

Artificial Neural Networks vs. Support Vector Machines for Skin Diseases Recognition

Michał Antkowiak

May 3, 2006

Master's Thesis in Computing Science, 20 credits
Supervisor at CS-UmU: Michael Minock
Examiner: Per Lindström

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN

Abstract

Artificial Neural Networks (ANNs) as well as Support Vector Machines (SVM) are very powerful tools which can be utilized for pattern recognition. They are being used in a large array of different areas including medicine. This thesis exemplifies the applicability of computer science in medicine, particularly dermatological tools and software which uses ANN or SVM. Basic information about ANN, including description of Back-Propagation Learning Algorithm, has also been introduced as well as information about SVM.

Finally the system for recognition of skin diseases, called *Skinchecker*, is described. *Skinchecker* has been developed in two versions: using ANN and SVM. Both versions have been tested and their results of recognition of skin diseases from pictures of the skin fragments are compared and presented here.

Contents

1	Introduction	1
2	Background	3
2.1	Computer science in medicine	3
2.1.1	Dermatological tools	3
2.1.2	ANN use in medicine	4
2.1.3	SVM use in medicine	4
2.2	Neural networks	5
2.2.1	Biological neural networks	5
2.2.2	Introduction to ANNs	6
2.2.3	Training of the ANN	9
2.3	Support Vector Machines	11
2.3.1	Kernel functions	11
2.3.2	Multi-class classification	12
3	Approach	15
3.1	Recognition of skin diseases	15
3.2	Chosen diseases	15
3.3	Dataset	18
3.3.1	Image preprocessing	19
3.4	<i>Skinchecker</i> with ANN	19
3.5	<i>Skinchecker</i> with SVM	19
4	Results	21
4.1	Results for ANN	21
4.1.1	Quickprop algorithm	21
4.1.2	Incremental algorithm	22
4.2	Results for SVM	23
4.3	Comparison	24
4.4	Cross validation testing	25

5	Conclusions	27
5.1	Limitations	27
5.2	Future work	27
6	Acknowledgements	29
	References	31
A	Back-Propagation Learning Algorithm description	35
A.1	Training	35
A.2	Classification	36

List of Figures

2.1	Two connected biological neurons	5
2.2	Model of an artificial neuron	6
2.3	Multilayer feed-forward ANN	7
2.4	Selection of the number of layers for solving different problems	8
2.5	Multilayer feedback ANN	8
2.6	Back-propagation of errors values	10
2.7	Optimal hyperplane	11
2.8	Transformation of input space into feature space	12
3.1	Exemplary pictures of diseases	17
3.2	<i>Skinchecker-DataSet</i> module	18
4.1	Training the ANN using quickprop algorithm	22
4.2	Training the ANN using incremental algorithm	23
4.3	Comparison of results	24

List of Tables

3.1 Diseases chosen for training the system.	16
4.1 Results for quickprop algorithm	21
4.2 Number of epochs for incremental training.	22
4.3 Results for incremental algorithm	23
4.4 Results for SVM	24
4.5 Cross validation testing	25

Chapter 1

Introduction

Skin diseases are now very common all over the world. For example the number of people with skin cancer has doubled in the past 15 years. “Currently, between 2 and 3 million non-melanoma skin cancers and 132,000 melanoma skin cancers occur globally each year. One in every three cancers diagnosed is a skin cancer and, according to Skin Cancer Foundation Statistics, one in every five Americans will develop skin cancer in their lifetime” [42]. Different kinds of allergies are also becoming more common. Many of these diseases are very dangerous, particularly when not treated at an early stage. Dermatologists have at their disposal large catalogues with pictures of skin segments, which they use as a reference for diagnosis of their patients cases’ skin ailments [10]. However it may be difficult even for experienced doctors, to make correct diagnosis because many symptoms look very similar to each other, even though they are caused by different diseases. All details such as colour, size or density of the skin changes are important. Modern medicine is looking for solutions, which could help doctors with any aspect of their work using the new technology. Such tools already exist, to our knowledge they concentrate on analysis of the colour of skin changes and UV photography [7, 38].

Although these tools are commercially available, there seems to be a lot of room for further improvement. The goal of this project is to make a system to recognize skin diseases using Artificial Neural Networks (ANNs) and Support Vector Machines (SVM). After testing both methods results will be compared. System should learn from the set of skin segments images taken by digital camera. After that it should return the probability of existence of any recognized disease, based on the same type of image made by user. Images should have the same size and should be taken from the same distance. Some practical study may display need of using other information (not only pictures) to train the network, for example part of the body where the symptoms were found or if the patient feels pain or tickle. Program can also provide some tools that will be found as useful during the study.

Artificial Neural Networks are one of the most efficient methods for pattern recognition [3, 4, 30, 35, 37]. They are used as a model for simulation of the workings of a brain. They cannot replace a doctor, but ANNs can help him in diagnosis [2]. Also SVM is used for pattern recognition and is achieving good accuracy. It expands very quickly and is more and more popular [34, 36]. It can be very useful in many types of applications, also in dermatology.

Chapter 2

Background

2.1 Computer science in medicine

Computer science has been widely adopted by modern medicine. One reason is that an enormous amount of data has to be gathered and analysed which is very hard or even impossible without making use of computer systems. The majority of medical tools are able to send results of their work directly to a computer facilitating significantly collection of necessary information. Computer systems can also lower the risk of misdiagnosis. In the experiment described in [27] an ANN-trained computer, an experienced dermatologist and an inexperienced clinician with minimal training were diagnosing melanoma. Results of the first two diagnoses were similar and were also better than the results of the inexperienced clinician. It shows that computer-aided diagnosis can be a very helpful tool, particularly in areas which lack experienced specialists.

A large number of such tools already exists and they provide an aid to the doctors in their everyday work [7]. In this paper we are concerned with dermatology, ANNs and SVM, therefore some tools applied in this area will be described.

2.1.1 Dermatological tools

Dermatology can be described as a branch of medicine primarily concerned with skin and its diseases. It is one of the most difficult areas in medicine. It demands detailed knowledge and experience [19]. Fortunately dermatologists are now aided by a number of modern inventions which makes their work easier and produces better results for patients.

A very important task of a doctor is to make the correct diagnosis of the illness as well as to estimate its progress. To remember and register the actual state of patients, doctors use some scoring systems. However, there are some factors which cause lack of objectivity in these systems. A problem can ensue when a patient changes his doctor. Previously given scores can be misunderstood because they may vary from doctor to doctor. Some discrepancy can also appear between scores taken by the same doctor in different circumstances. To avoid the human partiality in psoriasis scoring a computer system is proposed in [14].

Digital image analysis role in dermatology has increased recently. Many algorithms from this area are used to diagnose different diseases. A highly efficient algorithm for diagnosis of melanocytic lesions is described in [5]. After the experiment it turned out

that the algorithm developed by [5] can achieve similar accuracy as the dermoscopic classification rules applied by dermatologists. However, a dermatologist is still required to recognize if lesions are of a melanocytic nature.

Some dermatological problems are considered to be very difficult, for example the differential diagnosis of erythematous-squamous diseases. Differences between various types of this illness can be very small; therefore the probability of wrong diagnosis can be significant. To lower it, an expert system which consists of the three classification algorithms is proposed in [15]. A program, called DES¹, uses the nearest neighbour classifier, naive Bayesian classifier and voting feature intervals-5. A combination of these algorithms achieves better results and helps to avoid mistakes in diagnosis. According to [15] DES can be used not only by dermatologists but also by students for testing their knowledge. Another useful feature is a database of the patient records.

2.1.2 ANN use in medicine

ANNs' effectiveness in recognizing patterns and relations is a reason why they are being used to aid doctors in solving medical problems. Until 1996 they have not been used in this area frequently; they have been generally applied in radiology, urology, laboratory medicine and cardiology [20]. Recently ANNs have proven to be useful in many other fields of medicine including dermatology. They have shown large efficiency not only in diagnosis but also in modelling parts of the human body.

One of the most important dermatological problems is melanoma diagnosis. Dermatologists achieve accuracy in recognizing malignant melanoma between 65 and 85%, whereas early detection means decreasing of mortality. System using genetic algorithms and ANNs, proposed in [16], reached 97.7% precision. In the system proposed in [13] 85% sensitivity and 99% specificity were achieved for diagnosis of melanoma.

