Anotador de anomalías para imágenes médicas

Anomaly anotator for medical images

Alexander Falcón Ruiz[1*], Alberto Taboada Crispí[2],

Denis Hernández Pacheco[3], Maykel Orozco Monteagudo[4]

[1, 2, 3, 4]*Centro de Estudios de Electrónica y Tecnologías de la Información, Carretera a Camajuaní, km 5 ½, Santa Clara, Villa Clara, Cuba.*

afalcon@uclv.edu.cu

Resumen

Para comprobar la efectividad de los nuevos algoritmos se necesita de un estándar de oro o *ground truth*. En el caso de los algoritmos para detectar anomalías en imágenes médicas, debe emplearse un conjunto de imágenes anotadas por un personal médico especializado. La mayoría de las aplicaciones de software que permiten realizar estas anotaciones, vienen como valor agregado en los equipos médicos y acarrean problemas de licencia. Además, estas y otras herramientas disponibles carecen de la flexibilidad y funcionalidades requeridas en ciertos casos. En este trabajo se describe la implementación de un software para anotar anomalías en imágenes médicas en varios formatos, entre los que se incluye DICOM. Se puede cargar imágenes y realizar operaciones como hacer y deshacer, dibujar líneas, rectángulos, elipses, dibujar a mano alzada, rellenar, borrar, magnificar, usar varios colores y niveles de transparencia, entre otras.

Palabras clave: Anotación de anomalías, estándar de oro, software médico.

Abstract

In order to assess new algorithms, a ground truth is needed. Particularly, for algorithms that detect anomalies in medical images, a set of annotated images by medical specialists is necessary. Most software applications used in this kind of tasks comes as value added in medical equipment and carries license problems. These and others available tools do not have the flexibility and functionality required in some cases. In this paper the implementation of a software application for anomaly annotation in medical images is described. This program allows loading images, making annotations, undoing and repeating, drawing (lines, rectangles, ellipses, and free hand), filling, erasing, magnifying, and using different colors and transparency levels.

***Key words**: Anomaly annotation, ground truth, medical software.*

Introducción

En términos generales, las anomalías pueden ser consideradas como una pequeña fracción de un conjunto de datos, la cual difiere en algún sentido de la tendencia global o el patrón definido en dicho conjunto. Los conceptos de la detección de anomalías (DA) han sido aplicados a numerosos estudios de imaginología médica para buscar diferentes anomalías, asociadas a varias partes del cuerpo humano empleando varias modalidades de adquisición de imágenes (ver Fig. 1). La mayor parte de los estudios reportados han tratado con la detección de tumores de mamografía digital (De Santo, 2003; Astley, 2004; Huang, 2004; Selvi, 2005; Wei, 2005; Peng, 2006; Chiracharit, 2007; Ikedo, 2007; Karnan, 2007), imágenes de tomografía computarizada de pulmón (Minhas, 2005; Sluimer, 2006), e imágenes de resonancia magnética (Chen, 1999; Gering, 2003; Prastawa, 2004; Lee, 2005; Benamrane, 2006; Menze, 2006; Shinkareva, 2006; Bouix, 2007; Ekin, 2007), aunque se pueden mencionar muchas otras.

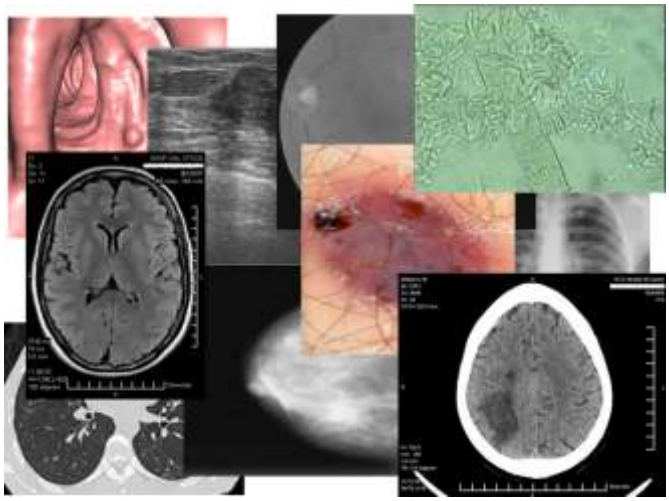

Fig. 1. Imágenes médicas empleadas en la detección de anomalías.

