A Modular Approach to Facial Expression Recognition

Michal Sindlar SINDLAR@PHIL.UU.NL
Cognitive Artificial Intelligence, Utrecht University, Heidelberglaan 6, 3584 CD, Utrecht

Marco Wiering MARCO@CS.UU.NL
Intelligent Systems Group, Utrecht University, Padualaan 14, 3508 TB, Utrecht

Abstract

We study the use of multi-layer perceptrons in applying artifical learning to the recognition of emotional expressions from frontal images of human faces. The perceptrons are trained using per-pixel luma data from the images' mouth and eye areas, and map the inputs to one of 6 emotions. We compare 3 different methods for processing input information: 1) one network module for all inputs; 2) one network module for both eyes, and one for the mouth; 3) one network module for the mouth, one for the left eye, and one for the right eye. Our results show that involving multiple modules leads to better results, resulting in an overall performance of 84% images classified correctly.

1. Introduction

Automated facial expression recognition from static images can be useful in a number of different applications, such as human-machine interaction, or detection of audience response. The goal of this research is to find out whether it is possible to successfully perform facial expression recognition using multi-layer perceptrons in a modular setup on practically unprocessed input data. For this specfic purpose a software tool named Narcissus[1] was developed.

The sections 2–7 deal with the following: Section 2 discusses the setup of the research and the image data used, section 3 is about the application used to process the data, section 4 briefly describes the training and testing procedure, in section 5 the tests and the results are discussed, and section 6 discusses our study in a context of related work. Section 7 concludes this paper.

[1] After the mythical figure who fell in love with himself after seeing the reflection of his own face in a pond.

2. Research setup

Since the main object is to construct an artifical classifier to use with static frontal images of human faces expressing emotions, a proper classifier as well as suitable images will be needed. This section discusses the selection of both.

2.1. Choice of classifier

We want a classifier that is robust, and relatively easy to implement, like a multi-layer perceptron (MLP) or radial basis function network (RBF). Since a study similar to this one obtained good results with MLPs, and results with RBFs were a bit worse (Gargesha & Kuchi, 2002), MLPs with a logistic sigmoid activation function are chosen as classifiers. Learning takes place through back-propagation. The classifier's outputs are Ekman's 6 basic emotions with clear facial signals: anger, disgust, fear, happiness, sadness, and surprise (Ekman, 1994). Other output categories are of course possible —such as the 2-dimensional model of emotion (Russell, 1980)— but Ekman's emotions are widely-used in research, and also the image data described in section 2.2 is based on them, so this model was adopted.

2.2. Image data

To investigate the expression of emotions in facial images, a sufficient amount of images expressing those emotions is of course required. Three sets were used, described in more detail in Sections 2.2.1, 2.2.2 and 2.2.3. Sample pictures are shown in Figure 1.

2.2.1. Cohn-Kanade image set

The Cohn-Kanade Facial Expression Database (Kanade et al., 2000) is a collection of approximately 2,000 grayscale image sequences from over 200 subjects. The images used were expertly analyzed with FACS for the occurrence of so-called action

Figure 1: *Sample images expressing surprise, one from the Cohn-Kanade database (left), one from the JAFFE database (middle), and one from the POFA set (right).*

units, as described in the Facial Action Coding System (Ekman & Friesen, 1978). The AU-codes were manually translated into Ekman's 6 basic emotions using the rules from the FACS Investigator's Guide (Ekman et al., 2002), and only the images that were expressing emotion according to this system were used in this research.

2.2.2. JAFFE image set

The Japanese Female Facial Expression (JAFFE) database (Lyons et al., 1998) consists of 213 grayscale images of Japanese women posing the 6 basic expressions used in this research, plus a neutral one. The images have been rated on a 5-point scale (from 1 to 5) for each of the 6 emotion categories by 60 female Japanese students. Each of the images was assigned to the category for which it achieved the highest overall rating.

2.2.3. POFA image set

The Pictures Of Facial Affect image set (Ekman & Friesen, 1976) contains grayscale photographs of 14 actors portraying expressions that are reliably classified by naive observers as the 6 basic expressions used in this study (the overall agreement is 91.6%).

3. Specifics of the application

The inner workings of the software tool Narcissus[2] are briefly described in this section. Section 3.1 describes the loading of images into the application for processing, in section 3.2 training one or more networks using data from the opened images is described, and section 3.3 describes testing one or more networks on the opened images.

