Ein Baukasten zur Analyse medizinischer Bilddaten mit Hilfe neuronaler Netze und Fuzzy-Logik

Jens Hiltner

Universität Dortmund – Lehrstuhl Informatik I
Otto-Hahn-Straße 16, 44221 Dortmund
Email: hiltner@ls1.cs.uni-dortmund.de

Zusammenfassung Für die Aufgabe der Bildanalyse ist die Erfahrung und das Wissen eines Experten von hohem Stellenwert. Die Analyse medizinischer Bilddaten stellt zudem hohe Anforderungen an die eingesetzten Methoden, die eine beträchtlichen Grad an Variabilität erlauben müssen. Die Vorgehensweise des Experten beruht auf Wissen, welches häufig nur in unscharfer Form beschreibbar ist. Auch ist teilweise die Beschreibung des Vorgehens nicht oder nur schwer verbal durchführbar, jedoch können anhand von Beispielen die gewünschte Ergebnisse beschrieben werden.

In diesem Beitrag soll der Prototyp eines Baukastens vorgestellt werden, der es erlaubt, einem Bildverarbeitungsexperten die Analyse medizinischer Bilddaten zu erleichtern. Dazu stellt der Baukasten verschiedene Funktionen zur Verfügung, deren Einsatzmöglichkeiten durch die Verwendung sogenannter CI-Methoden (Fuzzy-Methoden, künstliche neuronale Netze, evolutionäre Algorithmen) geprägt ist.

1 Einleitung

Die Analyse von Bilddaten stellt eine schwierige Aufgabe da, für die es kein allgemeines „Rezept" gibt. Für diese Aufgabe ist die Erfahrung und das Wissen des Experten, der die Analyse vornimmt, von hoher Bedeutung. Neben dieser benötigten Erfahrung bestehen – sowohl in der Auswertung neuartiger Bilddaten als auch in anderen Fragestellungen an die bekannten Bilddaten – Probleme, die ein experimentelles Vorgehen mit verschiedenen bekannten Verfahren erfordern, um zumindest in die Nähe eines gewünschten Rezeptes (Standardvorgehens) für genau diese Fragestellung zu kommen. Dabei bezieht sich der Begriff „Nähe" auf den benötigten Anteil an Interaktion mit dem System, der sich bei einem „guten Rezept" auf die Auswahl des Bildmaterials beschränken sollte.

Die Verwendung eines Baukastens, wie er hier vorgestellt werden soll, kann durch die Möglichkeit der Nutzung verschiedener „intelligenter" Methoden dazu beitragen, schnell Ergebnisse bei der Erstellung von Lösungen zu neuen Aufgabenstellungen erzielen zu können.

2 Baukasten

Der hier vorgestellte Baukasten besteht aus zwei Teilen. Zum einen existieren verschiedene Editoren zur Definition und Verwaltung der einzelnen Wissensinhalte. In einem Fenster kann ein Bild eingeladen werden, auf welches dann interaktiv realisierte Funktionen angewendet werden können. Zum anderen gibt es eine Programmierumgebung, die eine Vielzahl von Befehlen für den Zugriff auf Bilddaten, Wissenbasen oder neuronale Netze erlaubt. Der Baukasten stellt verschiedene Funktionen zur Verfügung, die auch teilweise schon aus anderen Werkzeugen (etwa Khoros) bekannt sind, deren Einsatzmöglichkeiten aber durch die ergänzende Verwendung sogenannter CI-Methoden [1] deutlich übersteigt.

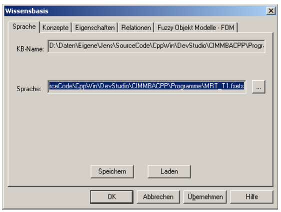
Als Standardfunktionen seien hier etwa vordefinierte Filter (Prewitt, Sobel, Laplace, etc.) genannt, ebenso besteht die Möglichkeit frei definierbarer Filter. Zur Segmentierung stehen neben kantenorientierten und bereichsorientierten auch modellbasierte Methoden zur Verfügung [2, 3]. Die Verwendung verschiedener Bildformate ist möglich (RAW-Format, VFF-Format, DICOM-Format), wobei die Bilddaten sowohl 2D als auch 3D sein dürfen.