Para evaluar el desempeño de los algoritmos de DA se debe disponer de un estándar de oro adecuado. El método más común en la evaluación de los resultados es comparar estos con los de un grupo de expertos (anotados o segmentados a mano) para un conjunto de imágenes (Chen, 1999). Los expertos humanos tienen excelentes capacidades de reconocimiento debido a su conocimiento global previo; sin embargo, su habilidad de delineación es pobre. Los coeficientes de variación intra-expertos han sido reportados alrededor del 6,5%, y los coeficientes de variación inter-expertos cerca del 22,1% de acuerdo a (Ashton, 2006).

A diferencia de los cientos de bases de datos de imágenes médicas disponibles para otras tareas de análisis de datos, son muy poco comunes las bases de datos anotadas para la prueba de algoritmos de DA. Esta carencia de estándares de oro, forzaron a Bouix et al. (2007) a enfocar su atención en un principio de acuerdo común para evaluar algoritmos para la clasificación de tejidos blandos de cerebro sin un estándar de oro, sin embargo no fue suficiente para una evaluación precisa del desempeño de dichos algoritmos.

Actualmente existen un conjunto de aplicaciones de software (Philips, 2008; AccuImage, 2008) que permiten realizar anotaciones sobre las imágenes médicas. Sin embargo, estas vienen como valor agregado en los equipos médicos y acarrean problemas de licencia. Peor aún, en muchos casos estas herramientas carecen de la flexibilidad y funcionalidades requeridas para hacer anotaciones adecuadas en la creación de un estándar de oro para la evaluación de algoritmos de DA. En este trabajo se describe la implementación de un software para anotar anomalías en imágenes médicas en varios formatos, entre los que se incluye DICOM (*Digital Imaging and Communication in Medicine*). El software permite cargar imágenes y realizar operaciones como deshacer y repetir, dibujar líneas, rectángulos, elipses, dibujar a mano alzada, rellenar, borrar, magnificar, usar varios colores y niveles de transparencia, entre otras. El software ha sido íntegramente programado en C# 2.0, cumple totalmente con especificaciones abiertas y es capaz de ejecutarse en varios sistemas operativos y plataformas de hardware.

Desarrollo

Materiales y métodos

Análisis para la selección de herramientas de desarrollo.

En la programación del software se han seleccionado herramientas que garantizan la portabilidad a varios sistemas operativos y plataformas de hardware, de modo que el programa pueda extenderse en su uso. Esto puede realizarse básicamente de dos maneras: (1) escribir versiones para cada variante, (2) escribir una versión y ejecutar en cualquier parte. Con el primer enfoque podría obtenerse el mejor rendimiento al compilar directamente al código nativo de cada plataforma en específico. Con la segunda variante tendríamos que sacrificar un poco de rendimiento, pero a cambio se obtiene un ahorro en el tiempo de desarrollo. Como este trabajo forma parte de una investigación para desarrollar algoritmos de DA, se ha escogido la segunda variante con el objetivo de disminuir el tiempo de desarrollo.

Una vez escogida la segunda variante es necesario emplear alguna de las tecnologías de entornos virtuales de ejecución existentes, entre los que se encuentran la plataforma Java de Sun, el *framework* .NET de Microsoft y el proyecto Mono (Sun, 2008; Microsoft, 2008; Mono, 2008). Optamos por .NET, puesto que los autores poseen experiencia en el manejo de herramientas y en el desarrollo de aplicaciones para este entorno. A pesar de solo existir implementaciones del *framework* .NET para el sistema operativo Windows, las aplicaciones desarrolladas también pueden correr sobre Mono. Para que esto último sea posible, en la programación deben usarse solo los tipos de datos que aparecen en la CTS (*Common Type System*, Sistema de Tipos Comunes), emplear únicamente código gestionado y respetar la compatibilidad con Mono. En la tabla 1 puede observarse las arquitecturas de hardware y sistemas operativos soportados por Mono.

Tabla 1. Arquitecturas soportadas por Mono (Mono, 2008).

