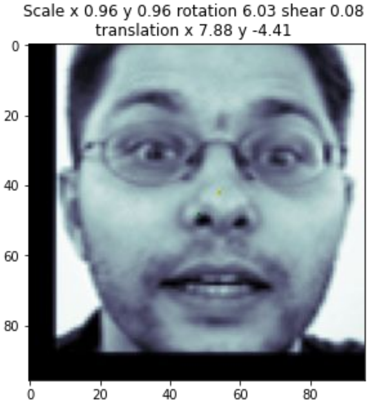
Example 3: Largest area, scaled

Augmentation

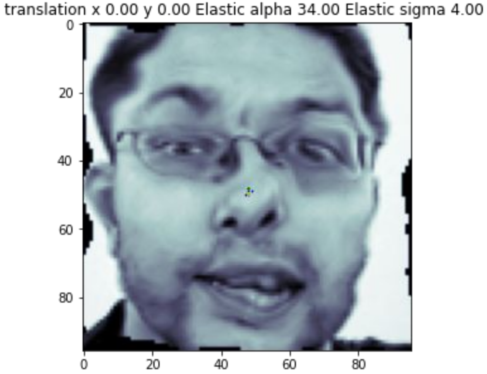
Transforming images



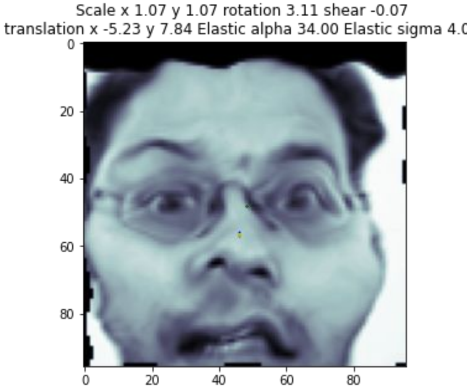
Affine



Elastic



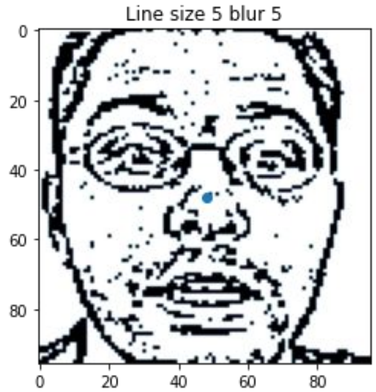
Both



To augment original images, transformations of the images were added to increase the sample size of training. The transformations that were included were:

1. Affine(shift, skew, rotate - preserve relationships)
2. Elastic (deform)[3]
3. Edge detection

Edge



[3] *"Best Practices for Convolutional Neural Networks applied to Visual Document Analysis", in Proc. of the International Conference on Document Analysis and Recognition, 2003.*

Augmentation

Benefit to model

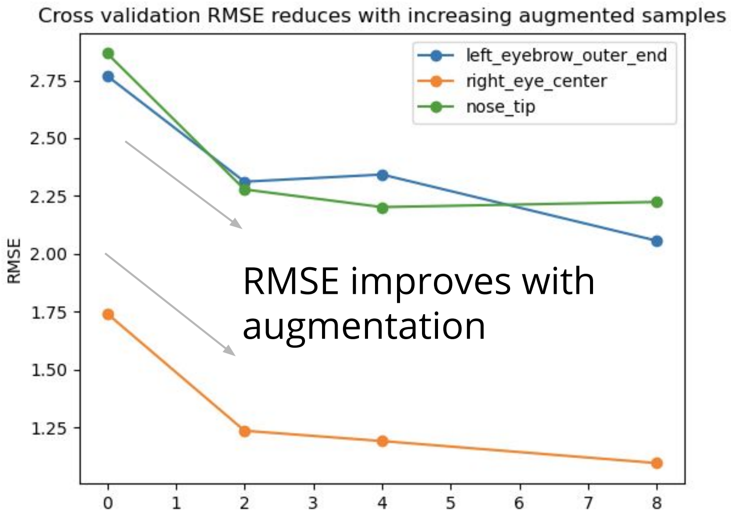
Cross validation RMSE reduces with increasing augmentation images

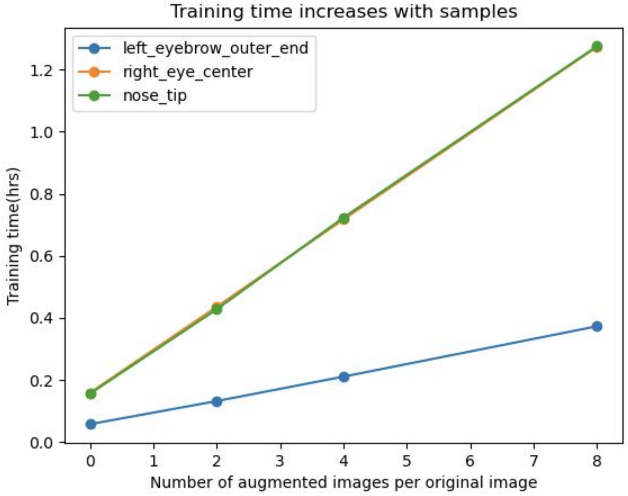
Some models benefit more than others

Training time increases with increasing samples, so we made a tradeoff of choosing

- 12 to 16 augmented samples/image for labels with low data points (Dataset A)
- 8 augmented samples/image for labels with high data points (Dataset B)

