

# **A Face Detection System using Neural Network Approach**

**Mohammad Inayatullah**

NWFP University of Engineering and Technology  
Peshawar, Pakistan  
babar@nwfpuet.edu.pk

**Shair Akbar Khan**

NWFP University of Engineering and Technology Peshawar  
Peshawar, Pakistan  
sakbar@nwfpuet.edu.pk

**Bashir Ahmad**

NWFP University of Engineering and Technology Peshawar  
Peshawar, Pakistan  
khansparty@hotmail.com

## **Abstract**

Detecting faces in images with different complex backgrounds and variation of the face in images is a complex job. In this paper, we present a neural network based upright frontal face detection system. In neural network based face detection approach, the neural network examines an incremental small window of an image to decide if there is a face contained in each window. To decrease the amount of time needed for detection, the algorithm is enhanced by processing the image before it is fed to the network. This result in even better performance as probability of error is considerably reduced.

## **1. Introduction**

In this paper, we present a neural network based algorithm to detect upright, frontal views of faces in both gray scale and color images. Several approaches to face detection have been used on the idea that a 'face' image is one event in the set of images. The neural network is trained to choose between two classes 'faces' and 'non-faces' images. But before training the neural network on these two classes, all images of the training set are first preprocessed in order to enhance images quality [1, 2]. Training a neural network for the face detection task is a challenging job due to the difficulty in characterizing 'non-face' images. It is easy to get a representative sample of images which contain faces but it is much harder to get a representative sample of those images which do not contain faces.

Due to much noise in images, extra patterns were also discovered by the neural network which made the neural network to take inaccurate decisions. In order to get rid of this problem, we added unknown variables with input nodes and assigned a very much larger value to these

unknown variables. These unknown variables attract extra attention of noise and thus avoiding the discovery of extra patterns and making the neural network results desirable and up to the mark [3].

## **2. Description of the System**

We have adopted a modular approach whereby the system was separated into smaller individual modules. Our system operates in two stages namely the Offline Stage and the Online Stage.

### **2.1 Offline Stage**

In the offline stage, a training set of two classes namely “non-faces” and “faces” were provided to the system with preprocessing techniques applied on them. Then these two classes were fed into the system randomly to train neural network on them. The offline stage is further sub divided in to two main modules.

#### **a. Preprocessing**

In the preprocessing the two classes both “non-faces” and “faces” were first subjected to the process of resizing by 20x20. So each image in the “non-face” and “face” class was first resize to 20x20 sizes. Then with the use of standard histogram equalization algorithm each image of both “non-face” and “face” classes were histogram equalized in order to correct brightness, contrast and equalize the different intensities level of the image.

After the histogram equalization, the technique of grayscale is applied on the images of both the classes in order to convert their color levels (i.e. RGB) to gray level.

The preprocessing as discussed above is applied to both classes before feeding them into the system because without preprocessing the network can not be efficiently trained. If the resizing is not done on the images then you have to create extra nodes in the neural network which will greatly reduce the efficiency of the neural network during training. Also if the images are not histogram equalized and not converted to grayscale then it will be difficult for the network to train on color images and also the different intensities of the images will train the network in improper way which will lead to incorrect results.

#### **b. Training the Neural Network**

Once the two classes both “non-faces” and “faces” are preprocessed then the training set is ready to train the neural network on them. But random images should be taken from training set and fed into the system. The process usually involves modifying the weights. Moreover, since the accumulated knowledge is distributed over all of the weights, the weights must be modified very gently so as not to destroy all the previous learning. A small constant called the *learning rate* ( $\xi$ ) is thus used to control the magnitude of weight modifications. Finding a good value for the learning rate is very important because if the value is too small, learning takes forever; but if the value is too large, learning disrupts all the previous knowledge. Unfortunately, there is no analytical method for finding optimal learning rate; it is usually optimized empirically by just trying different values.

During training you can easily observe training progress while checking the faces examples for positive response and the non-faces examples for negative response by feeding them to the neural network in the online stage. If the network in the online stage fails to recognize positive and negative responses then it means the network needs more training. Another processing step is needed where the network can differentiate between faces and non-faces or positive and negative responses, it is where you can save your network.

## 2.2 Online Stage

In the online stage, we use the trained network that has been trained in the offline stage to test a particular image to find out whether it contains a face or not. The stage has been sub divided in to a number of sub-modules which are given below.

- Sub-sampling and Localization.
- Preprocessing.
- Neural network.

