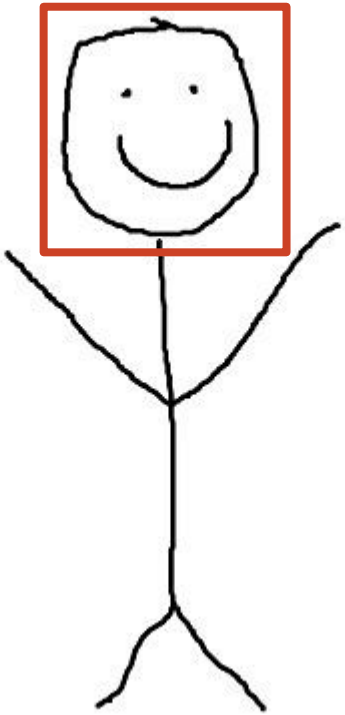
Face Detection with End-to-End Integration of a ConvNet and a 3D Model



Harish Pullagurla
Graduate Student , ECE ,NCSU

Guided by - Dr Tianfu Wu

What is this project about ?

Objective

This project is an attempt to implement the paper

"*Face Detection with End-to-End Integration of a ConvNet and a 3D Model* "

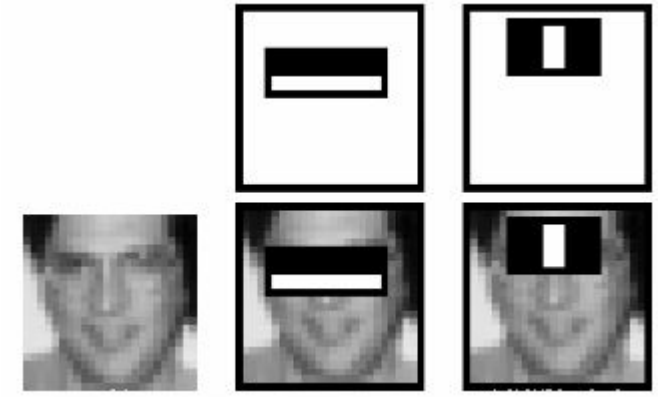
Yunzhun Li et.al

Primary aim is to detect faces given any image.

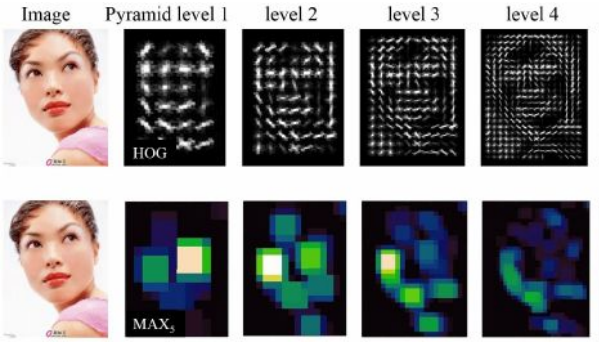
Reference code was available, but couldn't be executed because of change in Framework.

Hence, the project was reprogrammed.

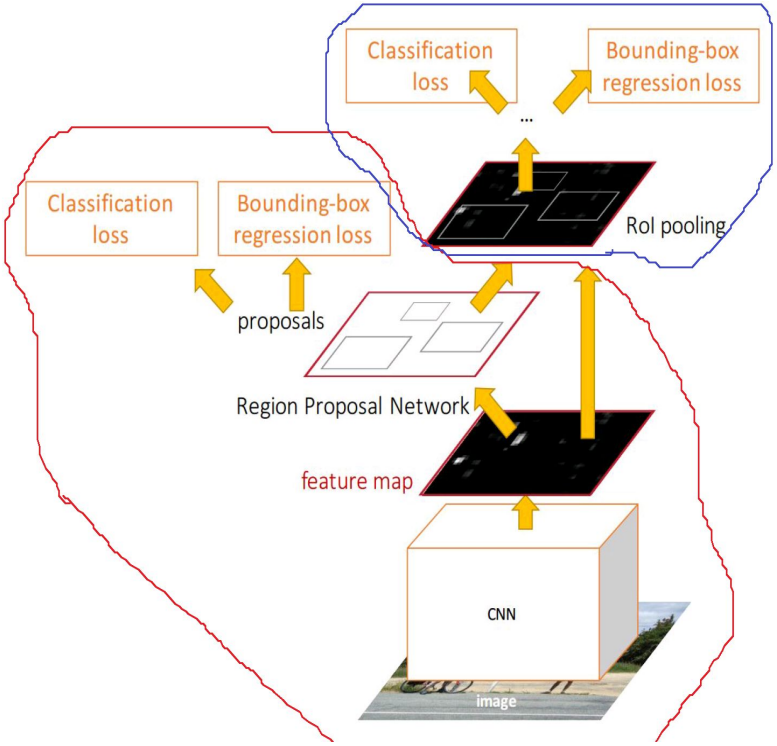
Related Work



Viola Jones - Adaboost of Haar Classifiers



DPM and HOG feature based Methods

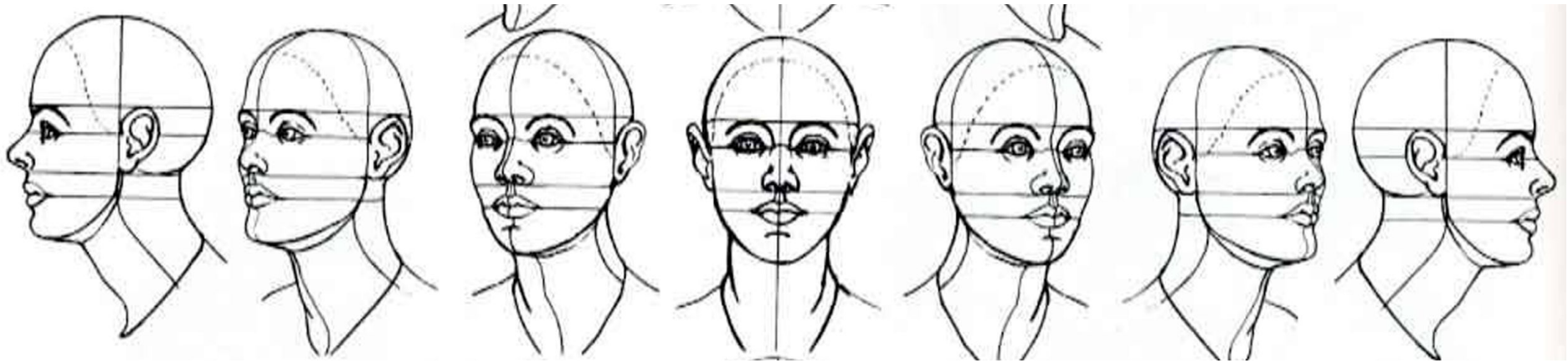


Faster RCNN Architecture

Lots of Face Detection methods already exist !
How is this one different ?

Introduction

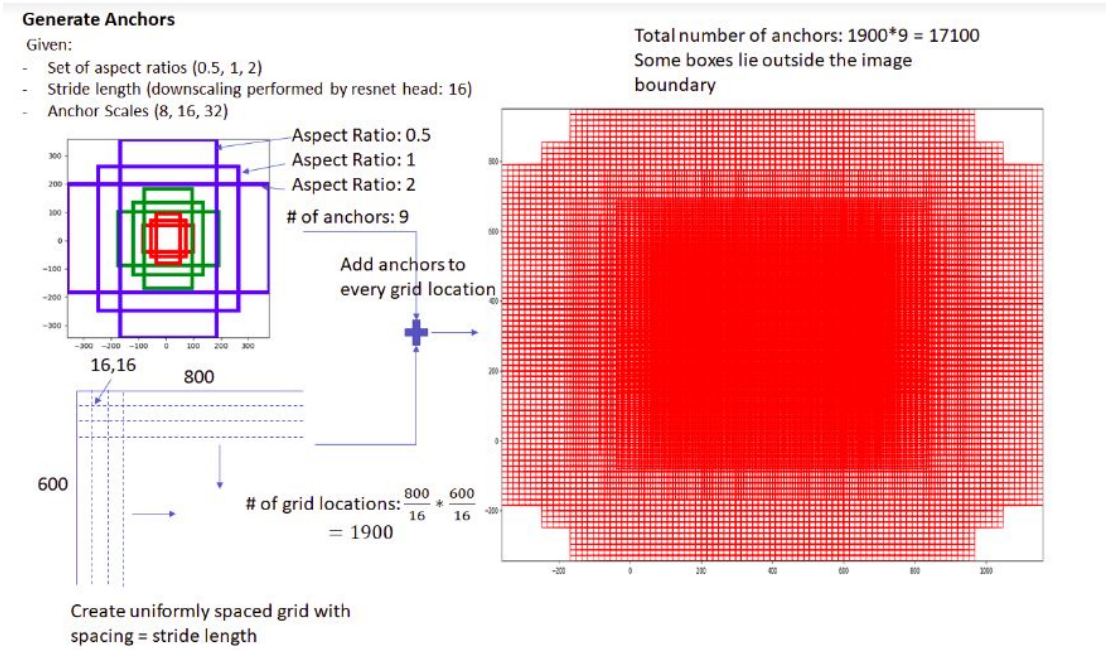
- An attempt is made to give reasoning to the outputs generated by the networks
- Considers each face as a projection from the 3D mean Face model
- Proposes an end to end, multi task learning framework, integrating 3D models into it.



