# Face Detection and Recognition Based on General Purpose DNN Object Detector

Veta Ghenescu
*Institute of Space Science*
Bucharest, Romania
ghenescu@spacescience.ro

Roxana Elena Mihaescu
*UTI Grup*
Bucharest, Romania
roxana.mihaescu@uti.ro

Serban-Vasile Carata
*Institute of Space Science*
*UTI Grup*
Bucharest, Romania
serban.carata@spacescience.ro

Marian Traian Ghenescu
*Institute of Space Science*
*UTI Grup*
Bucharest, Romania
ghenescu@spacescience.ro

Eduard Barnoviciu
*UTI Grup*
Bucharest, Romania
eduard.barnoviciu@uti.ro

Mihai Chindea
*UTI Grup*
Bucharest, Romania
mihai.chindea@uti.ro

*Abstract*—**Facial recognition is a very on-demand task today. What makes a facial detection and recognition system good is its robustness to changes in illumination, rotation and tilt, position and occlusion. In this paper we present a face recognition system trained on a proprietary datatabase. Our method is based on YOLO (You Only Look Once) model, which is a general object detection system, that by modifying and testing different internal parameters we have proven it to be adequate for face recognition. The database was created especially for this project, with images featuring 10 different subjects, different orientations, and illumination conditions, totalling to 120.000 samples. The models were based on 17 variations of the architecture of Darknet-19 neural network, from which we have chosen the one with the best results.**

*Index Terms*—**Face recognition, Deep Neural Network, YOLO - You Only Look Once, Darknet**

## I. Introduction

Face recognition is one of the most popular application of object recognition used nowadays. Even though these algorithms are in high demand, the problem of face recognition is not yet solved. This field of work is constantly developing, newer and better algorithms being implemented quite often.

Facial recognition has several real life applications, ranging from public to personal security. Those systems are used for identity verification, or in localizing and tracking a possible suspect in a crowd. Furthermore, face recognition is used in personal devices like laptops, smart-phones, etc., as a method to unlock them.

The most popular approach in face recognition is to train a neural network on a proper database. For this purpose were created several databases [1] that contain a large number of images. Those images are as varied as possible, in order to obtain a system that is robust to changes in illumination, rotation and tilt, position and occlusion.

Our face recognition method is based on YOLO (You Only Look Once) model, which uses the CNN(Convolutional Neural Network) [2] Darknet-19. YOLO is a general purpose object detection system, that we modified in order to recognize different instances of the same class.

This approach is defined by two main parts: the neural network and the database. Regarding the database, we have created one of our own, containing 120.000 samples from 10 subjects. Furthermore, we have modified the architecture of Darknet network, which has an input resolution of 416x416 pixels, and the following layer configuration: in the first convolutional layer, it has 32 filters, while in the last one, ends with 1024 filters. After we modified the number of filters and the input resolution, we obtained 17 different variations of this network, which we trained on our database.

The outline of the paper is as follows. The proposed method, including the second version of the YOLO model and our own configurations of the network are reviewed in Section II. Afterwards, we describe our proprietary database in Section III, while in Sections IV and V are presented the experimental results, and respectively, the conclusions.

## II. Proposed Method

In this section we will present our proposed algorithm, starting with the YOLO model and, in the final subsection, continuing with the description of the architectures that we built.

### A. YOLO model

In this paper we use the YOLOv2 model [3], an improved version of the original YOLO detection system [4]. Even though now is available YOLOv3 model, when we developed our algorithm, YOLOv2 was the last version of the model.

YOLOv2 optimizes the R-CNN (Regions-Convolutional Neural Network) [5], being a more general and faster detection system. This model handles object recognition as a single regression problem, and at the same time it analysis the whole image at once during the training stage.

Firstly, the second version brings improvements regarding the performance of the localization, while preserving the accuracy of classification. Secondly, YOLOv2 introduces many novel concepts that lead to better performances of the detection system.

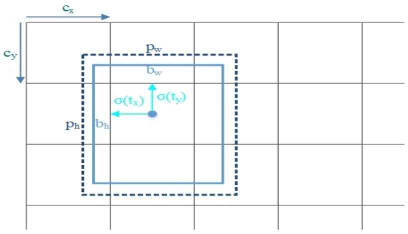

Fig. 1. Bounding box coordinates prediction [3]

Using *batch normalization* on the convolutional layers, has two effects: it enhances the convergence, and also excludes the need of other regularization methods [6]. Compared to the first model, which required an input resolution of 224 x 224 pixels, YOLOv2 uses *high resolution classifiers* (448 x 448 pixels).

