

# Schneller Aufbau medizinischer Diagnosesysteme mit ICE

Tim O. Müller, Rainer Stotzka, Michael Beller,  
Nicole V. Ruiter und Volker Hartmann

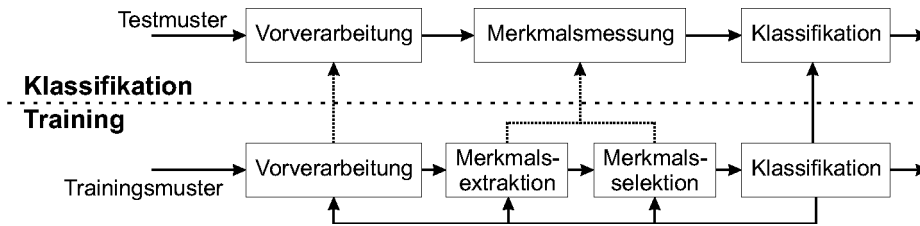
Institut für Prozessdatenverarbeitung und Elektronik,  
Forschungszentrum Karlsruhe, 76344 Eggenstein-Leopoldshafen  
Email: Tim.Mueller@ipe.fzk.de

**Zusammenfassung.** Für den Aufbau eines medizinischen Diagnosesystems muss stets ein neues Klassifikationssystem entworfen werden, da keine allgemeine und für beliebige Probleme verwendbare Lösung existiert. Aufgrund der vielen verschiedenen Kombinationsmöglichkeiten von Merkmalselektoren und Klassifikatoren ist der Entwurf eines guten Klassifikationssystems meist zeitaufwändig und umständlich. Hierfür waren bislang keine Softwarehilfsmittel verfügbar. Die Komponentensoftware ICE unterstützt sowohl den manuellen als auch erstmals den automatischen Entwurf medizinischer Diagnosesysteme, so dass in kurzer Zeit Systeme von guter Qualität erstellt werden können.

## 1 Einleitung

Der Zweck medizinischer Diagnosesysteme besteht darin, den Arzt als Experten bei seiner Arbeit zu unterstützen. Solche Systeme sind in der Lage, aufgrund gemessener Eingangsdaten medizinische Entscheidungen zu treffen. Diese Entscheidungen können von dem behandelnden Arzt ausgewertet werden und in dessen eigene Diagnose mit einfließen. Für die Brustkrebsdiagnose beispielsweise können medizinische Diagnosesysteme von großem Nutzen sein [1].

Der integrale Bestandteil medizinischer Diagnosesysteme ist ein Klassifikationssystem. Abbildung 1 zeigt, wie ein Klassifikationssystem im Allgemeinen aufgebaut ist [2]. Im Training muss das System mit den Mustern einer vorklassifizierten, repräsentativen Stichprobe trainiert werden, um zu lernen, wie sich die einzelnen diagnostischen Entscheidungen (die Klassen) voneinander unterscheiden. Die Trainingsmuster liegen beispielsweise in Form von Bildern oder bereits als extrahierte Merkmale vor. Entsprechend wird der Aufbau eines Klassifikationssystems mit der Vorverarbeitung oder erst mit der Merkmalsextraktion begonnen. Für jeden der vier Arbeitsschritte Vorverarbeitung, Merkmalsextraktion, Merkmalsextraktion und Klassifikation müssen zunächst geeignete Algorithmen ausgewählt werden. Die Qualität des erstellten Systems lässt sich anhand des geschätzten Generalisierungsfehlers messen, also anhand der Fehlklassifikation von Mustern, mit denen nicht trainiert wurde. Je geringer der Generalisierungsfehler, desto besser ist das System.

**Abb. 1.** Prinzipieller Aufbau eines Klassifikationssystems.

Für jeden Verarbeitungsschritt steht eine Vielzahl von Algorithmen zur Verfügung. Das Hauptproblem liegt im Auffinden einer guten Kombination von Merkmalsselektion und Klassifikation. Es existiert kein generisches Klassifikationssystem, das für beliebige Klassifikationsprobleme stets den besten Generalisierungsfehler liefert [2]. Daher muss für jedes neu gestellte Problem ein neues Klassifikationssystem entworfen werden. Die einzelnen Kombinationen führen für unterschiedliche Probleme zu unterschiedlich guten Generalisierungsfehlern. Um das Klassifikationssystem mit dem niedrigsten Generalisierungsfehler zu erstellen, müssten alle denkbaren Kombinationen von Merkmalsselektion und Klassifikation ausprobiert und getestet werden.