3.1. Opening images

Images of the formats GIF, JPEG and PNG are supported. These images can be opened into the application, and can be used either for creating and training a network, or testing an existing network. When images are opened for the first time, the areas containing the left eye, the right eye, and the mouth have to be selected by hand. These areas are the facial features that are expected to contain most emotion-related information. The selection procedure is manual: the user draws a selection rectangle to select the region where the features are. The aspects of these selection rectangles are constant, with a *horizontal* : *vertical* of $1.3 : 1$ for the eyes, and $2.2 : 1$ for the mouth. The aspects are kept constant, so that no distortion occurs when the feature images are scaled. The numbers of 1.3 and 2.2 were chosen more or less arbitrarily; the criterion being that the whole feature region (e.g. the left eye) could be fitted, without incorporating a lot of the surrounding 'noise'.

3.2. Training one or more networks

In training mode the images that have been opened can be used to train one or more networks. Table 1 shows the possible situations.

# of networks	input
1	both eyes and the mouth
2	both eyes into one network, the mouth into the other
3	each eye goes into a separate network, as does the mouth

Table 1: *Different networks and their input.*

When one network is selected, this network processes all image features. When two networks are selected, one network will be set up to process the left as well as the right eye, the other will process the mouth. In the case of three networks, each feature is processed by a separate network. From now on, when talking about the networks used by Narcissus, these will be referred to as 'modules'. The mode in which 2 modules are selected for processing — wherein one processes both eyes and the other the mouth — will be referred to as 'a system of 2 modules' or a '2-module system', and likewise for 3 modules.

After selecting the number of modules, the desired feature dimensions have to be set for the eyes (same dimensions for both) and the mouth. All features are scaled to these dimensions using the standard Java AffineTransform.getScaleInstance(x,y) scaling operation, to ensure that every feature image yields the same amount of inputs. The default setting is a dimension of $20 * 15$ pixels for the eyes and $40 * 18$ pixels for the mouth.

[2] Available at `http://narcissus.no-ip.org/`

For a 1-module system, this yields (including the bias):

$$(20 * 15) * 2 + (40 * 18) + bias = 1,321 \text{ inputs}$$

The data that is the actual input for the networks, is the *luma* (Y') value of each pixel, which is calculated with the following standard formula for obtaining luma values from non-linear RGB.

$$Y' = 0.299 * R + 0.587 * G + 0.114 * B$$

This comes down to converting the image from RGB color to grayscale. Using the luma values reduces the dimensionality of the data: instead of using three values (R, G, and B) from each pixel as input, we now only have to use one (Y'). This is also the case for images that already are in grayscale, which in RGB color space are represented as (Y', Y', Y'). The resulting input value is $0.299 * Y' + 0.587 * Y' + 0.114 * Y' = Y'$.

3.3. Testing one or more networks

In testing mode, the networks that were created in training mode can be evaluated. The necessary parameters, such as the feature dimensions, are taken from the network settings. Images can be evaluated with respect to a 1-, 2-, or 3-module system one by one or all at once. This mode provides a lot of visual feedback. The outputs of all networks are recorded and displayed, as well whether or not an image was classified correctly. Also, the outputs can be visualized in a graph, for easy viewing.

The face in Figure 2 was analyzed manually as described in 2.2.1 and reported to be expressing sadness. The output of the 3-module system as visualized in Figure 3 tells that two modules got it right, and one did not. There are six groups of bars; one for each emotion category.

Figure 2: *Sample image from the Cohn-Kanade database, expressing sadness.*

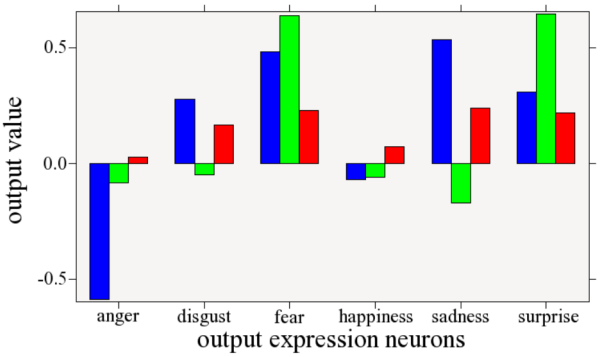

Figure 3: *Graph showing the output from three networks.*

From left to right: anger, disgust, fear, happiness, sadness, surprise. There are three bars: one for each module. The module analyzing the left eye is represented by the leftmost bar in each group (blue), the middle (green) bar stands for the right eye, and the module analyzing the mouth is the rightmost bar in each group (red). In this case, the left eye (left, blue) and mouth (right, red) correctly reported sadness (the 5^{th}group of bars), while the right eye module (middle, green) got it all wrong: it reported surprise, with fear coming in second by the smallest of margins. Sadness actually was the least likely option for this network.