Another developed system for computer-aided diagnosis of pigmented skin lesions is demonstrated in [17]. Diagnostic and neural analysis of skin cancer (DANAOS) showed the results comparable to results of dermatologists. It was also found that images hard to recognize by DANAOS differed from those causing problems to dermatologists. Cooperation between humans and computers could therefore lower the probability of mistakes. Results obtained are also dependent on the size and quality of the database used.

ANNs have also been adopted in pharmaceutical research [1] and in many other different clinical applications using pattern recognition; for example: diagnosis of breast cancer, interpreting electrocardiograms, diagnosing dementia, predicting prognosis and survival rates [38].

2.1.3 SVM use in medicine

SVM is not so well known as the ANN. However, it was also applied to some medical tools. One of them, proposed in [33], identifies patients with breast cancer for whom chemotherapy could prolong survival time. In the experiment SVM was used for classifying patients into one of three possible prognostic groups: Good, Poor or Intermediate. After classification of 253 patients, 82.7% test set correctness was achieved.

Another application of SVM in medicine is described in [18]. A system for cancer diagnosis used the DNA micro-array data as a classification data set. Diagnosis error

¹Diagnosis of Erythematous-Squamous

obtained by above mentioned system was smaller than in systems which uses other known methods. Reduction of the error achieved 36%.

SVM is relatively new method of classification and it expands very quickly. That will certainly cause wider use of SVM in different areas, also in medicine.

2.2 Neural networks

The ANNs consist of many connected neurons simulating a brain at work. A basic feature which distinguishes an ANN from an algorithmic program is the ability to generalize the knowledge of new data which was not presented during the learning process. Expert systems need to gather actual knowledge of its designated area. However, ANNs only need one training and show tolerance for discontinuity, accidental disturbances or even defects in the training data set. This allows for usage of ANNs in solving problems which cannot be solved by other means effectively [3, 35]. These features and advantages are the reason why the area of ANN's application is very wide and includes for example:

- Pattern recognition,
- Object classification,
- Medical diagnosis,
- Forecast of economical risk, market prices changes, need for electrical power, etc.,
- Selection of employees,
- Approximation of function value.

2.2.1 Biological neural networks

The human brain consists of around 10^{11} nerve cells called neurons (see Fig. 2.1). The

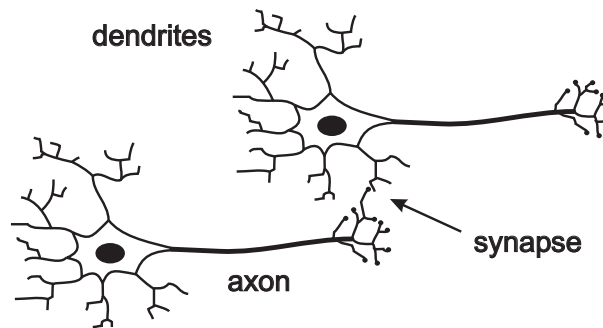


Figure 2.1: Two connected biological neurons. The black area inside a cell body is called nucleus. A signal is sent along the axon and through the synapse is transferred to dendrites of the other neuron (taken from [3] with permission).

nucleus can be treated as the computational centre of a neuron. Here the main processes take place. The output duct of a neuron is called axon whereas dendrite is its input. One neuron can have many dendrites but only one axon; biological neurons have thousands of dendrites. Connections between neurons are called synapses; their quantity in a human

brain is greater than 10^{14} . A neuron receives electrical impulses through its dendrites and sends them to the next neurons using axon. An axon is split into many branches ending with synapses. Synapses change power of received signal before the next neuron will receive it. Changing the strengths of synapse effects is assumed to be a crucial part of learning process and that property is exploited in models of a human brain in its artificial equivalent [3, 35].

2.2.2 Introduction to ANNs

Structure of an artificial neuron, worked out by McCulloch and Pitts in 1943 [22], is similar to biological neuron (see Fig. 2.2). It consists of two modules: summation module

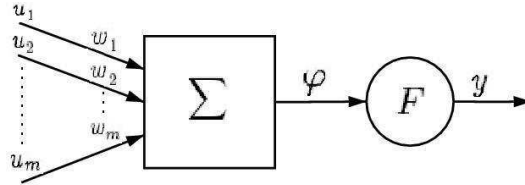


Figure 2.2: Model of an artificial neuron. The weighted sum of the inputs (see equation (2.1)) is transformed by the activation function to give the final output (see equations (2.2), (2.3), (2.4) and (2.5)).

Σ and activation module F . Roughly the summation module corresponds to biological nucleus. There algebra summation of weighted input signals is realised and the output signal φ is generated. Output signal can be calculated using the formula

$$\varphi = \sum_{i=1}^m w_i u_i = \mathbf{w}^T \mathbf{u} , \quad (2.1)$$

where \mathbf{w} is the vector of weights (synapses equivalent), \mathbf{u} - vector of input signals (dendrites equivalent), m - number of inputs. Signal φ is processed by the activation module F , which can be specified by different functions according to needs. A simple linear function can be used, then the output signal y has form

$$y = k\varphi , \quad (2.2)$$

where k is coefficient. Networks using this function are called Madaline and their neurons are called Adaline². They are the simplest networks, which have found practical application. Another type of the activation module function is a threshold function

$$y = \begin{cases} 1, & \varphi > \varphi_h \\ 0, & \varphi \leq \varphi_h \end{cases} , \quad (2.3)$$

where φ_h is constant threshold value. However, functions which describe a non-linear profile of biological neuron more precisely are: a sigmoid function

$$y = \frac{1}{1 + e^{-\beta\varphi}} , \quad (2.4)$$

²ADaptive LINear Element

where β is given parameter and a tangensoid function

$$y = \tanh\left(\frac{\alpha\varphi}{2}\right) \frac{1 - e^{-\alpha\varphi}}{1 + e^{-\alpha\varphi}}, \quad (2.5)$$

where α is given parameter [3, 35, 37].

Information capacity and processing ability of a single neuron is relatively small. However, it can be raised by the appropriate connection of many neurons. In 1958 the first ANN, called perceptron, was developed by Rosenblatt [31]. It was used for alphanumerical character recognition. Although the results were not satisfactory (problems appeared when characters were more complex or the scale was changed) it was successful as the first system built, which simulated a neural network. Rosenblatt has also proved that if a given problem can be solved by a perceptron, then the solution can be found in a finite number of steps [3, 35, 37].

After publication of the book “Perceptrons” by Minsky and Papert in 1969 [24] research in the area of ANNs had come to a standstill. The authors proved that perceptrons cannot solve linearly non-separable problems (e.g. compute an XOR-function). Research has been resumed after almost 15 years by a series of publications showing that these limitations are not applicable to non-linear multilayer networks. Effective learning methods have also been introduced [3, 35, 37].

Neurons in the multilayer ANNs are grouped into three different types of layers: input, output, and hidden layer (see Fig. 2.3). There can be one or more hidden layers

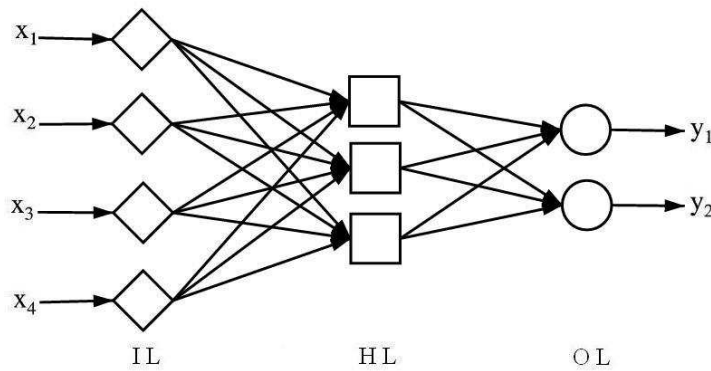


Figure 2.3: Multilayer feed-forward ANN. From the neurons in the input layer (IL) signals are propagated to the hidden layer (HL) and then finally to the output layer (OL).