## 2 Problemstellung und Stand der Technik

Es stehen bereits Hilfsmittel zur Lösung der einzelnen Teilaufgaben aus Abbildung 1 zu Verfügung. Diese müssen allerdings meist manuell verbunden werden und können dabei auch die Grenzen der jeweiligen Hilfsmittel überschreiten, in denen sie entwickelt wurden. Ein Softwaretool für den Entwurf von Klassifikationssystemen muss über eine große Anzahl von Merkmalsselektoren und Klassifikatoren verfügen. Idealerweise liegen diese in modularer Form, z.B. als Komponenten, vor, so dass ein Klassifikationsablauf wie mit einem Baukastensystem aus diesen Komponenten zusammengesteckt werden kann. Komponentenorientierte Hilfsmittel wie Khoros [3] oder Labview ermöglichen das modulare und visuelle Erstellen von Bildverarbeitungsketten, bieten jedoch keine Unterstützung für den Entwurf von Klassifikationssystemen und verfügen nur über eine geringe Anzahl verschiedener Klassifikationsalgorithmen. Lediglich das WEKA-System [5] bietet eine Vielzahl dieser Algorithmen an. Die manuelle Kombination von Merkmalsselektoren und Klassifikatoren ist mit diesem System teilweise möglich, jedoch stehen weder Algorithmen für die Bildverarbeitung und Merkmalsextraktion zur Verfügung, noch können solche integriert werden.

Von großem Nutzen ist ein Softwaretool, das den Entwurf medizinischer Diagnosesysteme erleichtert, indem es einerseits die Auswahl einer guten Kombination von Merkmalsselektion und Klassifikation unterstützt und andererseits den lückenlosen Aufbau eines Diagnosesystems aus Abbildung 1 ermöglicht. Bislang ist den Autoren kein Softwarehilfsmittel bekannt, das den Entwurf von Klassifikationssystemen in dieser Form unterstützt.

Um die Auswahl einer geeigneten Kombination mit einem Softwaretool zu ermöglichen, müssen die entsprechenden Algorithmen untergeordnete Algorithmen zur Klassifikation instanzieren und kontrollieren können. Dies wird als „self-guided assembly“ bezeichnet [9]. Mit dem kommerziellen Tool QuickCog [4] ließen sich Abläufe wie in Abbildung 1 modellieren, ein „self-guided assembly“ gestattet jedoch keines der bekannten Tools. Gerade bei kommerzieller Software sind Modifikationen in dieser Hinsicht meist nicht möglich. Erwünscht sind zudem Eigenschaften wie die freie Verfügbarkeit, Plattformunabhängigkeit und einfache Erweiterung bzw. Integration vorhandener Bibliotheken.