2.1 Editoren

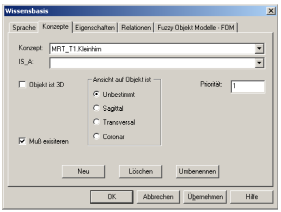
Die oben angesprochenen Erweiterungen zu bisher existierenden Systemen beziehen sich auf die umfangreiche Verwendung von unscharfen (Fuzzy-) Methoden und künstlichen neuronalen Netzen. Die Fuzzy-Methoden werden hauptsächlich zur Repräsentation unscharfen Wissens des Experten verwendet. Diesem ist es erlaubt, sein Domänenwissen in Wissensbasen unterschiedlicher Art zu speichern, etwa zur Beschreibung von Handlungswissen in Form von WENN-DANN-Regeln (wobei die Prämissen und Konklusionen unscharfe Terme sein dürfen) oder zur Beschreibung von Faktenwissen in semantischen Netzen (in denen die Konzepte ebenfalls unscharfe Begriffe verwenden dürfen). Auch medizinisches Fachwissen kann somit hier geeignet abgelegt werden. Diese Wissensbasen werden über eine Komponente des Baukastens verwaltet und können einfach strukturiert angelegt und gepflegt werden. In den Wissensbasen können Beschreibungen gesuchter Objekte mit natürlichsprachlichen Begriffen (Termen) hinterlegt werden, deren Interpretation mittels Fuzzy-Mengen ebenfalls mit Hilfe geeigneter Werkzeuge des Baukastens definiert werden können (Abb. 1).

Die Verwaltung der Interpretationen der verwendeten Sprache (die durch Fuzzy-Mengen hinterlegt sind) erfolgt ebenfalls mit Hilfe eines Editors (Abb. 2(a)). Dieser erlaubt die Definition der Interpretationen von den in der Wissensbasis genutzten Termen. Die Verwendung des gesamten Wissens erfolgt dabei kontextabhängig, um sicherzustellen, daß die jeweils gültige Interpretation eines Termes gewählt wird. Die gespeicherten Konzepte können ebenfalls in einem Kontext gespeichert werden, so daß bei der Auswertung eines Konzeptes immer die Interpretationen gewählt werden, in denen auch der Kontext von Konzept und Interpretation übereinstimmt.

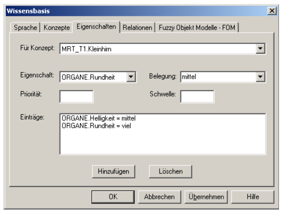
Weiterhin wird die Verwendung künstlicher neuronaler FF-Netze durch den Baukasten unterstützt. Diese können zum einen zur Klassifikation segmentierter



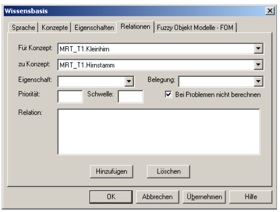
(a) Interpretation



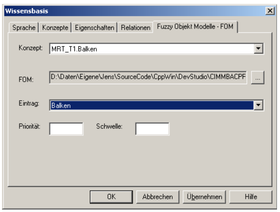
(b) Konzepte



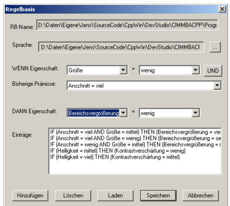
(c) Eigenschaften



(d) Relationen



(e) FOM



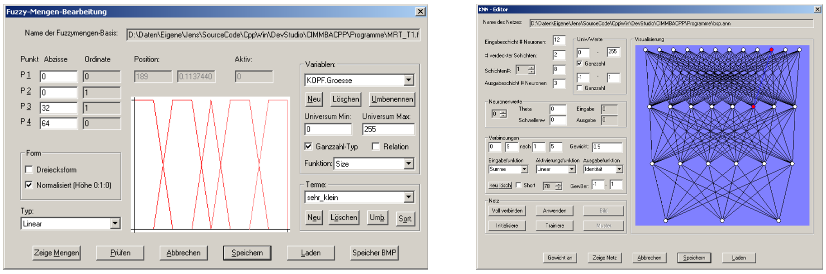
(f) Regelbasis

Abb. 1. Eingabemasken zur Erstellung und Verwaltung der Wissens- und Regelbasen.

