# Lightweight Two-Stream Convolutional Face Detection

Danai Triantafyllidou, Paraskevi Nousi and Anastasios Tefas
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
danaitri22@gmail.com, paranous@csd.auth.gr, tefas@aiia.csd.auth.gr

***Abstract*—Video capturing using Unmanned Aerial Vehicles provides cinematographers with impressive shots but requires very adept handling of both the drone and the camera. Deep Learning techniques can be utilized in this process to facilitate the video shooting process by allowing the drone to analyze its input and make intelligent decisions regarding its flight path. Fast and accurate on-board face detection for example can lead the drone towards capturing opportunistic shots, e.g., close ups of persons of importance. However, the constraints imposed by the drones' on-board processing power and memory prohibit the utilization of computationally expensive models. In this paper, we propose a lightweight two-stream fully Convolutional Neural Network for face detection, capable of detecting faces in various settings in real-time using the limited processing power Unmanned Aerial Vehicles possess.**

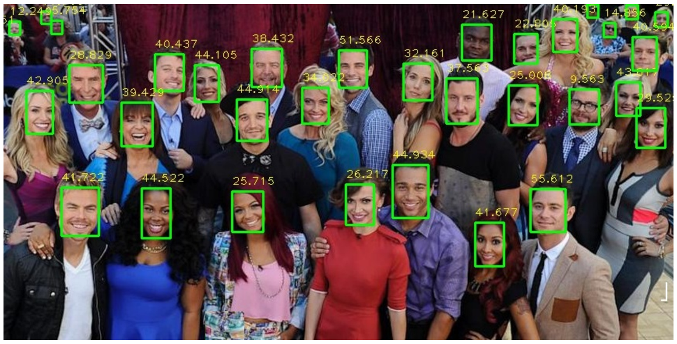


Fig. 1. An example of face detection in various poses and occlusions. The bounding boxes show output of the trained CNN. The scores shown are the result of the last convolutional layer of the face detection stream.

## I. Introduction

Video shooting using Unmanned Aerial Vehicles (UAVs), or drones, allows for capturing impressive shots, but requires adept manipulations of both the drone and the camera. Thus, capturing shots that are meaningful cinematography wise involves real-time control of the drone's course as well as the the camera's angle at the same time. There are also safety constraints to be taken into account, such as privacy issues or safe landing sites among others.

Certain aspects of the video shooting process with drones may be aided with Machine Learning techniques, as nowadays professional as well as commercial drones contain on-board processing units. Utilizing Machine Learning techniques in the video shooting process can assist the capturing of opportunistic shots, e.g., by first recognizing a person of importance the drone could then fly closer and capture close-up shots of that person. Drones with Graphics Processing Units (GPUs) in particular can be aided by Deep Learning techniques, as GPUs routinely speed up common operations such as matrix multiplications.

One such aspect of the video shooting process is that of face detection, as detecting and analyzing faces can lead the drone to capture important moments. Face detection is the first step to face and facial expression recognition and tracking, among other tasks which can be greatly beneficial to the quality of shots captured by the drone. For example, in sports events, if the drone detects the face of an important athlete, it can then be programmed to follow that person while capturing opportunistic shots, such as the athlete performing a remarkable feat or smiling after winning, etc.

Recently, Convolutional Neural Networks (CNNs) have been used for the task of face detection with great results [1]–[3]. However, using such models on drones for real-time face detection is prohibited by the hardware constraints, such as limited processing power and low memory, that drones impose. Fast execution time is of the essence for models designated to run on-board, especially as drones must also resolve other important issues such as obstacle avoidance, re-planning, SLAM [4], etc. Utilizing large models such as the aforementioned ones for face detection on drones becomes prohibitive by the low processing power and memory present on drones. The fully connected layers in the aforementioned models contribute the most to their time performance as they consist of significantly more parameters than their convolutional counterparts. Thus, a fully convolutional neural network, i.e., one that does not contain fully connected components, is better suited for the task of face detection when processing power is limited.

This observation comprises the intuition behind the design of our proposed fully convolutional model. Without fully connected layers, the proposed model is very light-weight and exhibits minimal computational complexity in both the training and testing process, while attaining great performance detection-wise. By utilizing multiple convolutional layers, our model is able to produce heatmaps which indicate both a) the existence or not of a face in the given image and b) the dimensions of a detected face. This is achieved by training a two-stream convolutional network with a number of common layers. More specifically, four layers of convolutions are applied to the input before the network breaks off into two streams, each consisting of three layers and responsible for its assigned task (i.e., face detection and size of face regression).