4. Training and testing procedure

The image set available for training and testing consists of 458 images in total. The images are spread unevenly over the 6 emotion categories , as follows (with the Cohn-Kanade / JAFFE / POFA ratio shown in parentheses): 60 in *anger* (23/24/13), 82 in *disgust* (51/18/13), 30 in *fear* (10/7/13), 116 in *happiness* (73/30/13), 55 in *sadness* (22/22/11) and 115 in *surprise* (77/25/13).

Since the number of samples is quite low for some categories, cross-validation has been adopted as training procedure. Cross-validation allows us to obtain valid results using only a small number of samples. This is done by dividing the data into S segments, using data from $S - 1$ of these segments for training, and testing performance with the remaining segment. This process is repeated S times, and the results of S runs are then averaged to obtain the final result. In our case the data has been split up in 10 sets, each containing approximately 90% of the samples as training data and the remaining 10% or so for testing. A certain network's performance has been defined as the average of its results on the 10 subsets.

5. Tests & results

The tests described in this section were performed following the procedure explained in section 4.

5.1. Determining optimal parameters

The following sections are about determining a function that returns a reasonable number of hidden units based on the number of inputs (5.1.1), and finding a feature size that has enough detail, without being unnecessarily large and slow to process (5.1.2).

5.1.1. Hidden units

The focus here is on calculating the number of hidden units as a function of the number of inputs. This number should not be too small because this allows for fewer possible mappings, and thus less expressive power, but there should not be too many hidden units either, because this increases processing time and can lead to overfitting the data.

No optimal feature size has been determined yet, so three sets of feature sizes have been considered, listed below in order of increasing number of inputs.

Low-detail: The dimension of the eyes is $5 * 4 = 20$ pixels and the dimension of the mouth is $10 * 5 = 50$ pixels. Including the bias, this amounts to $(2 * 20) + 50 + 1 = 91$ inputs.

Medium-detail: Eyes are $10 * 8$ and mouth is $20 * 9$. Total of $160 + 180 + 1 = 341$ inputs.

High-detail: Eyes are $40 * 31$ and mouth is $80 * 36$. Total of $2,480 + 2,880 + 1 = 5,361$ inputs.

All features are processed by one single network. For this test there was no need to do otherwise, because the number of hidden units applies to a network in general, and does not depend on the particular features it is processing. The learning rate was fixed at 0.02 and all networks ran 500 passes. If a 100% score was achieved on the training set before the 500^{th}pass, back-propagation (and thus further learning) stopped.

First the number of hidden units n_{hidden} was calculated from the number of inputs n_{input} using the function:

$$n_{hidden} = \sqrt[x]{n_{input}}$$

Then, 4 tests were performed using the aforementioned formula and the values 2, 3, 4 and 5 for x. This test yielded the results shown in Table 2. This table shows the average score over 10 cross-validation runs, and in parentheses the *standard deviation (σ)* of the 10 scores that make up the final score.

x	low	medium	high
2	70% (5.6)	79% (4.3)	81% (3.6)
3	64% (3.9)	78% (6.5)	81% (4.2)
4	57% (7.9)	73% (5.8)	81% (4.3)
5	48% (8.9)	57% (7.0)	77% (5.0)

Table 2: *Scores for the $\sqrt[x]{n_{input}}$ function.*

The most desirable results were obtained with $x = 2$ for the low-detail system, $x = 2$ for medium-detail, and $x = 4$ for high-detail. A function that yields a suitable number of hidden units, in a range comparable to the one given by $n_{hidden} = \sqrt[x]{n_{input}}$ with the aforementioned values for x, is

$$n_{hidden} = 3 * \ln n_{input}$$

This function returns a relatively large amount of hidden units for small networks, and (compared to $\sqrt[2]{n_{input}}$, for instance) a small n_{hidden} for large networks. Considering the results as seen in Table 2, this is what we want.