in the network but only one output and one input layer. The number of neurons in the input layer is specified by the type and amount of data which will be given to the input. The number of output neurons corresponds to the type of answer of the network. The amount of hidden layers and their neurons is more difficult to determine. A network with one hidden layer suffices to solve most tasks. None of the known problems needs a network with more than three hidden layers in order to be solved (see Fig. 2.4). There is no good recipe for the number of hidden neurons selection. One of the methods is described by formula

$$N_h = \sqrt{N_i N_o}, \quad (2.6)$$

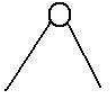
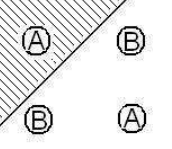
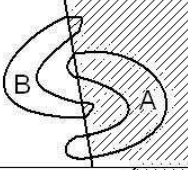
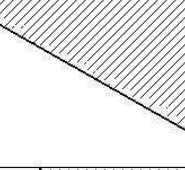
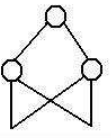
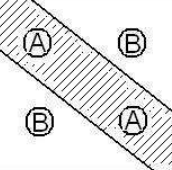
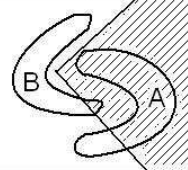
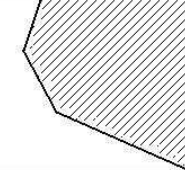
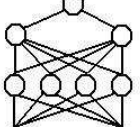
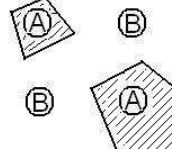
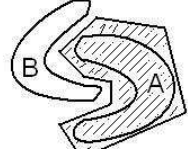
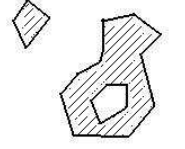
Structure	Type of Decision Regions	Exclusive-OR Problem	Classes with Mesned Regions	Most General Region Shapes
Single-layer 	Half plane bounded by hyperplane			
Two-layers 	Convex open or closed regions			
Three-layers 	Arbitrary (Complexity limited by number of nodes)			

Figure 2.4: Selection of the number of layers for solving different problems. More complicated networks can solve more complicated issues.

where N_h is the number of neurons in the hidden layer, and N_i and N_o are the corresponding numbers for the input and output layers, respectively. However, usually the quantity of hidden neurons is determined empirically [3, 35, 37].

Two types of a multilayer ANNs can be distinguished with regards to the architecture: feed-forward and feedback networks. In the feed-forward networks signal can move in one direction only and cannot move between neurons in the same layer (see Fig. 2.3). Such networks can be used in the pattern recognition. Feedback networks are more complicated, because a signal can be sent back to the input of the same layer with a changed value (see Fig. 2.5). Signals can move in these loops until the proper state is achieved. These networks are also called interactive or recurrent [35, 1].

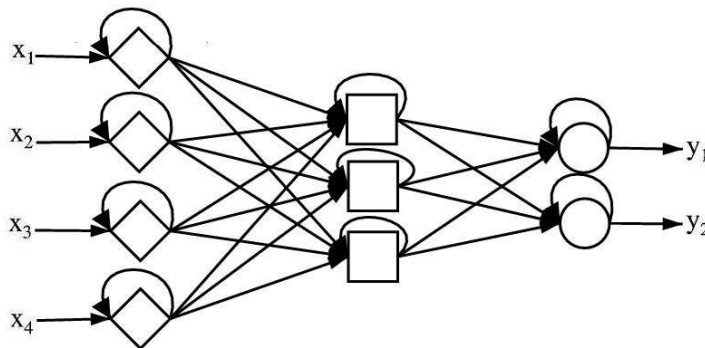


Figure 2.5: Multilayer feedback ANN. A signal can be returned to the same layer to adjust the proper state.

2.2.3 Training of the ANN

The process of training of the ANN consists in changing the weights assigned to connections of neurons until the achieved result is satisfactory. Two main kinds of learning can be distinguished: supervised and unsupervised learning. In the first of them external teacher is being used to correct the answers given by the network. ANN is considered to have learned when computed errors are minimized. Unsupervised learning does not use a teacher. ANN has to distinguish patterns using the information given to the input without external help. This learning method is also called self-organisation. It works like a brain which uses sensory impressions to recognise the world without any instructions [3, 35].

One of the best known learning algorithms is the Back-Propagation Algorithm (BPA). This basic, supervised learning algorithm for multilayered feed-forward networks gives a recipe for changing the weights of the elements in neighbouring layers. It consists in minimization of the sum-of-squares errors, known as least squares. Despite of the fact that BPA is an ill-conditioned optimization problem [12], thanks to specific way of the errors propagation, BPA has become one of the most effective learning algorithms [3, 35, 37].

To teach ANN using BPA the following steps have to be carried out for each pattern in the learning set:

1. Insert the learning vector \mathbf{u}^μ as an input to the network.
2. Evaluate the output values $u_j^{m\mu}$ of each element for all layers using the formula

$$u_j^{m\mu} = f(\varphi_j^{m\mu}) = f\left(\sum_{i=0}^{n_{m-1}} w_{ji}^m u_i^{(m-1)\mu}\right). \quad (2.7)$$

3. Evaluate error values $\delta_j^{M\mu}$ for the output layer using the formula

$$\delta_j^{M\mu} = f'(\varphi_j^{M\mu}) \delta_j^\mu = f'(\varphi_j^{M\mu})(y_j^{z\mu} - y_j^\mu). \quad (2.8)$$

4. Evaluate sum-of-squares errors ξ_μ from

$$\xi_\mu = \frac{1}{2} \sum_{j=1}^n (\delta_j^\mu)^2. \quad (2.9)$$

5. Carry out the back-propagation of output layer error $\delta_j^{M\mu}$ to all elements of hidden layers (see Fig. 2.6) calculating their errors $\delta_j^{m\mu}$ from

$$\delta_j^{m\mu} = f'(\varphi_j^{m\mu}) \sum_{l=1}^{n_{m+1}} \delta_l^{(m+1)\mu} w_{lj}^{(m+1)}. \quad (2.10)$$

6. Update the weights of all elements between output and hidden layers and then between all hidden layers moving towards the input layer. Changes of the weights can be obtained from

$$\Delta^\mu w_{ji}^m = \eta \delta_j^{m\mu} u_i^{(m-1)\mu}. \quad (2.11)$$

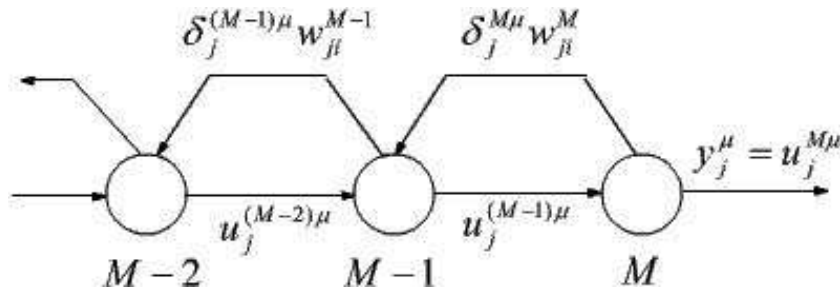


Figure 2.6: Back-propagation of errors values.

Above steps have to be repeated until satisfactory minimum of complete error function is achieved:

$$\xi = \sum_{\mu=1}^P \xi_{\mu} = \frac{1}{2} \sum_{\mu=1}^P \sum_{j=1}^n (y_j^{z\mu} - \varphi_j^{\mu})^2, \quad (2.12)$$

where the symbols used in equations (2.7), (2.8), (2.9), (2.10), (2.11) and (2.12) denote:

- P - number of learning patterns,
- μ - index of actual learning pattern, $\mu = 1, \dots, P$,
- M - number of layers (input layer is not included),
- m - index of actual layer, $m = 1, \dots, M$,
- n_m - number of elements (neurons) in layer m ,
- j - index of actual element, $j = 1, \dots, n_m$,
- φ_j^{μ} - weighted sum of input values for element j in layer μ ,
- f - activation function (see equations (2.2), (2.3), (2.4) and (2.5)),
- w_{ji}^m - weight between element j in layer m and element i in layer $m - 1$,
- $u_i^{(m-1)\mu}$ - output of element i in layer $m - 1$ for pattern μ ,
- δ_j^{μ} - learning error for element j for pattern μ ,
- $y_j^{z\mu}$ - expected network output value for element j for pattern μ ,
- y_j^{μ} - actual network output value for element j for pattern μ ,
- $\Delta^{\mu} w_{ji}^m$ - change of given weight for pattern μ ,
- η - proportion coefficient.

Every iteration of these instructions is called epoch. After the learning process is finished another set of patterns can be used to verify the knowledge of the ANN. For complicated networks and large sets of patterns the learning procedure can take a lot of time. Usually it is necessary to repeat the learning process many times with different coefficients selected by trial and error [3, 35, 37].

There is a variety of optimisation methods which can be used to accelerate the learning process. One of them is momentum technique [25], which consists in calculating the changes of the weights for the pattern $(k + 1)$ using formula

$$\Delta^{\mu} w_{ji}^m(k + 1) = \eta \delta_j^{m\mu} u_i^{(m-1)\mu} + \alpha \Delta^{\mu} w_{ji}^m(k) \quad (2.13)$$

where α is constant value which determines the influence of the previous change of weights to the current change. This equation is more complex version of equation (2.11).

