

# Face Keypoints Detection (W207 Final Project)

Veronika Nuretdinova

MIDS 2018

# Face Keypoints Detection: task and dataset

**Task:** Detect coordinates of 15 face keypoints on each image

**Evaluation criteria:**

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

**Train Images (2K)**



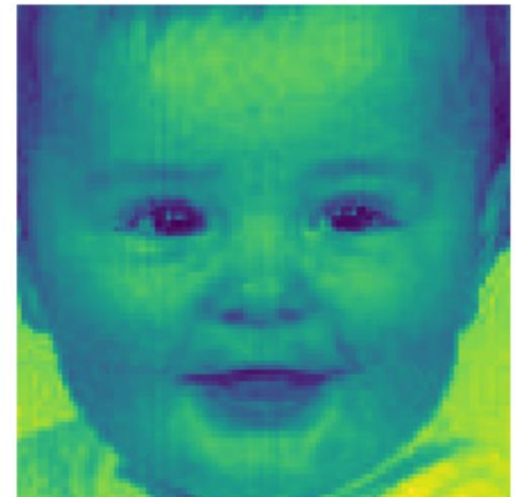
**Incomplete Train Images (5K)**



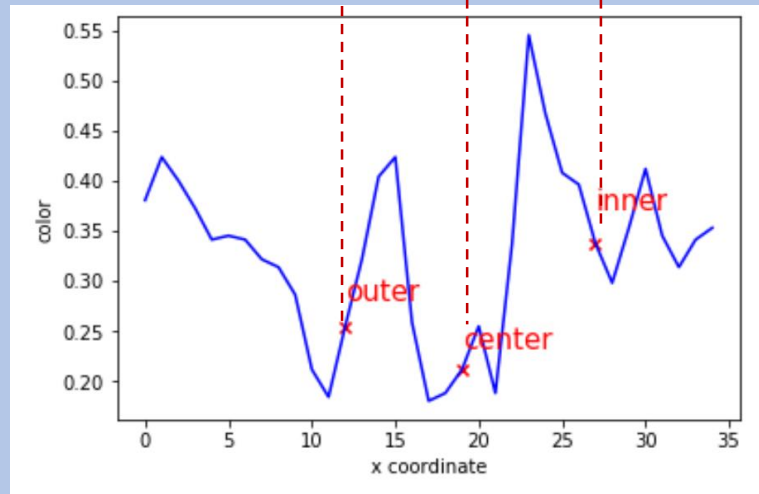
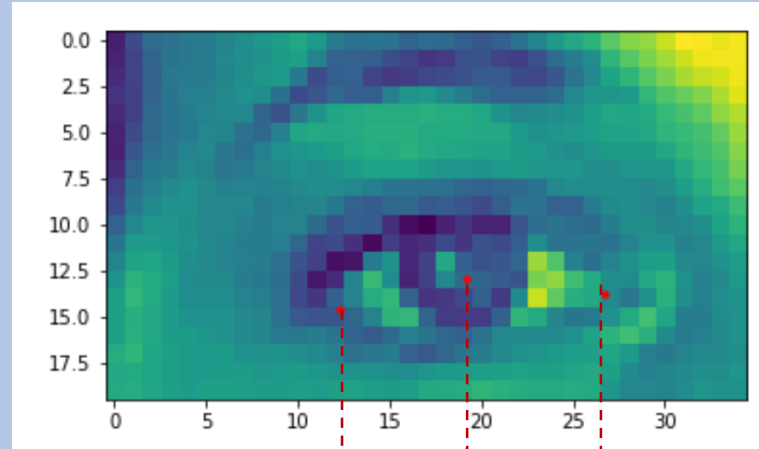
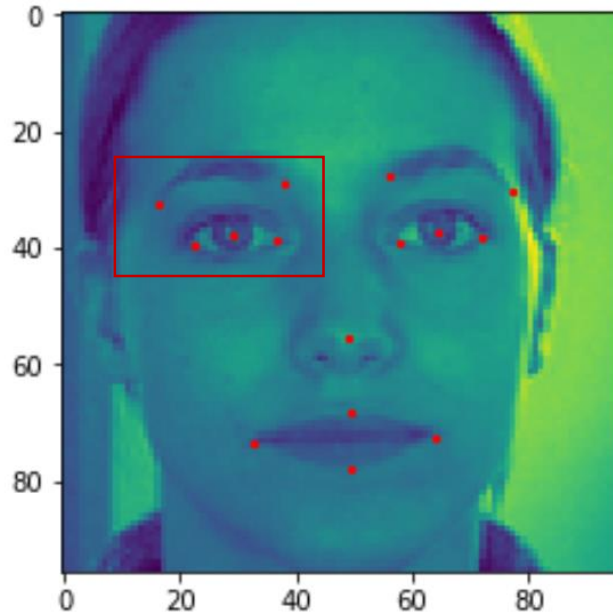
**Evaluation Images (5K)**



