# Mask R-CNN for Ear Detection

Matic Bizjak, Peter Peer and Žiga Emeršič
Faculty of Computer and Information Science
University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia
Email: mb5975@student.uni-lj.si, {peter.peer, ziga.emersic}@fri.uni-lj.si

***Abstract*—Ear detection is an important step in ear recognition pipeline as it makes or breaks the system. However, in the literature there is arguably the lack of ear detection approaches available. This poses a problem for opening ear recognition system to wider use and applications in commercial systems. To tackle this problem we present the use of Mask R-CNN for pixel-wise ear detection. Furthermore, we directly compare our approach to one of the previous best performing pixel-wise ear detection approach by using the same dataset and protocol. Our results with intersection over union score of 79.24% on AWE dataset show the superiority of our approach and present a viable approach for future use in ear recognition pipelines.**

***Keywords*—Mask R-CNN; Ear detection; AWE dataset**

## I. Introduction

Pipeline of a typical biometric system consists of raw data acquisition, detection, feature extraction, classification and system evaluation. Performance of each step can greatly influence the performance of the whole system. In this work, we focus on pixel-wise ear detection and evaluate the performance of a convolutional neural network (CNN) architecture, Mask R-CNN [1]. We use 2D images captured under uncontrolled conditions, proving the feasibility of this approach.

The rest of this paper is structured as follows: a review of related work is given in Section 2, and Section 3 describes the Mask R-CNN approach. Mask R-CNN is described as a combination of five parts. In Section 4, training dataset and performance metrics are described and a comparison with PED-CED approach is given to evaluate the performance of our approach. Finally, Section 5 provides the conclusions.

## II. Related work

Automatic ear detection approaches date all the way back to 2007, with the use of Hough Transform [2]. However, in this paper we overview only some of the more recent works. To get a more comprehensive overview of the field, the reader is referred to some of the ear detection surveys [3], [4]. In 2015 the authors of [5] presented an entropy-cum-Hough-transform-based ear detection approach. They used a combination of hybrid ear localizer and an ellipsoid ear classifier to enhance ear location predictions. Detection rate is defined as a ratio between number of successful ear localizations and number of all annotated ears. Ear localization is considered successful if detected area covers the entire ear and if the distance between center of the detected region and annotated ground truth is close enough. Authors achieved detection rate of 100.0%, 100.0% and 73.95% on UMIST [6], FEI [7] and FERET [8] datasets, respectively.

In 2016 authors of [9] proposed modified Hausdorff distance for automatic ear localization. This distance uses skin regions of side face image and ear template to locate the ear. Ear template was created by considering different structure of ears to detect ears of different shapes. To find the exact ear location authors automatically resized the ear template. Experimental results shows that the proposed approach is invariant to shape, pose, illumination and occlusion of ear images. Detection rate is again defined as ration between number of successful ear localizations and number of all annotated ears. Authors tested their approach on the CVL face database [10] and the ND-Collection E database [11] and obtained detection rates of 91.0% and 94.5%, respectively.

In 2017 authors of [12] improved traditional Faster Region-based Convolutional Neural Networks algorithm with Multiple Scale Faster R-CNN framework in terms of ear detection step. They evaluated their approach on three different databases, UBEAR [13], WebEar [12] and UND-J2 [14]. They achieved 100% accuracy on 1800 images from UND-J2 database, which includes ear images in controlled environment. On 200 ear images from WebEar database, which includes ear images from the web (uncontrolled environment), the approach achieved accuracy of 98%. Similar accuracy of 98.66% was achieved on 9121 ear images from UBEAR database. It must be noted that this approach uses bounding box detections.

In 2018 deep learning approaches for ear detection started to appear. The authors of [15] presented a novel ear detection technique based on convolutional encoder-decoder networks to address occlusions, ear accessories and variable illumination on images captured in unconstrained settings. Evaluation of this approach was tested on 250 ear images from AWE dataset [16], which consists of images gathered from the web (uncontrolled environment). The authors achieved accuracy of 99.4% and intersection over union score of 55.7%. As opposed to [12], this approach uses pixel-wise detections.