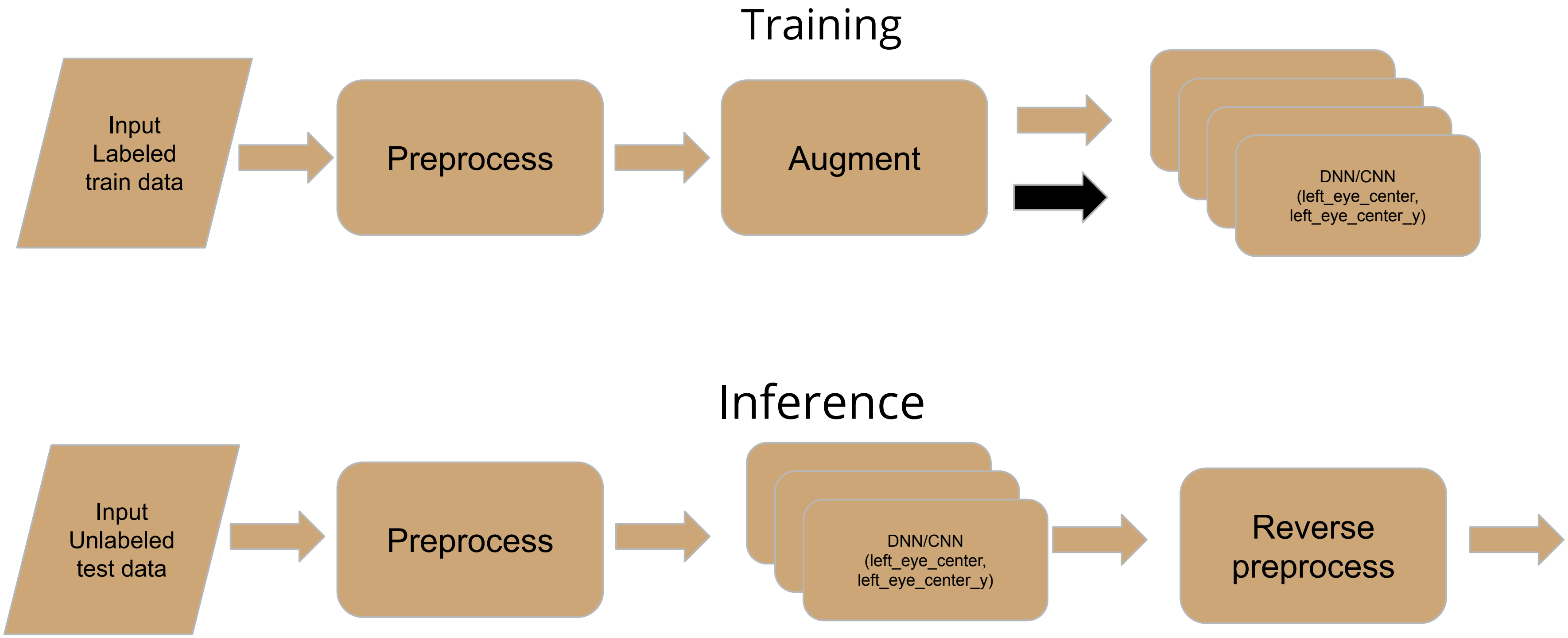
~training time 1.25hrs per model (15 models)