2.3 Support Vector Machines

SVM proposed by Vapnik [6, 39, 40, 41] was originally designed for classification and regression tasks, however later has expanded in another directions [34]. Essence of SVM method is construction of optimal hyperplane, which can separate data from opposite classes using the biggest possible margin. Margin is a distance between optimal hyperplane and a vector which lies closest to it. An example of such hyperplane is illustrated on figure 2.7. As it can be seen on the drawing, there can be many hyperplanes which

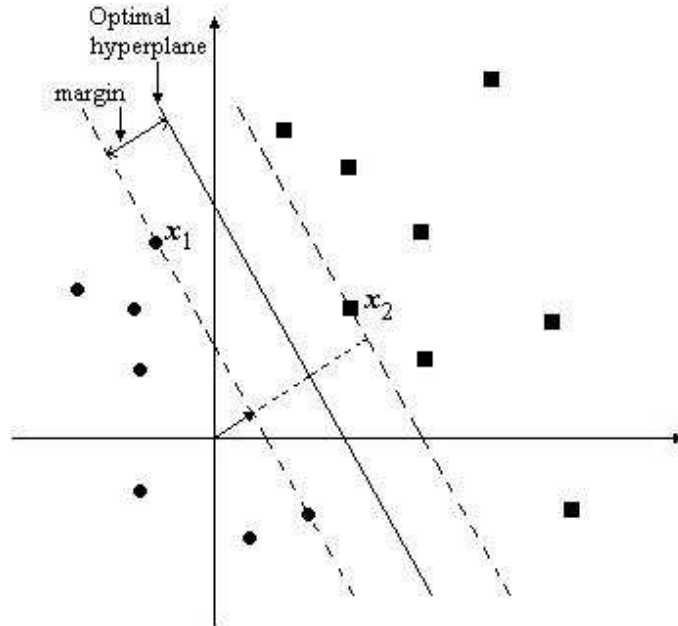


Figure 2.7: Optimal hyperplane separating two classes.

can separate two classes, but with regard to optimal choice, the most interesting solution can be obtained by gaining the biggest possible margin. Optimal hyperplane should satisfy

$$\frac{y_k F(\mathbf{x}_k)}{\|\mathbf{w}\|} \geq \tau \quad k = 1, 2, \dots, n, \quad (2.14)$$

where τ is a margin and $F(\mathbf{x})$ is defined as:

$$F(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b. \quad (2.15)$$

As we have mentioned in section 2.2.2, this function is not suitable for solving more complicated, linearly non-separable problems [32, 34, 21, 36].

2.3.1 Kernel functions

Possibility of occurrence of the linearly non-separability in the input space consist the cause why the idea of SVM is not optimal for hyperplane construction in the input space but rather in high-dimensional so called feature space Z . The feature space is

usually defined as a non-linear product of base functions $\phi_i(\mathbf{x})$, defined in the input space. Function of the optimal hyperplane is now:

$$F(\mathbf{x}) = \sum_{i=1}^n a_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \quad (2.16)$$

where $K(\mathbf{x}_i, \mathbf{x})$ is the inner product kernel of base functions $\phi_j(\mathbf{x})$, $j = 1, 2, \dots, m$. Inner product may be defined as:

$$K(\mathbf{x}, \mathbf{x}') = \phi_i(\mathbf{x})^T \phi_i(\mathbf{x}') \quad (2.17)$$

We are now looking for solution in other space, but the problem is linearly separable, so it is more effective, even if the problem was linearly non-separable in the input space (see Fig. 2.8).

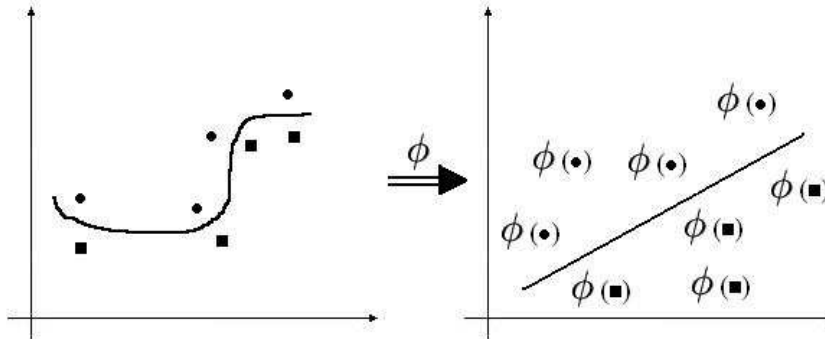


Figure 2.8: Transformation of input space into feature space.

Inner product $K(\cdot)$ can have many different forms. Some of the commonly used functions are: polynomial

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d, \quad (2.18)$$

Gaussian, which in practice gives the best results,

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right), \quad (2.19)$$

where $\sigma > 0$, and sigmoid kernels

$$K(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \langle \mathbf{x}, \mathbf{x}' \rangle + \vartheta), \quad (2.20)$$

where $\kappa > 0$ and $\vartheta < 0$ [34, 21, 36].

2.3.2 Multi-class classification

Although SVM method is naturally adapted for separating data from two classes, it can be easily transformed into very useful tool for the classification of more than two classes. There are two basic ways of solving the N -class problem:

- Solving N number of two class classification tasks,
- Pairwise classification,

The first method consists in teaching of many classifiers using *one versus the rest* method. It means that while solving every i -th task ($i = 1, 2, \dots, N$) we carry out the separation of one, current class from the other classes and every time new hyperplane comes into being

$$y(\mathbf{x}; \mathbf{w}_i, b_i) = \mathbf{w}_i^T \mathbf{x} + b_i = 0. \quad (2.21)$$

Support vectors, which belong to class i satisfy $y(\mathbf{x}; \mathbf{w}_i, b_i) = 1$, whereas the other ones satisfy condition $y(\mathbf{x}; \mathbf{w}_i, b_i) = -1$. If for a new vector we have

$$y(\mathbf{x}; \mathbf{w}_i, b_i) > 0, \quad (2.22)$$

then the vector is assigned for class i . However, it may happen, that (2.22) is true for many i or is not true for any of them. For such cases the classification is unfeasible [34, 36].

In pairwise classification N -class problem is replaced with $N(N-1)/2$ differentiation tasks between two classes. However, the number of classifiers is greater than in the previous method, individual classifiers can be trained faster and depending on the dataset this results in time savings [34]. Unambiguous classification is not always possible, since it may happen that more than one class will get the same number of votes [36]. Solution of that problem is described in [28].

Chapter 3

Approach

3.1 Recognition of skin diseases

Dermatology is considered to be heavily dependent on visual estimations. Even very experienced doctors have to continually verify their knowledge. The changes of the skin are visible but they are hard to identify because of the large number of different skin diseases, which have similar or identical symptoms.

“Dermatology is different from other specialties because the diseases can easily be seen. Keen eyes, aided sometimes by a magnifying glass, are all that are needed for a complete examination of the skin. Often it is best to examine the patient briefly before obtaining a full history. A quick look will prompt the right questions” [19, p.35].

As it was shown in the chapter 2, ANNs and SVM are very powerful tools for pattern recognition, which extent the role of magnifying glass in the smart way. During diagnosis of skin diseases, the image of the skin fragment has to be inspected. To help dermatologists with the diagnosis we developed the *Skinchecker* system.

3.2 Chosen diseases

From among 789 dermatological pictures received from professor Ryszard Żaba from the Dermatological Clinic of Poznań University of Medical Sciences we have chosen 215 with best quality and best applicability in pattern recognition area. Those pictures represent 7 different diseases (see Fig. 3.1) which are shortly described in this section. Numbers of pictures available for each of chosen diseases are presented in the table 3.1.

Acne Vulgaris (see Fig. 3.1a) is a chronic inflammatory disease characterized by cysts, open and closed comedones (plugged lesions containing a “cottage-cheese” like material), pus pockets, and raised red swellings. Often occurs in puberty, though may occur in the 20s or 30s. Several factors are involved in the pathogenesis: inheritance, increased sebum production, an abnormality of the microbial flora, cornification of the pilosebaceous duct and the production of inflammation [11, 19, 9].

Atopic¹ Dermatitis (see Fig. 3.1b), also known as Eczema, is generally attributed to a malfunction in the body’s immune system. It tends to occur in people who have a

¹Greek: a-topos - without a place.

