

Reconocimiento de Facial basado en FPGA

Face Recognition based on FPGA

Julio C. Molina¹, Miguel A. Risco^{1,2}

¹ Centro de Investigación y Desarrollo en Ingeniería (CIDI), Facultad de Ingeniería Electrónica y Mecatrónica de la Universidad Tecnológica del Perú (UTP)

² RVI Project from the Microprocessor Laboratory of the International Centre of Theoretical Physics (ICTP), Trieste, Italia

RESUMEN

El análisis de las imágenes es un tema en el que se está dando mucho énfasis con la finalidad de identificar parámetros, datos, características visuales en la imagen que proporcione datos del entorno que está representado en la imagen pero sin la intervención de una persona.

Varios métodos han sido desarrollados con el pasar de los años, cada uno buscando minimizar el error. La gran mayoría de estos sistemas de Reconocimiento Facial han sido implementados en una computadora de escritorio y pocos han sido sobre un hardware.

El presente trabajo presenta un sistema de Reconocimiento Facial basada en FPGA, La FPGA es configurada con el lenguaje VHDL, el sistema será independiente de una computadora, esto da al sistema tres principales ventajas, las cuales son la rapidez de procesamiento gracias a la capacidad de la FPGA de poder realizar procesos en paralelo, además de otras dos ventajas que son el peso y el tamaño que son importantes al tener aplicaciones muy limitadas en peso y tamaño.

Descriptor: Reconocimiento Facial, Análisis de Imágenes, FPGA, VHDL

ABSTRACT

The image analysis is a subject on which much emphasis is being given with purpose of identifying parameters, data, visual features in the image that provide us information about environment, it is represented in the image, but without the intervention of a person.

Many methods have been developed with passing of the years, each seeking to minimize the error. The vast majority of implementations of these Facial Recognition Systems have been in a computer and a few have been implemented on hardware.

This paper presents a Face Recognition System based on FPGA, The FPGA is configured with the language VHDL, this system is independent of a computer, this will give the system based on FPGA three main advantages, which include fast processing thanks to the ability of the FPGA that can perform parallel processes, and the other two advantages are the size and weight that are important to have more limited applications in weight and size.

Keywords: Facial Recognition, Images Analysis, FPGA, VHDL.

INTRODUCCIÓN

El reconocimiento de rostros se enfoca en el estudio e implementación de sistemas biométricos, que consiste en el estudio de características físicas únicas en cada persona. Esto se puede apoyar en las técnicas de procesamiento digital de imágenes las cuales puede extraer características biométricas.

La mayor parte de los sistemas de reconocimiento de rostros se realizan en una computadora de escritorio (es lo más común). Este trabajo busca que el Sistema de Reconocimiento de Rostros se realice

en una placa independiente de una computadora, para aplicaciones en donde variables importantes son el espacio y peso.

Con el pasar de los años se han presentado muchos métodos y técnicas para la detección de rostros como por ejemplo en [1], [2] donde el primero presenta una red neuronal vertical para un sistema de detección de rostros, esta red neuronal examina pequeños cuadros de una imagen, y analiza que cuadro contiene un rostro.

El segundo método plantea una segmentación de la imagen en áreas del color de piel, a la que se le

aplican filtros y algoritmos de detección de bordes y así detectar la presencia de ojos.

Uno de los métodos más usados actualmente es el método de Paul Viola y Michael Jones presentado con el nombre de “**Robust Real-time Object Detection**” [4], este método está basado en el método de Adaboost [3], es decir el método de Paul Viola y Michael Jones es una variante del método de Adaboost.

En el método de Adaboost de Paul Viola y Michael Jones se presenta un nuevo concepto llamado la “**Imagen Integral**”, también presenta una extracción de valores llamados “**Características Haar**”, que no son más que valores extraídos al aplicar filtros con base Haar sobre la imagen integral, otro aporte es la combinación de clasificadores en **cascada**.

En el presente documento se presenta la implementación de un detector de rostros, como fase inicial de un sistema de reconocimiento facial basado en FPGA.

Se divide en dos grandes etapas: Entrenamiento del sistema y Optimización del sistema.

MARCO CONCEPTUAL

¿Qué es una FPGA?

Una FPGA (*Field Programmable Gate Array*) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad se puede programar.

Una manera más práctica de entender que es una FPGA es imaginar que esta fuera una Protoboard gigante, pero en lugar de conectar nuestros componentes con cables, nosotros lo hacemos a través del lenguaje VHDL, el cual es un lenguaje en la que no programamos, sino correctamente dicho describimos el hardware, luego este será integrado en un chip que es la FPGA utilizando las herramientas del fabricante de la FPGA.

¿Por qué usar una FPGA?

Las ventajas de usar una FPGA son muy grandes, las principales son:

Nivel de integración: Es decir integrar un sistema digital grande a dimensiones microscópicas obteniendo así sistemas complejos, pequeños, ligeros y altamente funcionales.

Procesos Paralelos: Es decir la capacidad de dividir procesos enteros en bloques paralelos.