Arquitecturas Soportadas	*Sistema Operativo*
s390, s390x (32 y 64 bits)	Linux
SPARC (32)	Solaris, Linux
PowerPC	Linux, Mac OSX
x86	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows, Solaris, OS X
x86-64: AMD64 and EM64T (64 bits)	Linux, Solaris
IA64 Itanium2 (64 bits)	Linux
ARM: little y big endian	Linux
Alpha	Linux
MIPS	Linux
HPPA	Linux

Existen varios ambientes de desarrollo integrados (*Integrated Development Enviroment* (IDE)) para el desarrollo rápido de aplicaciones que compilan código para el *framework* .NET. Por otra parte, el lenguaje C# ha sido especialmente construido para adaptarse de manera natural a esta plataforma y aprovechar al máximo todas sus características. Entre los IDEs mencionados, se encuentran Microsoft Visual C# 2008 Express Edition y SharpDevelop, ambos compilan C#, son gratuitos y están disponibles para su descarga en la Web. Específicamente en este trabajo se empleó Microsoft Visual C# 2008 Express Edition.

Bibliotecas empleadas.

Para garantizar la portabilidad, se han empleado bibliotecas de clases escritas solo en código gestionado, independiente de cualquier plataforma. Adicionalmente se ha verificado que sean soportadas por Mono, haciendo uso de Moma (Mono, 2008), una herramienta que comprueba si un ensamblado escrito para .NET puede correr en Mono.

En la Web existen numerosos sitios de donde se puede descargar código para .NET. Entre los más conocidos están www.SourceForge.net, www.lawebdelprogramador.com y www.c-sharpcorner.com. Para este trabajo se ha descargado una biblioteca llamada OpenDICOM# de www.SourceForge.net. Esta proporciona una implementación totalmente nueva de DICOM para Mono y el *framework* .NET. El resto del software fue escrito empleando las bibliotecas del *framework* .NET.

Arquitectura del software.

La aplicación posee una arquitectura tradicional de tres capas (presentación, lógica y datos). El siguiente es un breve resumen de los detalles de implementación de cada una de las capas:

Presentación: implementada mediante un proyecto de aplicación de ventanas (Annotator.exe). Se emplean componentes visuales como menú principal (MenuStrip) con todas las opciones disponibles, barra de herramientas (ToolStrip) para un acceso rápido a

las funciones más usadas, barra de estado (StatusStrip) para brindar información y ayudas cortas al usuario, cajas de diálogos para abrir imágenes (OpenFileDialog) y seleccionar colores (ColorDialog), cuadros para mostrar imágenes (PictureBox), etc.

Lógica: implementada mediante un proyecto de librería de clases (Logic.dll) que contiene:

Clase que representa entidad del dominio (AnnImage.cs) donde se encapsula todo el procesamiento de imágenes.

Una clase que soporta las funcionalidades de deshacer/repetir del sistema (ImageClipboard.cs)

Clase que representa el gestor de las entidades del dominio (Annotator.cs).

Clase que representa los errores de la capa lógica (LogicException) y que serán enviadas por el componente hacia la capa superior en caso de que ocurra una excepción.

Datos: implementada mediante un proyecto de librería de clases (Data.dll), que contiene:

Una clase que abstrae a la aplicación del manejo de los datos (ImageFile.cs).

Una clase que representa un error ocurrido en la ejecución de los métodos de la capa de datos, y que será enviada hacia la capa superior (DataException).

Adicionalmente se emplea la biblioteca de clases opendicom-sharp.dll para el manejo de imágenes DICOM.

Herramientas de anotación.

Entre las herramientas de anotación se encuentra el lápiz, que permite dibujar trazos a mano alzada y marcar el contorno de cualquier anomalía. Además aparecen facilidades para el dibujo de líneas en el caso de contornos rectos y rectángulos y elipses para el marcado de áreas con formas semejantes a estas figuras. La herramienta de relleno permite rellenar las áreas marcadas previamente con el lápiz y las demás utilidades, con un color semitransparente que deja ver la imagen médica que se encuentra detrás. El borrador, de tamaño configurable, permite corregir las áreas previamente rellenadas para hacer una anotación más exacta.

En la imagen pueden aparecer anomalías muy pequeñas, en este caso la funcionalidad de acercamiento modifica el tamaño de la imagen a fin de conseguir una mejor anotación. También la imagen puede disminuirse o ponerse a su tamaño original.

En las anotaciones se pueden emplear varios colores, de modo que un color podría corresponderse con un tipo de anomalía en específico. Además se brinda la posibilidad de modificar el nivel de transparencia de las anotaciones.

Algunos detalles de implementación.