Modeling Architecture

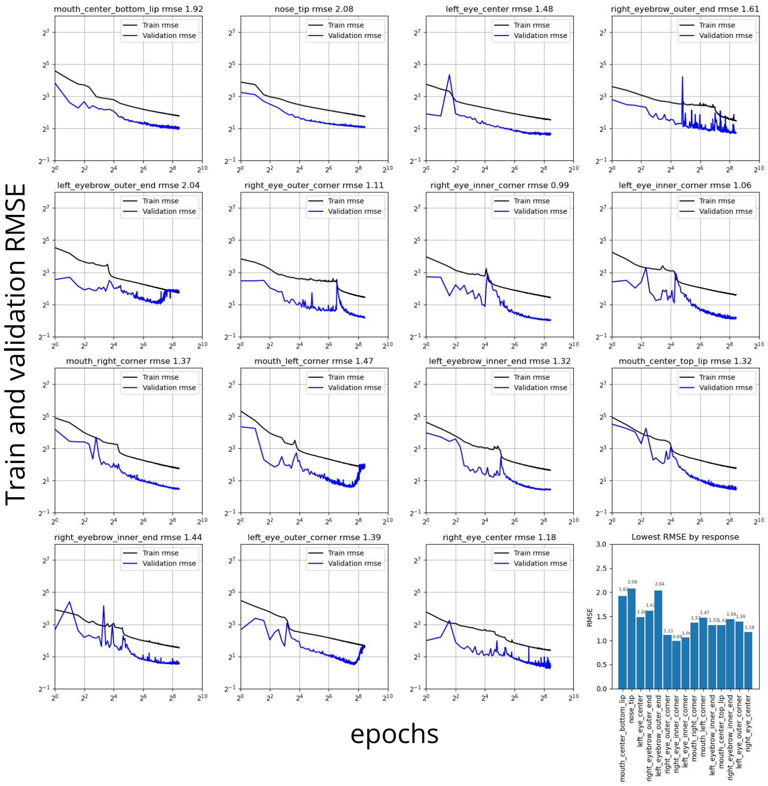
Cascade of modules

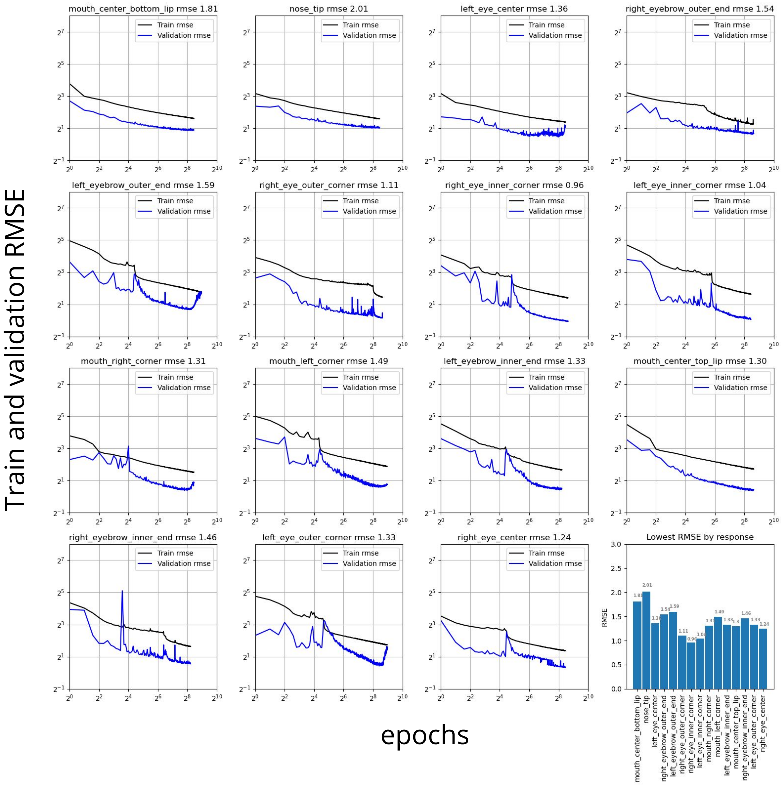


Modeling Architecture

Dashboard

Model learning plots

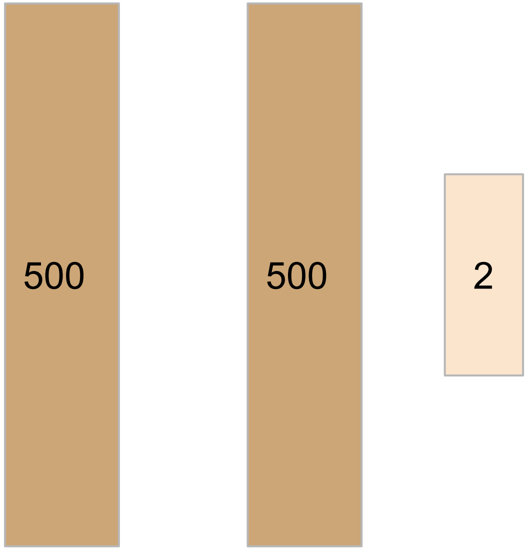




Model structure

DNN (300X300) densely connected

RELU activation for all layers



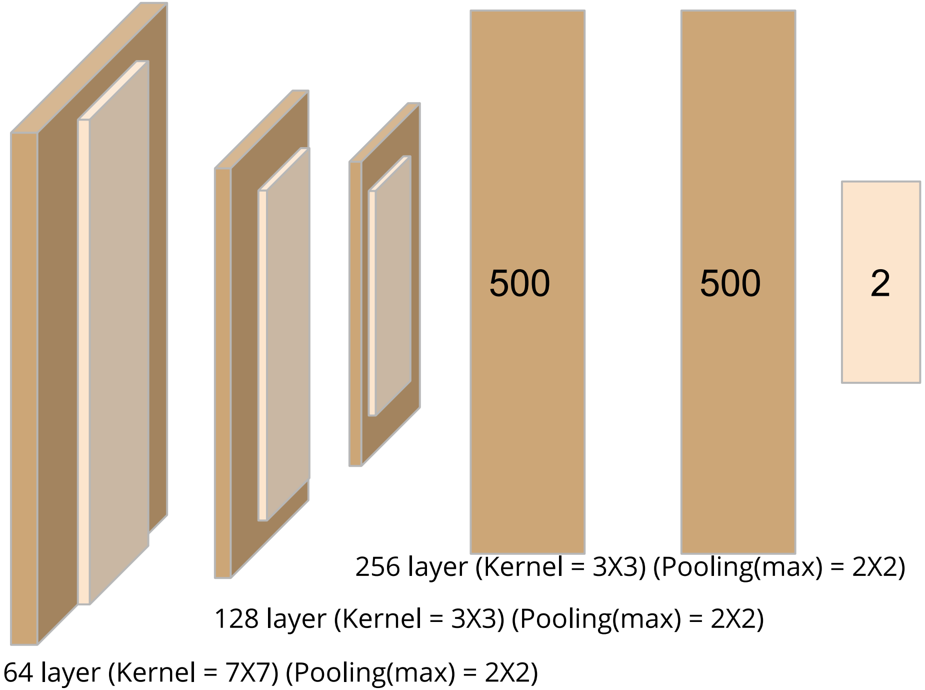
~10 min/model = ~2hrs for all models

CNN*

*Hands on Machine learning with Scikit-Learn, Keras & Tensorflow
Aurelien Geron (O'Reilly)