The new model does not predict the coordinates for bounding boxes from the fully connected layers, like the original model, and instead uses *anchor boxes*. This leads to an increased recall, at the cost of a slight decrease in accuracy. The network predicts 5 bounding boxes, each having 5 coordinates: $t_x$, $t_y$, $t_w$, $t_h$, $t_o$. The predictions $b_x$, $b_y$, $b_w$, $b_h$ are calculated using these coordinates, the offsets of the cell ($c_x$, $c_y$) and the dimensions of the bounding boxes ($p_w$, $p_h$), as shown in Figure 1. To compute the predictions, the following equations are used:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

$$Pr(object) * IOU(b, object) = \sigma(t_o) \tag{5}$$

By using anchor boxes, the input resolution is reduced at 416 x 416 pixels.

Beside the localization and detection precision, YOLOv2 is also faster than the previous model. It uses the Darknet-19 [7] neural network, whose architecture is shown in Figure 2. All the layers of the network are described in Table I.

Applying all of these novel techniques results in a faster model, which leads to increased performances in localization and detection.

### B. Proposed Darknet architectures

The purpose of this paper is to implement a face recognition system, based on the second version of YOLO model. In this section we describe the networks that we used in order to detect the most accurate one.

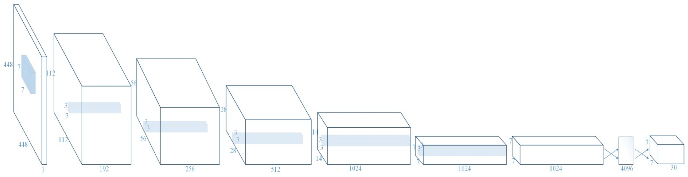

Fig. 2. Architecture of Darknet-19 neural network [3]

TABLE I
YOLO DARKNET INTERNAL LAYER DESCRIPTION

| Layer type | Number of filters | Dimension | Step | Output |
|---|---|---|---|---|
| Convolutional | 32 | 3 x 3 | | 224x224 |
| Maxpool | NaN | 2 x 2 | 2 | 112x112 |
| Convolutional | 64 | 3 x 3 | | 112x112 |
| Maxpool | NaN | 2 x 2 | 2 | 56x56 |
| Convolutional | 128 | 3 x 3 | | 56x56 |
| Convolutional | 64 | 1 x 1 | | 56x56 |
| Convolutional | 128 | 3 x 3 | | 56x56 |
| Maxpool | NaN | 2 x 2 | 2 | 28x28 |
| Convolutional | 256 | 3 x 3 | | 28x28 |
| Convolutional | 128 | 1 x 1 | | 28x28 |
| Convolutional | 256 | 3 x 3 | | 28x28 |
| Maxpool | NaN | 2 x 2 | 2 | 14x14 |
| Convolutional | 512 | 3 x 3 | | 14x14 |
| Convolutional | 256 | 1 x 1 | | 14x14 |
| Convolutional | 512 | 3 x 3 | | 14x14 |
| Convolutional | 256 | 1 x 1 | | 14x14 |
| Convolutional | 512 | 3 x 3 | | 14x14 |
| Maxpool | NaN | 2 x 2 | 2 | 7x7 |
| Convolutional | 1024 | 3 x 3 | | 7x7 |
| Convolutional | 512 | 1 x 1 | | 7x7 |
| Convolutional | 1024 | 3 x 3 | | 7x7 |
| Convolutional | 512 | 1 x 1 | | 7x7 |
| Convolutional | 1024 | 3 x 3 | | 7x7 |
| Convolutional | 1000 | 1 x 1 | | 7x7 |
| Avgpool | NaN | Global | NaN | 1000 |
| Softmax | NaN | NaN | NaN | NaN |

We have started from the Darknet network, which is described in Table I. When the network is trained for detection, the last convolutional layer with 1000 filters, is replaced by another three convolutional layers, each having 1024 filters, followed by a final layer, which has a variable number of filters. The number of filters of the last convolutional layer is calculated considering the number of classes that we have to recognize, according to the following equation:

$$N_{filters} = 5(5 + N_{classes}) \tag{6}$$

In this method we have 10 classes, so we set the number of filters of the last layer at 75 filters. In order to find the network that leads to best results regarding face recognition, we modified the other layers.

Firstly, we have changed the numbers of filters. So, in contrast to the original network which has 32 filters in the first convolutional layer and 1024 filters in the last one, we have generated several networks with variable number of filters. The first one has from 1 filter, to 32 filters, while the last generated network has from 32 filters, in the first layer, to 1024 in the last one. Secondly, we have changed also the input resolution. While the original network has an input of 416 x 416 pixels, we have used also a smaller resolution, 416 x 208 pixels. In this way, we have generated 17 different networks, which are described in Table II.