Disease	Number of pictures
Acne Vulgaris	29
Atopic Dermatitis	29
Granuloma Annulare	19
Keloid	29
Melanocytic Nevus	75
Melanoma Maligna	13
Nevus Pilosus	21

Table 3.1: Diseases chosen for training the system.

family history of allergies. Many children who get this disease outgrow it at puberty, but later develop dermatitis of the hands as adults. Risk factors may include exposure to tobacco smoke, skin infections, changes in climate, irritating chemicals, food allergies, and stress. It is manifested by lichenification, excoriation, and crusting involving mainly the face, neck and the flexural surfaces of the elbow and knee. The clinical picture varies greatly depending on the age of the patient, the course and duration of the disease. Atopic eczema may be distinguished from other types of dermatitis because of its typical course and associated stigmata. About 15% of the population has at least one atopic manifestation and its prevalence is rising [11, 19, 9].

Granuloma Annulare (see Fig. 3.1c) is characterized by a ring composed of smooth, firm papules or nodules on the skin which gradually enlarges. Multiple lesions may be present. In the generalized form, multiple small skin-coloured, erythematous or violaceous lesions appear in a symmetrical fashion on the trunk and, to a lesser extent, on the limbs, and the distinctive annular pattern is not always present [9].

Keloid (see Fig. 3.1d) is an excessive connective tissue response to skin injury characterized by firm red or pink plaques extending beyond the area of the original wound. They are most frequently found over the upper trunk and upper arms. Keloids are more common in darkly pigmented individuals, and most cases have been reported in patients between the ages of 10 and 30. There is a familial tendency to keloid formation. Normally, when the skin is cut, the cells of the underlying connective tissue rush in to repair the area by forming a collagen-based scar. Under normal circumstances, the cells know at what point they should stop making scar tissue. However, in some genetically predisposed individuals, the cells do not stop when they are supposed to, causing excess heavy scar tissue to form. Treatment of keloids is difficult, and as of now, the best treatment is prevention [11, 19, 9].

Melanocytic Nevus (see Fig. 3.1e) is some of the most common growth that occurs on the skin. Melanocytic nevi can be present at birth or around the time of birth. These tend to be larger moles and are referred to as congenital nevi. During childhood “regular moles” may begin to appear. Typically they are skin color to dark brown, flat to dome shaped growths that can appear anywhere on the skin. They are harmless, but sometimes they can be difficult to tell from skin cancer by lay persons. This can be especially true for a type of mole called a dysplastic nevus. Any growth that suddenly changes in size, color, shape, bleeds, itches on a regular basis or becomes inflamed or irritated needs to be evaluated by a dermatologist [23, 19].

Melanoma Malignas (see Fig. 3.1f) is a type of skin cancer that originates in the melanocytes, the skin cells containing pigment or color scattered throughout the body. In the United States alone, 32000 people are affected per year. Melanoma is one of the

fastest growing cancers, increasing at 4.3% per year. One person dies from melanoma per hour. The melanoma may proliferate locally, spread by satellite lesions, or extend via the lymphatics or blood stream, from which it may invade any organ of the body, especially the lungs, liver, brain, skin, and bones [11, 19, 9].

Nevus Pilosus (see Fig. 3.1g) is a pigmented area with increased hair growth. It is present at birth but may continue to develop during infancy. Usually, the light brown to black coloured lesions are raised, and deeper nodules may occur. The lesions vary in size from approximately 1 cm in diameter to large expanses, occasionally covering an entire trunk or leg (so-called giant or garment nevi) [9].

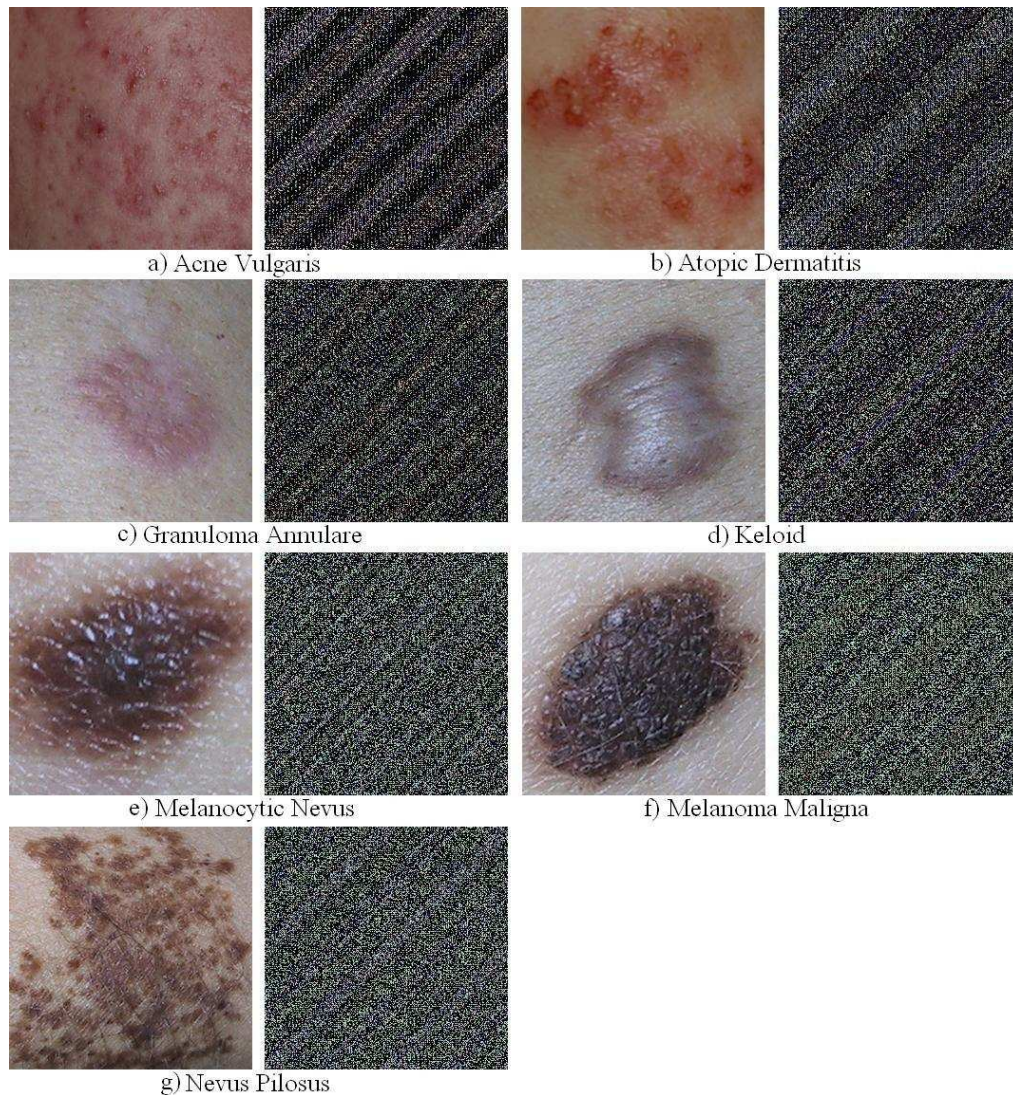


Figure 3.1: Exemplary pictures of skin diseases and the same images after Fourier transformation: a) Acne Vulgaris, b) Atopic Dermatitis, c) Granuloma Annulare, d) Keloid, e) Melanocytic Nevus, f) Melanoma Maligna, g) Nevus Pilosus.

3.3 Dataset

Before training programs can be used, all the pictures have to be compiled into proper dataset file. We have developed the *Skinchecker-DataSet* module (see Fig. 3.2) which provides all necessary, for our purposes, tools for image preprocessing. It also simplifies the organization of lists of all diseases and their images. User can simply add new diseases and assign pictures to them as well as remove any not used data. All lists can be saved and used in future, what is very useful when there is a need to prepare a dataset from the same collection of pictures but with another parameters or when some images need to be added to the list.