500 X 500 Dense layers with ReLU activation

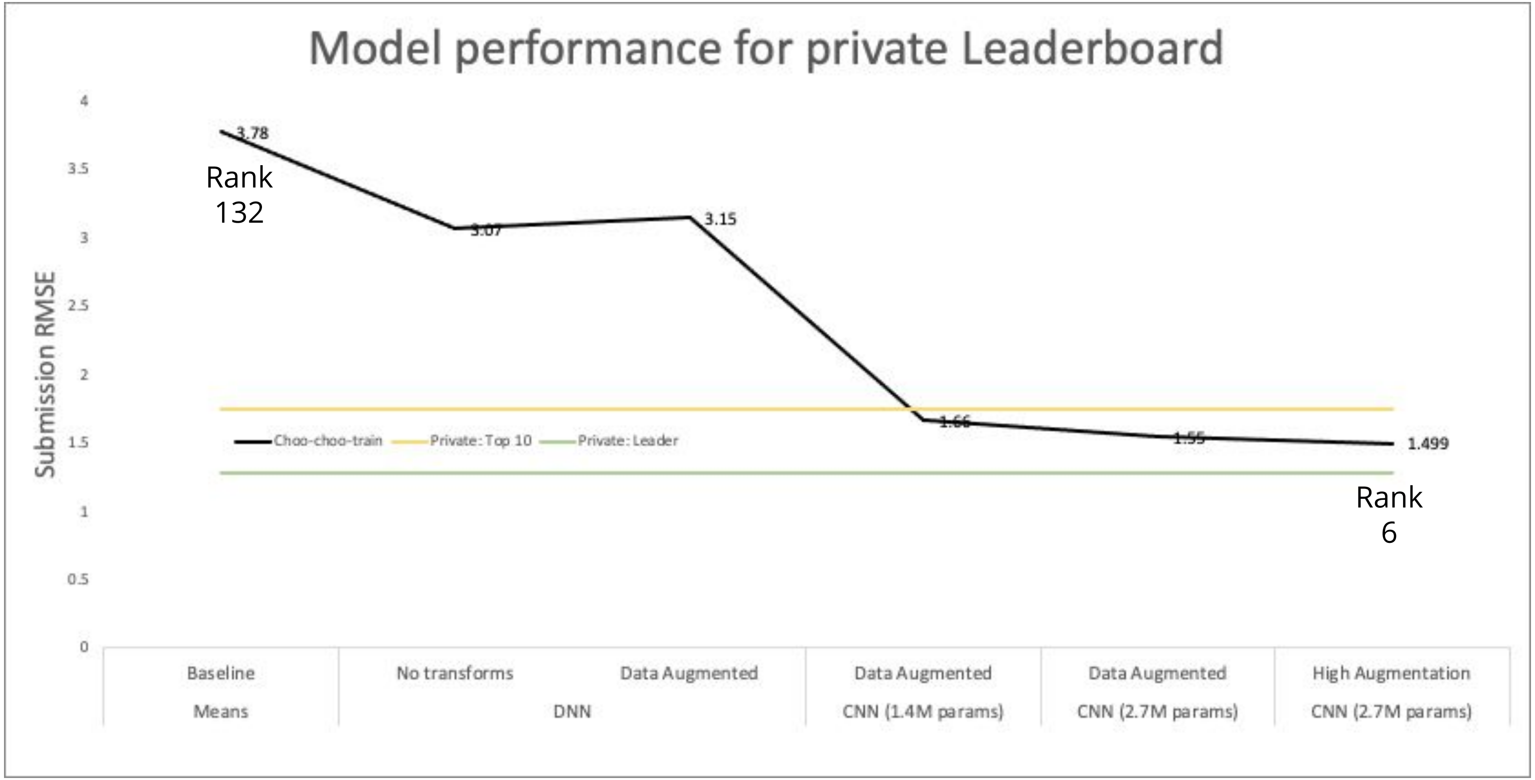
Convolutional pairs have a ReLu and a linear layer



~1 hrs/model = ~15 hrs for all models

Model performance

Through increasing complexity



Learning and experimentation

- Dense vs Convolution
- Semi-supervised Learning
- Pre-trained Models
- Augmentation
- Coordinate Auto-encoder (post-processing)
- Image Auto-encoder (pre-processing)
- PCA and Dimensionality Reduction

Thank you!