5.1.2. Feature size

As Table 2 shows, more inputs and a lot of hidden neurons seem to give the best results. However, there is a downside to having a large network. Table 3 shows the approximated processing times[3] for one pass on 414 examples *without* back-propagation.

x	low	medium	high
2	0.05	0.24	12.70
3	0.04	0.10	3.20
4	0.03	0.06	1.50
5	0.02	0.05	1.10

Table 3: *Approximated processing times in seconds for each of the networks.*

This means the network processing the low-detail features (91 inputs in total) with $\sqrt[2]{n_{input}}$ function for the hidden neurons took 50 milliseconds (0.05 seconds) for classifying 414 images, while the network processing high-detail features with $\sqrt[2]{n_{input}}$ hidden neurons took 12.7 seconds. Training both these network for 500 passes would take well over (because of back-propagation, which was not considered in Table 3) 25 seconds for the low-detail network, compared to 1 hour and 45 minutes for the high-detail network. Quite the difference!

The medium-detail network did not have as good results as the high-detail one, but the latter took a lot longer to process, without spectacular improvements in performance. Therefore an intermediate sized network was trained. This network has a $20 * 15$ size

[3] On an Athlon XP 3200+ with 1 gigabyte of memory.

for the eyes, and a $40 * 18$ size for the mouth. Including the bias, this amounts to 1,321 inputs. The $n_{hidden} = 3 * \ln n_{input}$ function returns 22 hidden neurons. This network has quite a good processing time: about 0.8 seconds per pass. The averaged score of 10 cross-validation runs is 82% with a standard deviation of 4.7, which is the best so far. In all following experiments this network (summarized in Table 4) is used.

eyes	mouth	n_{hidden}
$20 * 15$	$40 * 18$	$3 * \ln 1,321 = 22$

Table 4: *Optimal settings for feature size and n_{hidden}.*

5.2. Performance of networks

Using the parameters found in section 5.1, several systems are tested using the cross-validation procedure described in section 4.

5.2.1. Cross-module comparison

Table 5 shows the performance of each network from a 1-, 2-, and 3-module system as the overall percentage of correctly classified images, with the standard deviation σ on 10 cross-validation runs in brackets.

system	module	score (σ)
1 module	eyes & mouth	82% (4.7)
2 modules	eyes	68% (5.3)
	mouth	70% (7.9)
3 modules	left eye	62% (7.8)
	right eye	60% (5.0)
	mouth	69% (6.0)

Table 5: *Comparison of 1-, 2-, and 3-module systems.*

Clearly, the single-module system does best. It has the highest overall score of 82% correctly classified images, and the lowest σ (4.7), which means that it had the least amount of variation on the 10 cross-validation runs, and therefore is the most consistent of the 3 systems. Table 6 shows the highest and lowest scores for each of the networks in Table 5.

network	module	highest	lowest
1 module	eyes & mouth	89%	74%
2 modules	eyes	74%	59%
	mouth	85%	60%
3 modules	left eye	76%	54%
	right eye	65%	51%
	mouth	77%	61%

Table 6: *Performance high/low for 1-, 2-, and 3-module systems.*

Also note how the networks processing a single eye do not perform much worse than the network processing both eyes. The highest score for the 3-module system working on the left eye (76%) was even higher than the one for the 2-module network working on both eyes (74%). This is interesting, because it shows that even from a partially occluded face (where perhaps only half of the face is visible) expression recognition is possible when using a modular approach.

5.2.2. Module addition

Section 5.2.1 showed how a 1-module system outperformed the individual networks from the 2- and 3-module systems, and how the networks from the 2-module system also outperformed those from the 3-module system. Now let's see what happens when the individual modules are combined, so that they all 'cast a vote' in a single system. An easy way to achieve this is by simply adding up the 6 outputs of each system and pretending the resulting values are the outputs of a single system. Table 7 shows the results of this procedure for the 2- and 3-module systems, along with the result of the 1-module system (where no addition is possible) for comparison.

system	module	score (σ)
1 module	eyes & mouth	82% (4.7)
2 modules	eyes + mouth	82% (3.5)
3 modules	left eye + right eye + mouth	84% (5.3)

Table 7: *Performance of the 2- and 3-module systems after addition of output activations.*