DICOM es un estándar abierto y se ha convertido en el formato por excelencia para el intercambio de imágenes médicas. Adicionalmente a este formato, en el software se soportan los formatos BMP, GIF, JPEG, TIFF, PNG y EXIF, implementados en el *framework* .NET. En el fragmento de código siguiente se muestra como leer una imagen independientemente de su formato:

```
AcrNemaFile file = null;
if (DicomFile.IsDicomFile(fileName))
   file = new DicomFile(fileName, false);
else if (AcrNemaFile.IsAcrNemaFile(fileName))
   file = new AcrNemaFile(fileName, false);
if (DicomFile.IsDicomFile(fileName) || AcrNemaFile.IsAcrNemaFile(fileName)) // Archivo DICOM?
{
   imgWords = (ushort[])file.PixelData.ToArray()[0];  // Tomar la imagen del archivo DICOM
   img = new Bitmap(file.PixelData.Columns, file.PixelData.Rows);
   …
}
else
   img = new Bitmap(fileName); // Formatos soportados por el framework
```

Las imágenes en escala de grises, no son soportadas de forma directa por .NET. En este caso se puede realizar una conversión a una imagen RGB poniendo todas las componentes (roja, verde y azul) a un mismo valor obteniendo así una tonalidad de gris. Sin embargo, el contraste obtenido puede ser muy bajo, en especial si la imagen tiene más de 8 bits por cada píxel. Una solución puede ser escalar un rango del valor de los píxeles en la imagen a los 8 bits que se pueden mostrar en la imagen RGB. Esta operación es muy común en la manipulación de imágenes médicas donde se trabaja con los conceptos de ventana (*Windows*) y nivel (*Level*), o sea, a partir de un determinado nivel se abre una ventana de determinado ancho y estos valores son los que luego se escalaran. A continuación se muestra un fragmento de código que realiza las operaciones comentadas:

```
int inf = (int)(level + 1000 - window / 2);       // Level y Window expresados en unidades de Hounsfield
int sup = (int)(level + 1000 + window / 2);
int k = 0;
byte gray = 0;
for (int i = 0; i < img.Height; i++)
   for (int j = 0; j < img.Width; j++)
   {
      if (imgWords[k] < inf)
         gray = 0;
      else if (imgWords[k] > sup)
         gray = 255;
      else
         gray = (byte)(((imgWords[k] - inf) * 255) / (sup - inf));
      k++;
      img.SetPixel(j, i, Color.FromArgb((int)gray, (int)gray, (int)gray));
   }
```

Resultados y discusión

Se ha obtenido un software fácilmente extensible, con una interfaz de ventanas cómoda y amigable que le permite al usuario realizar anotaciones sobre una imagen médica de una manera muy fácil (ver Fig. 2). La aplicación posee un menú principal con todas las opciones disponibles en el sistema. Las funciones más utilizadas se han ubicado en una barra de herramientas para un acceso rápido a las mismas. Las imágenes cargadas por el software son mostradas sobre un panel que permite desplazar su contenido cuando la imagen excede el tamaño de la ventana. Las herramientas de dibujo se utilizan a través del ratón. Un ejemplo de lo anterior es el dibujo a mano alzada, donde se escoge esta utilidad en el menú principal o se pincha en el botón de la barra de herramientas que tiene un icono de lápiz. Luego, sobre la imagen mostrada se presiona el botón izquierdo del ratón y se arrastra por el lugar donde se quiere hacer el trazo, para terminar, se libera el botón. La herramienta para el ajuste Windows/Level de una imagen DICOM se emplea escogiendo dicha opción y posicionando el cursor sobre la imagen, presionamos el botón izquierdo del ratón y en la medida que arrastramos el cursor en dirección vertical se modifica el nivel (Level) y en la dirección horizontal se modifica la ventana (Windows). El resto de las herramientas también tiene un uso muy intuitivo, quedando todo explicado en la ayuda del software, aparte de la información mostrada en la barra de estado de la aplicación y los *tips* editados para cada componente visual.