Reconfigurable: Poder cambiar el hardware diseñado con unos pocos cambios en el código VHDL o Verilog.

Método de Adaboost (Viola & Jones)

El método de Paul Viola y Michael Jones está basado en el método de Adaboost, y fue presentado con el nombre de “Robust Real-Time Object Detection” [4].

Imagen Integral

La imagen integral está definida como la suma de todos los pixeles en la parte superior izquierda con respecto a un pixel en la posición (x, y), la suma incluye también al pixel de la posición (x, y).

Definición matemática:

Extracción de características

Durante el entrenamiento, las imágenes son convertidas a escala de grises con una resolución de 24x24 pixeles, para luego realizar la extracción de características aplicando filtros con base Haar [4]. En el siguiente cuadro se muestra la cantidad de valores obtenidos con los filtros con base Haar.

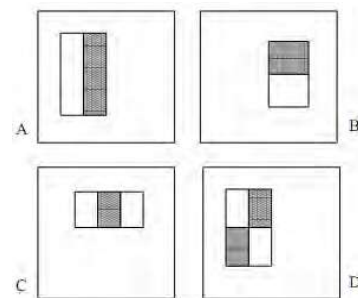


Figura 1: Filtros con base Haar

Tabla 1: Cantidad de datos Obtenidos con filtro Haar

	Tipo de características			
	Tipo A	Tipo B	Tipo C	Tipo D
Imagen 24x24 pixeles	43200 valores	43200 valores	27600 valores	20736 valores

Algoritmo de Adaboost

Consiste en los siguientes pasos:

- ❖ Al ingresar las imágenes, estas deben estar pre-procesadas, para luego calcular la

imagen integral y las características de la imagen.

- ❖ Se inician los pesos.
- ❖ Se escoge el número de Hipótesis T para el inicio del bucle.
 - Se normaliza los pesos
 - para cada característica, aplicamos clasificadores y evaluamos el error.

donde:

C : Clasificador débil.

F : Función que evalúa la imagen, en este caso es un valor característico de la imagen.

T : Umbral del clasificador débil.

- Se escoge el clasificador con el menor error, este error se utilizará para la actualización de los pesos.
- Se calcula α , este valor son los pesos de clasificador.
- ❖ Se obtiene el clasificador fuerte, que está formado por clasificadores débiles con umbrales calculados en paso anterior.

METODOLOGÍA

En este capítulo se describe el procedimiento seguido para realiza el diseño de un detector de rostro, el cual permitirá localizar un rostro dentro una imagen.

El proceso involucra la selección de la herramienta en donde se realiza la simulación, y la selección donde se realiza la implementación.

Análisis y selección del software

El software seleccionado es MATLAB, esta herramienta es seleccionada por su gran flexibilidad y facilidad en la programación, permitiendo la implementación y simulación del código de manera rápida, aquí algunas ventajas:

- Fácil de aprender.
- Instrucciones preparadas para el trabajo con matrices (muy usados en procesamiento de imágenes).
- Realizar gráficos en 2D y 3D con mucha facilidad.

Análisis y selección del hardware

La implementación final del sistema detector de rostros se realizará sobre una FPGA por las siguientes ventajas:

- Son comerciales.

- Nivel de Integración.
- Reconfigurables.
- Procesamiento paralelo.

La principal ventaja de la elección de la FPGA es el procesamiento paralelo, lo cual nos permite dividir el procesamiento total en bloques paralelos, esto no sería posible en las computadoras de escritorio, debido a que el procesamiento es realizado de forma serial.

Detector de Rostros

El Sistema Detector de Rostros es entrenado con una base de datos (rostros y no-rostros) para la obtención de valores característicos, estos serán usados para la detección de un rostro dentro de una imagen e indicar la ubicación por medio de un indicador.

El sistema realiza un escaneo a toda la imagen en diferentes escalas para la detección de rostros de distintos tamaños.

La implementación del Sistema Detector consta de dos etapas que son:

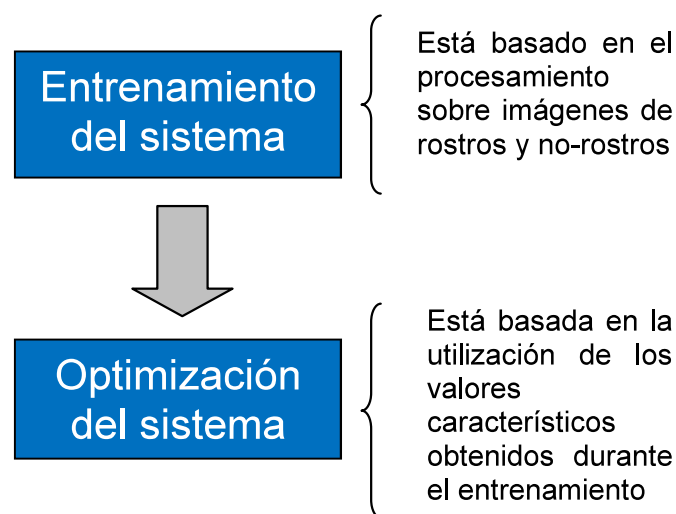


Figura 2: Etapas del Detector de Rostros

Entrenamiento del sistema

El entrenamiento del sistema se realiza mediante el análisis y procesamiento de imágenes, luego se aplica la técnica o algoritmo de aprendizaje de Adaboost.