After addition of output activations, the 3-module system suddenly performs best! Same as with the discussion of the separate modules, let's have a look at the highest and lowest scores of each of the systems. The 1-module system's scores are the same, of course. The results are shown in Table 8.

system	module	highest	lowest
1 module	eyes & mouth	89%	74%
2 modules	eyes + mouth	87%	77%
3 modules	left eye + right eye + mouth	93%	79%

Table 8: *Performance high/low for 1-, 2-, and 3-module systems after addition.*

Adding these systems up again produces a combined super-system in which all three systems are casting their vote. This system performs as shown in Table 9.

system	score	σ	highest	lowest
combined	85%	5.5	93%	75%

Table 9: *Score, σ and performance high/low for 1-, 2-, 3-module systems added together.*

This is the best system so far, but not by a great margin and at a price. It takes approximately three times as much time to run compared to the other systems,

because it actually consists of those systems. Its processing time is the processing time of the 1-module system, added up to that of the 2-module system, and again to that of the 3-module system. Therefore it's quite inefficient without major improvement over the other systems.

5.3. In-depth analysis

In this section the four systems discussed in section 5.2.2, which are the ones that yielded the best results, will be analyzed in-depth. The analysis is presented in form of a *confusion matrix*, as defined in (Kohavi & Provost, 1998). This matrix visualizes the *combined results* of the 10 cross-validation test runs in a diagram, whose row as well as column headings show a category label. A stands for anger, D for disgust, F for fear, H for happiness, Sa for sadness and Su for surprise. The row headings stand for the desired (correct) classification, the column headings for the actual classification. The cell values show how often a certain error (confusion) occurred. The values in the Σ-column's and Σ-row's cells show the summation over the preceding cells in their respective row and column. For the Σ-column, this can be interpreted as the *bias* towards a category, and for the Σ-row it represents the total number of *misclassifications* for images from this category. On the diagonal the percentage of *correctly* classified samples of a certain category is shown, and the final row (**T**) shows the total number of samples in each category, together with the total number of pictures. (Σ, Σ) shows the performance of this system as a percentage of correctly classified images, with the total number of *misclassified* images in brackets.

So, for example, by going to (Sa, A) in Table 10, we find that after 10 cross-validation tests for the 1-module system, 3 images in total that should have been classified as *Sadness* were confused for the category *Anger*.

(A, A) tells us that 63% of the test samples from the *Anger* category were correctly classified as such. This can be verified by checking the total number of misclassified *Anger* images in (A, Σ), which is 22, and indeed $\frac{60-22}{60} * 100\% = 63\%$.

The following sections 5.3.1 to 5.3.4 discuss the four addition systems from section 5.2.2.

5.3.1. 1-module system

As Table 10 shows us, the best-recognized category (happiness) was recognized much better than the worst-recognized (fear): 94% compared to 47%. Anger and disgust were often confused for each other: 15 times in total (add (A, D) and (D, A) together). Surprise is mistaken for disgust only once, and disgust never for surprise. In fact, only fear is often mistaken for surprise, and this quite often too (7 times). Another thing to note is that sadness is mistaken for happiness — which could be considered the opposite emotion — 5 times in total, which is a lot in this context.

	A	*D*	*F*	*H*	*Sa*	*Su*	Σ
A	*63%*	6	1	2	3	3	**15**
D	9	*85%*	2	0	3	1	**15**
F	2	0	*47%*	2	2	3	**9**
H	3	3	2	*94%*	5	2	**15**
Sa	6	3	4	3	*73%*	0	**16**
Su	2	0	7	0	2	*92%*	**11**
Σ	**22**	**12**	**16**	**7**	**15**	**9**	***82%*** (81)
T	*60*	*82*	*30*	*116*	*55*	*115*	***458***

Table 10: *Confusion matrix for the 1-module system.*

5.3.2. 2-module system[4]

	A	*D*	*F*	*H*	*Sa*	*Su*	Σ
A	*73%*	7	2	4	5	1	**19**
D	5	*84%*	1	2	1	0	**9**
F	3	0	*47%*	1	3	5	**12**
H	3	4	3	*93%*	6	1	**17**
Sa	5	2	4	1	*65%*	1	**13**
Su	0	0	6	0	4	*93%*	**10**
Σ	**16**	**13**	**16**	**8**	**19**	**8**	***82%*** (80)
T	*60*	*82*	*30*	*116*	*55*	*115*	***458***

Table 11: *Confusion matrix for the 2-module system, after addition of individual modules.*

Anger and disgust are mistaken for each other 12 times in total, again the highest score, while disgust and surprise are never mistaken for each other. Again, fear is often mistaken for surprise: 6 times. This is a lot, especially considering the fact that fear only has 30 samples. Sadness is mistaken for happiness 6 times.

