Dense Face Detection

Introduction and Rationale

Conventional face tracking and detection methods are limited to working with a few or just a single face. One aim of this project is to construct a dense face detector which is trained end-to-end using a fully convolutional neural network. The model should have capabilities of detecting a variable number of faces, in crowded scenes and for various scales, lighting and occlusions, for real time video. The purpose of this model is to serve as a baseline to be applicable for CCTV and other security systems, and areas such as person tagging on social media, face detection for digital cameras, and the general detection of humans.

Background

One-shot detectors were first introduced by YOLO [5] and later modified and improved by the models SSD [6], YOLO9000 [7], DSSD [8] and RetinaNet [9]. A one-shot detector predicts thousands of bounding box proposals in different spatial positions in an image, through a set of predetermined anchor boxes. The models regresses four coordinate offsets per bounding box and predicts probabilities that the boxes contains an object.

Method

We systematically constructed a face detector, coined FaceNet, by varying one new design decision at a time, keeping all other factors constant, iteratively improving the FaceNet object detector. For our baseline model a fully-connected architecture is used to enable dynamic scaling of the model. A convolution is defined by the following equations, and is trained using Stochastig Gradient Descent.

$$X^{(l+1)}_{r,c',h',w'} = \sum_{c=1}^{C^{(l)}} \sum_{j=1}^{k} \sum_{i=1}^{k} X^{(l)}_{r,c,(sh'-s+j),(sw'-s+i)} \, W^{(l)}_{c',c,j,i} \tag{1}$$

Here $X^{(l)} \in \mathbb{R}^{R \times C^{(l)} \times H^{(l)} \times W^{(l)}}$ is the neurons, and $W^{(l)} \in \mathbb{R}^{C^{(l+1)} \times C^{(l)} \times H^{(l)} \times W^{(l)}}$ is the convolutional weights, at layer l in the network. The batch size, channels, height and width at layer l is given by R, $C^{(l)}$, $H^{(l)}$, and $W^{(l)}$ respectively. $L(\theta)$ is the loss function. The kernel size and stride are given by k and s. Backpropagation is done using the following partial derivatives:

$$\frac{\partial L(\theta)}{\partial W^{(l)}_{c',c,h,w}} = \sum_{r=1}^{R} \sum_{h'=1}^{H^{(l+1)}} \sum_{w'=1}^{W^{(l+1)}} \frac{\partial L(\theta)}{\partial X^{(l+1)}_{r,c',h',w'}} X^{(l)}_{r,c,(sh'-s+h),(sw'-s+w)} \tag{2}$$

$$\frac{\partial L(\theta)}{\partial X^{(l)}_{r,c,h,w}} = \sum_{c'=1}^{C^{(l+1)}} \sum_{h'=1}^{H^{(l+1)}} \sum_{w'=1}^{W^{(l+1)}} \frac{\partial L(\theta)}{\partial X^{(l+1)}_{r,c',h',w'}} W^{(l+1)}_{c',c,(h+s-sh'),(w+s-sw')} \tag{3}$$

A ResNet-18 [4] was used as a backbone network, together with two sub-networks: A classification head and a regression head, constructed out of four residual bottleneck blocks [4]. Features from convolutional blocks *conv2* to *conv6* were used to predict bounding boxes of different scales. Anchor boxes of sizes 16^2 to 416^2 pixels were used to accomplish this.

The model was trained using an anchor box assignment strategy on the WIDERFace [10] dataset. An anchor box was assigned a positive label if its intersection over union (IoU) with a ground truth was greater than a threshold value (0.55). The IoU of set A and B is defined as:

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4}$$

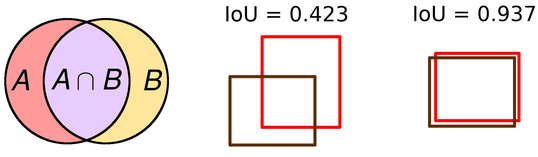

Fig. 7: IoU is defined as the size of the union divided by the size of the intersection of two sets A and B.