However, there are no standard benchmarks and evaluation methodology for ear detection. Existing approaches are evaluated on different performance metrics and also different databases which makes it harder to compare it among themselves. For example both [12] and [15] report evaluation on same performance metrics (accuracy, precision and recall). But authors in [12] use correct detections (true positives) on image level (number of correctly located ears) while authors in [15] use correct detections on pixel level (number of correctly classified pixels).

## III. Methodology

Mask R-CNN [1] extends Faster R-CNN [12], which uses bounding box detection. Authors of Mask R-CNN method added a branch for predicting an object mask along with the existing branch for bounding box detection in Faster R-CNN. This branch is a fully connected convolutional network, which is applied to each region of interest (RoI) and predicts a pixel to pixel segmentation mask. The main advantage of Mask R-CNN is locating exact pixels of each object instead of just bounding boxes. We begin with a quick overview of Mask RCNN's predecessor.

### *A. Faster R-CNN*

Faster R-CNN consists of two stages, one being Fast R-CNN [17] and the other being the region proposal network [12]. Fast R-CNN includes a feature extractor, classifier and bounding box regressor in a single convolutional neural network. Along with image, Fast R-CNN requires region proposals as an input. Authors of Faster R-CNN observed that convolutional feature maps calculated with Fast R-CNN could be used for generating region proposals. They managed to reuse results from Fast R-CNN and get the region proposals, which are candidates for object bounding boxes.

### *B. Mask R-CNN*

Mask R-CNN has similar stages as Faster R-CNN with added pixel-wise prediction branch. We can break down this method to five parts:

1) convolutional backbone architecture,
2) region proposal network,
3) region of interest (RoI) classifier,
4) bounding box regressor,
5) Detection pixel-wise masks.

The authors instantiated Mask R-CNN with multiple **backbone architectures**. They evaluated ResNet [18] and ResNeXt [19] networks of depth 50 or 101 layers and also explored Feature Pyramid Network (FPN), proposed in [20].

**Region proposal network** scans the image with sliding window over anchors (red squares on Fig. 1) with different size and aspect ratios [21].

As mentioned region proposal network does not scan over the image but uses one of the feature maps generated by convolutional neural network. For each anchor the region proposal network outputs foreground

Fig. 1. Red squares represent anchors, which are scanned by region proposal network. These squares are drawn on the original image for illustration. RPN actually scans over the feature maps but areas on the feature maps correspond to areas on original image. These anchors are of different sizes and usually they overlap to cover as much space as possible.

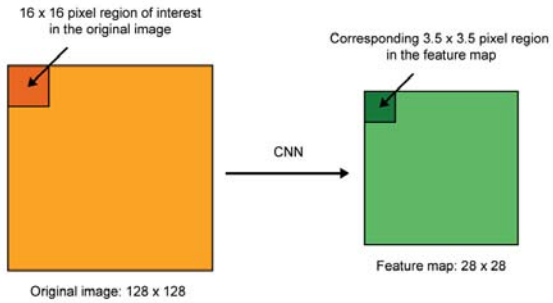



Fig. 2. Original image of size $128 \times 128$ pixels and corresponding feature map of size $28 \times 28$ pixels.

(object) or background class. Anchors that are most likely to contain objects are called regions of interest and are passed to RoI classifier [21].

The authors proposed **RoI classifier** called RoIAlign, which improves RoIPool [17] from Faster R-CNN. RoIPool is used for bounding box detections. Authors of Mask R-CNN realized that feature map calculated by RoIPool did not align with regions of the original image [22]. This is due to the fact that pixel-wise masks must be more precise that bounding box masks. RoIPool rounds RoI boundaries which leads to misalignment. If we take a look at Fig. 2, in RoIPool that would correspond to rounding region in the feature map from $3.5 \times 3.5$ pixel region to $3 \times 3$ pixel region. Contrary to RoIPool, in RoIAlign there is no rounding of RoI boundaries. However, in RoIAlign we use bilinear interpolation to get exact idea of what would be at pixel 3.5.