**Test Images (1.7K)**



# Creating features for keypoint detection



## Key features selected:

- Coordinates of the key point
- Color of the point
- Change of the color compared to the neighboring keypoint
- Is it near the extremum where the color radically changes

# First try: Random Forest

Train dataset  
(9K 96X96  
images)



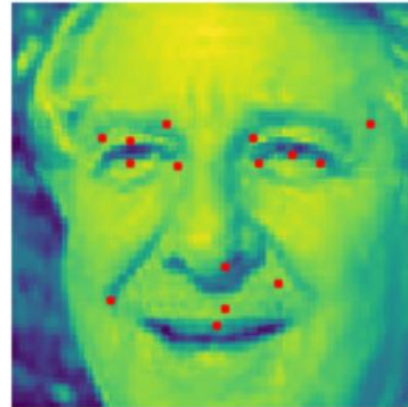
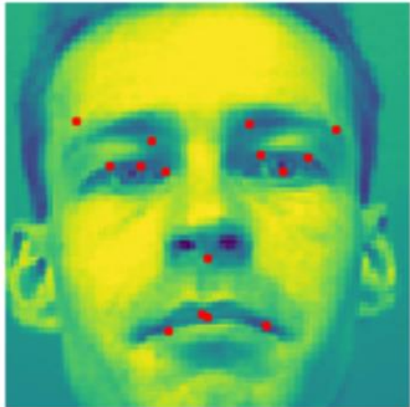
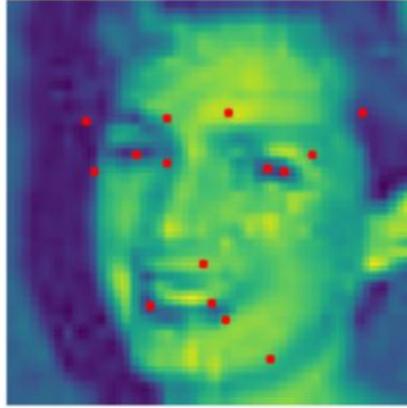
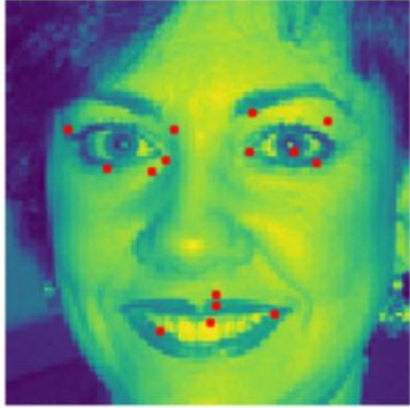
		Coordinates			Color change at the point		Is it at the extremum		Color	Keypoint?
	facepoint	image	x_index	y_index	x_delta	y_delta	x_change	y_change	value	target
0	left_eye_inner_corner	0	0	0	0.031373	0.031373	1	1	0.521569	0
1	left_eye_inner_corner	0	1	0	0.117647	0.074510	1	1	0.498039	0
2	left_eye_inner_corner	0	2	0	0.019608	-0.011765	1	1	0.454902	0
3	left_eye_inner_corner	0	3	0	-0.023529	0.031373	1	1	0.435294	0
4	left_eye_inner_corner	0	4	0	-0.015686	-0.066667	1	1	0.435294	0

- 8 features + target
- 9K images\* 96\*96 pixels \*15 features = 15.7 M observations