Backup

Agenda

Objective

Exploratory data analysis

- Description
- Characteristics

Data Augmentation

Data preprocessing

Modeling Architecture

Model Structure

Model Performance

Under evaluation

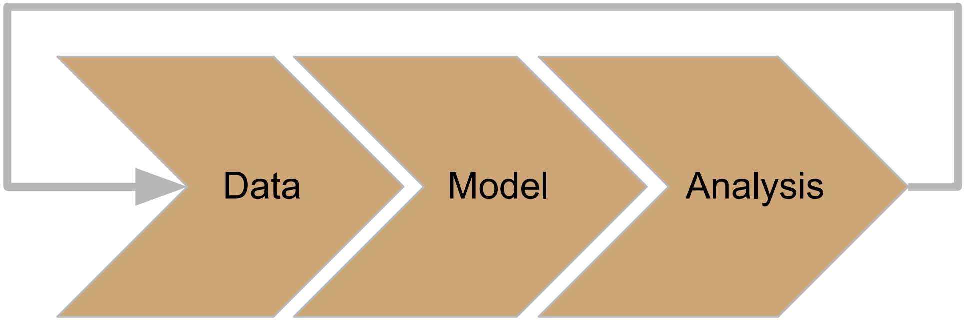
Improvements

Modeling strategy

1. Baseline Model

2. Cascade model
   a. Phase 0: OLS
   b. Phase 1: DNN
   c. Phase 2: CNN
   d. Phase 3: CNN with transfer learning

Loss: RMSE



Data

Understanding data characteristics

Augmenting data with more samples

Creating more samples for misclassified images

Model

Comparison of architectures

Model architecture optimization

Hyperparameter optimization

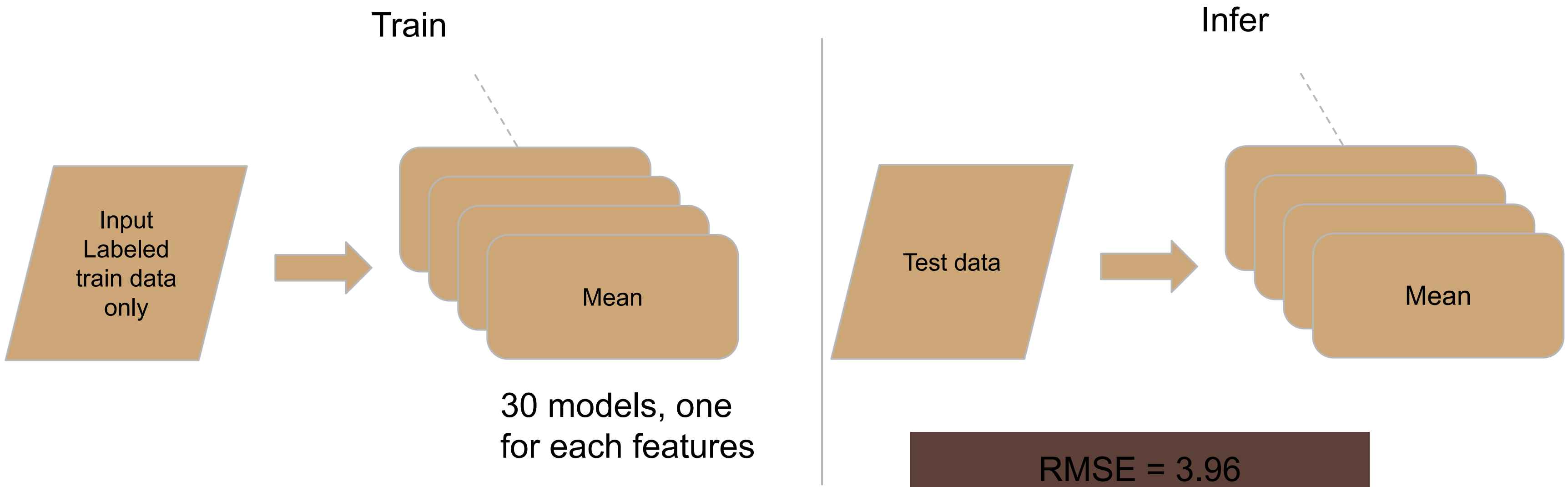
Model dashboard

Analysis

Visualizing misclassifications

Root-cause analysis

Baseline - Location Means



RMSE = 3.96

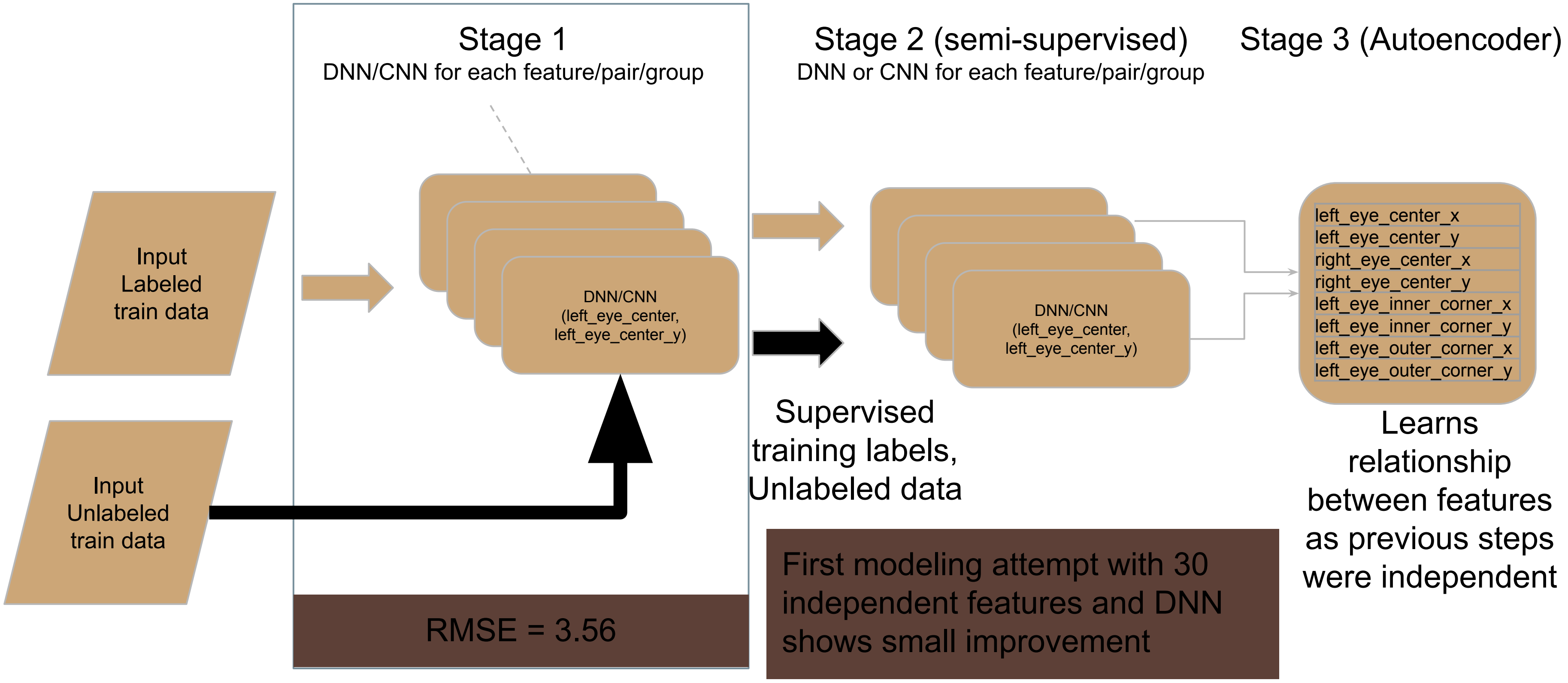
Your most recent submission

Name	Submitted	Wait time	Execution time	Score
MySubmission.csv	just now	1 seconds	1 seconds	3.96243

