CS 283 Paul Debevec
Prof. Richard Fateman
Fall 1992 UC Berkeley

CS283 Course Project

A Neural Network for Facial Feature Location

Introduction

In order for computers to continue their march into the mainstream of human activity, they must continue to improve their ability to interact with humans. This means that they need to discern human writing, understand human speech, and recognize human faces. My project deals with the recognition of human faces. One stage of recognizing a face is to figure out how the face is oriented, and to do this one needs to know where the facial features are in the image. For my project, I implemented a neural network to locate the eyes, nose, and mouth of facial images (“mug shots”) taken from a particular high school yearbook.

Goal

The basic goal of this project was to train a neural network to be capable of locating the eyes, nose, and mouth in a person’s picture. The network was trained by repeatedly showing it positive and negative examples of eyes, noses, and mouths from the training set. The output of running the network on an image is more images, one for each feature. Each feature’s output image shows to what degree the neural network believes each point in the input image is part of such a feature. For example, if the output image corresponding to the nose exhibits values close to 1 in a region near the center of the image and values close to 0 everywhere else, then the network is indicating that the nose is definitely located in that region near the center. Such an output image could be converted into a coordinate pair of the location of the feature by finding the center of the largest connected region exhibiting a response close to 1.

Method

Acquisition of images

The images I used for the network were the underclass portraits of the 1987 University High School Yearbook. These were chosen to provide a reasonably homogeneous set of images that had similar contrast, lighting, and subject orientation. They were scanned in on a Macintosh using a Microtek flatbed scanner at 96 horizontal by 128 vertical pixels of resolution, in grayscale with 8 bits/pixel. This resolution was the minimum necessary to clearly see details of all the facial features. I used 101 of these images for the network, with 97 in the training set and four in the testing set.

Specification of feature coordinates

For the 97 images in the training set, I needed to record the locations of the eyes, nose, and mouth. I wrote an interactive program for the Silicon Graphics machine to display the image in a window and let the user click on each of the features, with the ability to fix mistakes. To get more accurate data, the image is displayed at five times normal size and the coordinates are recorded to within 0.2 pixels. Using the program, it only took twenty minutes to locate all of the features for all of the images.

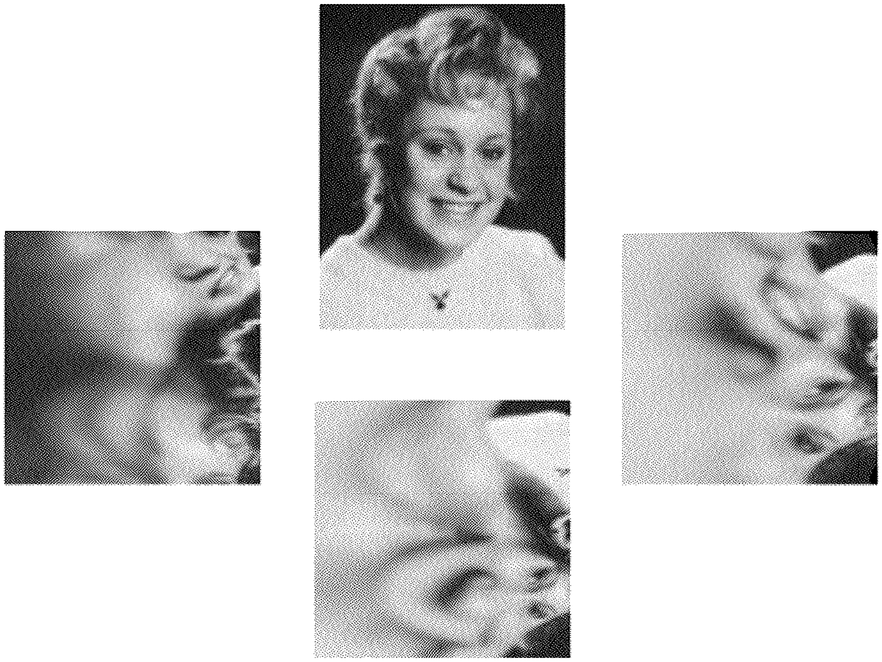
Generation of the training set

The amount of time and storage required to train and operate a neural network is at best linear in the number of input units. To keep the network training time reasonable, I decided to limit the number of input neurons to 64. The challenge of generating the training set was to find a way to amply embody the essence of a facial feature in an image with as few numbers as possible.