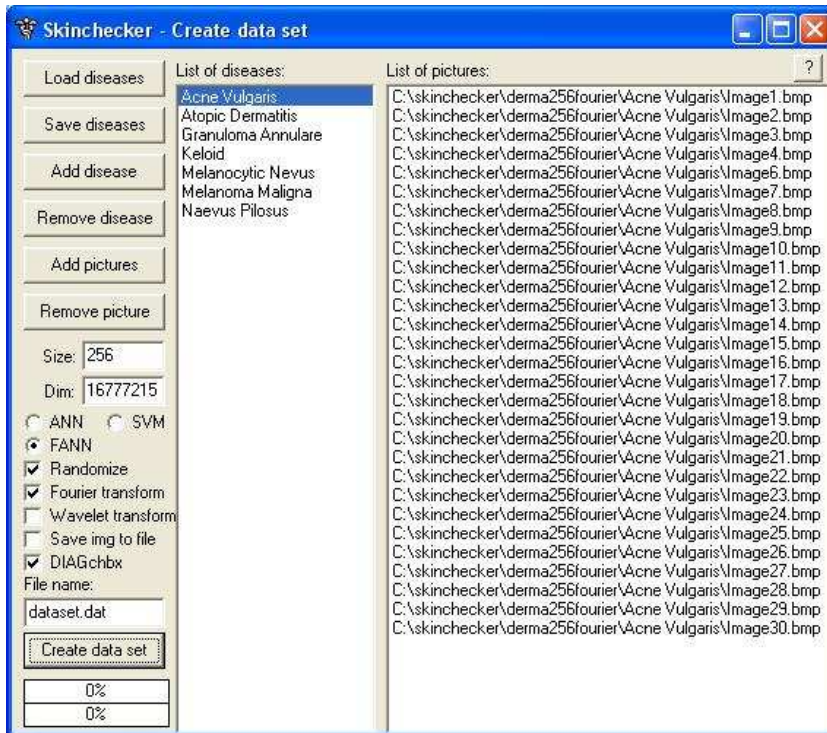


Figure 3.2: Interface of *Skinchecker-DataSet* module. It builds dataset files from images of different diseases given in the list.

When the list of diseases and their images is ready, some options should be also set. First of all, user should decide if the dataset will be used for ANN or SVM version of *Skinchecker*, because training files differ slightly between two versions. User can also choose whether images will be ordered in the dataset file as they are in the list of images or if they should be compiled randomly, what can be significant for some learning algorithms. Fourier or Wavelet transforms can be also performed on pictures, what is explained in section 3.3.1. There is also an option to save images which were previously transformed. After clicking “Create data set” button dataset file will be created. That file can be then used by ANN or SVM version of *Skinchecker* training module.

3.3.1 Image preprocessing

After some trials with training using whole images were made, it appeared that there is a need to perform some additional image preprocessing. First tests took large amount of time and were not effective, because of large input vectors (n^2 input neurons for $n \times n$ pictures). The total network error could not go below 0,3. The situation has changed after performing Fourier transform [29] (see Fig. 3.1). After taking the average value of diagonal lines of pixels from transformed images we have obtained spectrum, which we have used as a pattern for recognition ($2n$ input neurons for $n \times n$ pictures). Such system can be trained in reasonable time and gives satisfactory results. However, there is also a limitation, because of Fast Fourier Transform (FFT) algorithm requirements the size of an image must be power of 2 [29]. We have used 256×256 pictures, therefore the number of input neurons is 512.

3.4 *Skinchecker* with ANN

This version of *Skinchecker* uses Fast Artificial Neural Network Library (FANN) to create and train the network. FANN is a free open source neural network library, which implements multilayer artificial neural networks in C [26].

In our tests we have used 2 algorithms from among those available in FANN: quick-prop and incremental. The best results have been achieved for incremental algorithm, which is standard backpropagation algorithm, where the weights are updated many times during a single epoch. We have build ANN with 512 neurons in the input layer, 7 in the output layer and 12 in the hidden layer. Results of testing the network are presented in the next chapter.

3.5 *Skinchecker* with SVM

This version of *Skinchecker* uses LIBSVM library [8], which provides the support vector classification (SVC) in two versions: C-SVC and ν -SVC. After using the C-SVC method it has appeared that results were not satisfactory. Therefore we have decided to use ν -SVC and we have obtained satisfactory results for $\nu < 0,3$ (presented in next chapter). More information about above mentioned SVC methods can be found in [8, 36].

Chapter 4

Results

4.1 Results for ANN

We used method described in the chapter 3. We have built the 3-layers ANN with 512 inputs, 7 outputs and 12 neurons in hidden layer. In this section results of testing that ANN are presented.

4.1.1 Quickprop algorithm

While training the network, we have set the desired total network error to 0,001. We have set also the maximum number of epochs to 3000. Unfortunately it was impossible for quickprop algorithm to achieve that error, however obtained values are near those desired. They vary between 0,003 and 0,005 according to the momentum value. The Fig. 4.1 presents how the learning proceeded. It shows the total network error in consecutive epochs. We can see that the best error flow was for $\alpha = 0,5$, it is near the perfect desired flow.

Disease	NoP	$\alpha = 0,2$	$\alpha = 0,4$	$\alpha = 0,5$	$\alpha = 0,6$	$\alpha = 0,7$
Acne Vulgaris	29	93,6%	96,0%	94,4%	95,2%	96,0%
Atopic Dermatitis	29	95,3%	94,0%	95,3%	96,0%	93,6%
Granuloma Annulare	19	94,4%	94,3%	93,4%	95,2%	93,9%
Keloid	29	95,2%	94,5%	95,0%	94,7%	94,1%
Melanocytic Nevus	75	96,7%	97,5%	97,0%	96,4%	96,0%
Melanoma Maligna	13	95,4%	94,3%	92,0%	93,0%	93,6%
Nevus Pilosus	21	95,1%	94,9%	95,5%	95,9%	94,1%
Total	215	95,1%	95,1%	94,7%	95,2%	94,5%

Table 4.1: Results of testing the ANN with quickprop algorithm for different momentum values. NoP - number of pictures used for training the ANN.

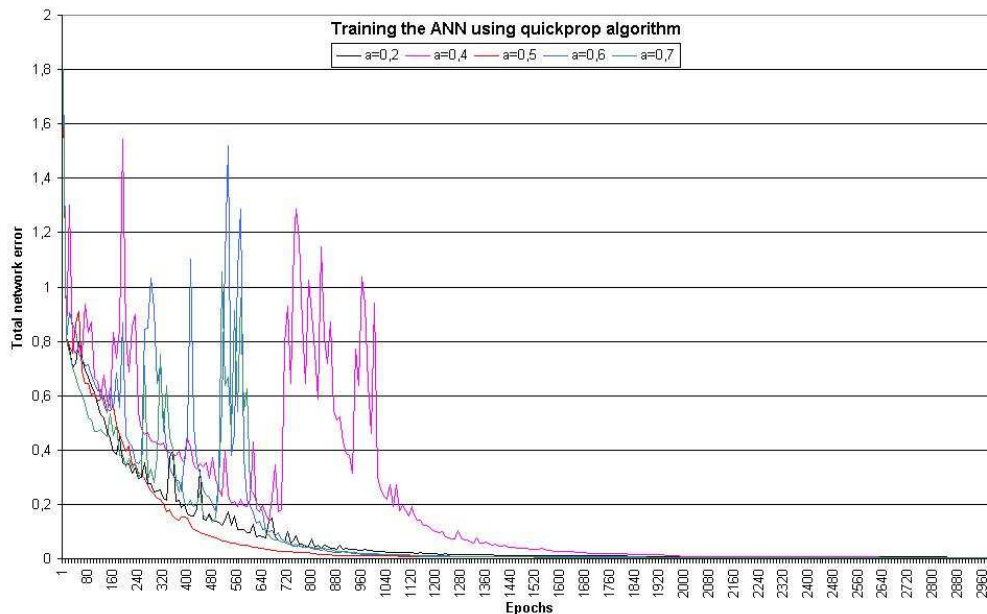


Figure 4.1: Training the ANN using quickprop algorithm. The chart presents total network error through epochs for different momentum values.

4.1.2 Incremental algorithm

Using incremental algorithm we were able to achieve desired total network error in less than 3000 epochs. The exact number of epochs needed to achieve desired error for each momentum value is presented in table 4.2. The Fig. 4.1 presents how the learning proceeded. It shows the total network error in consecutive epochs. We can observe that the error plots are here sharper than it was in previous algorithm. It is because in incremental algorithm weights are changed after every training vector (what accelerate the training process), whereas in quickprop algorithm after entire epoch.

α	Epochs
0,4	2821
0,5	1728
0,6	786
0,7	884
0,8	1123

Table 4.2: Number of epochs needed to achieve desired error ($\xi = 0,001$) for each momentum (α) value using incremental algorithm.