This paper's main contributions include the proposition of a very light-weight model which consists of only 106.123 free parameters while still accurately detecting faces of variable sizes, capable of performing both face detection and face size regression at the same time by utilizing a two-stream convolutional neural network. The model is comprised by ten convolutional layers in total, with each stream containing three layers, making the network deep. However, the number of channels per convolutional layer does not exceed 48, contributing to the model's fast performance. Furthermore, a novel training method which involves the addition of progressively harder positive and negative examples is proposed to train these lightweight models effectively. Properly training a small and lightweight network can lead to improved performance of the network over larger architectures.

The proposed fully convolutional two-stream neural network is evaluated on the challenging WIDER dataset [5], which contains faces with high variations in size, pose, occlusions etc, as well as on the FDDB dataset [6]. An example of face detection where the faces exhibit various poses and occlusions is shown in Figure 1. Because of the network's computationally lightweight components, the model is suitable for on-board use on drones to achieve very fast and accurate face detection and facilitate the video shooting process.

The rest of this paper is organized as follows. Section II presents previous work related to the task of face detection. Section III analyzes our proposed method, which is evaluated in Section IV. Last, our conclusions are summarized in Section V.

## II. Related Work

The work proposed by Viola and Jones [7], was the first method to apply Haar-like features in a cascaded classifier and made real-time face detection possible. Thereafter, the main line of research was focused on the combination of robust descriptors with classifiers. Much effort has been also devoted to replacing Haar-like features with more complicated ones like SURF [8], HOG, ACF [9], and NPD [10]. In [11], a joint-cascade method achieved excellent results by introducing an alignment step in the cascade structure. However, complex cascade methods increase the computational cost and often require pose/orientation annotations. Another common approach to face detection is to deploy a deformable parts-based model to model the information between facial parts [12], [13]. Such methods, have critical drawbacks as they lack in computational efficiency and are prohibitive of real-time detection.

In recent years, Deep Convolutional Neural networks (CNNs) have dominated many tasks of computer vision as they, in most cases, significantly outperform traditional methods. Along with the popularity of deep learning in computer vision, deep learning approaches have been explored for face detection tasks. A deep network named Alexnet [14], which was trained on ILSVRC 2012 [15], rekindled interest in convolutional neural networks and outperformed all other methods used for large scale image classification. The R-CNN method proposed in [16] generates category-independent region proposals and uses a CNN to extract a feature vector from each region. Then it applies a set of class-specific linear SVMs to recognize the object category. In [3], a cascade of CNNs was proposed which consists of 6 CNNs and operates on multiple resolutions. In [1] a deep CNN with three output branches for face/non-face classification, face pose estimation and facial landmarks localization was proposed. The model consists of three convolutional layers each followed by a max pooling layer and the last pooling layer is followed by a fully connected layer, whose output comprises the input of the three aforementioned branches. Recently, a face detector called DDFD, [2], showed that a CNN can detect faces in a wide range of orientations using a single model. The model accepts input images of size 227 × 227, and scales images up or down to detect faces larger or smaller than this size respectively.

All the above methods use very complex networks having more than 1M number of parameters that renders them inappropriate for large-scale visual information analysis or real-time face detection with constrained computational power. In contrast, our proposed model is capable of running on drones with limited processing power.

## III. Proposed Method

### A. CNN Architecture

We trained a fully convolutional neural network comprised of ten convolutional layers interspersed by dropout layers. The architecture of the CNN is summarized in Figure 2, where each convolutional layer is accompanied by a PReLU activation and a dropout layer. Table I displays the number of trainable parameters, where *conv* accompanied by an index denotes a convolutional layer and *prelu* accompanied by the same index denotes the respective activation layer. Our model has two output branches, one for face/no face classification (layers *conv#-det* and *prelu#-det*) and one for bounding box regression (layers *conv#-reg* and *prelu#-reg*). A softmax function is applied to the output of the last convolutional layer of the detection branch, which produces probabilities for the face/non-face task. Channel-wise parameters were learned for each PReLU layer. The regression branch was trained with positive examples of size 32 × 32 to predict the width to height ratio of face examples and was connected to the fourth layer of the CNN trained for the task of face detection in a parallel manner as shown in Figure 2. Training this network requires the optimization of 106.123 free parameters in total.

In all our experiments, we start with a learning rate of 0.001 for the first 200.000 iterations and then lower it to 0.0001. The CNN was trained using Stohastic Gradient Descent (SGD). The weights of the network were initialized using the Xavier method [17].