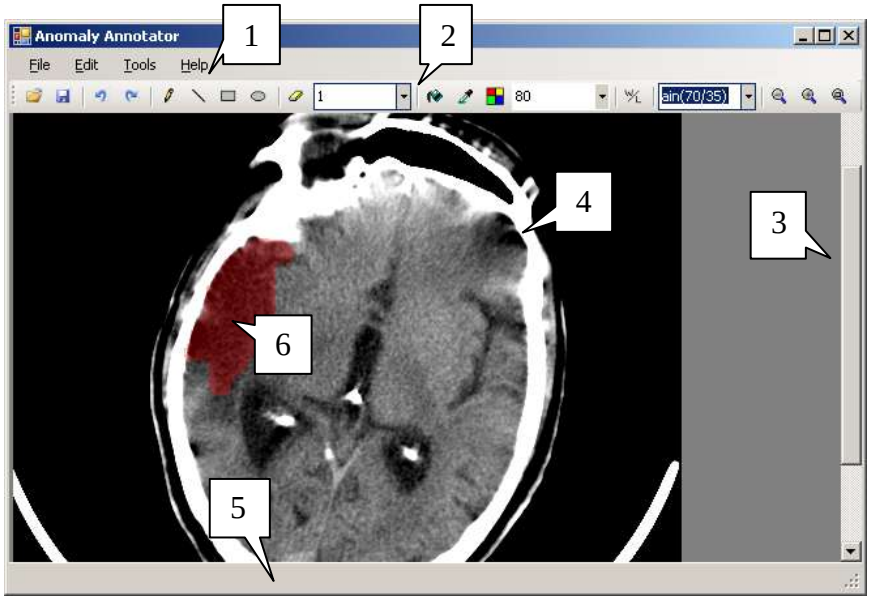

Fig. 2. Ventana principal de la aplicación.

A continuación se brinda una explicación más detallada de las partes señaladas en la Fig. 2:

Menú principal

File: operaciones con archivos

Open: abrir imagen (varios formatos soportados)

Save: salvar anotaciones realizadas

Exit: salir de la aplicación

Edit: operaciones de edición

Undo: deshacer anotación

Repeat: repetir anotación

Clear: borrar todas las anotaciones

Tools: herramientas para la anotación

Pencil: anotación a mano alzada

Line: dibujo de líneas

Rectangle: dibujo de rectángulos

Ellipse: dibujo de elipses

Eraser: borrar en la anotación actual

Fill with color: rellenado de regiones

Pick color: seleccionar un color de las anotaciones realizadas

Select color: seleccionar color de cuadro de dialogo

Windows/Level: ajuste de Windows/Level en imágenes DICOM

Zoom in: aumentar imagen

Zoom out: disminuir imagen

Actual size: tamaño original de imagen

Image Info: muestra información sobre la imagen, cabeceras DICOM.

Help: ayuda de la aplicación

Contents: muestra archivo de ayuda de Windows con ayuda de uso de la aplicación (Anotador.chm)

About: muestra datos acerca de la aplicación

Barra de herramientas con funciones más usadas.

Panel con barras deslizantes para desplazar las imágenes que excedan el tamaño de la ventana.

Cuadro de Imagen: se muestra la imagen médica en el campo *BackgroundImage* y las anotaciones en el campo *Image*.

Barra de estado para mostrar informaciones y ayudas cortas al usuario.

Anotación de color rojo (nivel de transparencia 80)

Debe destacarse que las operaciones de rellenado de regiones grandes y la modificación en tiempo real del Windows/Level de las imágenes DICOM no han tenido el mejor desempeño, aspectos que deben mejorarse en futuras versiones de la aplicación.

Conclusiones

Las bibliotecas de clases disponibles en el *framework* .NET (y en Mono) permiten realizar poderosas operaciones de procesamiento de imágenes, sin embargo presentan algunas limitaciones como por ejemplo el no dar soporte a imágenes en escala de grises y la no disponibilidad de funciones para el rellenado de regiones.

Respetando algunas reglas en la programación, puede llegar a escribirse código realmente portable empleando la plataforma .NET, con todas las ventajas que ello representa.

La aplicación obtenida pudiera usarse para otros fines de los propuestos en los objetivos, como por ejemplo la segmentación de imágenes para diferentes aplicaciones.

Referencias Bibliográficas

AccuImage, 2008. [www.accuimage.com]

Ashton, E., Method and system for automatic identification and quantification of abnormal anatomical structures in medical images, United States Patent 7,103,224, 2006.

Astley, S.M., Computer-based detection and prompting of mammographic abnormalities, The British Journal of Radiology, 2004, 77, 194–S20.

Bouix, S., Martin-Fernandez, M., Ungar, L., Nakamura, M., Koo, M.S., McCarley, R.W., Shenton, M.E., On evaluating brain tissue classifiers without a ground truth. NeuroImage, 2007, 36, 1207–1224.

Chen, M., Kanade, T., Pomerleau, D., & Rowley, H.A., Anomaly Detection through registration, Pattern Recognition, 1999, 32, 113-128.