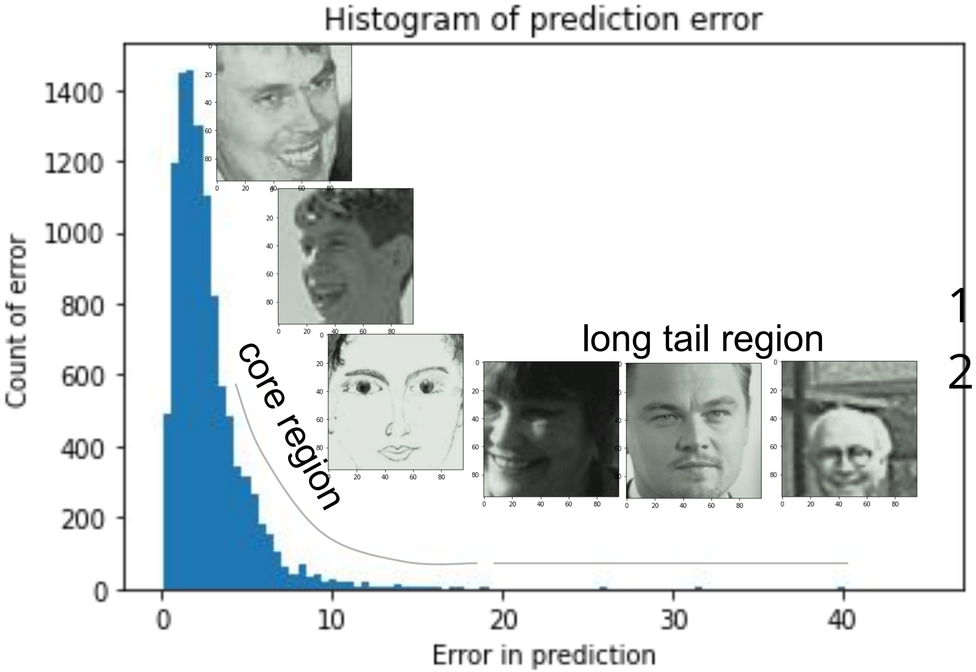
Complete

Jump to your position on the leaderboard

Cascade model

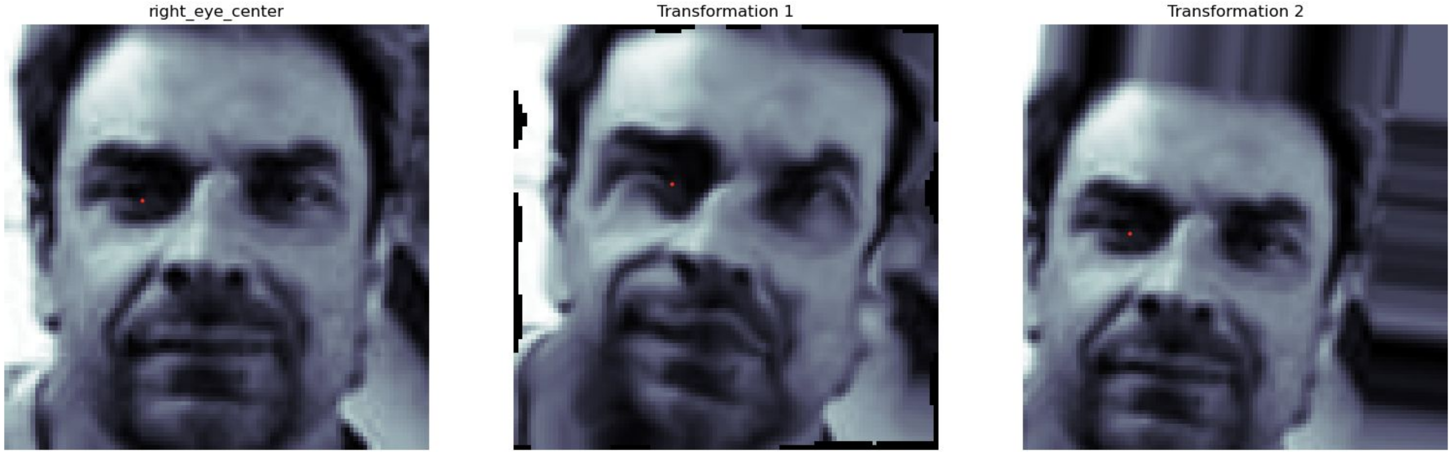


Analysis of predictions (Stage 1: DNN)



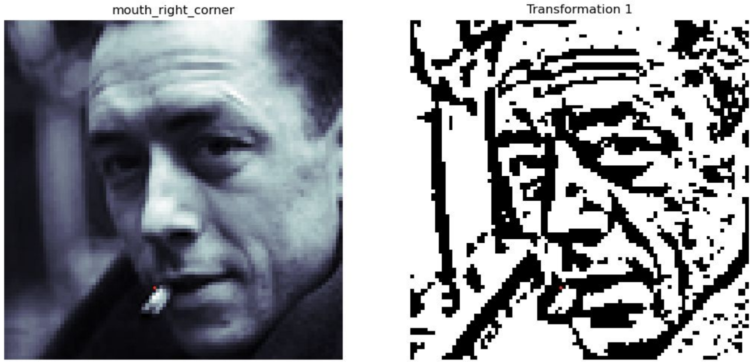
1. Long tail region
2. Core region

Data Augmentation



To augment original images, transformations of the images were added to increase the sample size of training. The transformations that were included were:

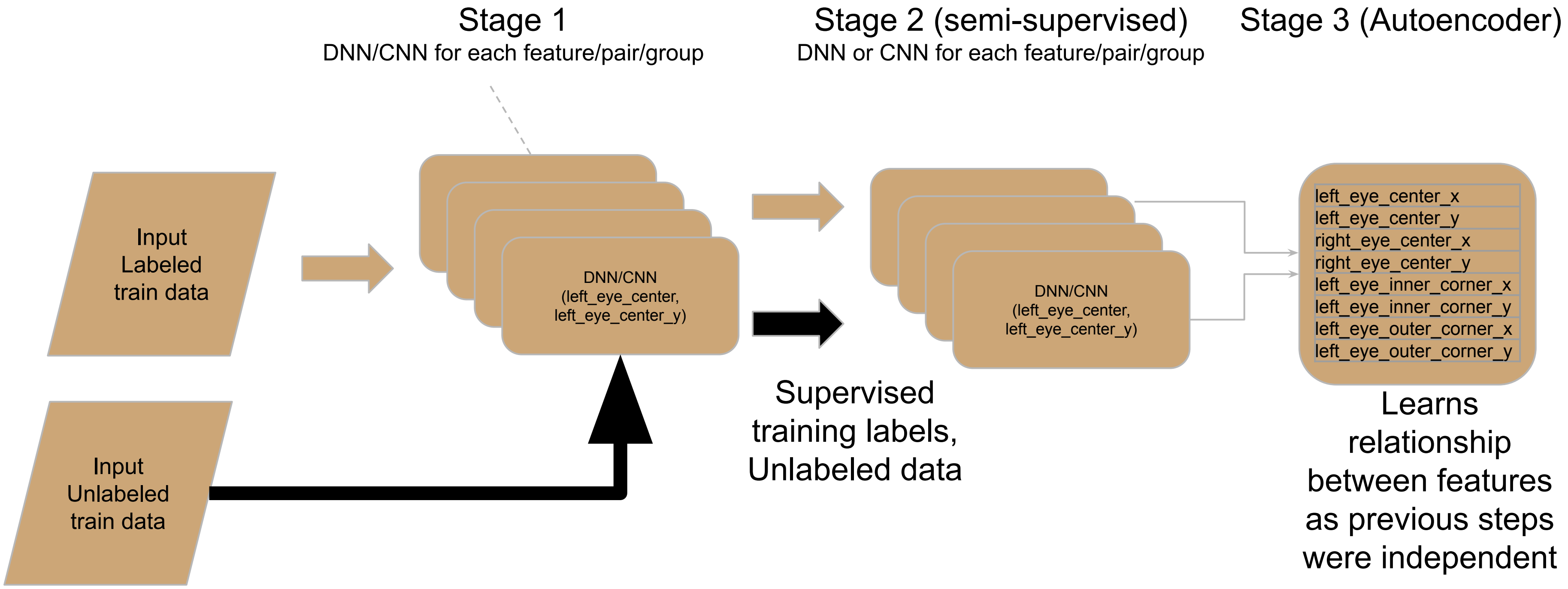
1. Affine(shift, skew, rotate - preserve relationships)
2. Elastic (deform)*
3. Cartoon transformation



*"Best Practices for Convolutional Neural Networks applied to Visual Document Analysis", in Proc. of the International Conference on Document Analysis and Recognition, 2003.

Model architecture

Cascade of models



Data Augmentation

- Model performance improves with increased training data
- Challenge: Limited training data w.r.t. number of features to train.
- Solution: Create new training examples via data augmentation

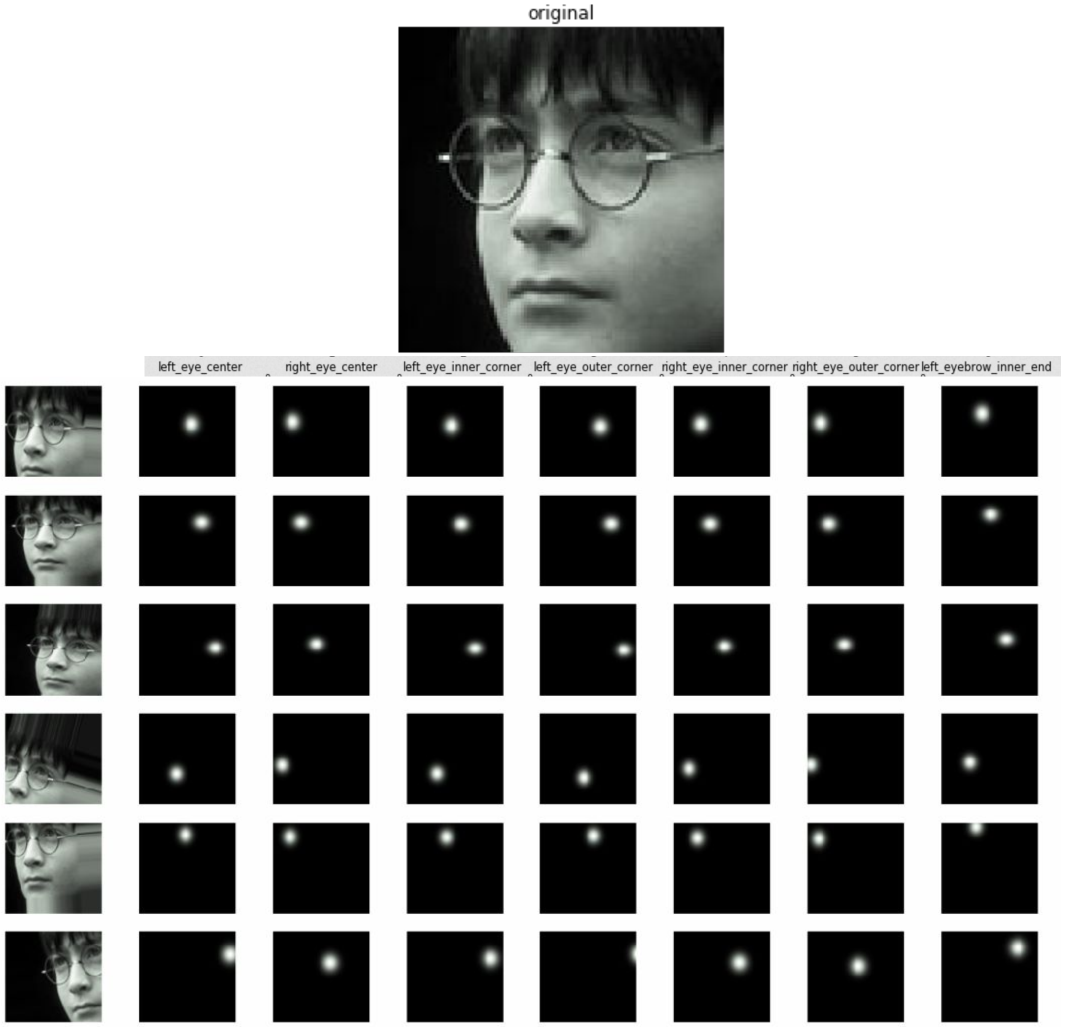
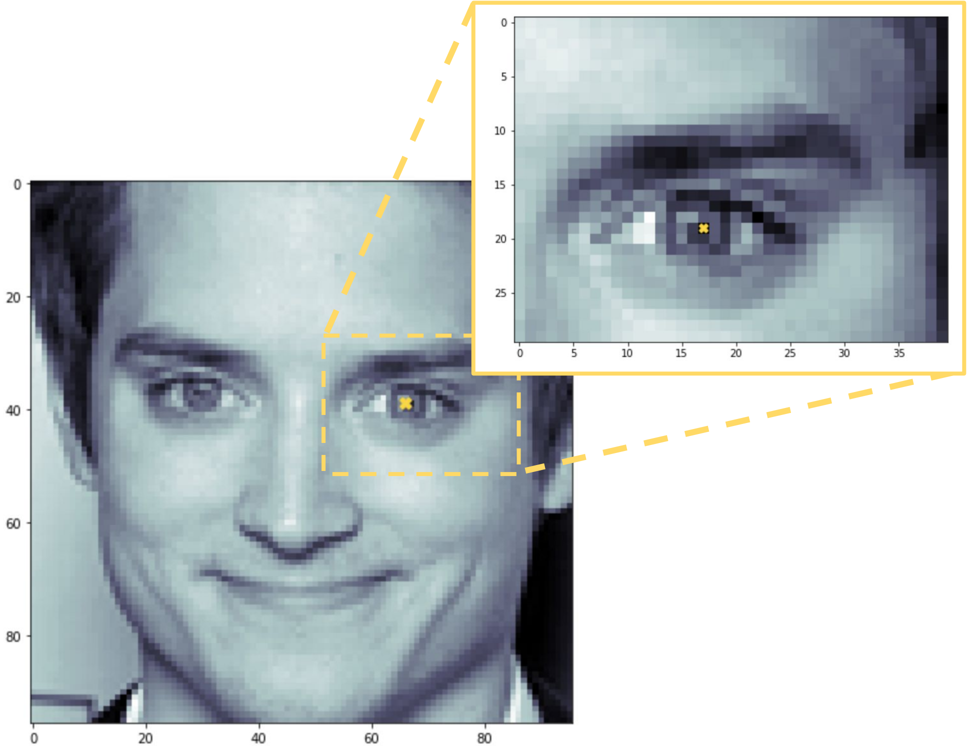