### 3 Integrated Component Environment – ICE

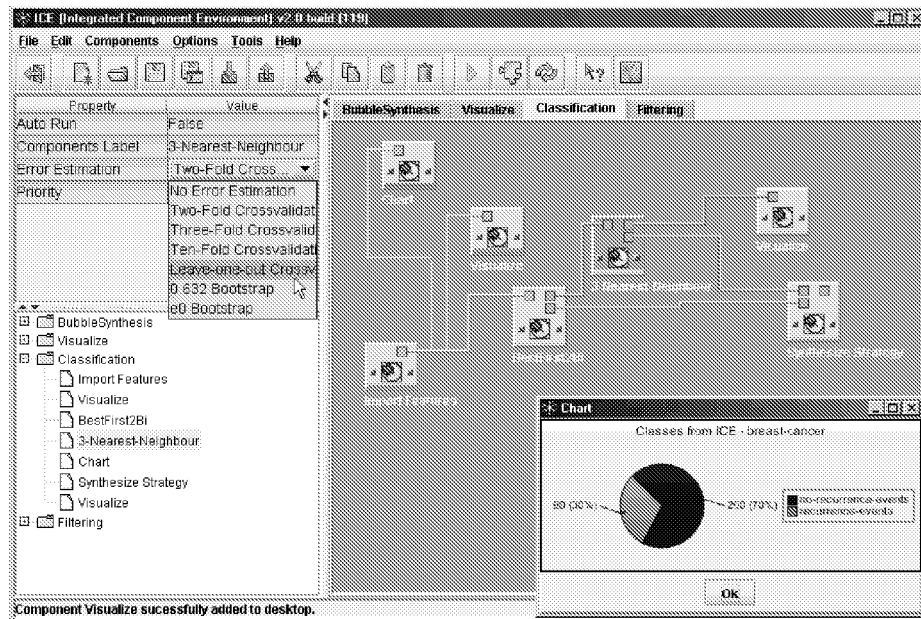
Das „Integrated Component Environment“ (ICE) stellt ein solches Softwaretool dar. Es wurde am Forschungszentrum Karlsruhe entwickelt und in Form einer objektorientierten Komponentensoftware auf Basis von Java bzw. JavaBeans umgesetzt. Das Weka-System bildet mit seinen Klassifikationsalgorithmen den Kern von ICE. Das Konzept der JavaBeans wurde nach eigenen Anforderungen erweitert, um eine einfache Integration bereits vorhandener Bibliotheken beispielsweise für die Bildverarbeitung zu ermöglichen. Abbildung 2 zeigt die graphische Oberfläche von ICE, die ähnlich wie Khoros eine visuelle Programmierung gestattet. Aufgrund der Eigenentwicklung wird den Komponenten die Steuerung der Komponentensoftware zugänglich gemacht [7]. Daher ist mit diesem Tool auch das „self-guided assembly“ möglich. In ICE sind mehr als 100 Komponenten für die Verarbeitungsschritte aus Abbildung 1 implementiert, unter anderem 23 Merkmalsselektoren und 13 Klassifikatoren.

Besonders hervorzuheben ist, dass in ICE Algorithmen integriert sind, die den Entwickler bei der Suche nach einer guten Kombination von Merkmalsselektion und Klassifikation unterstützen. Aus Komplexitätsgründen ist es unmöglich, die beste Kombination zu finden. Daher wurden die folgenden Einschränkungen getroffen: Es werden nur Kombinationen aus genau einem Merkmalsselektor und einem Klassifikator in Betracht gezogen. Weiterhin wird keine Parameteroptimierung der einzelnen Algorithmen durchgeführt.

### 4 Ergebnisse und Diskussion

Mit ICE wurden verschiedene Klassifikationsprobleme medizinischer Motivation untersucht. Hierzu wurde ein handelsüblicher PC mit 733 MHz eingesetzt und Java Version 1.3 unter Windows 2000 und Linux verwendet. Zur Detektion von Mikroverkalkungen in der weiblichen Brust wurde eine vollständige Verarbeitungskette entsprechend Abbildung 1 ausgehend vom Import der digitalen Bilddaten bis hin zur Klassifikation aufgebaut [6]. Die Klassifikationsergebnisse sind in Tabelle 1 unter „Brustkrebsdiagnose B und C“ zu finden. Die weiteren Klassifikationsprobleme standen bereits als extrahierte Merkmale zur Verfügung. Vorverarbeitung und Merkmalsextraktion waren hierfür nicht notwendig.

**Abb. 2.** Graphische Benutzeroberfläche des Integrated Component Environments ICE.



In Tabelle werden die benötigten Zeiten, erreichte und bislang beste Generalisierungsfehler zusammengefasst. Die Generalisierungsfehler der bislang besten Klassifikationssysteme entstammen entweder einem manuellen Aufbau oder sind der Literatur entnommen [5]. Für die meisten Probleme konnten mit ICE in Minuten Klassifikationssysteme erstellt werden, die den bis zu diesem Zeitpunkt bekannten gleichwertig oder sogar überlegen waren. Gegenüber den manuellen Entwürfen, für die je mehrere Tage benötigt wurden, stellt dies eine enorme Verbesserung dar. Durch die erwähnten Einschränkungen wird die Komplexität der Kombinationsmöglichkeiten drastisch reduziert. Möglicherweise wird nicht mehr das optimale Klassifikationssystem erstellt, die Ergebnisse zeigen jedoch, dass in akzeptabler Zeit eine gute Kombination von Algorithmen ermittelt wird.

