**Face Detection with Deep Learning**

*Pramit Gopal Yeole*[*yeole.pr@northeastern.edu*](mailto:yeole.pr@northeastern.edu) *Northeastern University, Boston*

*Start date: 19th May 2023 Completion Date: 26th May 2023*

**ABSTRACT**

**INTRODUCTION**

Face detection is the technique to identify human faces in photos/videos.

Face detection uses machine learning and deep learning algorithms, statistical analysis, image processing techniques to pinpoint human faces within larger images. These larger images may include objects such as landscapes, sceneries, and anything other than human faces.

This uses machine learning and Neural networks for the task of detecting faces and tracking them in real-time.

Some of the applications of face detection can be in 24x7 surveillance, biometrics, law enforcement, social media, and many more.

Some of the existing methods for face detection are:

1. Viola-Jones Algorithm
2. Knowledge/rule-based face detection
3. Feature Based or feature invariant
4. Template matching
5. Appearance based
6. CNNs (Convolutional Neural Networks)
7. Single-shot detectors (SSD)

Other methods include the use of feature-based cascade classifiers using the opencv library

The state-of-the-art methods for face detection involve the use of Deep learning techniques such as Multi-Task Cascade Convolutional Neural Network (MTCNN).

Kalman Filter tracking is used to predict face position. This increases the face detection rate and also meet the real time detection requirements.

Methods for face detection:

1. Training simple classifier (Mukherjee et al. ([2017](https://link.springer.com/article/10.1007/s10462-018-9650-2#ref-CR27)))
2. Neural Networks (Mukherjee et al. ([2017](https://link.springer.com/article/10.1007/s10462-018-9650-2#ref-CR27)))
3. Convolutional Neural Networks combined with Kalman Filtering (Ren et al. ([2017](https://link.springer.com/article/10.1007/s10462-018-9650-2#ref-CR29))): CNNs are used here to detect the face in a video. But when a face is largely deflected or severely occluded, Kalman Filter tracking is used to predict face position. This increases the face detection rate and meet the real time detection requirements.
4. Deep Cascaded detection method – exploits bounding-box regression, a localization technique, to approach the detection of potential faces in images. This involves cascaded architecture with 3 stages of deep convolutional networks to predict existence of faces.

Challenges in Face Detection [1]:

These are mainly the reasons that reduce the accuracy and face detection rate.

1. Odd human expressions in a digital image
2. Face occlusion: face hidden by other objects
3. Lighting effects – varying Illuminations
4. Complex backgrounds
5. Too many faces in the image
6. Low resolution images
7. Varying skin color
8. Face orientation
9. Distance from camera

Latest advances in technology have propelled people and industries to offer the face detection as a service in the form of APIs. Some of the notable ones are Amazon Recognition, Microsoft Face API, IBM Watson visual recognition, Google cloud vision.

**DATASET/DATABASES**

Some of the standard datasets/databases of different types of faces that have been used as benchmarks for the state-of-art models are mentioned below [1]:

[Face Recognition Homepage - Databases (face-rec.org)](https://www.face-rec.org/databases/)

| **Database** | **Website** | **Description** |
| --- | --- | --- |
| MIT dataset | <http://cbcl.mit.edu/softwaredatasets/FaceData2.html> | 19 × 19 Gray-scale PGM format images. Training set: 2429 faces, 4548 non-faces. Test set: 472 faces, 23,573 non-faces |
| PIE database, CMU | [www.ri.cmu.edu](http://www.ri.cmu.edu/) | A database of 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions |
| FERET database | [www.itl.nist.gov/iad/humanid/feret/feret\_master.html](http://www.itl.nist.gov/iad/humanid/feret/feret_master.html) | It consists of 14,051 eight-bit gray-scale images of human heads with views ranging from frontal to left and right profiles |
| The Yale face database | [www.face-rec.org/databases/](http://www.face-rec.org/databases/) | Contains 165 gray-scale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink |
| Indian face database | [www.pics.stir.ac.uk/Other\_face\_databases.htm](http://www.pics.stir.ac.uk/Other_face_databases.htm) | 11 images of each of 39 men, 22 women from Indian Institute of Technology Kanpur |
| AR database | <http://www2.ece.ohio-state.edu/~aleix/> | It contains over 4000 color images corresponding to 126 people’s faces (70 men and 56 women). Features based on frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf) |
| SCface—surveillance cameras face database | [www.scface.org](http://www.scface.org/) | Images were taken in uncontrolled indoor environment using five video surveillance cameras of various qualities. Database contains 4160 static images (in visible and infrared spectrum) of 130 subjects |

For the purpose developed a standard face detection model, a custom dataset comprising of photos taken via laptop’s webcam has been prepared, and would be used to train a deep learning model for the downstream task of face detection.

**DATA DESCRIPTION**

A custom dataset containing live images taken through the webcam at different poses is utilized. The process of data collection is described below:

1. 3 batches of 30 images are captured through the web camera of the laptop/personal device. This is done using the Python “OpenCV” library.
2. Using python “labelme” software program, the captured images were labeled as having a “face” or not having a face. Iteratively bounding boxes were drawn for each image containing a face.

**METHODOLOGY – AN OVERVIEW**

Initially, a VGG16 pretrained image classification model using the Keras API has been used to classify the images based on the presence of a face or not. This is then followed by a regression model that determines the coordinates of the bounding box for the detected face. Additional layers for classification and regression are added as the final 2 layers for the task of face detection.

Loss functions:

1. Binary Cross Entropy for classification
2. Localization Loss function: this measures the error of the X and Y coordinates and the height and width of the bounding box with the actual and predicted values.

The results of implementing the model will 5 values:

1. First set of value will be whether a face was detected or not
2. The next set will be 4 values of the X, and Y coordinates of our bounding box.

For the purpose of face detection, we primarily use

**PRE-PROCESSING**

**MODELS IMPLEMENTED**

**RESULTS**

**CONCLUSION & LIMITATIONS**

**REFERENCES**

1. Kumar, A., Kaur, A. & Kumar, M. Face detection techniques: a review. *Artif Intell Rev* **52**, 927–948 (2019). <https://doi.org/10.1007/s10462-018-9650-2>
2. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.