This classifier returns the class of the object in the region of interest. Unlike region proposal network, which has two classes, this stage is capable of classifying region of interest to more classes, such as person, car, airplane, etc.

The main purpose of **bounding box regressor** is to further refine the coordinates for the bounding box once the object has been classified.

**Detection pixel-wise masks** are the main advantage that extends Faster R-CNN. Mask branch is convolutional network which takes regions selected by ROI classifier and generates low resolution 28x28 pixels

masks for these regions [21].

### C. Implementation

To perform ear detection on ear images we use Mask R-CNN implementation available online [23]. The implementation used, however, has some differences from the paper.

Mask R-CNN implementation uses standard convolutional neural network ResNet101 proposed in [18]. In addition to ResNet101 authors of [23] included feature pyramid network from [20], which uses connections to build in-network feature pyramid [1]. As already explored in [1], excellent gains in accuracy and speed are achieved using a ResNet-FPN backbone for feature extraction with Mask R-CNN. Feature pyramid network enables better representation of objects at different scales.

All images are resized to $1024 \times 1024$ pixels to support training multiple images per batch. If image is not square it is padded with zeros, to preserve the aspect ratio [23]. Image is then converted to a feature map of shape $32 \times 32 \times 2048$ while passing through the ResNet backbone.

Instead of RoIAlign authors of implementation use TensorFlow's *crop_and_resize* function for simplicity.

## IV. Experimental results

In this section we describe images used for training, training protocol and performance metrics used to evaluate and compare Mask R-CNN ear detection approach with PED-CED approach.

### A. Training dataset

For the purpose of training Mask R-CNN for ear detection, ear images were collected from the web. In that way the images were gathered in unconstrained environment and of different resolutions. RefineNet-based [24] detector was used to detect ears on all images. We examined results and annotated which ears were correctly detected. Next step was to remove failed detections after which we ended up with 12,500 ear images with their pixel-wise detection masks which were used as our ground truth. Images were split into train and validation set of 9,500 and 3,000 images, respectively.

### B. Training

We train our model on Nvidia GeForce GTX 1070 graphics card with 8GB of memory. One image per GPU is used and we set number of epochs to 200 and number of steps per epoch to 30. For training starting point we use weights from https://github.com/matterport/Mask_RCNN/releases/download/v2.1/mask_rcnn_balloon.h5.

### C. Performance metrics

We used the same performance metrics as in [15]. We have only two classes, ear and non-ear. We define:

- TP (true positives): number of pixels that are correctly classified as a part of an ear,
- TN (true negatives): number of pixels that are correctly classified as non-ear pixels,
- FP (false positives): number of non-ear pixels that are classified as a part of an ear,
- FN (false negatives): number of ear pixels that are classified as non-ear pixels.

First we define detection accuracy as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (1)$$

This accuracy has large TN value, since the majority class is non-ear. Hence we expect this measure to have value close to 1. Second metric is intersection over union (IoU), which is defined as

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \quad (2)$$

IoU represents a ratio between pixels that are in annotated and detected ear regions (intersection) and pixels that are in union of annotated and detected ear regions. Perfect overlap gives us score of 1 while completely failed detection gives us score of 0. Third metric is recall, defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (3)$$

which tells us how many of ear pixels were actually detected. On the other hand we define precision as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (4)$$

which tells us how many of the detected ear pixels were indeed ear pixels.