The first idea I had was to use an 8 by 8 pixel neighborhood of each feature to describe it to the neural network. Although this representation requires only 64 numbers, it gives the neural network a very myopic view of the feature in question—none of the context of the rest of the face is available to the network. This is undesirable because the context around a facial feature can serve as useful means of recognizing that feature. An example is that the nose is always located below two dark regions to either side of it—the eyes. It would be nice for the neural network to have access to the general image context around a feature while still having a good sense of the feature itself.

The solution I used was the same that the human visual cortex uses. Instead of sampling the neighborhood of a feature with a tiny rectilinear grid of points, I sampled the neighborhood of a feature with concentric circles, centered about the coordinates of the feature and radiating outward with exponentially increasing radii. This is generally referred to as a log-polar mapping. I did this with 32 circles, the smallest being one pixel in radius and the largest being fifty pixels, enough to span the image. Around each circle I sampled the image at 32 equally-spaced points, creating a 32 by 32 pixel representation of the feature. To reduce this to 64 numbers, I averaged 4 by 4 pixel blocks to get a 8 by 8 pixel representation.

This procedure is best described with some pictures. Below is a picture of Steffie Kovaks, class of 1989. To the left of Steffie’s picture are log-polar maps centered about her left eye, her nose, and her mouth. Points from the same circles appear on the same vertical lines in the maps; points from the same angles on their circles appear on the same horizontal lines. Thus the left half of each mapping consists of points sampled from near the feature, and the right half consists of points sampled from all across the rest of the face. The input to the neural network is formed by scaling the polar-log mapping down to 8 by 8 pixels.



A demonstration of the log-polar mapping function. Top-original image, Left—log-polar map of the left eye, Bottom—log-polar map of the mouth, Right—log-polar map of the nose.

Along with each list of 64 inputs, the training set had to specify the desired output of the neural network when given those inputs. I decided to have there be four output units, one for each feature. The desired output for a left eye was always set to (1 0 0 0), for a right eye (0 1 0 0), for a nose (0 0 1 0), and for a mouth (0 0 0 1).

While writing the training set generation code, I realized that network would only be getting information about eyes, noses, and mouths. This limited viewpoint would almost certainly give the neural network the impression that the sole qualification for being a nose were not being an eye or a mouth. But I wanted the network to be able to correctly classify other points of the picture that weren't in the list of facial features. To this end, I changed the training set generator to flavor its output with a certain proportion of maps of randomly chosen points in the images that weren't close to any of the feature points. These random negative examples would hopefully teach the neural network that not only is a person's left eye not a nose, but that neither is the person's forehead, chin, or shirt. The desired output for each negative example was set to (0 0 0 0).

Training the network

The neural network I used has 64 inputs and four outputs, one for each of the facial features (the left eye, the right eye, the nose, and the mouth). I chose to have one hidden layer of ten neurons between the input and output layers. I made this decision since I doubted that a network without hidden units would perform well, and I thought two or more hidden layers would introduce needless complexity. These intuitions came from reading descriptions of other successful neural networks.