5.3.3. 3-module system

The anger-disgust confusion is lower for the 3-module system, only 8 mistakes. Sadness is mistaken for happiness 7 times. Disgust and surprise are confused only once. Fear is mistaken for surprise 5 times.

5.3.4. Combined super-system

Since this system reflects the previously discussed three systems, there are no real surprises.

The overall pattern for the four systems considered is

[4] For (Σ, Σ) in Table 11, $(378/458) * 100\% = 83\%$ and not 82%. This is not an error, but a reflection of the fact that the performance has been calculated as the average of the performance of individual cross-validation tests, which in this case leads to a discrepancy of 1%.

	A	*D*	*F*	*H*	*Sa*	*Su*	Σ
A	*70%*	3	1	2	5	0	**11**
D	5	*85%*	3	3	1	1	**13**
F	2	2	*53%*	1	3	1	**9**
H	5	6	3	*93%*	7	1	**22**
Sa	5	1	2	1	*69%*	1	**10**
Su	1	0	5	1	1	*97%*	**8**
Σ	**18**	**12**	**14**	**8**	**17**	**4**	***84%*** (73)
T	*60*	*82*	*30*	*116*	*55*	*115*	***458***

Table 12: *Confusion matrix for the 3-module system, after addition of individual modules.*

	A	*D*	*F*	*H*	*Sa*	*Su*	Σ
A	*72%*	5	1	1	3	0	**10**
D	6	*87%*	3	1	2	1	**13**
F	1	0	*50%*	1	3	3	**8**
H	3	4	3	*96%*	6	1	**17**
Sa	5	2	4	2	*73%*	0	**13**
Su	2	0	4	0	1	*96%*	**7**
Σ	**17**	**11**	**15**	**5**	**15**	**5**	***85%*** (68)
T	*60*	*82*	*30*	*116*	*55*	*115*	***458***

Table 13: *Confusion matrix for the combined super-system.*

that fear is recognized worst by far, and that performance on happiness and surprise is best. There is a clear correlation between performance and the total number of samples for a certain category: the more samples, the better the performance. For all cases considered, bias towards happiness was significantly higher than bias towards surprise. Perhaps this is because the expression of surprise has a more unique signal (wide-open mouth), only to be confused with fear, while the expression of happiness comes in many different forms.

6. Related work

In the first part of this section, related studies are briefly reviewed. In the second part, this study is compared to similar ones, and a comparison is drawn. Finally, suggestions for further research are given, as well as possible improvements for the current approach.

6.1. Review of related work

The past decade has seen a lot of activity in the field of (semi-) automatic facial expression recognition, using widely differing approaches. This brief review will focus on a neural network-based method operating on input data gathered from static facial images. Other possibilities are analysis of image sequences (with output to FACS AU's, for instance), and analysis of static images using template- or rule-based methods. For a concise overview of the myriad of possible approaches, see (Pantic & Rothkrantz, 2000).

Zhang et al. (Zhang et al., 1998) compare geometry-based and Gabor wavelets-based approaches to facial expression recognition using multi-layer perceptrons. Their findings are that Gabor wavelets are much more powerful than geometric position. They achieve an overall score of 90.1% using combined information from Gabor wavelets and geometric position, with 7 hidden units and output to 7 categories (anger, disgust, fear, happiness, sadness, surprise, and neutral), using 213 images from the JAFFE database. Fear is problematic: when excluding it, the results are 92.3% correctly classified images, and human agreement with the expressors' intention rises with 6% to 85.6%.

A very interesting approach is taken by Dailey et al. (Dailey et al., 2002) in a system called EMPATH. They present an artificial approach to facial expression recognition, modelled on the human perceptual system. Their system has three major layers. The *perceptual layer* uses Gabor filters and represents the human complex cells in the visual cortex. The *Gestalt layer* performs principal component analysis using linear hidden units, and is comparable to the 'face cells' in the human inferior temporal cortex. The *category layer* is the output layer, and has the 6 categories of anger, disgust, fear, happiness, sadness and surprise. This system's performance on the POFA image database as input was as depicted in Table 14. Again, performance on fear is particularly bad.