## Suggested Machine Learning approaches:

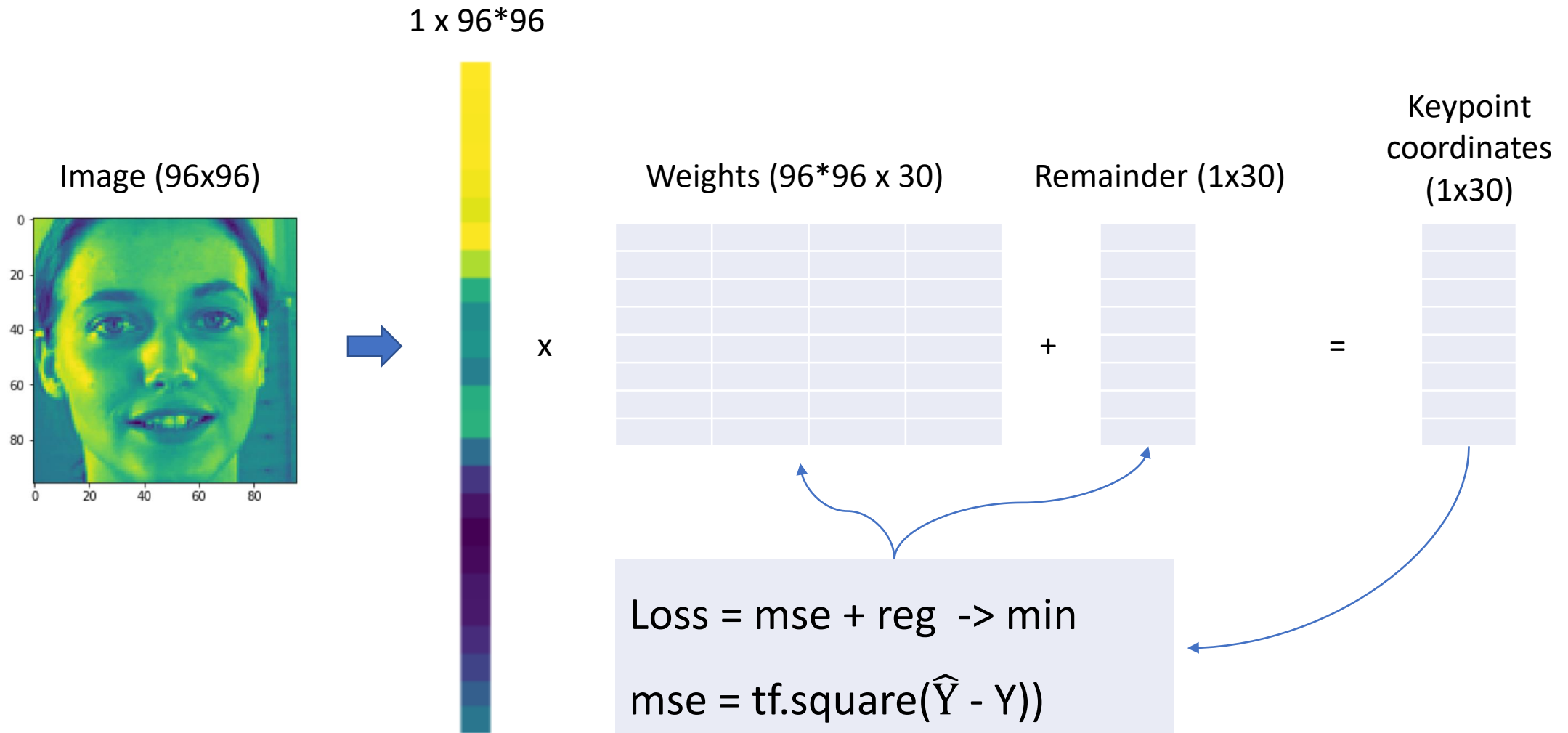
- Probability methods (Naïve Bayes, Bernoulli) to estimate the keypoint probability for each pixel
- Tree methods to find out feature combinations to detect the keypoints

# Random Forest method result: very approximate detection



- RMSE = 3.57
- Kaggle public score = 4.98

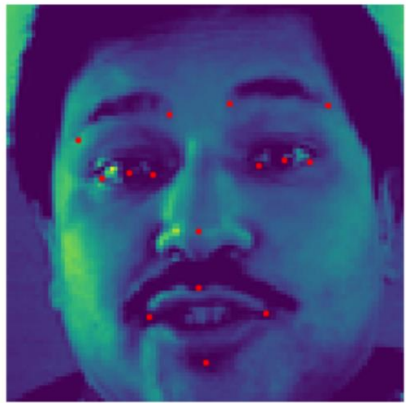
# Second try: Neural net using Tensorflow