TABLE II
DARKNET ARCHITECTURES USED IN OUR PROPOSED METHOD

| Network Index | Input resolution | Number of filters | |
|---|---|---|---|
| | | First layer | Last layer |
| 1 | 416x416x3 | 1 | 32 |
| 2 | 416x208x3 | 1 | 32 |
| 3 | 416x416x3 | 4 | 128 |
| 4 | 416x208x3 | 4 | 128 |
| 5 | 416x416x3 | 8 | 256 |
| 6 | 416x208x3 | 8 | 256 |
| 7 | 416x416x3 | 12 | 384 |
| 8 | 416x208x3 | 12 | 384 |
| 9 | 416x416x3 | 16 | 512 |
| 10 | 416x208x3 | 16 | 512 |
| 11 | 416x416x3 | 20 | 640 |
| 12 | 416x208x3 | 20 | 640 |
| 13 | 416x416x3 | 24 | 768 |
| 14 | 416x208x3 | 24 | 768 |
| 15 | 416x416x3 | 28 | 896 |
| 16 | 416x208x3 | 28 | 896 |
| 17 | 416x416x3 | 32 | 1024 |

Each of the described networks has been trained on a data set, for a number of 300.000 generations. Comparing all the trained models, we were able to determine the network that leads to the best performances.

## III. DATABASE

In this section we will describe the database that we have created. The existing publicly available data sets contain images with small resolutions, while our proprietary database consists of images with higher resolution (1920x1080 pixels).

The database has an important role in face detection and recognition. It has to contain a variety of samples for every class, in order to obtain a robust system, invariant to light, position or size changes, and rotations of the head.

### A. Factors that influence the face recognition system

The performance of a face recognition system is influenced by several factors [8] as:

**Input resolution.** Generally, as the input resolution increases, so does the accuracy [9] of the system, but also is increased the processing time.

**Face resolution.** An increased resolution of the region of interest leads to better performances.

**Image quality.** The accuracy is affected when the SNR(Signal-to-Noise Ratio) decreases.

**Lighting conditions.** Depending on how dark the images are, the performances of the system will vary. A brighter image will lead to better results.

**Rotation angle.** The positions of faces can be very diversified, and this can make the detection and recognition more difficult. As the vertical and horizontal deviations increase, the performances drop.

**Occlusions.** The presence of accessories, can hide important facial features, making the recognition more difficult.

To make the model as robust as possible, when we created our database, we included images with a large variety of those factors.

### B. Creating the database

The first step, before creating the database, was to choose the conditions in which the samples were taken, and the subjects. Regarding the subjects, we chose them to be as visually varied as possible(different skin tone, facial hair, presence of accessories). The database contains images with 10 people: 4 women and 6 men. In Table III we described all of the variations regarding the distance between the camera and the subject and the horizontal and vertical deviations. Those variations were applied for every subject from the database.

TABLE III
VARIATIONS OF THE SUBJECTS POSITION

| d | Vertical Deviation | Horizontal Deviation | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1m | -10° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |
| | 0° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |
| | 10° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |
| 3m | -10° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |
| | 0° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |
| | 10° | -90° | -45° | -15° | 0° | +15° | +45° | +90° |

where d represents the distance between the camera and the subject.

Finally, we obtained around 12000 samples for each class, for a total of 120000. In Figure 3 are shown some samples with one of the 10 subjects, extracted from our database.

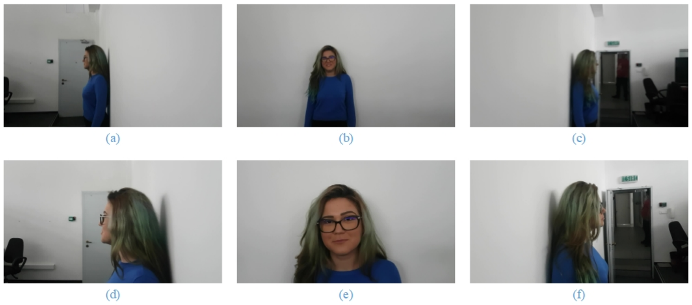
(a)
(b)
(c)
(d)
(e)
(f)

Fig. 3. Sample from our database

## IV. EXPERIMENTAL RESULTS

In this section we will describe the results obtained after the testing stage.

The first step was to train all of the neural networks from Table II with a training set of 15000 images from the database. For the training stage we chose an equal number of images for each class. Those images were chose to be as visually varied as possible. All the networks were trained for 300.000 generations each. The training stage was realized on Nvidia 1080Ti graphics card. For each network, the training lasted almost 30 hours.