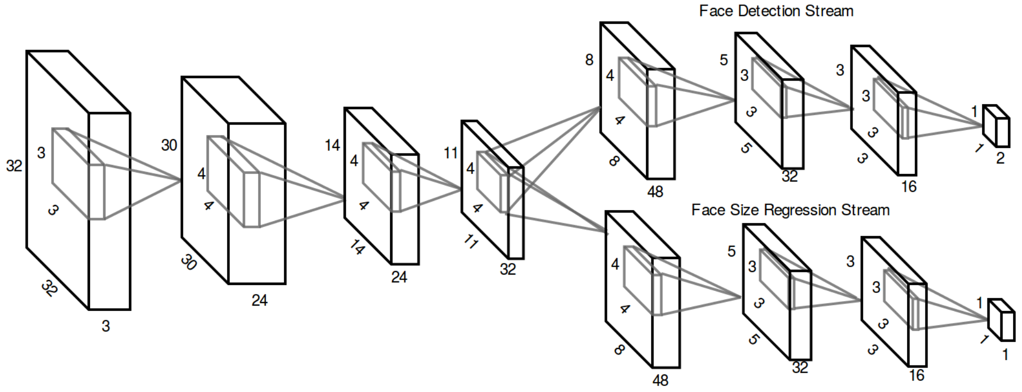


Fig. 2. Architecture of the proposed two-stream convolutional face detector.

TABLE I
CNN ARCHITECTURE

| layer | kernel | filters | output | parameters |
|---|---|---|---|---|
| Common Convolutional Layers | | | | |
| conv1 | $3 \times 3$ | 24 | $30 \times 30 \times 24$ | 648 |
| prelu1 | | | $30 \times 30 \times 24$ | 24 |
| conv2 | $4 \times 4$ | 24 | $14 \times 14 \times 24$ | 9216 |
| prelu2 | | | $14 \times 14 \times 24$ | 24 |
| conv3 | $4 \times 4$ | 32 | $11 \times 11 \times 32$ | 12288 |
| prelu3 | | | $11 \times 11 \times 32$ | 32 |
| conv4 | $4 \times 4$ | 48 | $8 \times 8 \times 48$ | 24576 |
| prelu4 | | | $8 \times 8 \times 48$ | 48 |
| Bounding Box Regression Convolutional Layers | | | | |
| conv1-reg | $4 \times 4$ | 32 | $5 \times 5 \times 32$ | 24576 |
| prelu1-reg | | | $5 \times 5 \times 32$ | 32 |
| conv2-reg | $3 \times 3$ | 16 | $3 \times 3 \times 16$ | 4608 |
| prelu2-reg | | | $5 \times 5 \times 32$ | 32 |
| conv3-reg | $3 \times 3$ | 1 | $1 \times 1 \times 1$ | 144 |
| Face Detection Convolutional Layers | | | | |
| conv5-det | $4 \times 4$ | 32 | $5 \times 5 \times 32$ | 24576 |
| prelu5-det | | | $5 \times 5 \times 32$ | 32 |
| conv6-det | $3 \times 3$ | 16 | $3 \times 3 \times 16$ | 4608 |
| prelu6-det | | | $3 \times 3 \times 16$ | 16 |
| conv7-det | $3 \times 3$ | 2 | $1 \times 1 \times 2$ | 288 |

### *B. Progressive positive and hard negative example mining*

The light-weight architecture of the proposed model establishes the need for an effective training methodology, to allow the model to accurately detect faces and correctly localize the predicted bounding boxes. Intuitively, the model should learn easier positive examples first, followed by progressively harder positive examples as the training process proceeds and converges. As the network learns to accurately detect easy examples, slightly harder ones are added to the training dataset. The difficulty of a training example is determined by the probability produced by the network itself for this particular training image. We call this method of adding positive examples to the training set *progressive positive example mining*.

A similar intuition is followed for the purpose of collecting negative examples. Hard negative examples must be collected in conjunction to the positive ones to avoid false positives and force the training process to differentiate between faces and examples mistaken for faces. Given some images which serve as negative examples, the network produces scores that represent the probability that these images depict faces. The higher the score, the harder the example is for the network to distinguish. Thus, such examples must be added to the training set of the network first to guide the training process. This process simulates *hard negative example mining*.

Given an initial set of both positive and negative examples, a new set of negative and positive examples are fed into the network and scores are produced for both sets. False positives from the set of negative examples for which the network produces a high score are considered as hard negative examples and added to the dataset. Respectively, false negatives from the set of positive examples for which the network produces a high score are considered as easy positive examples and appended to the dataset. The network is trained with the newly augmented dataset and the process is repeated iteratively.