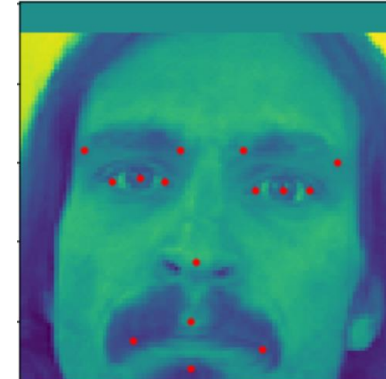
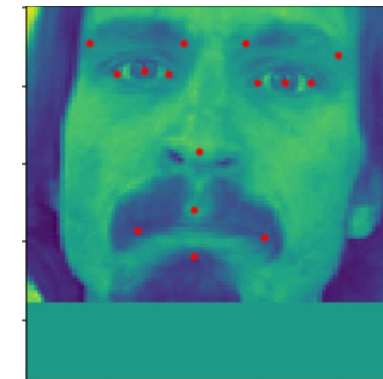
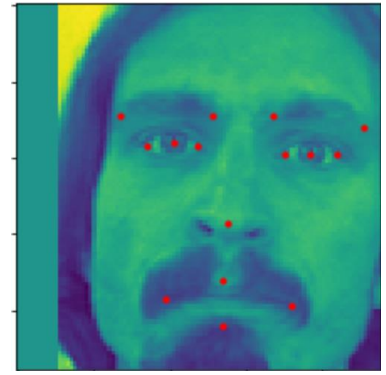
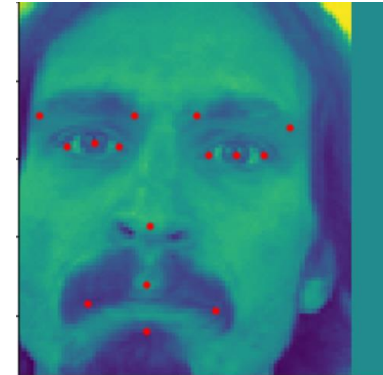
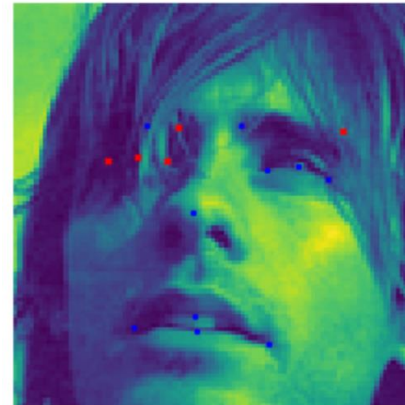
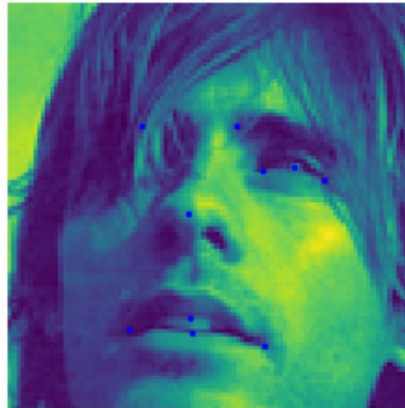