Image Source: https://fairyonice.github.io/Achieving-top-5-in-Kaggles-facial-ke

Image Preprocessing

- Models perform better with less noise and features
- Challenge: Lots of unnecessary features assessed by each model
- Solution: Crop Images for each feature to reduce number of features to train



Transfer Learning

- Model Parameters are Critical
- Challenge: Training each CNN is very time consuming
- Solution: Leverage pretrained existing models via transfer learning

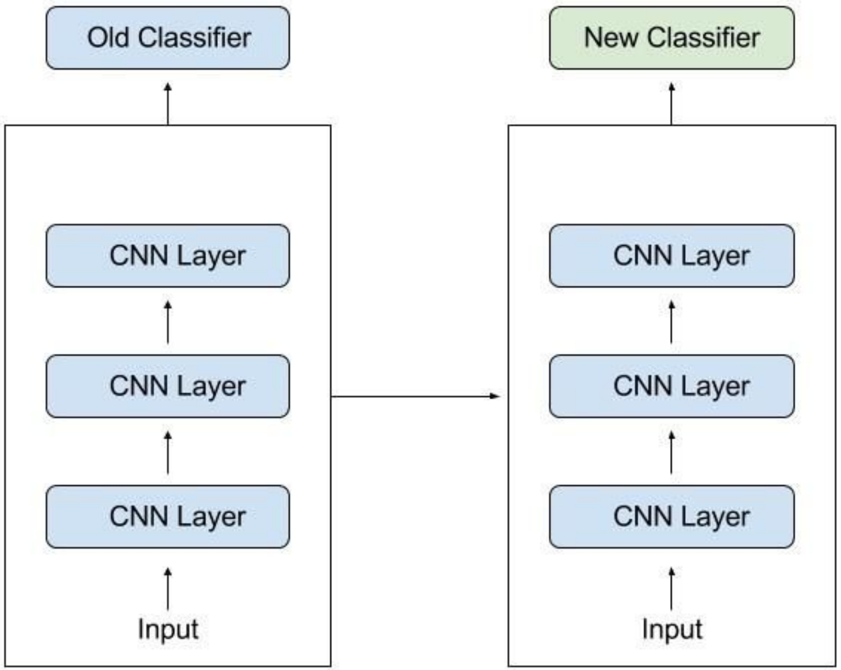


Image Source: https://builtin.com/data-science/transfer-learning

Tasks

Data exploration

- Sweta and Leon to look at

Operating model

- Final submission in notebook
  - Only add final(or working) version to the main submission notebook
- Create functions in your own notebook or import as a py file
- Store all work in Git (including your own work-in-progress code too)

Model

Model dashboard

- Learning curves

Model architecture and hyperparameter optimization

- Genetic algo optimization?

Transfer learning

Batch normalization

Dropout

Reference

Augmentation

- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0
- https://fairyonice.github.io/Data-augmentation-for-facial-keypoint-detection.html
-

- GANN for increasing training data on misclassified samples
-