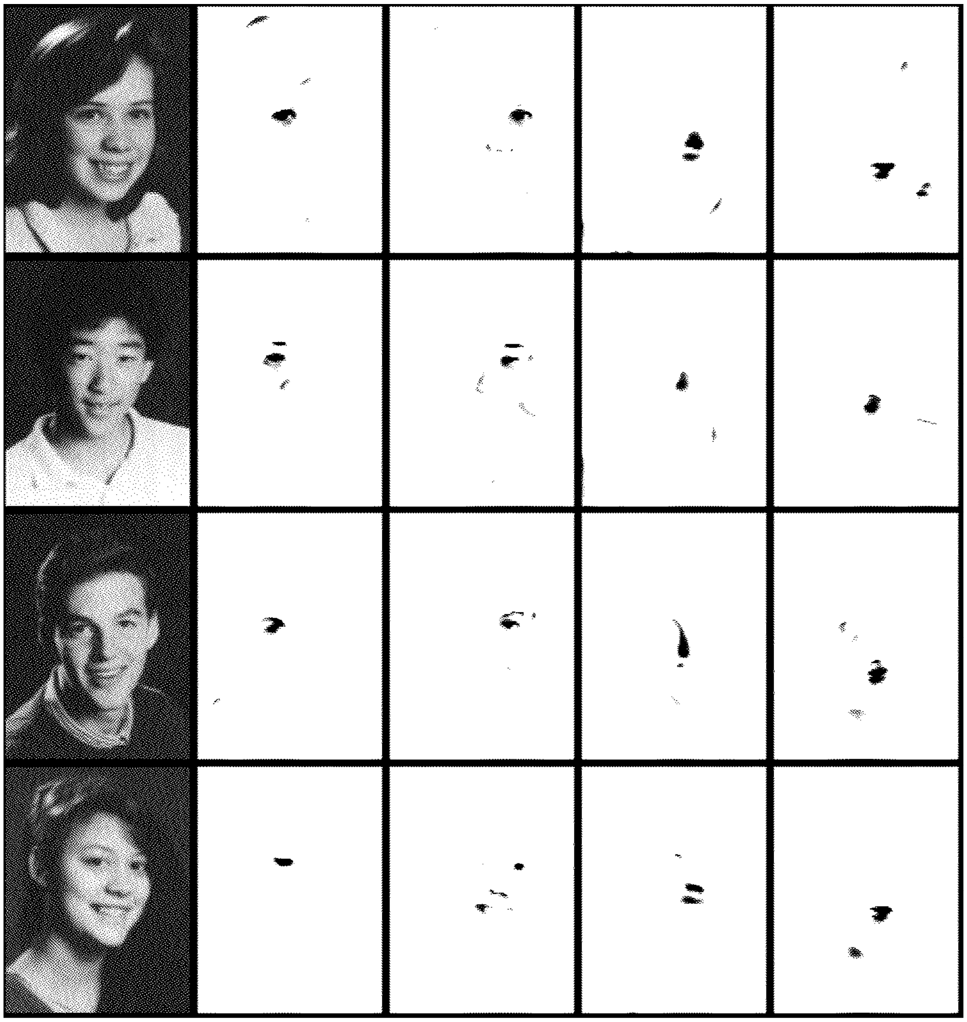
To train the network, I used the standard back-propagation algorithm as described in Rich and Knight. All of the connection weights were initialized to small random values. Then each sample in the training set was successively shown to the network and the weights were adjusted slightly to bring the actual output closer to the desired output for that sample. After all the samples in the training set were shown to the network, they were shown again. This continued until the weights in the neural network stabilized, which was typically after approximately 500 times through the training set. Training the neural network typically took on the order of an hour running on one of the HP snake machines.

Generating the feature maps

To see how well the neural network performed, I wrote another Silicon Graphics program to display the neural network's opinion of the locations of a person's facial features in an image. This was done by running the log-polar mapping on every single point in the input image, feeding each map into the neural network, and recording the four neural network outputs into four new images. The image corresponding to a particular feature is black wherever the neural network output was close to 1, white wherever it was close to 0, and shades of gray for intermediate values. Thus a successful neural network would generate four completely white images, except for a black spot at the coordinates of the corresponding facial feature.

Results

The results improved greatly when there was a large proportion of random negative samples in the training set. The best results I got were with a network that was trained with three times as many random negative samples as feature samples. Here are the four images in the testing set, along with their feature maps as computed by this neural network. Keep in mind that the network did not see these images when it was in the training stage.