ICE wird beständig erweitert, beispielsweise wird für die Vorverarbeitung eine merkmalsgesteuerte Segmentierung entwickelt und implementiert [8]. Eine Freigabe als OpenSource-Projekt ist geplant. In ICE ist eine große Anzahl von Komponenten aus dem Bereich der Bildverarbeitung, Registrierung, Mammadiagnostik, Lymphozytenidentifikation, usw. integriert. ICE wird außerdem in zwei Projekten der Mikrosystemtechnik für die Datenakquisition und Vorverarbeitung eingesetzt, weil neue Komponenten einfach zu integrieren sind und einheitliche Schnittstellen vorliegen. Mit ICE lassen sich Verarbeitungsketten aus Abbildung 1 schnell und lückenlos aufbauen lassen. Die Auswahl einer guten Kombination von Merkmalsselektion und Klassifikation wird von ICE unterstützt. Der prototypische Entwurf medizinischer Diagnosesysteme wird dadurch erleichtert.

**Tabelle 1.** Benötigte Erstellungszeiten und Generalisierungsfehler GF mit Hilfe von ICE erstellter von Klassifikationssysteme. (Mu=Muster, Me=Merkmale, Kl=Klassen)

Klassifikationsproblem	Benötigte Zeit	Erreichter GF	Bisher bester GF
Brustkrebsdiagnose A 286 Mu, 9 Me, 2 Kl	3 Minuten	24,1 %	22 %
Brustkrebsdiagnose B 1004 Mu, 85 Me, 2 Kl	2 Stunden	6,5 %	21 %
Brustkrebsdiagnose C 256 Mu, 4096 Me, 2 Kl	17 Stunden	35 %	33 %
Lymphozytenklassifikation 189 Mu, 45 Me, 2 Kl	6 Minuten	1,6 %	2,5 %
Hepatitis 155 Mu, 19 Me, 2 Kl	11 Minuten	11,5 %	17 %
Kontaktlinsen 24 Mu, 4 Me, 3 Kl	1 Minute	16,7 %	17 %
Diabetes 768 Mu, 8 Me, 2 Kl	6 Minuten	19 %	24 %
Herzerkrankungen 302 Mu, 13 Me, 5 Kl	3 Minuten	15 %	21 %
Lymphographie 148 Mu, 18 Me, 4 Kl	6 Minuten	12 %	15 %

## Literaturverzeichnis

1. Giger ML: Computer-aided diagnosis of breast lesions in medical images. *Computing in Medicine*. IEEE Computer Society, 5:39–45, 2000.
2. Jain AK, Duin RPW, Mao J: Statistical Pattern Recognition: A Review. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 1(1):4–36, 2000.
3. D. Argiro, K. Farrar and S. Kubica: Cantata: The Visual Programming Environment for the Khorus System. *Visualization, Imaging and Image Processing Conference Proceedings*, 2001
4. König A, Eberhardt M and Wenzel R: QuickCog Self-Learning Recognition System – Exploiting machine learning techniques for transparent and fast industrial recognition system design. *PennWell, Image Processing Europe*, 10–19, 1999
5. Witten IH and Frank E: *Data Mining*. Morgan Kaufman, 2000
6. Müller TO, Stotzka R, Höpfel D und Yang H: Texturanalyse zur Detektion gruppierter Mikroverkalkungen bei der Brustkrebsfrüherkennung. *Bildverarbeitung für die Medizin BVM*. Springer, 342–346, 2001.
7. Müller TO und Stotzka R: ICE: Komponentensoftware für die computergestützte Diagnose. *Biomedizinische Technik BMT*. Schiele & Schön, 46:378–379, 2001.
8. Beller M, Stotzka R und Gemmeke H: Merkmalsgesteuerte Segmentierung in der medizinischen Mustererkennung. *Bildverarbeitung für die Medizin*. Springer, 2004.
9. Szyperski C: *Component Software – Beyond Object-Oriented Programming*. ACM Press Books, 1999