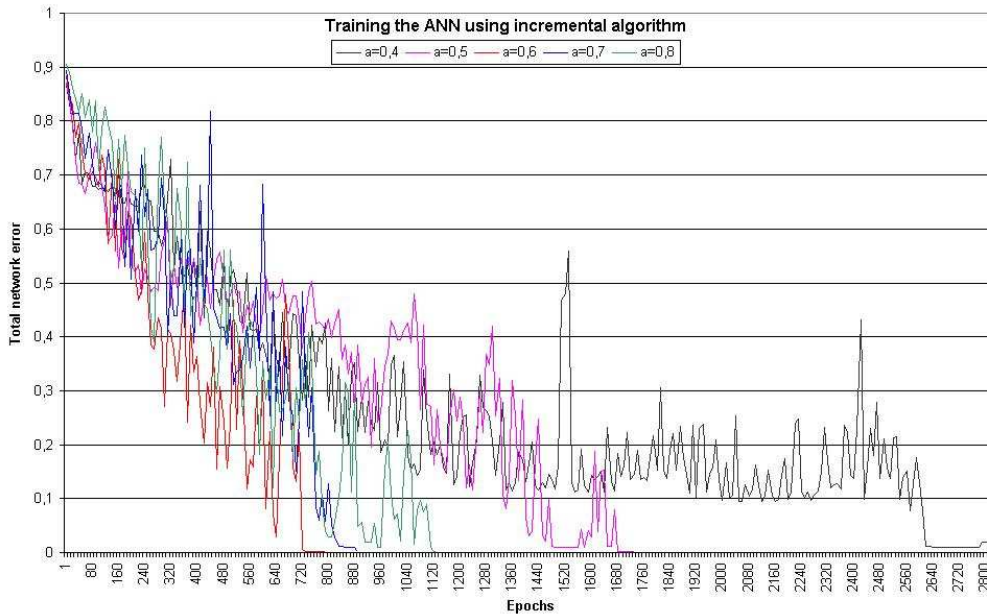


Figure 4.2: Training the ANN using incremental algorithm. The chart presents total network error through epochs for different momentum values.

Disease	NoP	$\alpha = 0,4$	$\alpha = 0,5$	$\alpha = 0,6$	$\alpha = 0,7$	$\alpha = 0,8$
Acne Vulgaris	29	96,7%	97,4%	98,9%	98,4%	99,6%
Atopic Dermatitis	29	99,8%	98,6%	99,2%	99,0%	99,9%
Granuloma Annulare	19	95,0%	96,6%	96,6%	96,4%	99,8%
Keloid	29	99,8%	97,4%	96,1%	96,8%	99,8%
Melanocytic Nevus	75	99,4%	98,7%	99,9%	99,0%	99,8%
Melanoma Maligna	13	98,1%	97,7%	98,7%	97,3%	97,6%
Nevus Pilosus	21	98,9%	99,8%	99,2%	98,7%	95,5%
Total	215	98,2%	98,0%	98,4%	97,9%	98,9%

Table 4.3: Results of testing the ANN with incremental algorithm for different momentum values. NoP - number of pictures used for training the ANN.

4.2 Results for SVM

We have used ν -SVC for classification of pictures, as it was outlined in chapter 3. The results of testing the SVM for different ν value are presented in the table 4.4. Results for Melanoma Maligna are much worse comparing to other diseases: partially because it's the smallest training set but also because those pictures are quite similar to Melanocytic Nevus, which is the biggest set. This is also a common problem in dermatology to differentiate between Melanocytic Nevus which is not very dangerous and Melanoma Maligna which can be lethal.

Disease	NoP	$\nu = 0,05$	$\nu = 0,1$	$\nu = 0,15$	$\nu = 0,2$	$\nu = 0,25$
Acne Vulgaris	29	89,3%	89,9%	89,4%	89,6%	89,5%
Atopic Dermatitis	29	84,2%	85,0%	84,8%	84,9%	84,8%
Granuloma Annulare	19	91,3%	90,8%	90,7%	90,9%	90,8%
Keloid	29	92,5%	92,9%	93,0%	93,0%	92,7%
Melanocytic Nevus	75	89,1%	89,4%	89,6%	87,2%	82,3%
Melanoma Maligna	13	57,3%	58,1%	60,4%	43,2%	18,9%
Nevus Pilosus	21	88,9%	90,1%	90,0%	90,1%	91,7%
Total	215	84,7%	85,2%	85,4%	82,7%	78,7%

Table 4.4: Results of testing the SVM with ν -SVC for different ν values. NoP - number of pictures used for training the SVM.

4.3 Comparison

The Fig. 4.3 presents a comparison of the best results achieved by each method. It appears that much better results in classification were obtained using ANN than SVM. It seems also that ANNs are more resistant to insufficient data amount, because even for small set of Melanoma Maligna pictures results were satisfactory. That cannot be said about SVM, which had a problem with classification of above mentioned disease and mislead it with Melanocytic Nevus.

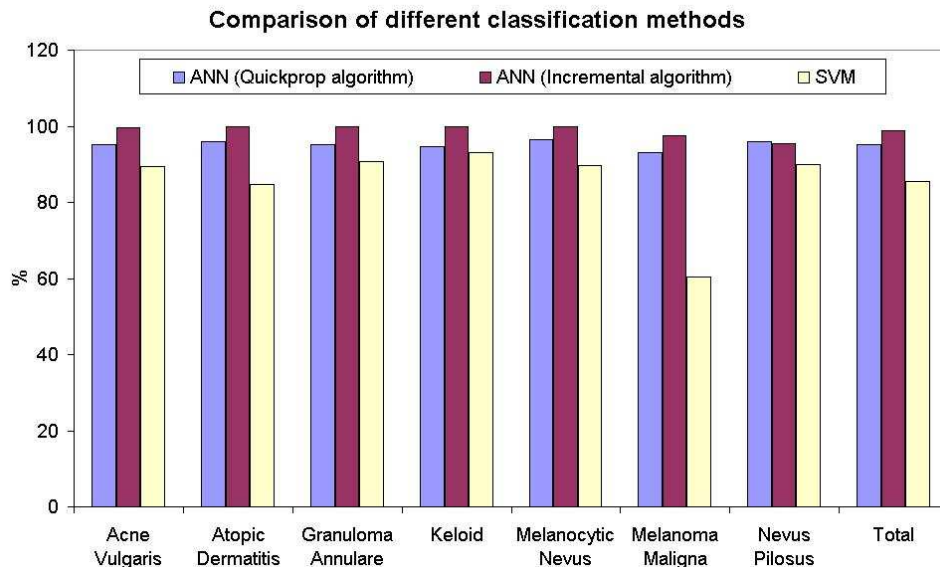


Figure 4.3: Comparison of results.

Comparing both used ANN's algorithms we can observe that better results were obtained using incremental algorithm than quickprop algorithm. However, it can change when bigger, more complicated datasets will be used. Incremental algorithm is faster, but if there will be more diseases to classify it may appear that quickprop algorithm will be more effective.

4.4 Cross validation testing

We have used holdout method to perform the cross validation. From all 215 pictures we have randomly selected 129 (60%) for training and 86 (40%) for validation. Results are presented in the table 4.5. We can observe that changes in ANN's results are not significant, whereas SVM's accuracy has decreased over 30%.

Disease	SVM			ANN			
	TS	VS	$\nu = 0, 15$	TS	VS	Quick	Incr
Acne Vulgaris	16	13	45,1%	15	14	96,1%	99,6%
Atopic Dermatitis	20	9	42,1%	20	9	96,3%	100%
Granuloma Annulare	15	4	42,9%	15	4	94,0%	98,7%
Keloid	18	11	55,0%	18	11	95,4%	97,2%
Melanocytic Nevus	38	37	64,1%	45	30	97,4%	99,5%
Melanoma Maligna	9	4	48,4%	7	6	96,8%	97,5%
Nevus Pilosus	13	8	66,1%	9	12	97,4%	100%
Total	129	86	52,0%	129	86	96,2%	98,9%

Table 4.5: Results of testing the SVM and ANN using cross validation. TS - number of samples used for training, VS - number of samples used for validation, Quick - results for quickprop algorithm with $\alpha = 0,5$, Incr - results for incremental algorithm with $\alpha = 0,8$.

Chapter 5

Conclusions

ANNs are very efficient tools for pattern recognition and they can be successfully used in dermatological applications. They can increase the detectability of dangerous diseases and lower the mortality rate of patients. *Skinchecker* can recognize diseases but the doctor has to decide if there is a need to analyse some part of the skin. After using the program, the doctor has to decide what to do with the results. The data obtained from the program can be helpful. Although computers cannot replace the dermatologist, they can make his work easier and more effective. Proposed system might be also very useful for general practitioners, who do not have wide knowledge about dermatology.