### a. Sub-sampling and Localization:

In sub-sampling and localization, 20x20 sized small images are extracted from the reduced sized image and fed to the neural network. If an 80x80 window is run through the original image to locate face through localization then the original image is first reduced through a specific ratio where 20x20 images will be extracted from the reduced sized image. The size reduction of the original image is done through sub-sampling. The ratio is calculated as if 80x80 sized is window is applied to the original image then what will be the width and height of reduced size image when 20x20 sized window is applied to the reduced sized image. So it is achieved by multiplying the original image width and height by 20 each, and then dividing each one by 80, which will give the new width and height of the reduced sized image. Once the image size has been reduced through subsampling then a window of 20x20 will be run through the reduced sized image (i.e. 20x20 sized small images will be extracted from the reduced sized image) and then these 20x20 sized small images will be given to the neural network one by one which will give result 1 if the 20x20 sized image contain face otherwise 0 result will be returned and this process is done through localization.

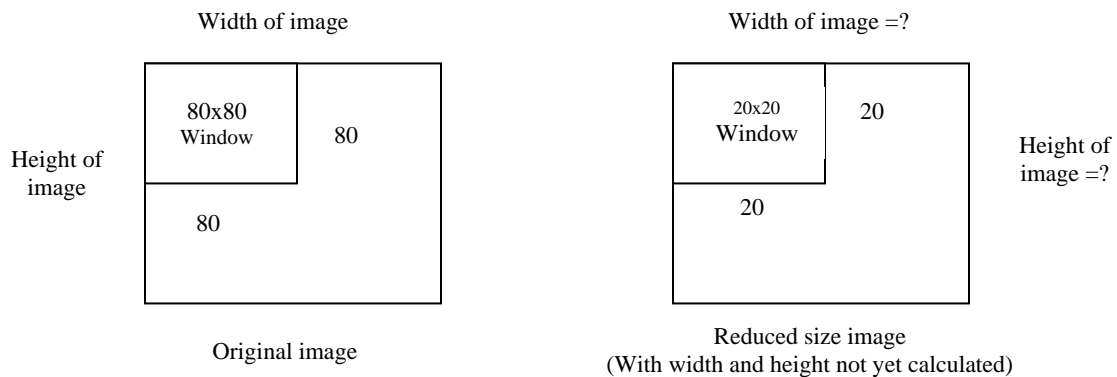


Fig 1: Images before Sub-Sampling

Sub-sampling is calculated by the formula as given below:

$$\text{New width of image} = \text{width of image (original image)} * 20/80$$

$$\text{New height of image} = \text{height of image (original image)} * 20/80$$

After Sub-sampling:

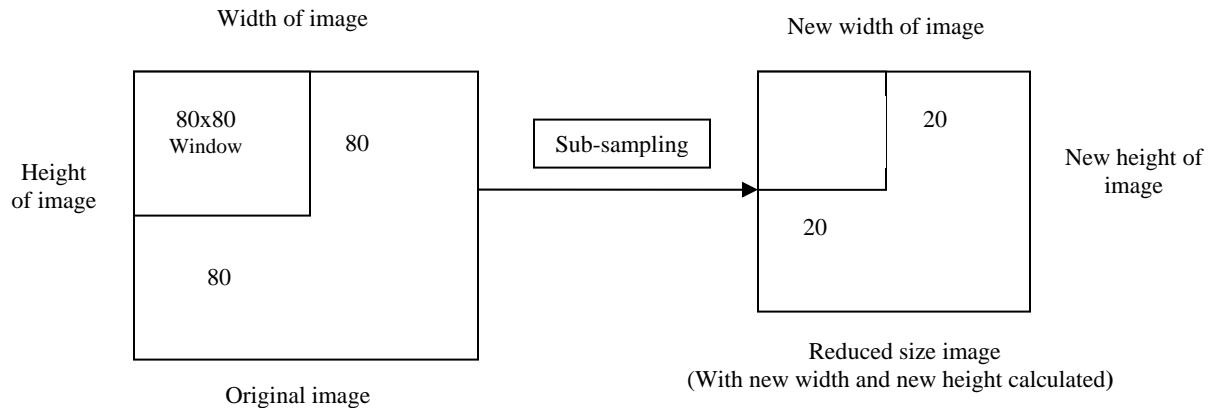


Fig 2: Images after Sub-Sampling

Now if the subsampling and localization fails at the 80x80 window sized then the window sized is incremented by 10 in an iterative loop and the whole process of sub-sampling and localization is repeated until the face is located in image and result of 1 is returned by the neural network. This incriminations of window size maximally goes up to 150x150 window sized.

Algorithm for sub-sampling and localization:

- 1) Start window size at 80x80 in original image and iterate successively through 150x150.
- 2) Sub-sample the image according to the window size set in step 1.
- 3) Apply localization on the sub-sampled image obtained in step 2 (i.e. extract 20x20 portion from the resized image obtained in step 2).
- 4) Preprocess the 20x20 image obtained in step 3.
- 5) Repeat step 3 and step 4 until all 20x20 images are extracted from sub-sampled image obtained in step 2.
- 6) Repeat step 1 to step 5 if at 80x80 window size the algorithm fails to find the face and if face found then indicate the face by rectangular box.