# Expanding the number of available observations by 23K images

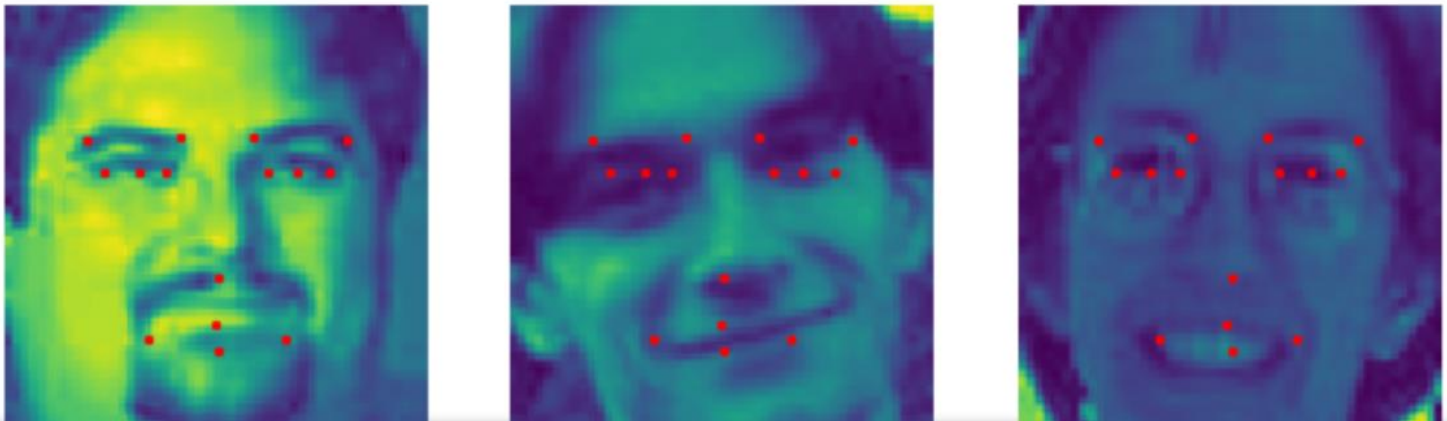
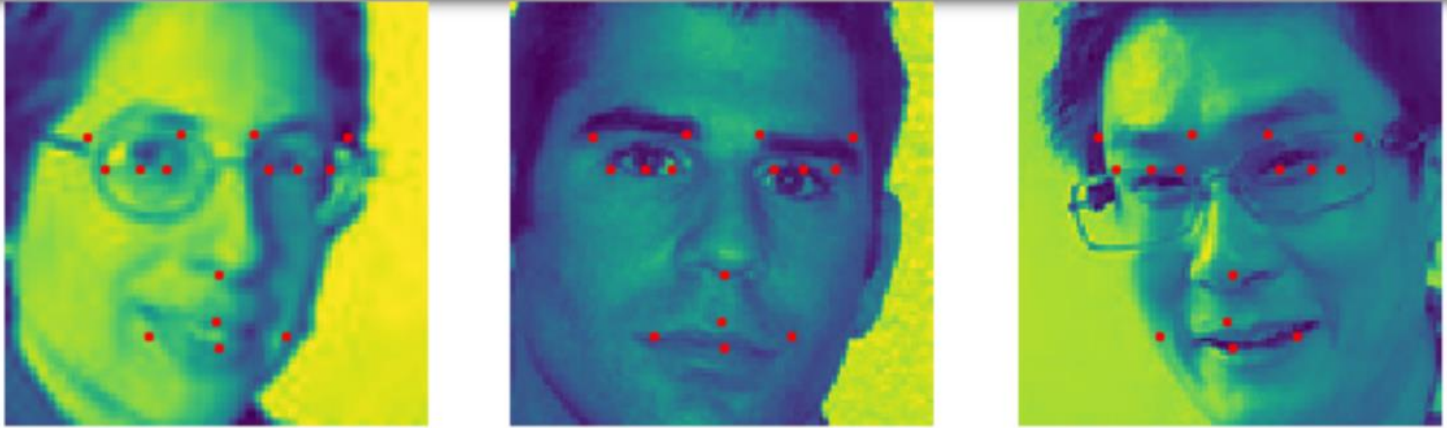
- Flipping (+2k)
- Complete incomplete observations (+5k)
- Shift images in 4 directions (+16K)