Expression	*System performance*	*Human agreement*
Happiness	100.0%	98.7%
Surprise	100.0%	92.4%
Disgust	100.0%	92.3%
Anger	89.1%	88.9%
Sadness	83.3%	89.2%
Fear	67.2%	87.7%
Average	**90.0%**	**91.6%**

Table 14: *Performance of EMPATH and human agreement on image data from the POFA database.*

6.2. Comparison to related work

Compared to other approaches, ours is simple and straightforward because it uses hardly any preprocessing (apart from converting all data to grayscale, which doesn't affect much since all image sets used have grayscale images). Still, it achieves pretty good results, with a high around 85%. However, feature selection is manual, and one could wonder how much bias is introduced by hand-selecting the features.

An interesting phenomenon is the (relatively) poor detection of fear by artificial neural network-based systems, as well as human observers (as illustrated by Table 14 with the EMPATH results). In our system, detection of fear was quite terrible too, which we first

attributed to the low number of training samples. Now it seems that a higher amount of fear-examples maybe could improve results, but that fear also has inherent characteristics which make it harder to recognize.

It should be noted that we used three different image collections, and that most other studies use only one. The sample images from a single set for a single category can be quite low; e.g. 7 samples from the JAFFE database for fear. Most likely our results could improve by using more samples per image set, since it is much harder for artificial learning methods to generalize when learning from a low number of samples.

7. Discussion

Our goal of creating a working system to classify emotions from static frontal images, using hardly preprocessed images and multi-layer perceptrons, has been met quite well. Most notable of the method used in this study is perhaps the use of a separate network module per facial feature (left eye, right eye, and mouth). Separate feature modules yield reasonable results (between 60% and 70%), and simple addition of output values improves results to well over 80%). This could be useful in processing facial images for recognition of emotional expression, where features have been (partially) obscured. It is very probable that our approach could be successful in *identity recognition* from (partially obscured) facial images as well.

Using preprocessing, such as application of Gabor filters, seems to improve results. This study could be extended by applying a Gabor wavelet-based approach and a modular one to build a more robust Gabor-based system. It would also be interesting to see how a modular approach would perform when automatic face and/or feature detection is applied. Moreover, to obtain a generally applicable emotion-recognition tool, training should be performed using many samples showing expressions in many different circumstances. This means using different image collections that involve multi-cultural subjects and different lighting conditions, and perhaps also lateral facial images, subjects with partially obscured faces, or subjects wearing glasses.

References

Dailey, M., Cottrell, G., Padgett, C., & Adolphs, R. (2002). EMPATH: A neural network that categorizes facial expressions. *Journal of Cognitive Neuroscience*, *14*, 1158–1173.

Ekman, P. (1994). All emotions are basic. *The Nature of Emotions; Ekman, P. and Davidson, R.J., eds.*, pages 15–19.

Ekman, P., & Friesen, W. (1976). *Pictures of facial affect*. Consulting Psychologists Press, Palo Alto, CA.

Ekman, P., Friesen, W., & Hager, J. (2002). *Facial Action Coding System Investigator's Guide*. A Human Face, Salt Lake City, UT.

Ekman, P., & Friesen, W. V. (1978). *Facial Action Coding System*. Consulting Psychologist Press, Palo Alto, CA.

Gargesha, M., & Kuchi, P. (2002). Facial expression recognition using artificial neural networks. `http://www.public.asu.edu/~pkuchi/ExpressionRecognition.pdf`.

Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), March 2000, Grenoble, France*.

Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, *30*, 271–274.

Lyons, M. J., Akamatsu, S., Kamachi, M., & Gyoba, J. (1998). Coding facial expressions with Gabor wavelets. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, April 14-16 1998, Nara Japan*, pages 200–205.

Pantic, M., & Rothkrantz, L. (2000). Automatic analysis of facial expressions: the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 1424–1445.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*, 1161–1178.

Zhang, Z., Lyons, M., Schuster, M., & Akamatsu, S. (1998). Comparison between geometry-based and Gabor wavelets-based facial expression recognition using multi-layer perceptrons. *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, 454–459.