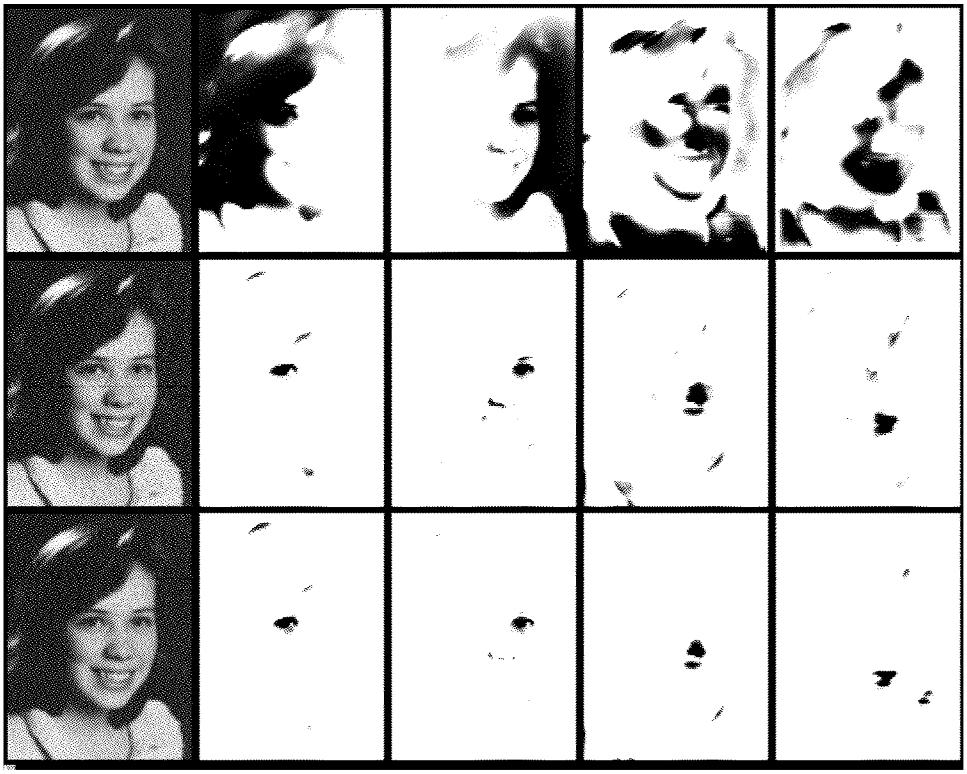


The four images in the testing set and the neural nework's corresponding feature maps, shown to the right of each image. Their order, from left to right, is left eye, right eye, nose, and mouth. "Left eye" actually refers to the person's right eye, which appears closer to the left of the image.

The neural network did a good job of finding the features. Each feature map gives a positive response in a small region of its feature, and zero response to the other features. Additionally, the feature maps are zero at almost all of the other points in the image. In every feature map, the largest connected black blob is centered over the location of the desired feature, which is the criterion for having successfully located a feature.

The fact that the neural network did an excellent job of distinguishing the right eye from the left eye in each image suggests that it made use of both global and local information, since locally the eyes appear similar. In the cases where there were other significant black blobs that did not correspond to the correct feature point, the other black blobs were generally located close to the feature point and had the same local intensities. This suggests that local intensity and relative position to the face were both weighted highly by the neural network.

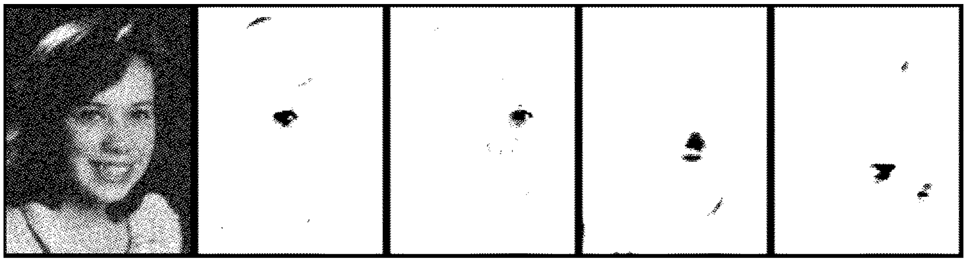
The images above were all done with twelve random samples for each training image. To determine the effect of the random negative samples, I trained three other neural networks with fewer such samples. The images below were generated with zero, four, and eight random samples per image.



The effect of adding different proportions of randomly chosen negative samples to the training set: Top—no samples, Middle—four samples per training image, Bottom—eight samples per training image.