Benamrane, N., Aribi, A.., & Kraoula, L., Fuzzy Neural Networks and Genetic Algorithms for Medical Images Interpretation, Proceedings of the Geometric Modeling and Imaging— New Trends (GMAI'06), 2006, IEEE Computer Society.

Bouix, S., Martin-Fernandez, M., Ungar, L., Nakamura, M., Koo, M.S., McCarley, R.W., Shenton, M.E., On evaluating brain tissue classifiers without a ground truth. NeuroImage, 2007, 36, 1207–1224.

Chen, M., Kanade, T., Pomerleau, D., & Rowley, H.A., Anomaly Detection through registration, Pattern Recognition, 1999, 32, 113-128.

Chiracharit, W., Sun, Y., Kumhom, P., Chamnongthai, K., Babbs, C.F., & Delp, E.J., Normal mammogram detection based on local probability difference transforms and support vector machines, IEICE Transactions on Information and Systems, 2007, E90-D(1), 258-27.

De Santo, M., Molinara, M., Tortorella, F., & Vento, M., Automatic classification of clustered microcalcifications by a multiple expert system, Pattern Recognition, 2003, 36, 1467 – 1477.

Ekin, A., Jasinschi, R., van der Grond, J., & van Buchem, M., Improving information quality of MR brain images by fully automatic and robust image analysis methods, Journal Of The Society For Information Display, 2007, 15 (6), 367-376.

Gering, D.T., Recognizing deviations from normalcy for brain tumor segmentation, PhD Thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2003.

Huang, Y.L., & Chen. D.R., Watershed Segmentation For Breast Tumor In 2-D Sonography, Ultrasound in Med. & Biol., 2004, 30(5), 625–632.

Ikedo, Y., Fukuoka, D., Hara, T., Fujita, H., Takada, E., & Endo, T. et al., Development of a fully automatic scheme for detection of masses in whole breast ultrasound images, Medical Physics, 2007, 34(11), 4378-4388.

Karnan, M., & Thangavel, K., Automatic detection of the breast border and nipple position on digital mammograms using genetic algorithm for asymmetry approach to detection of microcalcifications, Computer Methods and Programs in Biomedicine, 2007, 87(1), 12-2.

Lee, C.-H., Schmidt, M., Murtha, A., Bistritz, A., Sander, J., & Greiner, R., Segmenting Brain Tumors with Conditional Random Fields and Support Vector Machines, Lecture Notes in Computer Science, 2005, 3765, 469-478.

Menze, B.H., Lichy, M.P., Bachert, P., Kelm, B.M., Schlemmer, H.-P., & Hamprecht, F.A., Optimal Classification of Long Echo Time In Vivo Magnetic Resonance Spectra in the Detection of Recurrent Brain Tumors, Multidimensional Image Processing, IWR, University of Heidelberg, 2006.

Microsoft, 2008. [www.microsoft.com]

Minhas, A.S., & Reddy, M.R., Neural network based approach for anomaly detection in the lungs region by electrical impedance tomography, Physiol. Meas., 2005, 26, 489–502

Mono, 2008, [www.mono-project.com]

Philips, 2008. [www.medical.philips.com]

Prastawa, M., Bullit, E., Ho, S., & Gerig, G., A brain tumor segmentation framework based on outlier detection, Medical Image Analysis, 2004, 26(8), 275–283.

Selvi, S.T., Rejani, Y.I., Sowmy, A., Breast Cancer Detection Using MMRF, Proceedings of the International Conference on Information and Automation, December 15-18, Colombo, Sri Lanka, 2005, 127-132.

Shinkareva, S.V., Ombao, H.C., Sutton, B.P., Mohanty, A., & Miller, G.A., Classification of functional brain images with a spatio-temporal dissimilarity map, NeuroImage, 2006, 33, 63–71.

Sun Microsystems, 2008 [www.sun.com]

Sluimer, I., Schilham, A., Prokop, M., & Ginneken, B., Computer Analysis of Computed Tomography Scans of the Lung: A Survey, IEEE Trans. on Medical Imaging, 2006, 25(4), 385-405.

Wei, L., Yang, Y., Nishikawa, R.M., Wernick, M.N., & Edwards, A., Relevance vector machine for automatic detection of clustered microcalcifications, IEEE Transactions on Medical Imaging, 2005, 24(10), 1278-1285.