Introduction

Extends the existing Faster RCNN architecture, with 2 key modifications

1. New Region Proposal Network
   a. Faster RCNN uses pre configured Anchor boxes as input to the network



Introduction

Extends the existing Faster RCNN architecture, with 2 key modifications

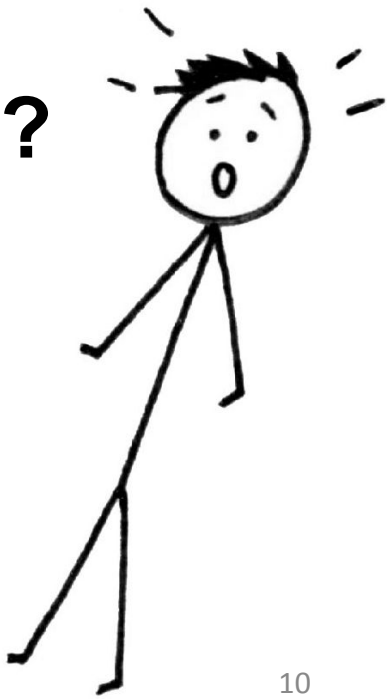
1. New Region Proposal Network
    a. Faster RCNN uses pre configured Anchor boxes as input to the network
    b. This network generates region proposals based on learning **keypoint** projections from the 3D mean face model and giving **faceness scores** to each of them

Introduction

Extends the existing Faster RCNN architecture, with 2 key modifications

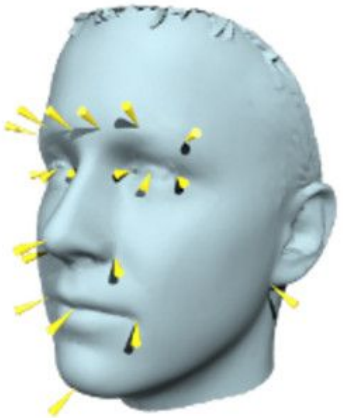
1. New Region Proposal Network
   a. Faster RCNN uses pre configured Anchor boxes as input to the network
   b. This network generates region proposals based on learning **keypoint** projections from the 3D mean face model and giving **faceness scores** to each of them
2. RoI Pooling Layer
   a. Faster RCNN, resizes each region proposal to a fixed size box, taking samples from equally divided areas
   b. This network - proposes **configuration pooling** layer , which takes samples , giving importance to each keypoint location.

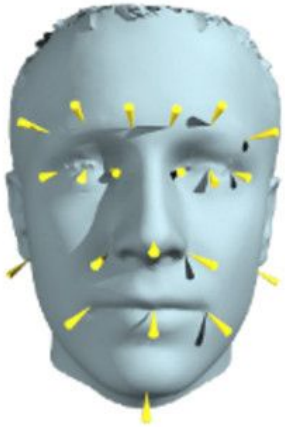
Couldn't understand anything in the previous slide !!
What are key points ?
What dataset is being used ?

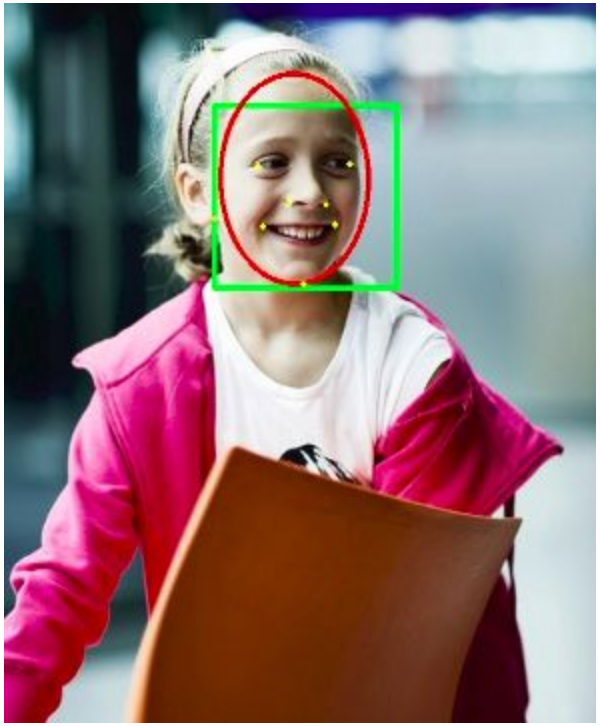


Data Set Description

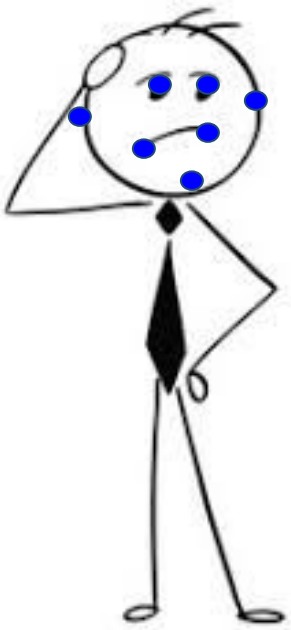
Annotated Facial Landmarks in the Wild (AFLW) - 25k annotated faces with upto 21 Landmark points