Strukturen verwendet werden, aber auch zur Vorsegmentierung von Bilddaten wurden sie erfolgreich eingesetzt [4]. Ein Editor (Abb. 2(b)), der eine Definition der Struktur, der Gewichte des Netzes sowie der Aktivierungsfunktionen erlaubt, wird vom Baukasten zur Verfügung gestellt. Die Trainingsdaten für das jeweils konstruierte Netz können in einer Datei hinterlegt werden, die ggf. auch von anderen Programmen erzeugt wurden (bspw. SNNS – Stuttgarter Neuronale Netze Simulator). Auch zellulare neuronale Netze können zur Bildvorverarbeitung eingesetzt werden [5, 6], wobei sich allerdings bei letzteren die Auswahl auf bereits gelernte Funktionen beschränkt. Dadurch ist die Schnittstelle zu einer geplanten Implementierung einer Lernkomponente gewährleistet.

2.2 Programmierumgebung

Neben der interaktiven Nutzung des Baukastens zur Analyse des Bildmaterials ist sicherlich die eigens implementierte Programmiersprache CIM^{2}BA/Prg zu nennen, die alle vom Baukasten zur Verfügung gestellten Methoden anbietet. Diese Programmiersprache erlaubt das in der Experimentierphase gefundene „Rezept“ zu hinterlegen und in Form einer Stapelverarbeitung automatisiert für eine Fülle von Bilddaten anzuwenden, die einer gleichen Aufgabenstellung unterliegen. Dabei können sowohl die Standardmethoden auf einen Bilddatensatz angewendet werden, als auch die unscharfen Beschreibungen der Wissensbasis und die künstlichen neuronalen Netze. Aber auch die Programmierumgebung kann natürlich experimentell eingesetzt werden, durch das Speichern von Zwischenergebnissen können zudem bereits erzielte Ergebnisse verwendet werden.



(a) Fuzzy-Editor

(b) KNN-Editor

Abb. 2. Masken zur Erstellung und Verwaltung der Fuzzy-Mengen und künstlichen neuronalen Netze.

Die Sprache unterstützt neben verschiedenen Standarddatentypen (Zahlen, Zeichenketten) die Datentypen `Image`, `Segment` und `SegmentList`. Insbesondere sind die Verwendung von Konzepten, Fuzzy-Mengen und neuronalen Netzen, auf die in der Programmiersprache direkt zugegriffen werden kann, zu nennen. Durch Schleifen und bedingte Ausführungen von Programmteilen kann der Analyseprozeß ergebnisabhängig gesteuert werden. Dabei kann auch auf die in der Regelbasis hinterlegten Regeln zugegriffen werden. Ein Konzept beschreibt in der Regel Eigenschaften eines gesuchten Objektes und Relationen, die dieses Objekt zu anderen Objekten im Bild hat. Diese Eigenschaften sind unscharf beschrieben, etwa `Brightness=medium`. Durch Zugriffsoperationen auf Konzeptinhalte können so die Werte abgefragt und zur Auswertung verwendet werden, etwa liefert `GetValue(Brainstem,brightness)` den Helligkeitswert eines Konzeptes *Brainstem* zurück. Der Zugriff auf neuronale Netze erfolgt derart, daß die Eingabeneuronen mit Werten belegt werden können (`SetVal(Net,Neuron,Val)`). Anschließend kann das Netz mit `ApplyNet(Net)` berechnet und mit `GetVal(Net,Neuron)` der Ausgabewert abgefragt werden.

3 Abschlußbemerkungen

Das Baukastenprinzip wurde schon mehrfach in verschiedenen Anwendungen vorgestellt, weshalb hier nur auf die entsprechende Literatur verwiesen werden soll. In [7] wird eine Anwendung zur Analyse von MRT-Bilddaten auf pathologische Veränderungen beschrieben, [8] beschreibt die Anwendung zur Erkennung von Koronararterienverkalkung, die Verwendung von neuronalen Netzen zur Vorsegmentierung wird in [5] beschrieben.