5.1 Limitations

Currently *Skinchecker* can be used only on Windows platform, it was tested on Windows XP. However it can be easily transferred to any other platform supported by FANN library or LIBSVM.

5.2 Future work

Although *Skinchecker* has classified diseases correctly, there are still a lot of improvements which can be done to increase its accuracy. Some additional information can be added to the training set, for example age of the patient, colour of the skin, etc. Bigger datasets with more different diseases should be used to make the use of *Skinchecker* reasonable, however it is hard to obtain enough pictures currently. Also a tool, which will help the doctor to prepare the picture for classification would be desirable, as well as some hardware to obtain proper images of the skin. Set of instructions about how to take usable pictures, including information about light, distance from a patient, etc. should improve the usability of *Skinchecker*.

Chapter 6

Acknowledgements

I would like to thank Michael Minock from Umeå University and Michał Mucha from Adam Mickiewicz University in Poznań for ideas and help with realisation of this project. I would also like to thank professor Ryszard Żaba from the Dermatological Clinic of Poznań University of Medical Sciences for dermatological pictures and for all medical advices. Many thanks to Asim Sheikh for checking my english. Also to Per Lindström for his help during my study period in Umeå and Jürgen Böstler for teaching me how to write publications.

References

- [1] S. Agatonovic-Kustrin and R. Beresford. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, 22:717–727, 2000.
- [2] Michał Antkowiak. Recognition of skin diseases using artificial neural networks. In *Proceeding of USCCS'05*, pages 313–325, 2005.
- [3] Chris M. Bishop. Neural networks and their applications. *Review of Scientific Instruments*, 65(6):1803–1832, 1994.
- [4] Chris M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, UK, 1995.
- [5] A. Blum, H. Luedtke, U. Ellwanger, R. Schwabe, G. Rassner, and C. Garbe. Digital image analysis for diagnosis of cutaneous melanoma. development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions. *British Journal of Dermatology*, 151:1029–1038, 2004.
- [6] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifier. In *5th Annual ACM Workshop on COLT*, pages 144–152, 1992.
- [7] Canfield Scientific. Home Page. <http://www.canfieldsci.com>, accessed April 27, 2005, 2005.
- [8] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [9] DermIS. Dermatology Information Service. <http://www.dermis.net/>, accessed March 30, 2006, 2006.
- [10] Anthony du Vivier. *Atlas of Clinical Dermatology*. Churchill Livingstone, 2002.
- [11] eCureMe. Medical Dictionary. <http://www.ecureme.com/>, accessed March 30, 2006, 2006.
- [12] Jerry Eriksson, Mårten Gulliksson, Per Lindström, and Per-Åke Wedin. Regularization tools for training large feed-forward neural networks using automatic differentiation. *Optimization Methods and Software*, 10:49–69, 1998.
- [13] Monika Gniadecka, Peter Alshede Philipsen, Sigurdur Sigurdsson, Sonja Wessel, and et al. Melanoma diagnosis by Raman spectroscopy and neural networks: Structure alterations in proteins and lipids in intact cancer tissue. *J Invest Dermatol*, 122:443–449, 2004.

- [14] David Delgado Gomez, Toke Koldborg Jensen, Sune Darkner, and Jens Michael Carstensen. Automated visual scoring of psoriasis. <http://www.imm.dtu.dk/visiondag/VD02/medicinsk/articulo.pdf>, 2002.
- [15] H. A. Güvenir and N. Emeksiz. An expert system for the differential diagnosis of erythematous-squamous diseases. *Expert Systems with Applications*, 18:43–49, 2000.
- [16] H. Handels, Th. Roß, J. Kreusch, H. H. Wolff, and S. J. Pöpl. Feature selection for optimized skin tumor recognition using genetic algorithms. *Artificial Intelligence in Medicine*, 16:283–297, 1999.
- [17] K. Hoffmann, T. Gambichler, A. Rick, M. Kreutz, and et al. Diagnostic and neural analysis of skin cancer (DANAOS). A multicentre study for collection and computer-aided analysis of data from pigmented skin lesions using digital dermoscopy. *British Journal of Dermatology*, 149:801–809, 2003.
- [18] Ming Huang and Vojislav Kecman. Gene extraction for cancer diagnosis by support vector machines. *Artificial Intelligence in Medicine*, 35:185–194, 2005.
- [19] J. A. A. Hunter, J. A. Savin, and M. V. Dahl. *Clinical Dermatology*. Blackwell Science, 1995.
- [20] Dipti Itchhaporia, Peter B. Snoe, Robert J. Almassy, and William J. Oetgen. Artificial neural networks: Current status in cardiovascular medicine. *J Am Coll Cardiol*, 28(2):515–521, 1996.
- [21] Norbert Jankowski. *Ontogeniczne sieci neuronowe*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, PL, 2003.
- [22] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [23] John L. Meisenheimer. Melanocytic Nevus. <http://www.orlandoskindoc.com>, accessed March 30, 2006, 2006.
- [24] M. Minsky and S. Papert. *Perceptrons*. Cambridge, Mass.: MIT Press, 1969.
- [25] M. Moreira and E. Fiesler. Neural networks with adaptive learning rate and momentum terms. Technical Report 95-04, Institut Dalle Morre D’Intelligence Artificielle Perceptive, Martigny, Valais, Switzerland, 1995.
- [26] Steffen Nissen. Implementation of a fast artificial neural network library (fann). Technical report, Department of Computer Science University of Copenhagen, Copenhagen, DK, 2003.
- [27] D. Piccolo, A. Ferrari, K. Peris, R. Daidone, B. Ruggeri, and S. Chimenti. Dermoscopic diagnosis by a trained clinician vs. a clinician with minimal dermoscopy training vs. computer-aided diagnosis of 341 pigmented skin lesions: a comparative study. *British Journal of Dermatology*, 147:481–486, 2002.
- [28] J. C. Platt, N. Christianini, and J. Shave-Taylor. Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems*, pages 547–553, 2000.

- [29] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 1992.
- [30] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.
- [31] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
- [32] Bernhard Shölkopf. Statistical learning and kernel methods. Technical Report MSR-TR-2000-23, Microsoft Research Limited, Cambridge, UK, 2000.
- [33] Bernhard Shölkopf. Survival-time classification of breast cancer patients. Technical Report 01-03, National Taiwan University of Science and Technology, Taipei, Taiwan, 2001.
- [34] Bernhard Shölkopf and Alex Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [35] Christos Stergiou and Dimitrios Siganos. Neural networks. *SURPRISE 96 Journal*, 4, 1996.
- [36] Piotr Szczepaniak. *Obliczenia inteligentne, szybkie przekształcenia i klasyfikatory*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, PL, 2004.
- [37] Ryszard Tadeusiewicz. *Sieci neuronowe (Neural Networks)*. Akademicka Oficyna Wydawnicza, Warszawa, PL, 1993.
- [38] Andrew Tomlinson. Medical applications for pattern classifiers and image processing. <http://www.railwaybridge.co.uk>, accessed April 27, 2005, 2000.
- [39] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 1995.
- [40] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, USA, 1998.
- [41] V. Vapnik. The support vector method of function estimation. *Neural Networks and Machine Learning*, pages 239–268, 1998.
- [42] World Health Organisation. Ultraviolet radiation and the INTERSUN Programme. <http://www.who.int>, accessed April 27, 2005, 2005.

Appendix A

Back-Propagation Learning Algorithm description

A.1 Training

```
for each layer (except input layer):
  for each neuron in layer:
    for each weight of neuron:
      set random weight;

while total network error is greater than 0.01:
  for each picture in training set:
    read picture's histogram and correct answer;
    scale values to the range between -1 and 1;
    for each layer (except input layer):
      for each neuron in layer:
        sum all ratios of weights and last layer outputs;
        compute outputs of this layer;

    for each neuron in output layer:
      compute neuron's errors
    for each hidden layer:
      for each neuron in layer:
        compute neuron's errors;

    for each layer (except input layer):
      for each neuron in layer:
        for each weight of neuron:
          compute new weight values;

for each picture:
  for each neuron in output layer:
    sum output errors;
compute total network error;
```

A.2 Classification

```
read picture's histogram and correct answer;
scale values to the range between -1 and 1;
for each layer (except input layer):
    for each neuron in layer:
        sum all ratios of weights and last layer outputs;
        compute outputs of this layer;

sort and show the results;
```