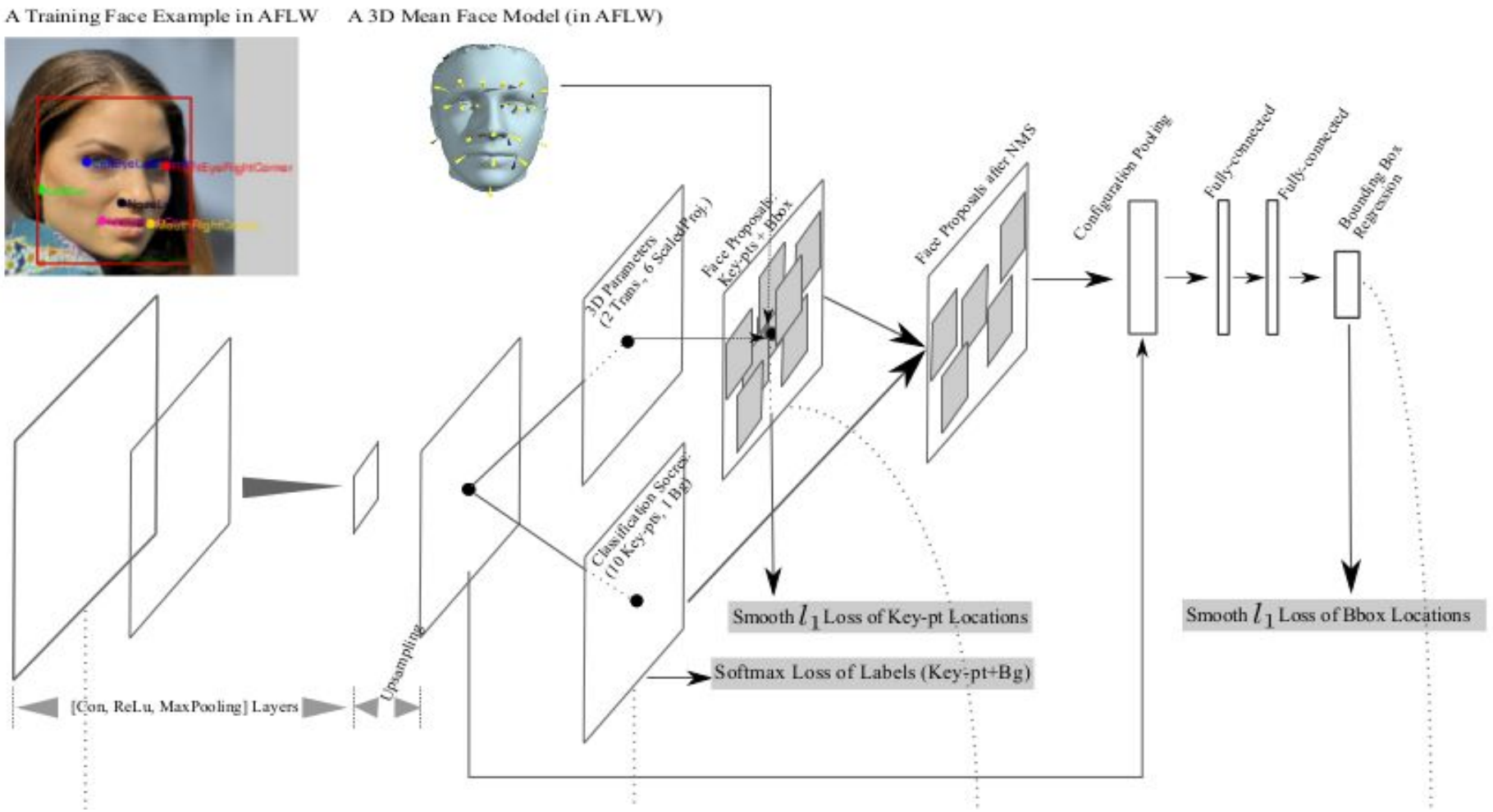




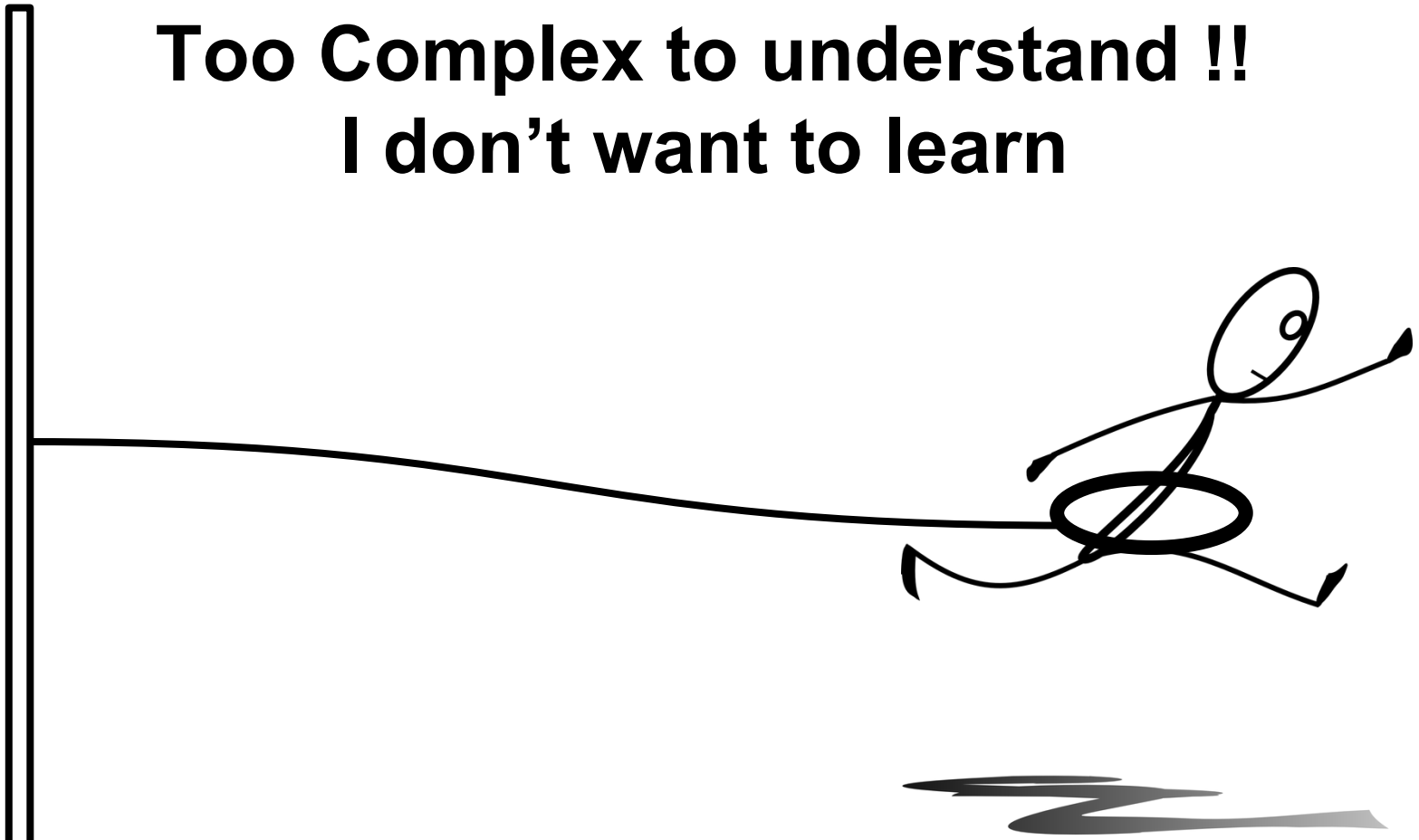
I have keypoints with me !
How is this data used to learn faces ?
What is the Network Architecture ?



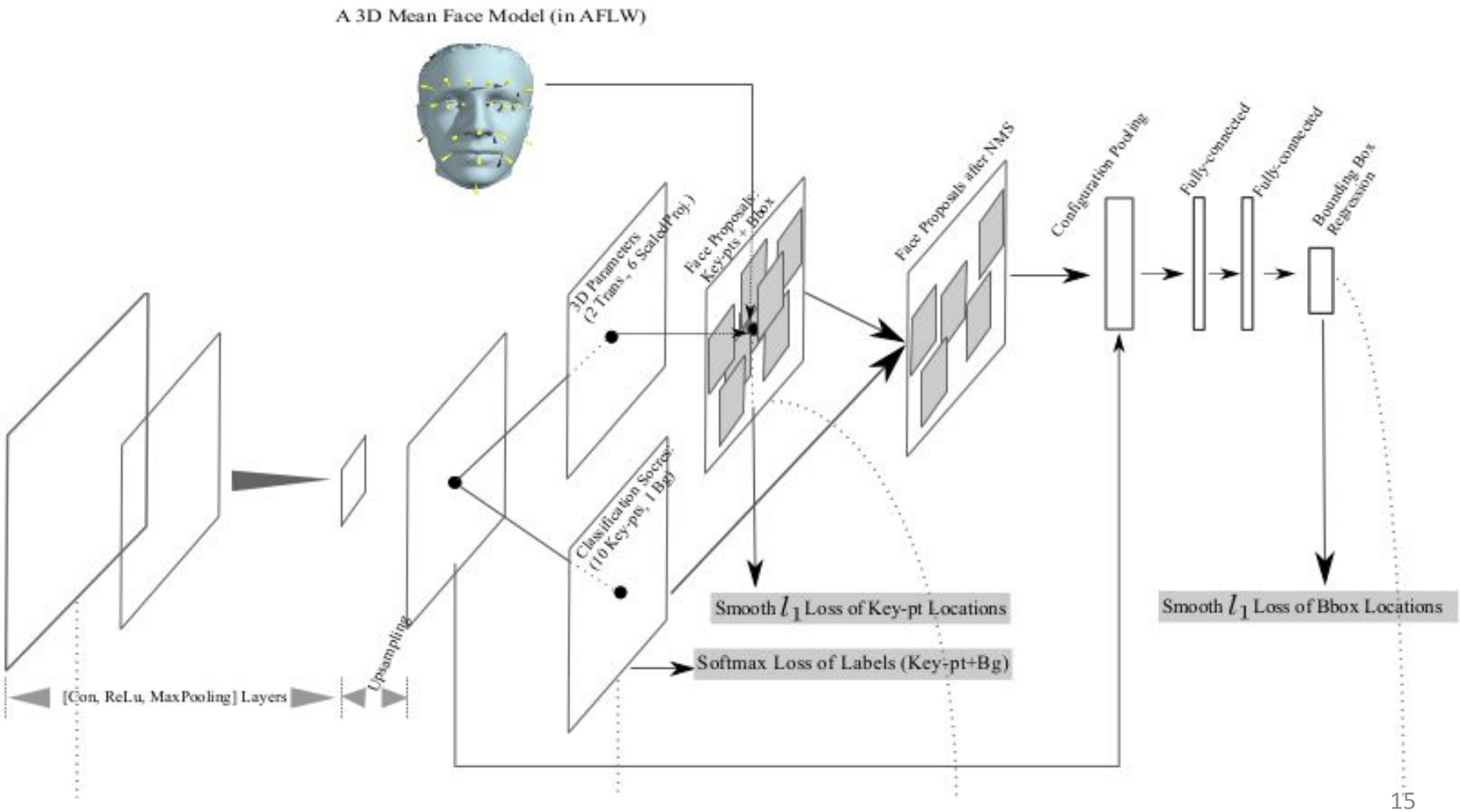
Network Architecture



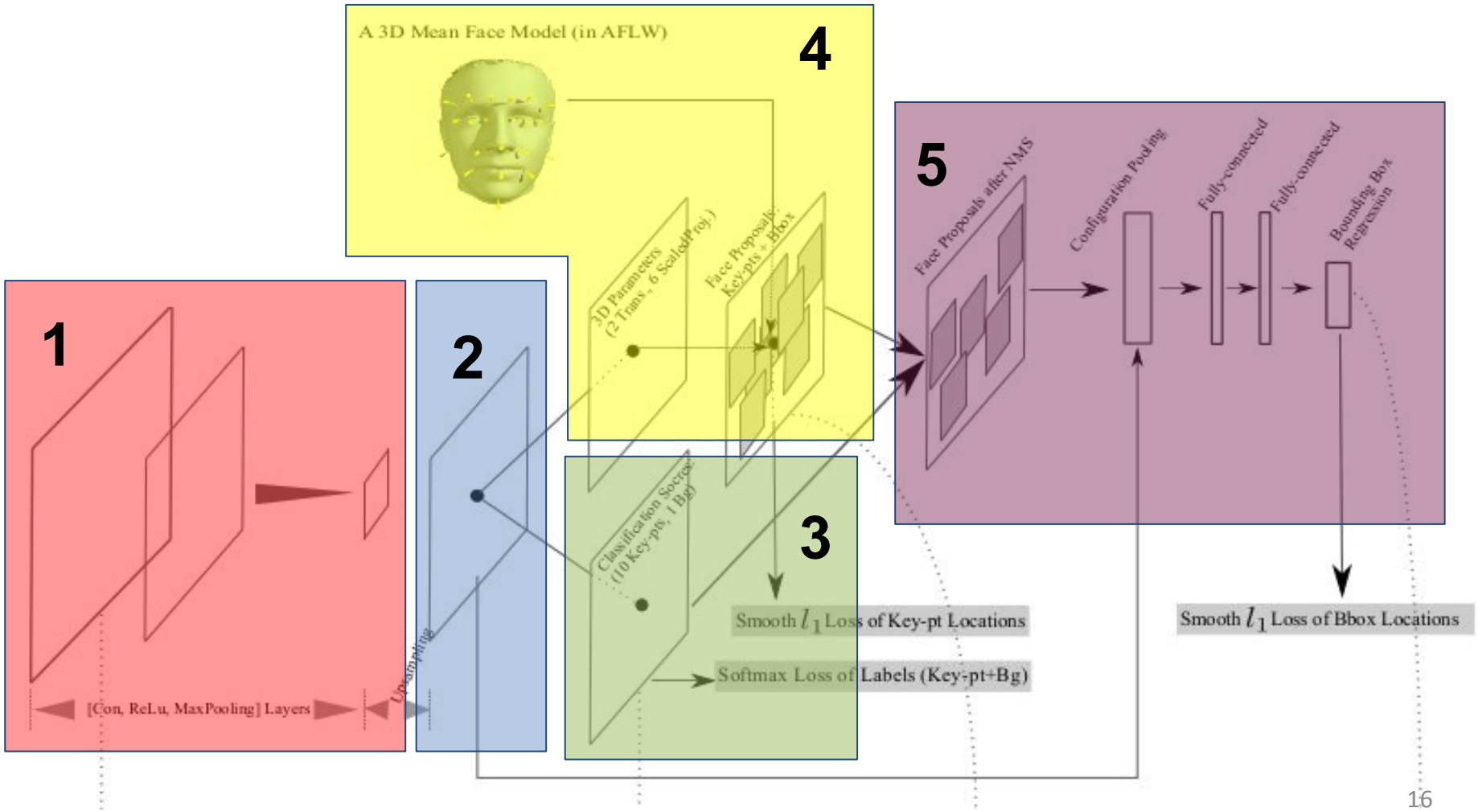
Too Complex to understand !!
I don’t want to learn