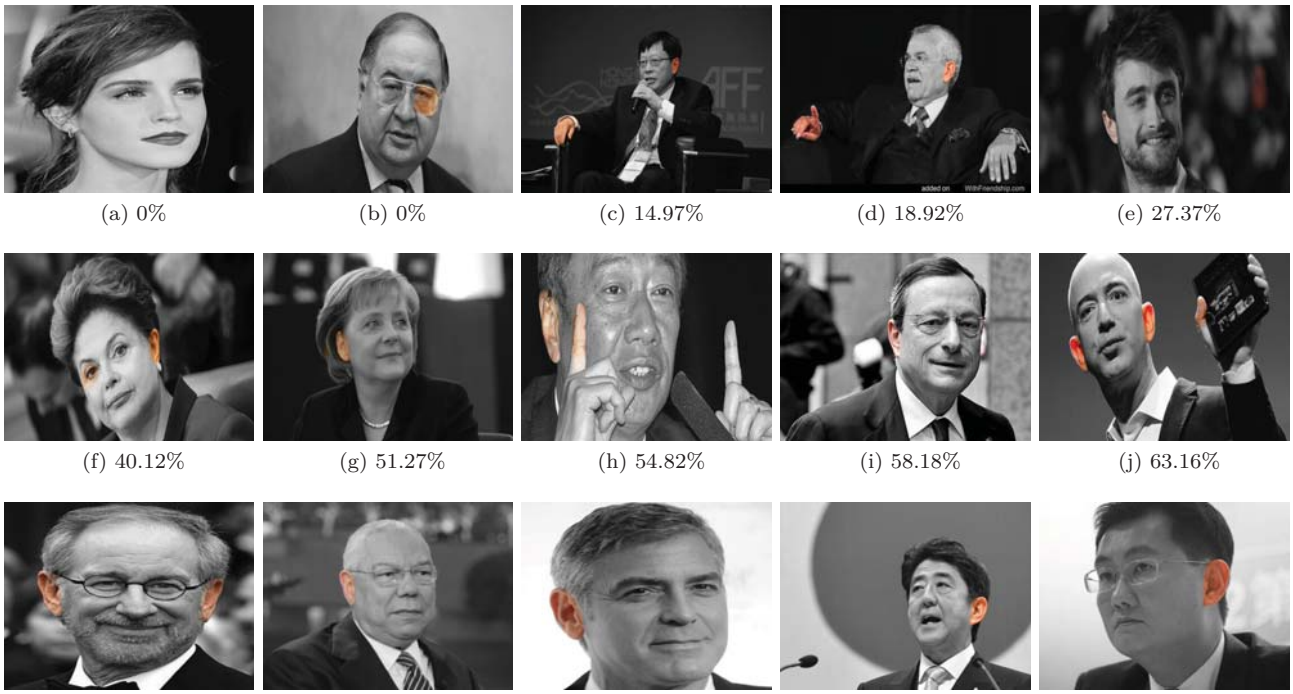
### D. Detection evaluation

We evaluate our approach on 250 test images of AWE dataset [16] and compare it to [15]. We achieve similar detection accuracy of 99.7%, which is partly the consequence of background (non-object) being the majority class. Performance comparison on AWE dataset is shown in Table I.

TABLE I
Performance comparison of Mask R-CNN and PED-CED on AWE dataset.

| Approach | IoU [%] | Precision [%] | Recall [%] |
|---|---|---|---|
| PED-CED | 55.7±25.0 | 67.7±25.7 | 77.7±32.8 |
| Mask R-CNN | **79.24±0.19** | **92.04±0.16** | **84.14±0.20** |

In addition we evaluate our approach in terms of correctly detected ears. Ears on image are correctly detected if intersection over union is above 0.5. Our approach correctly detects ears on 232 images (out of 250 test images of AWE dataset) thus achieving detection accuracy of 92.8%.



(a) 0% (b) 0% (c) 14.97% (d) 18.92% (e) 27.37%

(f) 40.12% (g) 51.27% (h) 54.82% (i) 58.18% (j) 63.16%

(k) 90.81% (l) 92.36% (m) 93.63% (n) 95.03% (o) 96.13%

Fig. 3. Detection results ordered in terms of their increasing values of intersection over union score. Colored regions are classified as ears. In the top row there are some of the worst detections, the middle row shows average detection and the last row shows some of the best detections made by Mask R-CNN model.