Our FaceNet v1.0 baseline used the total loss function $L(\theta)$, which is the mean of the regression losses L_r of all positively assigned anchors plus the mean of the classification losses L_c of all anchors. Let $\hat{r}_a$ and r_a be the predicted and ground truth coordinate offsets for anchor box a, and $\hat{p}_a$ and p_a be the predicted and ground truth probability score respectively, for anchor a to contain an object. Let N be the set containing all anchor boxes a, and N_{pos} be the set containing every positively assigned anchor box.

$$L_r(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{5}$$

$$L_c(p, \hat{p}) = -p \log \hat{p} \tag{6}$$

$$L(\theta) = \frac{1}{|N|} \sum_{a \in N} L_c(p_a, \hat{p}_a) + \frac{1}{|N_{pos}|} \sum_{a \in N_{pos}} L_r(r_a - \hat{r}_a) \tag{7}$$

All FaceNet versions were evaluated on the easy, medium, and hard splits of the WIDERFace dataset [10], using the mean average precision (mAP) metric. It is defined as the integral of a model's precision $P(\theta)$ with respect to its recall $R(\theta)$.

$$\text{mAP} = \int_0^1 P(\theta) \, \mathrm{d}R(\theta) \tag{8}$$

Focal Loss versus Binary Cross Entropy

Focal loss, used in RetinaNet [9], introduced a loss function that focused on the training examples which were hard to classify. It was shown that using focal loss instead of cross entropy for multi-class object detection increased the final accuracy of the model. FaceNet v1.1 and v1.2 experimented using binary cross entropy L_{BCE}, and Focal Loss L_{FL} respectively as their classification losses, defined as:

$$L_{BCE}(p, \hat{p}) = -p \log(\hat{p}) - (1 - p) \log(1 - \hat{p}) \tag{9}$$

$$L_{FL}(p, \hat{p}) = -p(1 - p)^{\gamma} \log(\hat{p}) - (1 - p) p^{\gamma} \log(1 - \hat{p}) \tag{10}$$

Model	mAP_e	mAP_m	mAP_h
v1.0	0	0	0
v1.1	37.4	21.7	9.7
v1.2	22.4	12.1	5.1

Table 3: Versions 1.0 to 1.2. Results on the easy (e), medium (m) and hard (m) WIDERFace splits.

The baseline model, FaceNet v1.0, converged to predict a probability of zero that there existed an object for every bounding box. Later versions showed that Focal Loss performed considerably worse compared to binary cross entropy.

Adding Feature Pyramids

Our experiments showed that an addition of a Feature Pyramid Network (FPN) [11] architecture leads to a dramatic increase in accuracy. FaceNet v1.3 and v1.4 used binary cross entropy and Focal Loss respectively with an FPN. Versions 1.5 and 1.6 experimented using a new pyramid-level-specific loss function $L(\theta)_{level}$. Here L is the set of all pyramid levels P, and N^P is the set of all created anchor boxes at every spatial position at pyramid level P:

$$L(\theta)_{level} = \sum_{P \in L} \frac{1}{|N^P_{pos}|} \Big(\sum_{a \in N^P} L_c(p_a, \hat{p}_a) + \sum_{a \in N^P_{pos}} L_r(r_a - \hat{r}_a) \Big) \tag{11}$$

Model	mAP_e	mAP_m	mAP_h
v1.3	89.3	85.9	65.9
v1.4	81.0	81.0	64.2
v1.5	87.2	82.5	50.4
v1.6	85.1	80.0	50.6

Table 4: Versions 1.3 to 1.6. Results on the easy (e), medium (m) and hard (m) WIDERFace splits.

Adding Online Hard Example Mining

FaceNet v1.7 and v1.8 used Online Hard Example Mining (OHEM) [12], training on the top 128 classification losses, after non-maxima supression was applied. Version 1.7 applied OHEM separately on every example in the mini-batch, while version 1.8 used the whole mini-batch.