Network Architecture

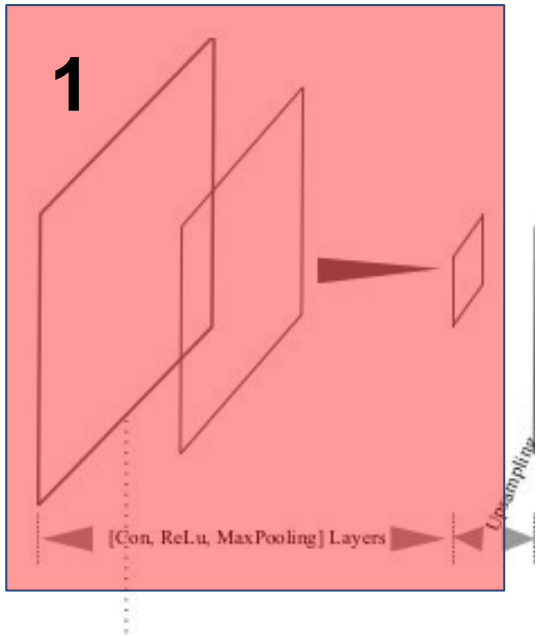


Network Architecture



Network Architecture

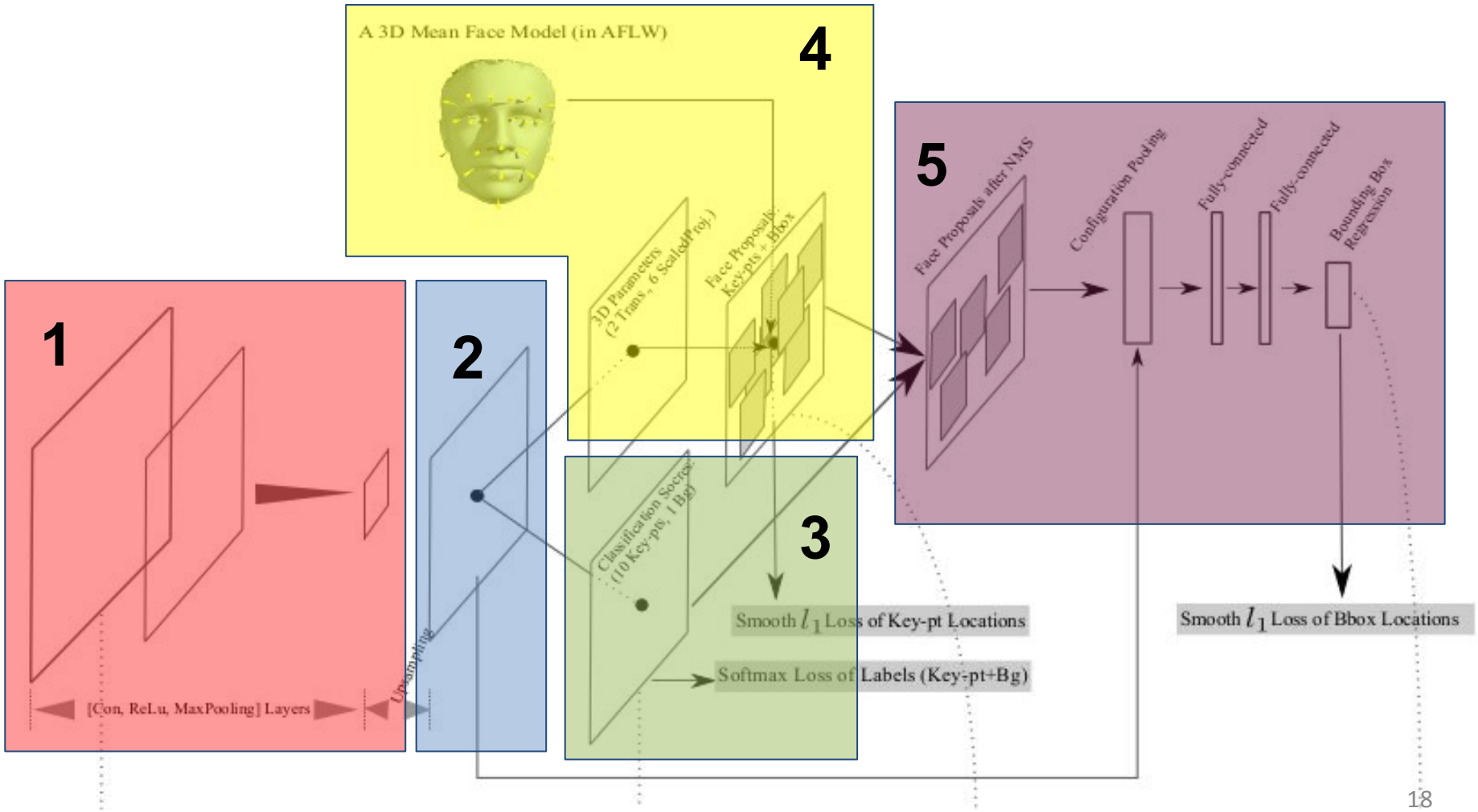
Part 1 Feature Backbone



- This Sub network takes input an Image and outputs a feature map for it to it
- Networks such as Resnet , VGG pretrained on relevant datasets could be used for this purpose

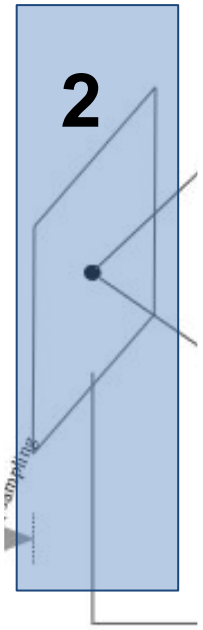
- VGG 16 - pre trained on imagenet is used
- **Input Dimensions - [w ,h] x 3**
- **Output Dimensions - [w/16, h/16] x 512**