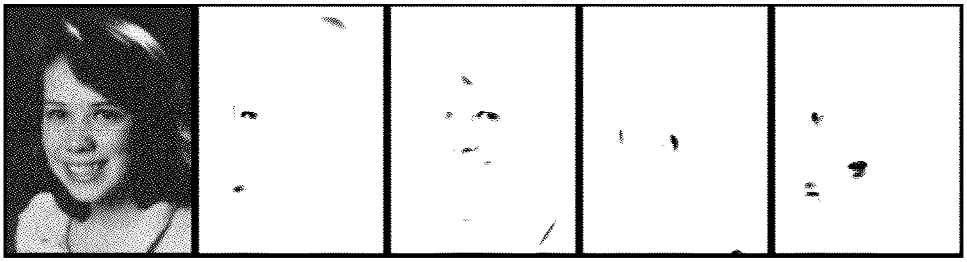
Obviously, the random samples were crucial in getting the neural network to respond correctly to the non-feature areas of the image. With no random samples, the neural network does a fine job of classifying the features with respect to each other, but as feared it happily identifies vast regions of the image to each particular feature. With four random samples per image, the problem is reduced to just a few errant black patches. With eight and twelve random samples, the black blobs are significantly reduced and the individual features are clearly dominant.

To better understand exactly what the network is and is not capable of, I modified one of the testing images in various ways and ran the network on it. I first tested the network's ability to deal with random noise in the testing image. I added 30% random noise to each pixel in the image (considerably worse than what is present in a video signal) and got the results shown below. The network produced nearly indistinguishable results from those of the original test image, demonstrating a good ability to deal with noise.



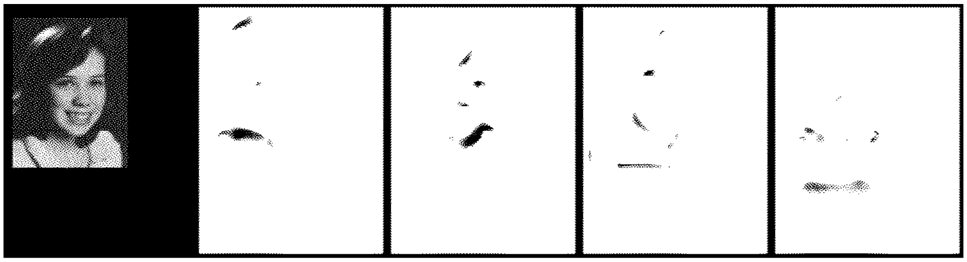
The effect of adding random noise to a test image.

I suspected that the neural network took advantage of the consistent lighting in the set of training images. To find out to what degree my suspicion was correct, I mirrored an image in the testing set and ran it through the neural network. The result is shown below. The network was still able to locate the eyes, but its response to them is considerably weaker. The same is true of the mouth. The nose is not very well identified. Thus the lighting angle and the resultant pattern of shadows is very important.



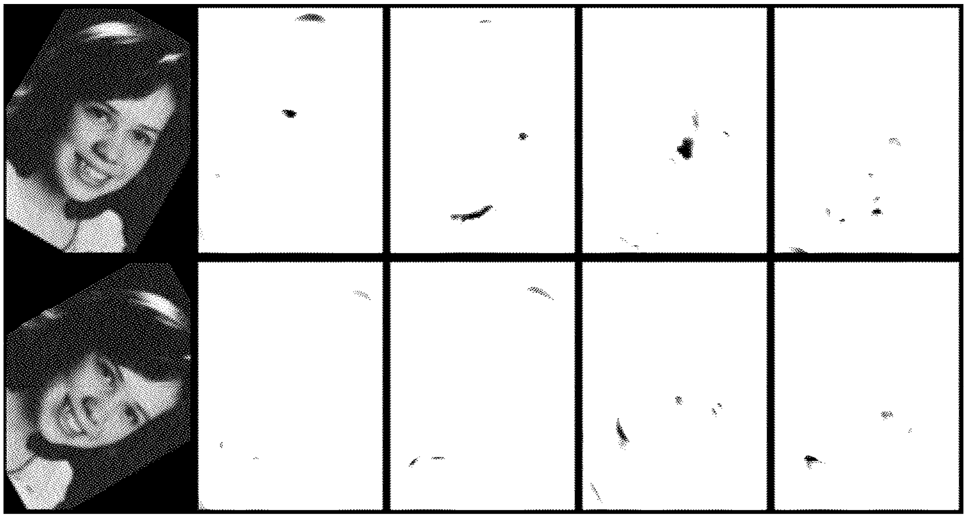
The effect of mirroring a test image.