The best ear detection on AWE dataset has IoU score of 96.13%. Fig. 3 summarizes some of the best ear detections, some average ear detections and some of the worst ear detections. From 3a we see that occlusions in form of hair covering part of the ear is, in some cases, still a problem for our model. Mask R-CNN also incorrectly classifies glasses and eyes as ears, which can be seen on 3b and 3f, respectively. In addition, fingers (3h and 3j) and hands (3c and 3d) are also classified as ears in some cases. In these cases creases on skin might be interpreted as pinna and therefore classified as ear. Hand and finger regions are elongated shaped just like ears in most of the cases. Our model learns that ears usually appear on both sides on portrait images. We can see from 3i that Mask R-CNN tries to find ears on both sides of portrait image. It correctly finds left ear but incorrectly classifies background object as right ear due to it's specific position on the image. From last row we can see that our model correctly detects one and more ears on the image with high accuracy (in terms of IoU score).

## V. Conclusion

In this paper we describe Mask R-CNN and evaluate this approach for ear detection. We report pixel-wise performance in terms of detection accuracy, intersection over union precision and recall. We also report detection accuracy image-wise. We show that this approach performs very well for ear detection step, achieving detection accuracy of 99.7%, IoU score of 79.24%, precision of 92.04% and recall of 84.14% on AWE dataset. IoU score, precision and recall were all shown to be significantly higher than those achieved by PED-CED approach, while detection accuracy stays almost the same.

## References


[1] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[2] B. Arbab-Zavar and M. S. Nixon, "On Shape-Mediated Enrolment in Ear Biometrics," in *International Symposium on Visual Computing*, Springer. Springer, 2007, pp. 549–558.

[3] S. Prakash and P. Gupta, *Ear Biometrics in 2D and 3D: Localization and Recognition*. Springer, 2015, vol. 10.

[4] A. Pflug and C. Busch, "Ear biometrics: A survey of detection, feature extraction and recognition methods," *IET Biometrics*, vol. 1, no. 2, pp. 114–129, 2012.

[5] P. Chidananda, P. Srinivas, K. Manikantan, and S. Ramachandran, "Entropy-cum-hough-transform-based ear detection using ellipsoid particle swarm optimization," *Machine Vision and Applications*, vol. 26, no. 2-3, pp. 185–203, 2015.

[6] U. of Sheffield, "The sheffield (previously umist) face database," https://www.sheffield.ac.uk/eee/research/iel/research/face, 1998, accessed 25 December 2018.

[7] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, pp. 902–913, 2010, https://www.sheffield.ac.uk/eee/research/iel/research/face, accessed 25 December 2018.

[8] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[9] P. P. Sarangi, M. Panda, B. P. Mishra, and S. Dehuri, "An automated ear localization technique based on modified hausdorff distance," in *Proceedings of International Confer-*

*ence on Computer Vision and Image Processing*. Springer, 2016, pp. 229–240.
[10] P. Peer, "Cvl face database," http://www.lrv.fri.uni-lj.si/facedb.html, 1999, accessed 25 December 2018.
[11] U. of Notre Dame, "Nd-collection e database," https://cvrl.nd.edu/projects/data/#nd-collection-e, 2002, accessed 25 December 2018.
[12] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497
[13] R. Raposo, E. Hoyle, A. Peixinho, and H. Proença, "Ubear: A dataset of ear images captured on-the-move in uncontrolled conditions," in *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*. IEEE, 2011, pp. 84–90.
[14] P. Yan and K. W. Bowyer, "Biometric recognition using 3d ear shape," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 8, pp. 1297–1308, 2007.
[15] Ž. Emeršič, L. L. Gabriel, V. Štruc, and P. Peer, "Convolutional encoder–decoder networks for pixel-wise ear detection and segmentation," *IET Biometrics*, vol. 7, no. 3, pp. 175–184, 2018.
[16] Ž. Emeršič, V. Štruc, and P. Peer, "Ear Recognition: More Than a Survey," *Neurocomputing*, 2017.
[17] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385
[19] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016. [Online]. Available: http://arxiv.org/abs/1611.05431
[20] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144
[21] W. Abdulla, "Splash of color: Instance segmentation with mask r-cnn and tensorflow," https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46, 2018.
[22] D. Parthasarathy, "A brief history of cnns in image segmentation: From r-cnn to mask r-cnn," https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4, 2017.
[23] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.
[24] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." in *Cvpr*, vol. 1, no. 2, 2017, p. 5.