Network Architecture



Network Architecture

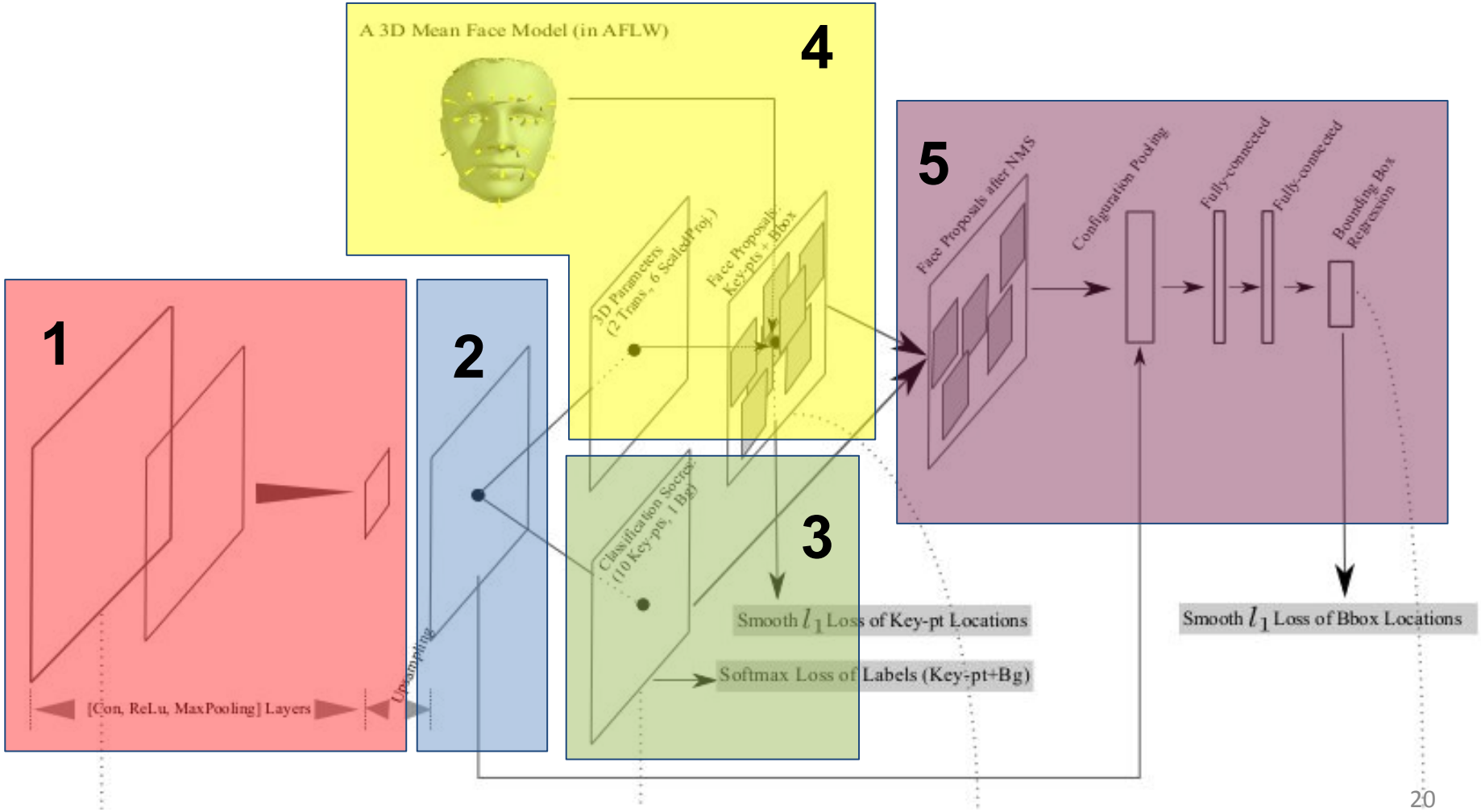
Part 2
Upsampling Block + Convolution



- Outputs obtained from module 1 is scaled down by 16 times wrt input.
- Key points that are close to each other merge together, when scaled down.
- Hence, for scaling up Deconvolution module, is used. (*8 time scale up*)
- Conv - Relu - follows it

- **Input - [w/16 ,h/16] x 512**
- **Output - [w/2, h/2] x 64**

Network Architecture



Network Architecture

Part 3
Classification Head

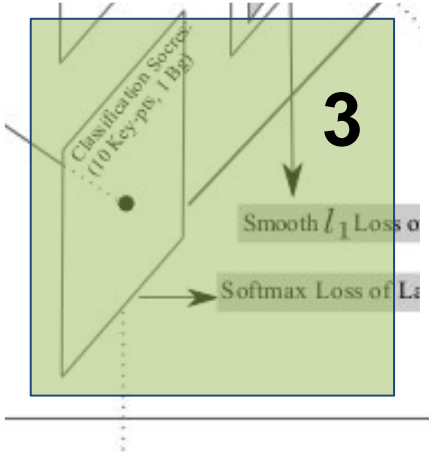
- Input is a Scaled up feature map - **[w/2, h/2] x 64**
- Convolution filter [depth 11] - represent (10 + 1) keypoint types

Output :-

- Task :- Multi class classification
- Loss function - Softmax -
  - Minimise Cross Entropy Loss Func

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

Modul Task :- Classify each pixel, into a keypoint type label



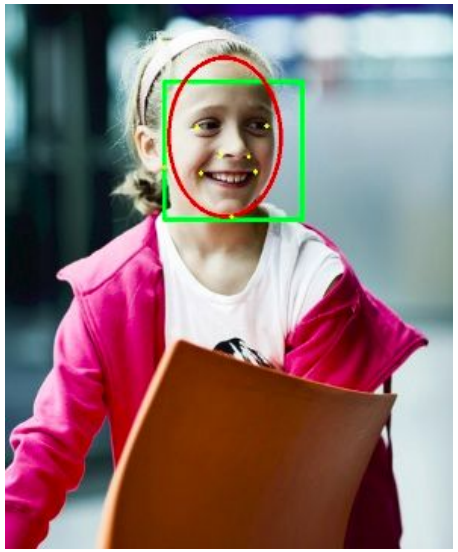
$$\mathcal{L}_{cls}(\omega) = -\frac{1}{2m}\sum_{i=1}^{2m}\log(p_{\ell_i}^{\mathbf{x}_i})$$

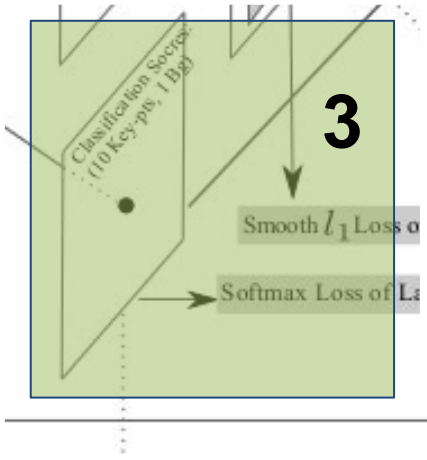
Network Architecture

Part 3
Classification Head

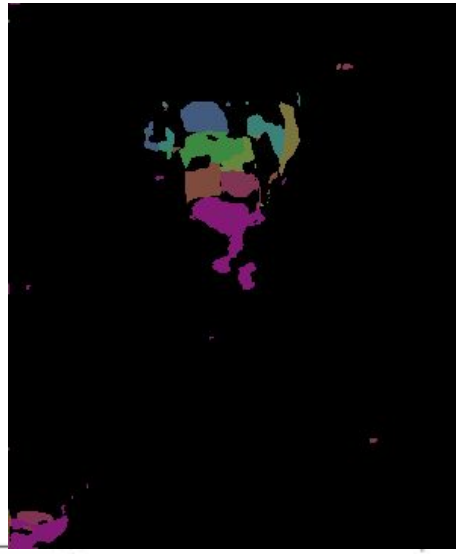
Output - **[w/2, h/2] x 11**
Each channel corresponds to the probability of each class
Max at each pixel location gives us the label map