Model	mAP_e	mAP_m	mAP_h
v1.7	46.0	51.8	42.3
v1.8	55.9	36.8	15.6

Table 5: Versions 1.7 and 1.8. Results on the easy (e), medium (m) and hard (m) WIDERFace splits.

New Features, Color Jitter, and Depth

Versions 1.9 to 2.5 evaluated additional anchor box scales, IoU thresholds, deeper features, random color jitter and network depth. Most notable of these were the following versions: FaceNet v2.1 used features from *conv3* to *conv7* to increase computational speeds and to allow for deeper features. Bilinear interpolation was added to both sub-networks to offset the decrease in scale. Version 2.3 used additional data augmentation by randomly adding noise and shifting hues. To increase detection results on smaller faces, FaceNet v2.4 used anchors of sizes 8^2 to 416^2, and features from *conv2* to *conv7*.

Model	mAP_e	mAP_m	mAP_h
v2.1	89.1	86.6	64.4
v2.3	92.2	90.3	70.1
v2.4	89.6	86.8	75.8

Table 6: Notable results from versions 1.9 to 2.5. Results on the easy (e), medium (m) and hard (m) WIDERFace splits.

Qualitative Results

Fig. 8: Final results of FaceNet on the publically available WIDERFace [10] validation dataset.

Conclusion

Our final model is capable of dynamically detecting thousands of faces in a crowded scene for various scales, lighting and occlusions, with an inference time of 22 ms. In contrast to previous research, Focal Loss and Online Hard Example Mining was shown to decrease the accuracy of the object detector. The final model is sufficiently computationally efficient and accurate, achieving a mean average precision of 92.2, to serve as a baseline model for any system which incorporates surveillance, security or the detection of humans.

References

All figures in this project were made by the project author Nikita Zozoulenko.

[1] *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.* S. Bai, Z. Kolter, and V. Koltun arXiv preprint arXiv:1803.01271, 2018.

[2] *Dilated Recurrent Neural Networks.* S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. Hasegawa-Johnson, T. Huang arXiv preprint arXiv:1710.02224, 2017

[3] *Microsoft COCO Captions: Data Collection and Evaluation Server.* X. Chen and H. Fang and TY Lin and R. Vedantam and S. Gupta and P. Dollár and C. L. Zitnick. arXiv preprint arXiv:1504.00325, 2015

[4] *Deep residual learning for image recognition.* K. He, X. Zhang, S. Ren, and J. Sun. arXiv preprint arXiv:1512.03385, 2015.

[5] *You only look once: Unified, real-time object detection.* J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. arXiv preprint arXiv:1506.02640, 2015.

[6] *SSD: Single Shot MultiBox Detector.* W. Liu and D. Anguelov and D. Erhan and C. Szegedy and S. Reed and C. Fu and A. Berg. arXiv preprint arXiv:1512.02325, 2015

[7] *YOLO9000: Better, Faster, Stronger.* J. Redmon and A. Farhadi. arXiv preprint arXiv:1612.08242, 2016.

[8] *DSSD : Deconvolutional Single Shot Detector.* C. Fu and W. Liu and A. Ranga and A. Tyagi and A. C. Berg arXiv preprint arXiv:1701.06659, 2017

[9] *Focal Loss for Dense Object Detection.* Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He and Piotr Dollár. arXiv preprint arXiv:1708.02002, 2017

[10] *WIDER FACE: A Face Detection Benchmark.* Yang, Shuo and Luo, Ping and Loy, Chen Change and Tang, Xiaoou IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[11] *Feature Pyramid Networks for Object Detection.* Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan and Serge J. Belongie. arXiv preprint arXiv:1612.03144, 2016

[12] *Training Region-based Object Detectors with Online Hard Example Mining.* A. Shrivastava, A. Gupta, and R. Girshick. arXiv preprint arXiv:1604.03540, 2017