The process of gradual training in stages, as described, resolves a significantly important issue which was indeed validated in practice: in the event of a training set being unequally distributed between the two classes, a training batch may contain little to no actual samples of one of the classes. As a result, the network may be deprived of the presence of samples of said class and, by extension, the ability to identify between the two classes may be negatively impacted.

### *C. Training dataset*

The CNN was trained with positive examples extracted from the AFLW [18], MTFL [19] and WIDER FACE [5] datasets. The training images include real world examples and are rich in with expression, pose, gender, age and ethnicity variations. The first consists of 21K images with 24K face annotations, MTFL consists of 12K face annotations, and WIDER FACE contains 32K images with about 390K face annotations, with half of these intended for training. The number of the training images was increased by applying horizontal mirroring (flip) so as to achieve better generalization of the CNN.

## IV. EXPERIMENTS

### *A. System Analysis*

The first output branch of our model contains the probability scores of the CNN for every $32 \times 32$ region in the original input image with a stride of 2 pixels. As stated previously, our model is fully convolutional and therefore able to produce a classification heatmap of the given input. Non-maximum suppression (NMS) is applied to eliminate highly overlapped detection regions. Firstly, we sort all the bounding boxes according to their score. Let $max\_score$ be the maximum score of all boxes and $s_i$ the score of the $i^{th}$ bounding box. If $s_i < 0.95 * max\_score$ the bounding box is removed. Secondly, we group the remaining bounding boxes using OpenCV [20]. The final position of the bounding box is calculated by averaging all the bounding boxes. The same applies for the calculation of the final probabilty score.

In order to detect faces smaller or larger than $32 \times 32$ we scale the original image up or down respectively. An image pyramid is built from the image to cover faces at different scales. At each level $i$ of the pyramid, the image is resized by a factor of $2^{-i/step}$. Positive values of $i$ scale down the original image, while negative values allow the detection of faces smaller than $32 \times 32$. In our experiments, the $step$ parameter is set to 8. However, even by setting it to 4 it can provide formidable detection results. During deployment of the CNN, we add an extra average pooling layer to the classification output. It has been verified that this layer reduces the number of false positives as only the heatmap pixel coordinates having neighboring coordinates with similar values are stored.

The second output branch of our model contains a map with the regression scores of the CNN. The regression layers are connected to the output of the 4-th convolutional layer of the detection branch and therefore calculate the regression scores for a volume of size $8 \times 8 \times 48$. During deployment, we multiply each dimension of the predicted bounding box by the regression score in order to acquire rectangular detection boxes.

### B. Evaluation

The proposed detector is evaluated on the FDDB dataset [6], which depicts of about 5 thousand faces. For evaluation, the toolbox provided by [21] which includes corrected annotations for the aforementioned benchmark was used. Our model achieves a 91.7% recall rate on this dataset, as shown in Figure 3. Recently several CNNs that are based on VGG or on ResNet50 with a corresponding number of parameters exceeding 1M have been proposed to deal with small resolution faces using image pyramids. The resulting models are very slow and cannot perform real-time face detection even when state-of-the-art desktop GPUs are used, making them unsuitable for use on drones due to the existing hardware limitations.

The detector is also evaluated on the WIDER Face Dataset [5]. The images are split into 61 event categories containing 32K images in total which depict about 393K faces. The dataset is split into training, validation and testing sets, each containing 50%, 10% and 40% respectively of the images corresponding to each event category. Figure 4 shows the precision recall curves for this dataset. Our model (ts-CFD) achieves a recall rate of 75.2%, 71.9% and 49.3% on WIDER easy, medium and hard partitions respectively.

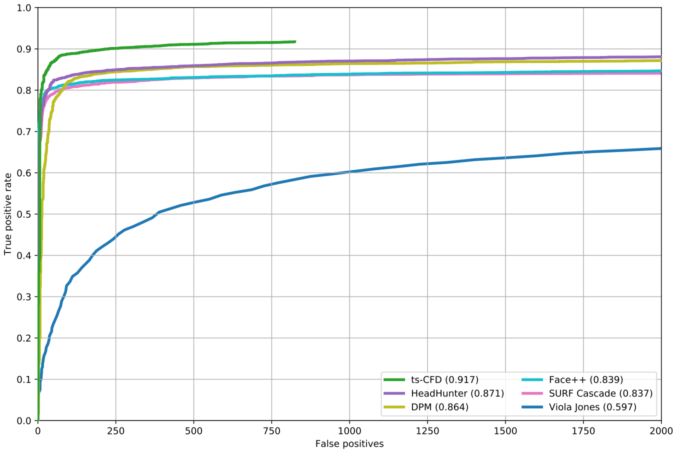


Fig. 3. Comparison of ROC curves on the FDDB dataset. Our model achieves a 91.7% recall rate.