Input Image

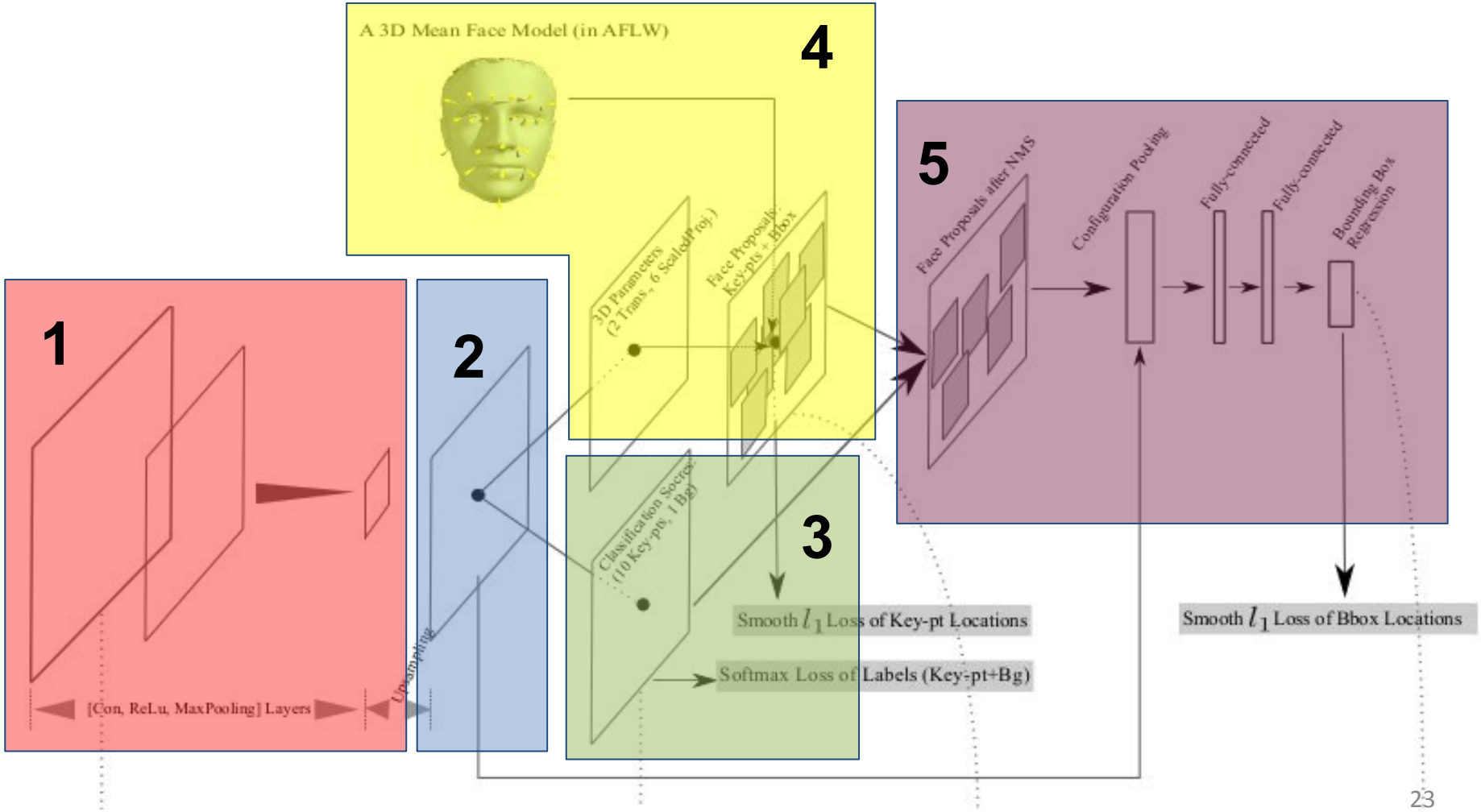




Intermediate Label map output

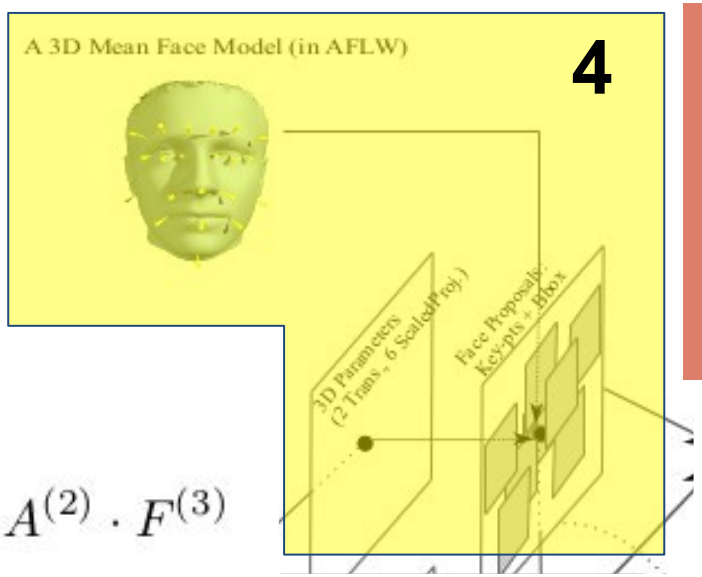


Network Architecture



Network Architecture

Part 4 - Region Proposal Network



Module Task :- Provide Promising bounding box proposals from the full image for further processing

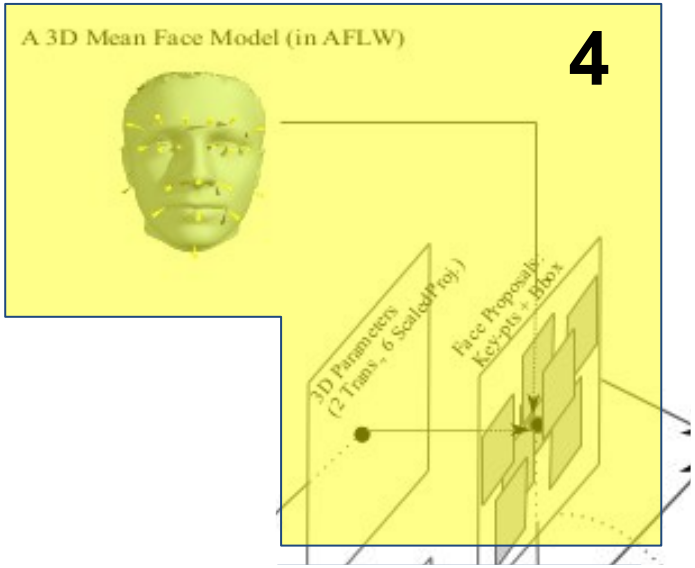
$$\hat{F}^{(2)} = \mu + A^{(2)} \cdot F^{(3)}$$

A 2D face can be obtained by projecting the mean 3D face.
Here F(3) - Keypoint location in 3D space, obtained from the mean face model
U - Translation coefficient A - Transformation Matrix 2x3 dimension
F(2) is the point in 2D space

8 parameters are to be learnt, inorder to project a point

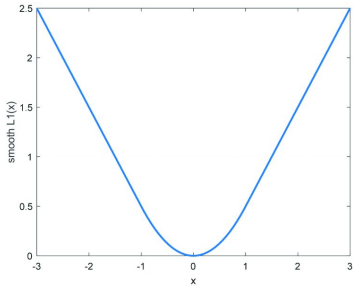
Network Architecture

Part 4 - Region Proposal Network



$$\mathcal{L}_{loc}^{pt}(\omega) = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{t\in\{x,y\}} \text{Smooth}_{l_1}(t_i - \hat{t}_{i,j})$$

$$\text{Smooth}_{l_1}(a) = \begin{cases} 0.5a^2 & \text{if } |a| < 1 \\ |a| - 0.5 & \text{otherwise.} \end{cases}$$



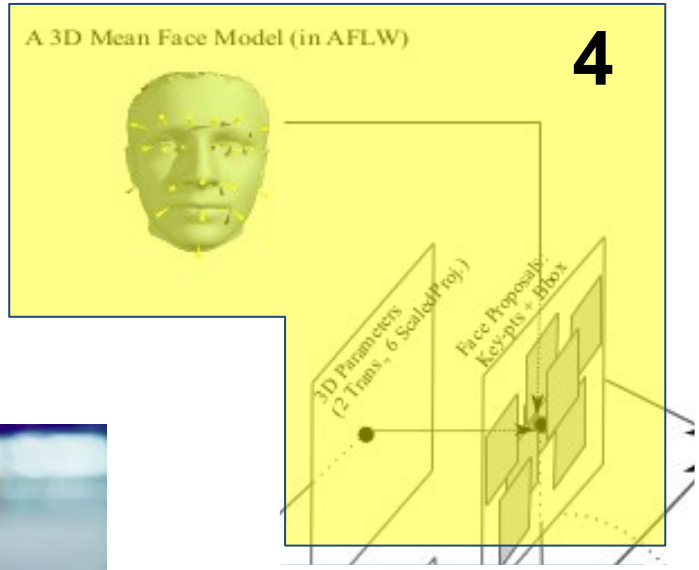
Conv filter - 8 depth - projection parameters

Face keypoint projections locations (x,y) For each pixel

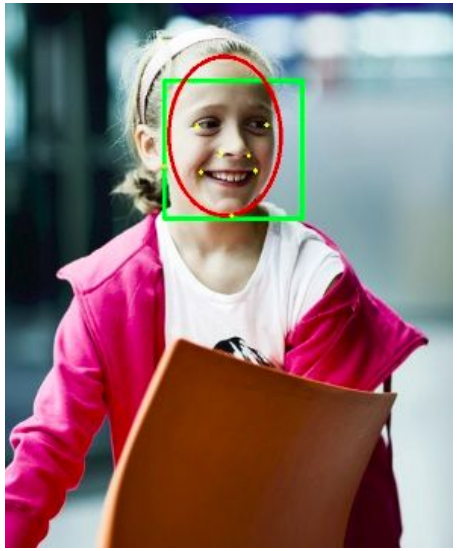
Smooth L1 loss for each of the projected location