Run first TF model based on 4K images and apply it to incomplete images



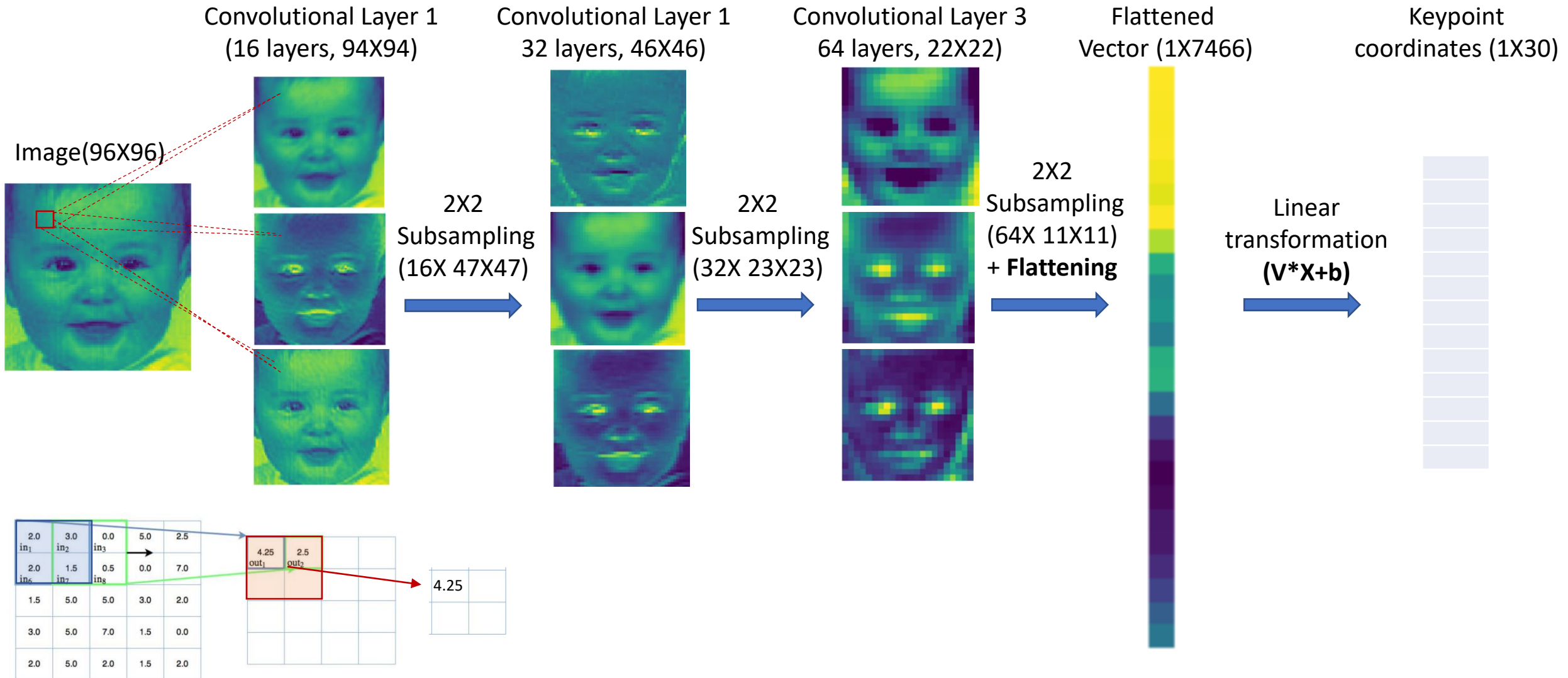
1-layer neural net result: better result. Poorer performance on inclined faces



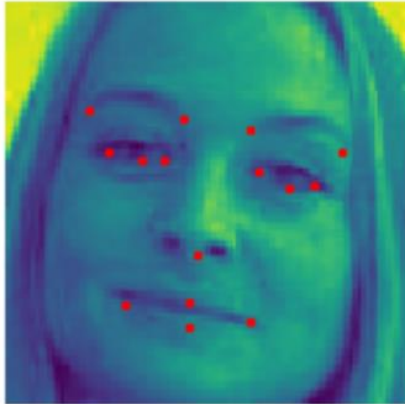
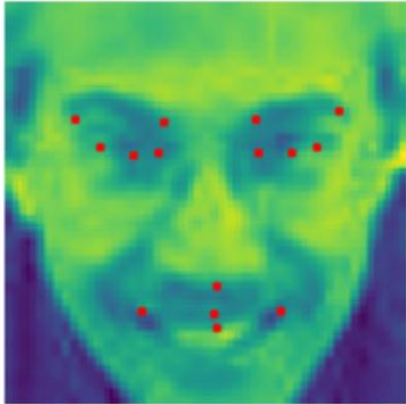
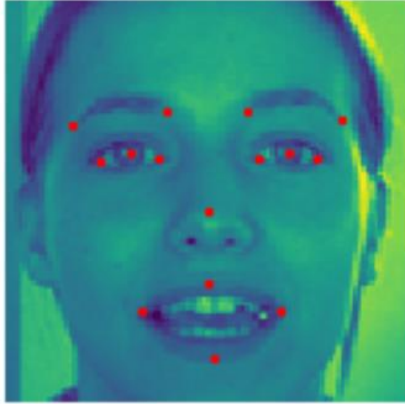
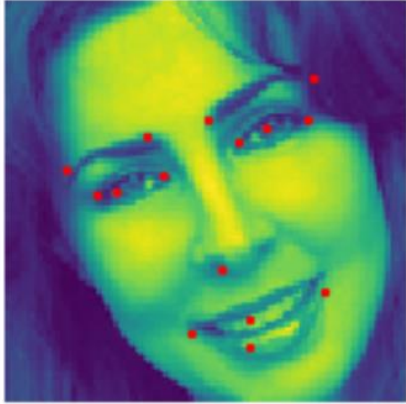
- RMSE = 2.91
- Kaggle public score = 3.98