Der hier beschriebene Baukasten ist derzeit prototypenhaft in C++ implementiert (die genannten Komponenten sind weitestgend verfügbar). Er stellt sowohl eine Experimentierumgebung dar, welche die Auswertung von Bilddaten mit neuartigen Aufgabenstellungen erleichtert, aber auch mittels der Programmiersprache eine spätere Routinenutzung erlaubt, wobei jeweils auf Wissensbasen und neuronale Netze zurückgegriffen werden kann. Die Implementierung weist derzeit noch den Nachteil auf, daß die Sprache interpretiert wird und dadurch einen Geschwindigkeitsnachteil gegenüber kompilierter Software aufweist. Neben einer Erweiterung des Befehlsumfanges der Programmiersprache wird somit auch über die Entwicklung eines Compilers für diese Sprache nachgedacht. Detailiertere Informationen sind unter [9] zu finden.

Literatur

1. J. C. Bezdek: *What is Computational Intelligence?* In: J. M. Zurada, R. J. Marks und C. J. Robinson (Herausgeber): *Computational Intelligence – Imitating Life*, Seiten 1–12. IEEE Press, 1994.
2. J. Hiltner: *Operatoren zur deskriptiven und modellbasierten unscharfen Wissensbeschreibung in der medizinischen Bildverarbeitung.* In: T. Lehmann, V. Metzler, K. Spitzer und T. Tolxdorff (Herausgeber): *Bildverarbeitung für die Medizin 1998*, Seiten 114–118, Berlin – Heidelberg, März 1998. Springer-Verlag.
3. T. Stüttgen, J. Hiltner, M. Fathi und B. Reusch: *Fuzzy Snakes: A New Approach of Initializing and Optimization Active Contour Models.* In: M. Jamshidi, P. Borne, A. Maciejewski, S. Nahavandi, R. Lumia, M. Fathi und T. Furuhashi (Herausgeber): *World Automation Congress - 3rd International Symposium on Soft Computing for Industry*, Seiten ISSCI118 – 1–8, Albuquerque, USA, Juni 2000. TSI Press.
4. J. Rittscher, J. Hiltner und C. Moraga: *Künstliche Neuronale Netzwerke zur Vorhersage der Hirnkontur.* In: H. Evers, G. Glombitza, T. Lehmann und H.-P. Meinzer (Herausgeber): *Bildverarbeitung für die Medizin 1999*, Seiten 302–306, Berlin – Heidelberg, März 1999. Springer Verlag.
5. J. Hiltner, I. Aizenberg, E. Meyer zu Bexten und C. Moraga: *Neural Networks and Fuzzy Logic in Medical Image Processing.* In: T. Yamakawa und G. Matsumuto (Herausgeber): *Methodologies for the Conception, Design and Application of Soft Computing*, Seiten 325–328, Iizuka, Japan, Oktober 1998. World Scientific Publishing Co. Pte. Ltd.
6. I. N. Aizenberg: *Neural Networks Based on Multi-valued and Universal Binary Neurons: Theory, Application to Image Processing and Recognition.* In: B. Reusch (Herausgeber): *Computational Intelligence – Theory and Applications*, Band 1625 der Reihe *Lecture Notes in Computer Science*, Seiten 306–316, Berlin – Heidelberg, 1999. Springer-Verlag.
7. B. Reusch, M. Fathi und J. Hiltner (Herausgeber): *PG Anomalia – Entwicklung eines wissensgesteuerten Bildverarbeitungssystems zur Erkennung von pathologischen Strukturen in radiologischen Bilddaten.* Projektgruppenendbericht. Universität Dortmund, Dortmund, 1998.
8. J. Hiltner, M. Wawro, M. Fathi, B. Reusch, P. Kriener, J. Holstein und D. Grönemeyer: *Automatische Detektion und Auswertung von Verkalkungen der Koronararterien in EBCT- und Spiral-CT-Bilddaten.* In: A. Horsch und T. Lehmann (Herausgeber): *Bildverarbeitung für die Medizin 2000*, Seiten 341–345, Berlin – Heidelberg, März 2000. Springer Verlag.
9. J. Hiltner: *CIM^{2}BA-Homepage.* `http://ls1-www.cs.uni-dortmund.de/~hiltner/CIMMBA/index.html`.