Network Architecture

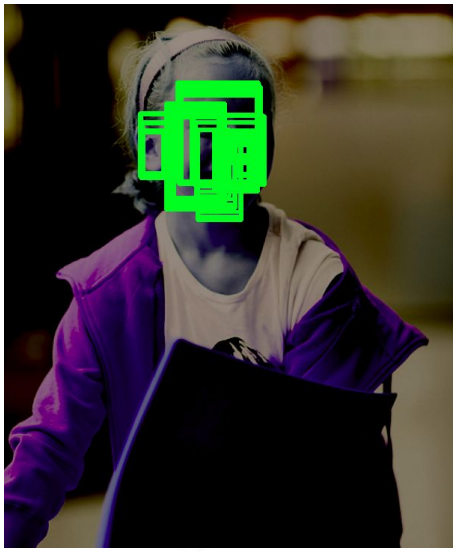
Part 4 - Region Proposal Network



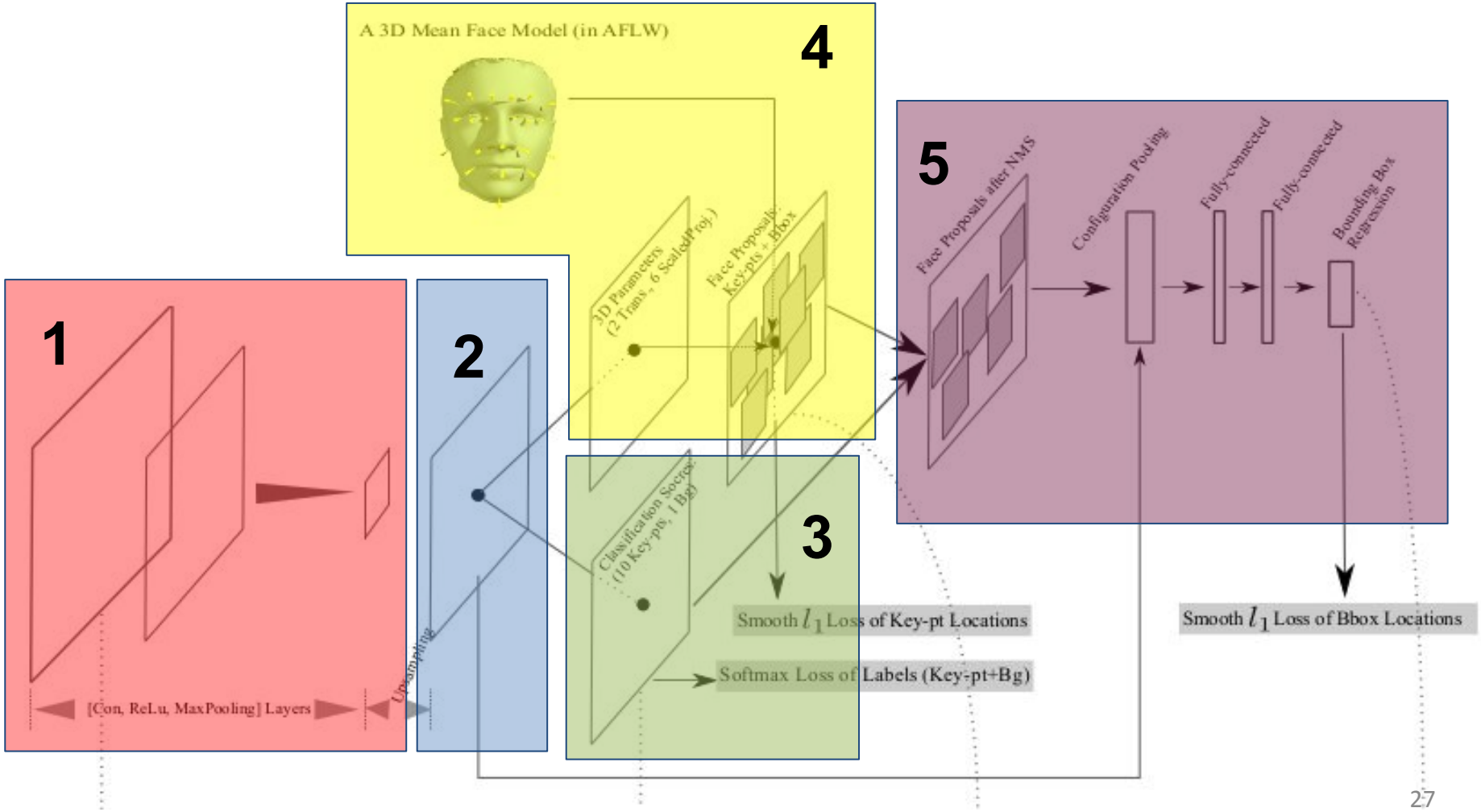
Input Image



Intermediate Proposal map output

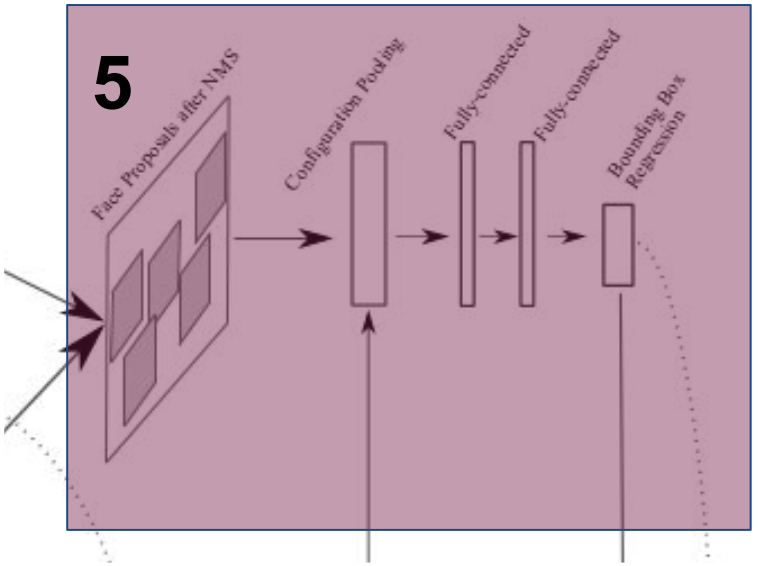


Network Architecture



Network Architecture

Part 5 - BBox Fine Tuning Unit



This Submodule takes as input region proposals
Passes it through ROI Pooling kind of layer - **this gives fixed size outputs for each proposal box**

2 Fully Connected Layer

Bounding Box regression (Loss) to better fit the proposal layer

$$\mathcal{L}_{loc}^{box}(\omega) = \frac{1}{K}\sum_{k=1}^{K}\sum_{i\in\{x,y,w,h\}} \mathrm{Smooth}_{l_1}(t_i - v_i)$$

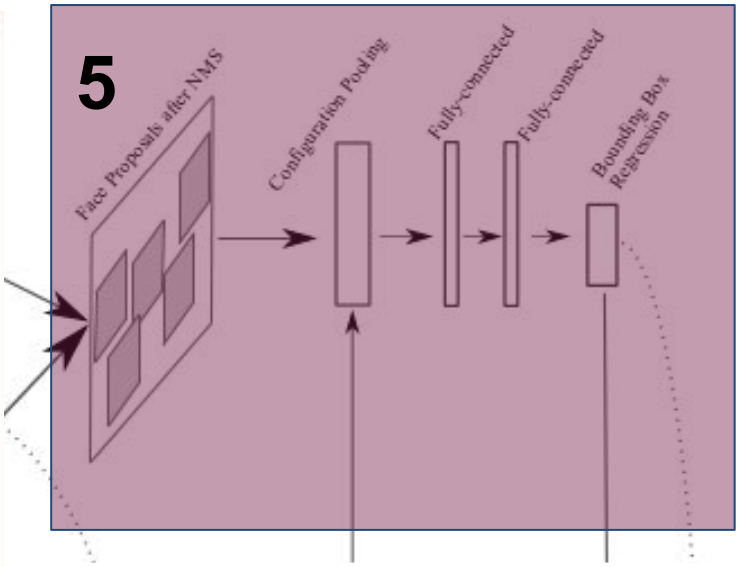
Network Architecture

Scale invariant bounding box regression function

$$
\begin{aligned}
t_x &= (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \\
t_w &= \log(w/w_a), \quad t_h = \log(h/h_a), \\
t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\
t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),
\end{aligned} \tag{2}
$$

where x, y, w, and h denote the box's center coordinates and its width and height. Variables x, x_a, and x^* are for the predicted box, anchor box, and ground-truth box respectively (likewise for y, w, h).

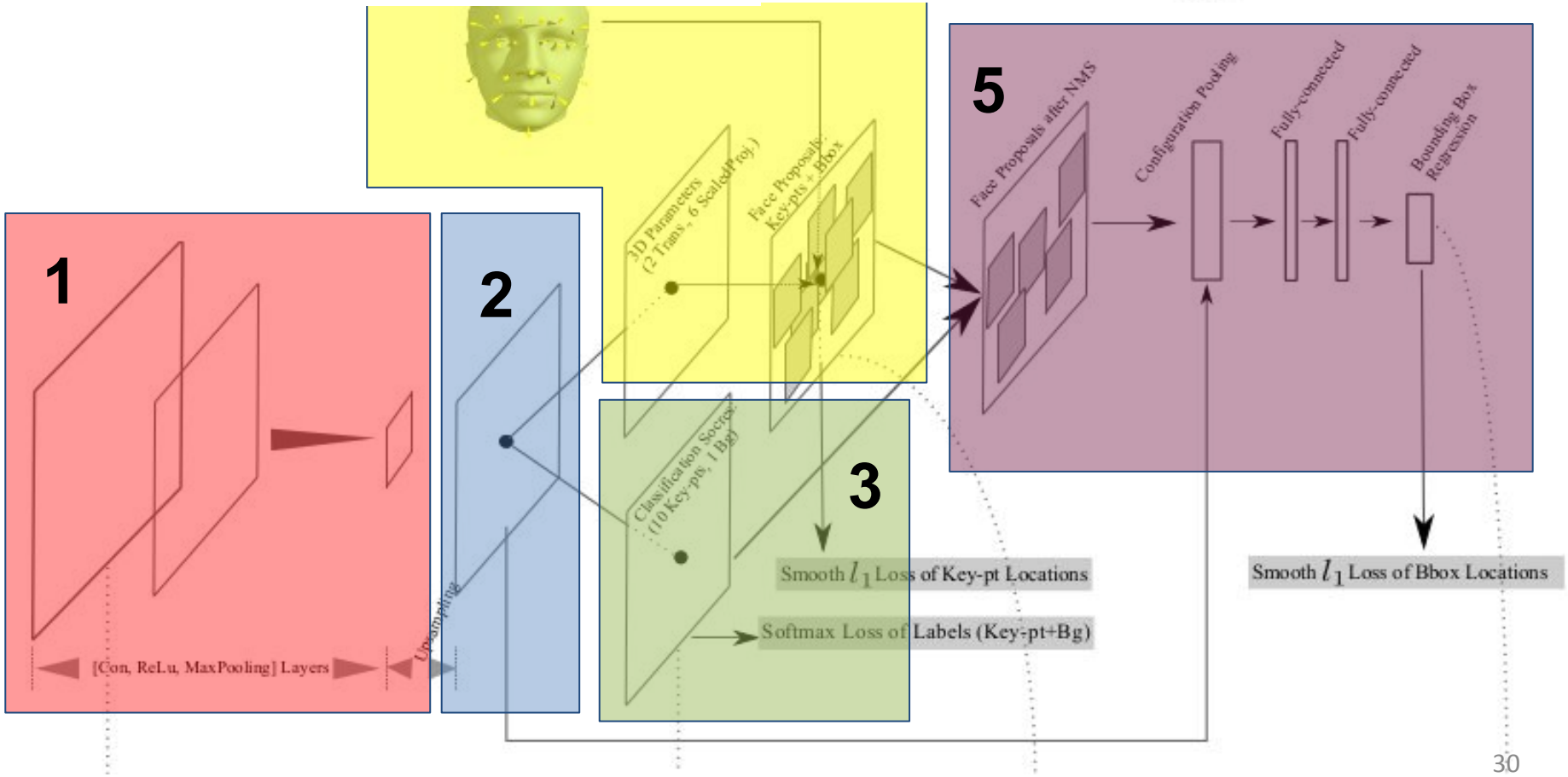
Part 5 - BBox Fine Tuning Unit



Network Architecture

Over all Loss Function

$$\mathcal{L}(\omega) = \mathcal{L}_{cls}(\omega) + \mathcal{L}_{loc}^{pt}(\omega) + \mathcal{L}_{loc}^{box}(\omega),$$