# Third try: deep learning through convolutional neural networks (CNN)

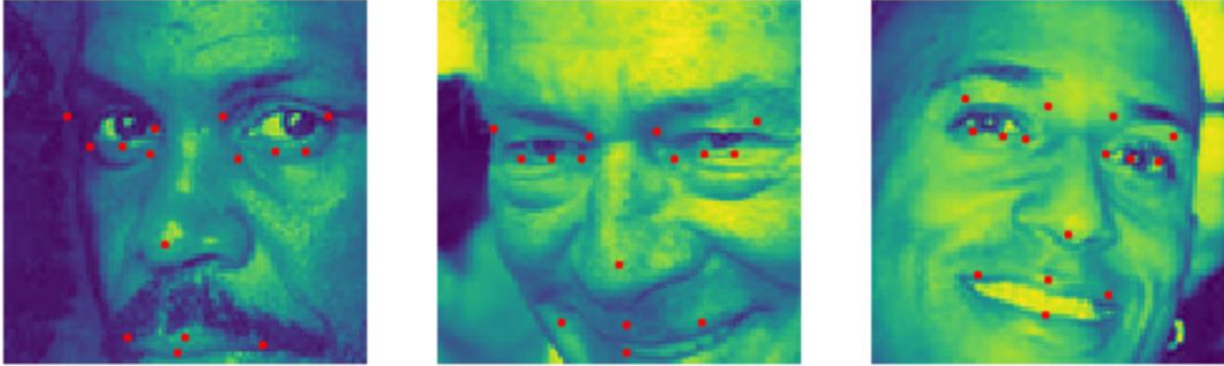


# Convolutional neural net result. Works with inclined faces

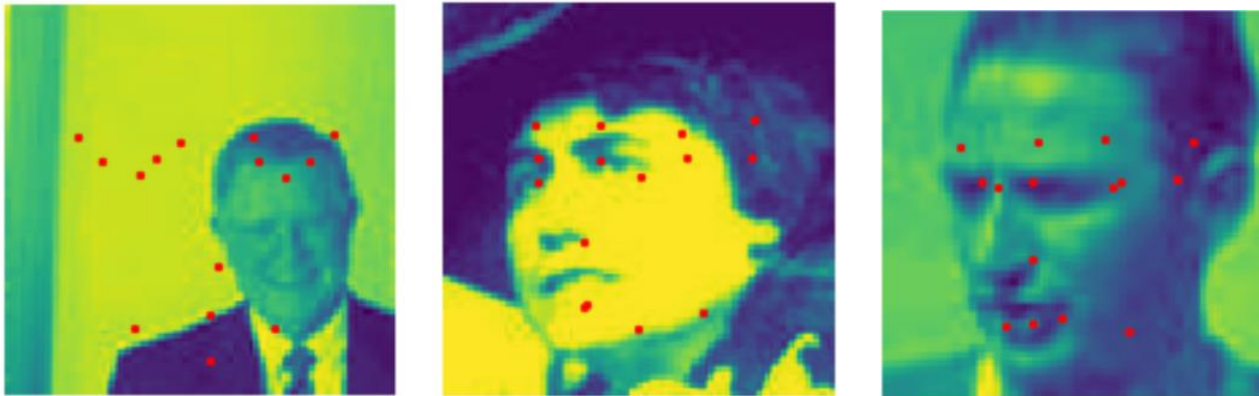


- RMSE = 2.28
- Kaggle public score = 2.45

# How I would continue to improve the results



- Expand and shift images (to train the model on “non-full” and non-centered faces)



- Squeeze and shift by diagonal (to train the model on “small” and non-centered faces)