I tested the neural network's scale invariance by shrinking a testing image to half size and running it through the network. The result is shown below. The network did a poor job of recognizing the facial features in the scaled image. The only feature that was identified was the right eye (relative to the viewer), but it was not the dominant positive response in the image. Thus the neural network is not scale invariant.



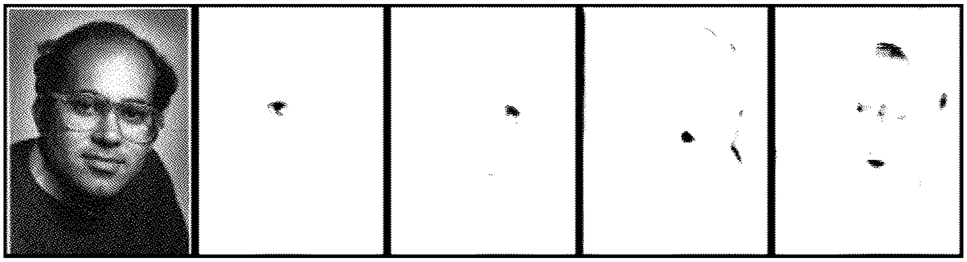
The effect of scaling a test image.

I tested the neural network's rotational invariance by rotating a testing image by several angles and running these images through the network. The results are shown below for rotation angles of 30° and 60°. The neural network performed badly in both cases. At 30° the network was only capable of recognizing the nose, and at 60° no features were correctly identified. Thus the network is not rotation invariant.



The effect of rotating a test image. Top—30°, Bottom—60°.

As a final *coup de grace*, I ran the neural network on an image that was not only not in the training set, but that wasn't even in the Uni High Yearbook. The result is shown below. Surprisingly, the neural network did an excellent job of recognizing all of the features. Both eyes are perfectly located and the nose shows up quite clearly. While the mouth is correctly identified, there are other regions of positive response that dominate, most notably near the top of the head. I believe that this is because the neural network had only been exposed to high school students, none of whom were bald.



The results of running the neural network on a test image from a different set of images than the training set.

Conclusion

The neural network I trained did an excellent job of locating the eyes, nose, and mouth in typical mug shots. It proved to be resistant to a reasonable amount of noise in the test images. It was not good at recognizing images that were significantly rotated, scaled, or differently lit from the images in the training set. Humans are able to recognize faces in the presence of noise, scaling, and in a variety of lighting conditions. However, a human's ability to recognize faces is severed impaired by rotations of over 45°. This is because humans encounter scaled and differently lit faces throughout their daily lives, whereas rotated faces are far rarer than correctly oriented ones. The question remains whether a larger neural network, if shown many faces with varying lighting and at varying scales, might be able to correctly identify features in arbitrary images.

Acknowledgments

I would like to thank Randy Katz for the use of his scanner, Jitendra Malik for the use of the robotics lab equipment, Richard Fateman for sanctioning this non-LISP project, and the students of University High School for their unwitting participation. I would also like to thank Raph Levien for the use of his digital halftoning algorithm to prepare the images used in this report.

Bibliography

Cavanagh, Patrick. "Size and Position Invariance in the Visual System." *Perception*, Vol. 7, 1978, pp. 167-177.

Hertz, John, Anders Krogh, and Richard G. Palmer. Introduction to the Theory of Neural Computation. Addison-Wesley, 1991.

Reitboeck, H. J., and J. Altmann. "A Model for Size- and Rotation-Invariant Pattern Processing in the Visual System." *Biological Cybernetics*. Springer-Verlag, 1984.

Rich, Elaine, and Kevin Knight. *Artificial Intelligence*, 2ed. McGraw-Hill, 1991, pp. 492-509.

Watson, Mark. Common Lisp Modules: *Artificial Intelligence in the Era of Neural Networks and Chaos Theory*. Springer-Verlag, 1991.