Inference Pipeline

Inference is similar to the learning pipeline

Few additional computations for region proposals

1. Faceness Score

$$\text{Score}(\hat{\mathbf{x}}_i, \hat{\ell}_i) = \sum_{i=1}^{10} \log(p_{\hat{\ell}_i}^{\hat{\mathbf{x}}_i})$$

2. Non Maximal Suppression(NMS) on proposal boxes

Also NMS is performed on the final list of bounding boxes obtained to remove multiple similar outputs

Implementation Details

Implementation Details

The program was written in python using Mxnet 1.0 as the deep learning framework and other packages such as numpy.

Custom operators were used, hence framework has to be rebuilt from sources upon addition of new operators.

70 % of AFLW dataset is being used for training, remaining will be used for testing.

Implementation Details

GPU's - Aws is being used

Instance Details :-
P2.xlarge - Comes pre installed with Frame works
1 Nvidia - K80 GPU
4 CPU cores , 64 GB RAM
Reserved Instance : 0.9 $/hr Spot Instance : ~0.3 $/hr

S3 data storage volume could be added to the instance based on requirements

Implementation Details

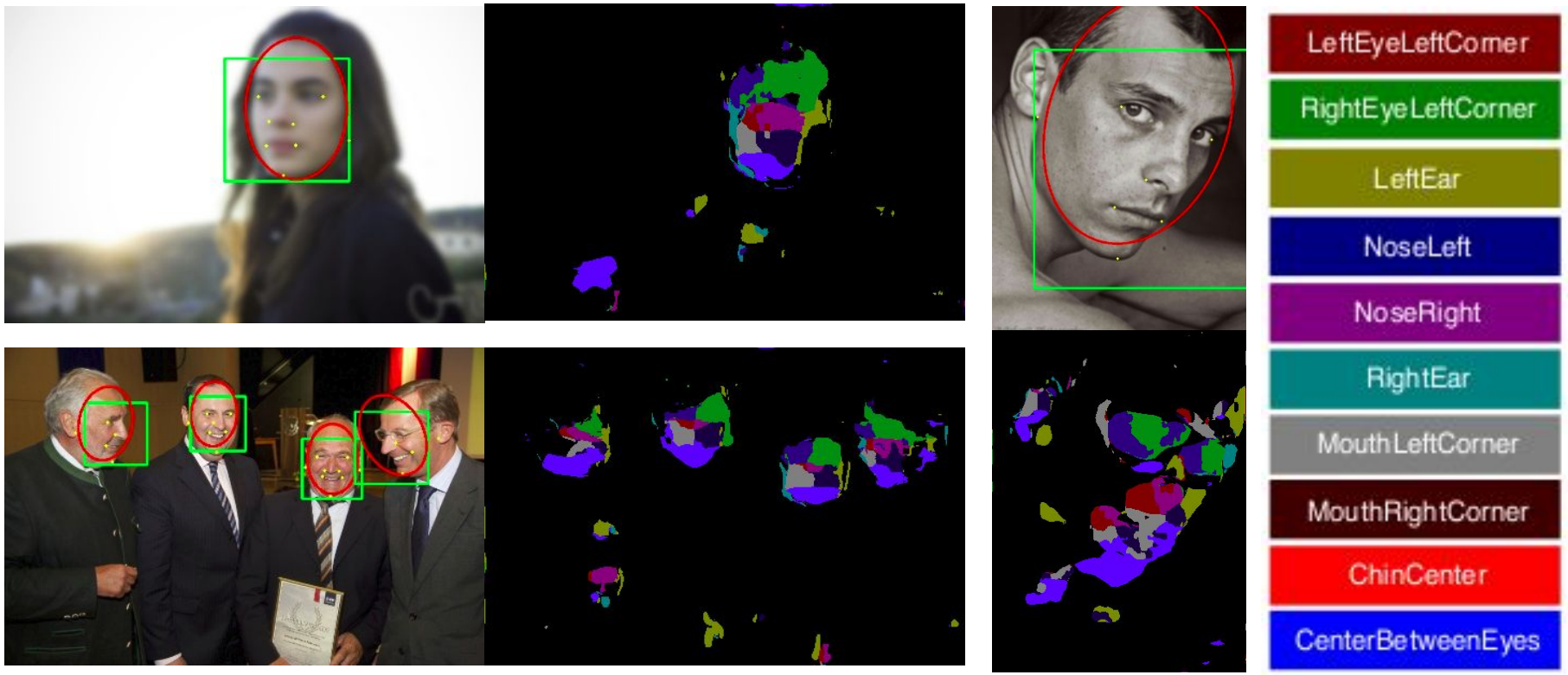
Weight update -

- learning rate was started at 0.01, with decay after certain epochs
- Momentum optimizer, with weight of 0.9 for it.
- Batch size of 1
- Network was run for 10 epochs

Evaluation -

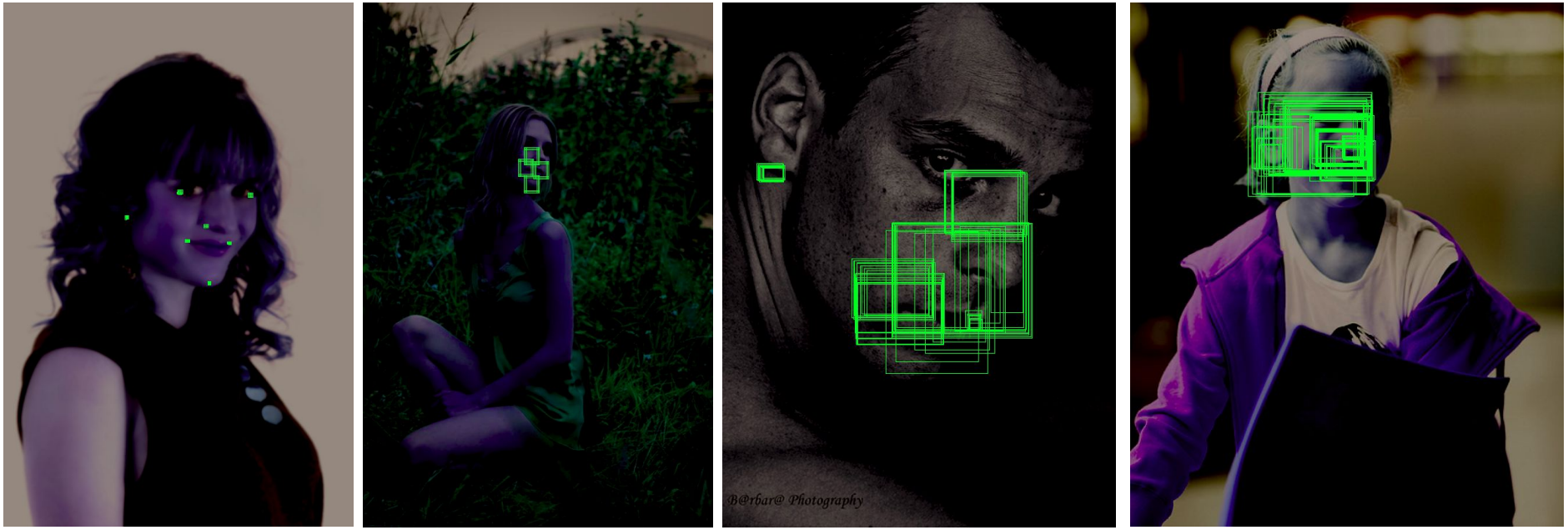
- Intersection over union (IOU) for Bounding Box Reg
- Accuracy measure for key point class, classification problem

Results



Example heat Map Outputs

Results



Region proposal bounding boxes from different learning instances starting from the left

Conclusions and Future Work

- New experimental network architecture proposed, has been reprogrammed to reproduce the results.
- Results obtained for heatmaps and region proposals were comparable to the original paper.
- Work shall be extended to complete the fine tuning of the region proposals to obtain the final face bounding box
- Network can be extended to generalize to any other object type like cars.

Main References

1. [1] Li Y., Sun B., Wu T., Wang Y. (2016) Face Detection with End-to-End Integration of a ConvNet and a 3D Model. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9907. Springer, Cham
2. [2] Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
3. [3] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
4. [4] Girshick, R.: Fast R-CNN. In: ICCV (2015)
5. [5] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)


Thank you!!