#### **b. Preprocessing:**

To reduce the variation caused by lighting or camera differences, the images are preprocessed with standard algorithms such as histogram equalization to improve the overall brightness and contrast in the images. The grayscale process is applied to the histogram equalized images so that the neural network can process them efficiently as without these corrections the processing time of neural network will increase resulting in lower efficiency. When these images are histogram equalized then grayscale were applied on these images. After applying the grayscale, the image is then intersected with an “oval mask” for ignoring background pixels.

Oval mask for ignoring background pixels



Original window



Histogram equalized window



Apply Grayscale



Intersection with oval mask



Fig 3: Under Process Images shown at different phases

Algorithm for preprocessing:

- 1) Apply histogram equalization on images received from sub-sample and localization phase.
- 2) Apply Grayscale on image of step 1.
- 3) Intersect the image of step 2 with oval mask.
- 4) Convert the image of step 3 to an array.
- 5) Fed the array of the step 4 to the neural network.
- 6) If face contains in the image neural network returns 1 otherwise 0.

**c. Neural Network**

After applying the preprocessing steps these images are then fed in to the neural network and on the basis of trained network in Offline stage the neural network decides whether the window contains face or not.

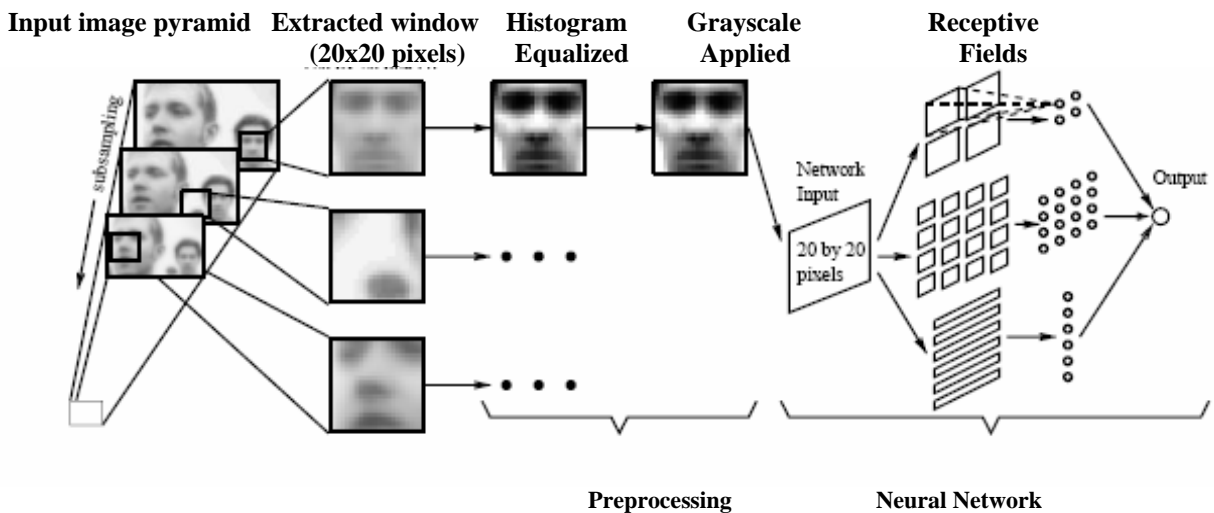


Fig 3: Preprocessing and Neural Network Application Phase

### **3. Conclusions and Future Work**

Our algorithm can detect between 80% and 90% of the faces, with an acceptable number of false detections. The main limitation of the current system is that it only detects upright frontal faces. There are a number of directions for future work. One of the assumptions of our system was that the face in an input image should not be tilted/rotated, this constraint which can be overcome if we include second neural network for rotations.

As our system scans every area of an input image which is very time consuming in order to shorten the search area if we can somehow roughly predict the area where face might be present this will save the time consumption.

System performance can be further increased if a system detect an area as a face and actually it is not face then adds that area to nonfaces examples of training set and train the system again on that training set.

### **References**

- [1] Maya Choueiri, Nassib El-Sayegh, and Wassim Said “Real-Time Face Detection and Recognition”, Electrical and Computer Science Department, American University of Beirut, Beirut-Lebanon.
- [2] Rowley, Baluja, and Kanade “Neural Network-Based Face Detection”, (PAMI, January 1998).
- [3] Jingtao Yao, Nicholas Teng, Hean-Lee Poh and Chew Lim Tan “Forecasting and Analysis of Marketing Data using Neural Networks”, journal of Information science and Engineering 14,843-862 (1998).
- [4] “Introduction to Artificial Neural Systems” a book written by Jacek and M.Zurada.