In order to find the most accurate network, we have tested all of the networks and compared the results obtained. For the first test, we used 3000 images from the database, with all the subjects. We compared the predictions made by each network, with the manually annotated images, and we obtained the results from Table IV.

TABLE IV
RESULTS OBTAINED DURING THE FIRST TESTING STAGE

| Network Index | Results (%) after | | |
|---|---|---|---|
| | 100.000 generations | 200.000 generations | 300.000 generations |
| 7 | 61.607678 | 59.148170 | 54.769046 |
| 9 | 64.607079 | 76.544691 | 82.723455 |
| 10 | 50.989802 | 46.070766 | 49.430114 |
| 11 | 53.149370 | 71.685663 | 40.491902 |
| 15 | 81.463727 | 52.909418 | 65.086983 |
| 16 | 36.832633 | 60.407918 | 44.991002 |
| 17 | 46.970606 | 51.649670 | 35.392921 |

In Table IV were presented only the significant results. We selected and continued testing only the networks with an accuracy greater then 80%. In contrast to the first testing stage when we tested at every 100.000 generations, now we performed tests at every 20.000 generations, to get a more detailed view on how do the two networks evolve. The results obtained during this second stage are shown in Table V.

TABLE V
RESULTS OBTAINED DURING THE SECOND TESTING STAGE

| Number of training generations | Results (%) for | |
|---|---|---|
| | 9h network | 15th network |
| 40.000 | 38.152370 | 77.864426 |
| 60.000 | 80.773845 | 78.464307 |
| 80.000 | 86.004799 | 52.519496 |
| 100.000 | 64.367127 | 81.463707 |
| 140.000 | 31.043791 | 72.075585 |
| 180.000 | 49.610078 | 75.884823 |
| 200.000 | 76.184763 | 53.599280 |
| 220.000 | 81.853629 | 72.3775525 |
| 240.000 | 72.675465 | 78.794241 |
| 260.000 | 45.530894 | 76.064787 |
| 280.000 | 89.652070 | 72.135573 |
| 300.000 | 82.333533 | 64.877025 |

As can be seen in Table V, we obtained the best results in face detection and recognition, by using the 9th configuration. This network starts with 16 filters in the first convolutional layer and ends with 512 filters in the last one, having an input resolution of 416x416 pixels.

We trained further this network, up to 600.000 generations, using this time a training data set containing an additional 30.000 images. Contrary to the first 15.000 images that were manually annotated, these images were obtained by using automatic annotation on the remaining 105.000 images. The used method was Haar-Cascade facial detection method [10], with a 28.57% accuracy.

The result obtained by further testing this network did not show any significant improvements, the best accuracy remaining 89.65%.

## V. CONCLUSIONS

In this paper, we have shown that the YOLO model, which is a general object detection system, can be trained to recognize several instances of the same class, in this case being used for facial recognition. For our method, we conceived 17 different configurations based on Darknet-19 neural network, by varying the input resolution and the number of filters from convolutional layers. These networks were trained on our proprietary database. The database was created as varied as possible, in order to obtain a system that is robust to position, orientation and lighting variations. As demonstrated during test results, our method is able to efficiently detect and recognize the subjects from the database, leading to a top accuracy of 89.65%. This result was obtained with the 9th configuration, which has 16 filters in the first convolutional layer, and ends with 512 filters in the last one, and an input resolution of 416x416 pixels.

Furthermore, besides the high accuracy in detecting and recognizing the subjects, our proposed system is fast, providing a good balance of speed and accuracy. It can process 45 frames per second. These features make the system described in this paper to be appropriate for real-time applications [11].

## ACKNOWLEDGMENT

This work was supported by the Romanian Space Agency (ROSA), contract STAR-C3-160/2017 and Romanian UEFISCDI Agency under projects PN3-483/2017 and PN3-PED-147/2017.

## REFERENCES

[1] R. Gross, "Face databases," in *Handbook of Face Recognition*, A. S.Li, Ed. New York: Springer, February 2005.

[2] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[3] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[7] J. Redmon, "Darknet: Open source neural networks in c," http://pjreddie.com/darknet/, 2013–2016.

[8] J. R. Beveridge, G. H. Givens, P. J. Phillips, and B. A. Draper, "Factors that influence algorithm performance in the face recognition grand challenge," *Computer Vision and Image Understanding*, vol. 113, no. 6, pp. 750–762, 2009.

[9] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[11] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction." in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, vol. 5. IEEE, 2003, pp. 53–53.