The complexity of the competitive algorithms is very large compared to the proposed network. Our model consists of 106.123 whereas the fully convolutional AlexNet architecture proposed in [2] had 60 million parameters. Table II shows execution times for our model (denoted by ts-CFD) and the aforementioned face detector (denoted by DDFD) for RGB images of size $227 \times 227$. A variant of the Faster R-CNN for face detection [22], based on the VGG-16 architecture, is also compared (denoted by FRCNN). These experiments were conducted on a mid-range GPU with computational capabilities similar to those of on-board embedded GPUs. The results indicate that our proposed network is capable of processing 129 images per second, significantly outperforming the other techniques.

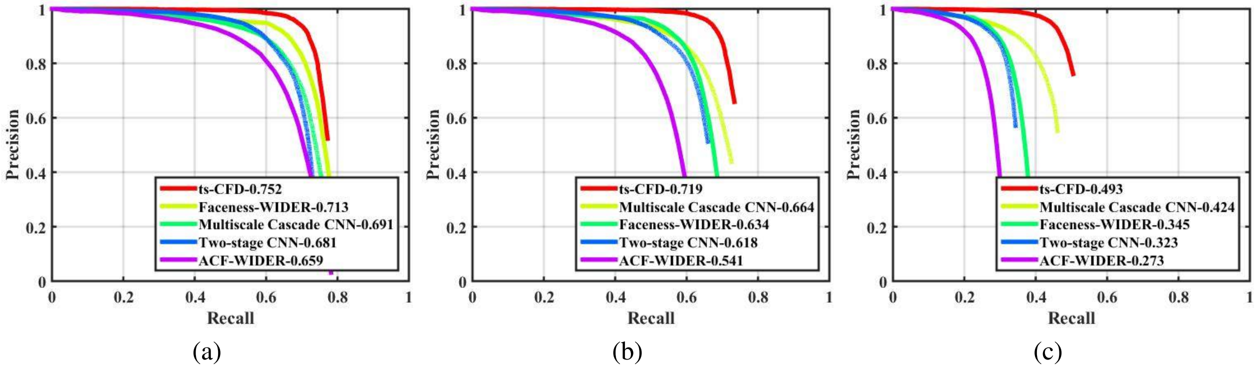
TABLE II
EXECUTION TIMES, FLOPS AND FRAMES PER SECOND COMPARISON BETWEEN THE PROPOSED CNN, DDFD AND VGG-16 ARCHITECTURE.

| | DDFD | FRCNN | ts-CFD |
|---|---|---|---|
| **Floating point operations** | 224B | 634B | 393M |
| **Time** (in seconds) | 0.035108 | 0.132594 | 0.007749 |
| **Frames per second** | 28.5 | 7.5 | 129 |

## V. CONCLUSIONS

A very fast and light-weight two-stream fully convolutional face detector was proposed in this paper, capable of predicting both the existence or not of a face in a given image as well as the size of the predicted face. The model thus encapsulates both face classification and face size regression into a common framework with a significantly small number of trainable parameters.

The model was evaluated on the very challenging WIDER dataset and found capable of detecting faces of various sizes, poses and occlusions while maintaining very low execution times. This is indicative of the fact that the model is suitable for on-board use in drones with limited processing power for real-time face detection.



(a) (b) (c)

Fig. 4. Comparison of different face detectors on WIDER dataset for the (a) easy, (b) medium and (c) hard subset of faces.

## Acknowledgment

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

## References

[1] C. Zhang and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 1036–1041.

[2] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the ACM on International Conference on Multimedia Retrieval*, 2015, pp. 643–650.

[3] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[4] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.

[5] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[6] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[8] J. Li and Y. Zhang, "Learning surf cascade for fast and accurate object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3468–3475.

[9] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proceedings of the IEEE International Joint Conference on Biometrics*, 2014, pp. 1–8.

[10] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 211–223, 2016.

[11] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 109–122.

[12] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in *Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems*, 2015, pp. 1–8.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[16] 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.

[17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[18] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2011, pp. 2144–2151.

[19] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 94–108.

[20] G. Bradski, *Dr. Dobb's Journal of Software Tools*.

[21] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 720–735.

[22] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," *arXiv preprint arXiv:1606.03473*, 2016.