Introduction

Conventional face tracking and detection methods are limited to a few or just a single face. One aim of this paper is to construct a dense face detector using a fully convolutional neural network capable of detecting a variable number of faces in real time video. This method is to be served as a baseline to be applicable for CCTV or other security systems, or areas such as person tagging on social media and face detection for digital cameras.

There exists multiple moderns techniques and network architectures used for the training of multi-class object detectors. The additional aim of this paper is to systematically and empirically investigate how the modern methods and techniques, such as Feature Pyramid Networks [**?**], Focal Loss [**?**], Online Hard Example Mining [**?**], and data augmentation, affects the performance of a single class object detector.

Background

7. References

Dense Face Detection

1 Dense Face Detection

In this section we systematically and empirically investigate how to construct and improve a convolutional-neural-network-based model to detect a variable number of faces in an image. Every factor and suggestion which affects the model performance is evaluated one by one and added to a baseline model, which we name FaceNet.

1.1 Background

1.1.1 WIDERFace

For the task of detecting a variable number of faces in an iamge, the WIDERFace dataset [**?**] was used. It is a dataset consisting of 32 203 images of a total of 393 703 number of faces at different scales, lighting and occlusions. Every training example consists of a number of bounding boxes which describes all the faces in the image. A bounding box consists of four coordinates specifying its location: two for the upper left corner and two for the bottom right corner of the bounding box. The images are labeled by humans and is called the ground truth of the image.

1.1.2 One-shot detectors

One-shot detectors were first introduced by the model YOLO [**?**] and later modified and improved by the models SSD [**?**], YOLO9000 [**?**], DSSD [**?**] and RetinaNet [**?**]. A one-shot detector works by predicting up to tens of thousands of bounding boxes in different spatial positions in an image, that sets out to detect every object in that image. For every set of predetermined spatial positions in the image, the model predicts 4 bounding box coordinates and probability scores that there exists an object inside the specified bounding box.
The model is trained by assigning every bounding box either a negative or a positive label during training, depending on if the bounding box contains a ground truth object or not. A bounding box is said to contain an object if the intersection over union (IoU) [**?**] of the area of the bounding box and the area of the ground truth is bigger or equal to a threshold value (0.55). The intersection over union is used to determine how similar two sets A and B are (see figure 1). It is bounded by the interval $[0, 1]$ and is defined by the following equation:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

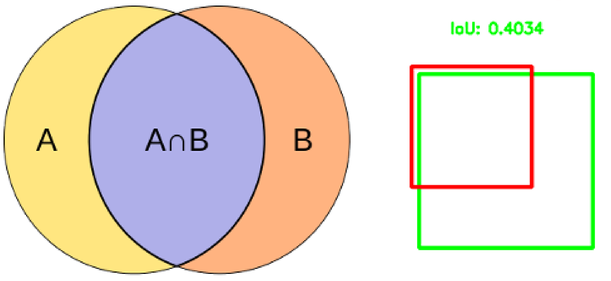

Figure 1: IoU [**?**] is defined as the size of the union divided by the size of the intersection of two sets A and B. A bigger IoU implies that the predicted bounding box is closer to the ground truth.

1.2 Method

1.2.1 FaceNet Baseline Model

The baseline model uses the deep residual convolutional neural network ResNet-18 [**?**] as a backbone network and feature extractor. It uses a total of 18 convolutional layers throughout blocks named *conv1* to *conv5* to extract features from the input image. An additional *conv6* block is created after the last layer to create one additional scale of feature maps. The block consists of one convolutional layer with stride 2, and 2 residual bottleneck layers used by Resnet-50 up to ResNet-152. The feature maps created by the backbone model is then fed into two smaller sub-networks, called the classification head and the regression head. They are constructed out of 4 residual bottleneck layers.
Bounding boxes are created from a number of anchor boxes at every spatial position (width and height) from a set of feature maps. An anchor box is a bounding box of a specific size and scale, used to base the bounding box predictions off. Anchor boxes used by FaceNet are set to be of sizes in the range $[16, 406]$ pixels, each increasing by a factor of $2^{1/3}$ for every new anchor box size. Two ratios of width:height are used at every scale, namely $1 : \frac{2}{3}$ and $1 : 1$, to enable the detection of faces facing sideways and directly into the camera. Having densely packed anchors makes it that every ground truth bounding box (of sufficient size) has an IoU$>$ 0.55 with at least one anchor box. The anchor boxes are spaced out to include 3 scales with 2 aspect ratios across feature maps from the levels *conv2* to *conv5*, to enable the detection of faces of different scales, similar to RetinaNet [**?**].
The regression and classification sub-networks take in the last convolutional output from a single network level (*conv2* to *conv5*) as input. Let W and H be the width and height of the tensor of activations at a single network level, and C be the amount of classes to classify. FaceNet uses $C = 2$, one for a background class (no face) and one for a foreground class (a face). The classification head predicts C probabilities for every anchor A, that there exists a face in the anchor box at every spatial location $(W \times H)$, for a total of $KWHA$ values. The regression head predicts 4 coordinate offsets for every point of the anchor box, for it to match the ground truth as closely as possible, for a total of $4WHA$ coordinate offsets. Additionally, similar to RetinaNet [**?**], the last conventional layer which predicts class probabilities uses a bias $b = -\log \frac{C(1-\pi)}{\pi}$, such that the model, when first initialized, predicts a probability of π for the case of there existing a foreground class (a face) in every bounding box. The value $\pi = 0.001$ was used.

1.2.2 Training

If not otherwise stated, each version of FaceNet was trained for 70 000 iterations on an Nvidia GTX 1080ti with an initial learning rate of 0.0001, and lowering the learning rate by a factor of 10 after 50 000 and 60 000 iterations. Due to memory limitations a batch size of 8 is used. The models are trained on square image crops, randomly picked to be of between 0.33 and 0.95 of the short size of the image, and resized to be of size 512^2 pixels. This is done to artificially increase the size of the training data. Additionally, the image is flipped horizontally with a probability of 0.5.
The models are evaluated on the WIDERFace validation set, using the easy, medium and hard splits provided by the WIDERFace evaluation server. The evaluation metric used is mean average precision (mAP). It is calculated by plotting the accuracy of the model as a function of its recall. The mean average precision is given by the integral of the accuracy with