Las imágenes para el entrenamiento son pre-procesadas cambiando la imagen de color a una imagen a escala de grises y luego realizar el redimensionamiento a 24x24píxeles.

Luego del pre-procesamiento de la imagen, se realiza una transformación de la imagen a una llamada “**Imagen Integral**” resultando una imagen

con la misma cantidad de píxeles, la imagen integral es usada para la extracción de valores característicos (característica Haar)

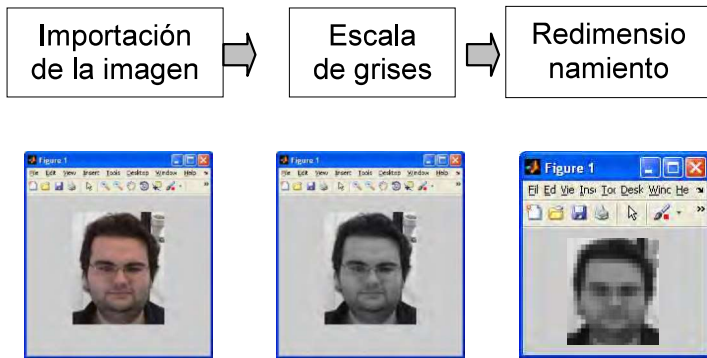


Figura 3: Pre-procesamiento de la imagen

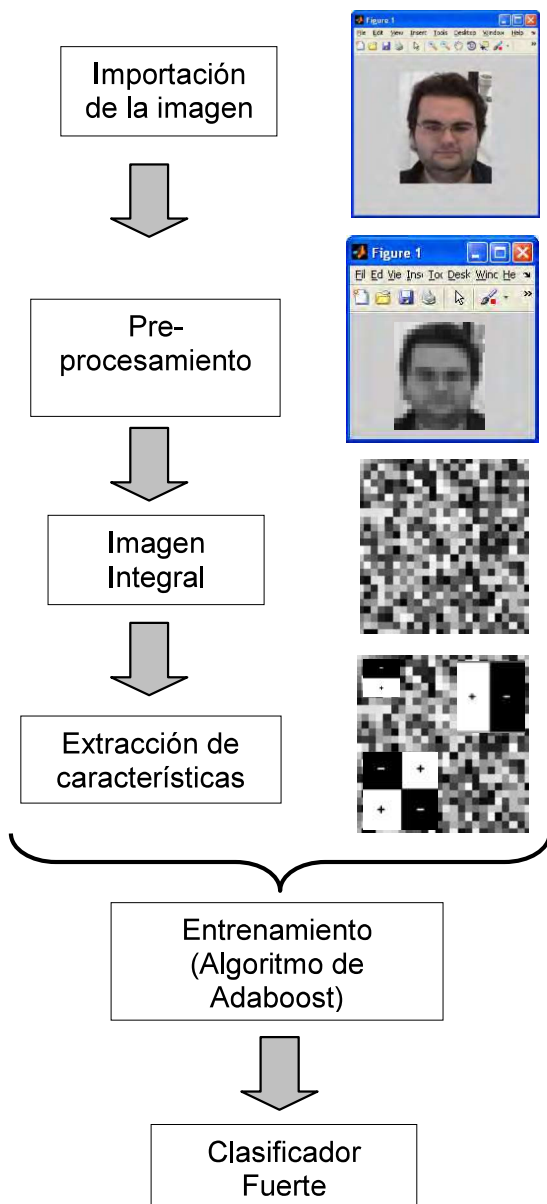


Figura 4: Proceso de entrenamiento

Optimización del sistema

La optimización del sistema está basada en la utilización de los valores característicos obtenidos durante el entrenamiento, a partir de estos valores se obtiene un **umbral** el cual es modificado durante la optimización.

Escenario de prueba:

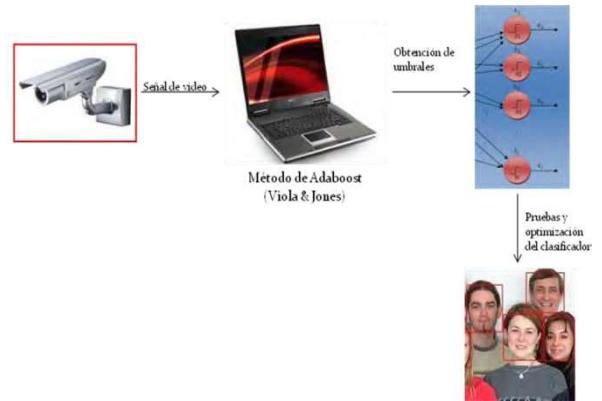


Figura 5: Escenario de simulación

Implementación en la FPGA

La implementación del sistema dentro de la FPGA cuenta con varios bloques. En el esquema de pruebas presentado se realiza las pruebas del sistema.

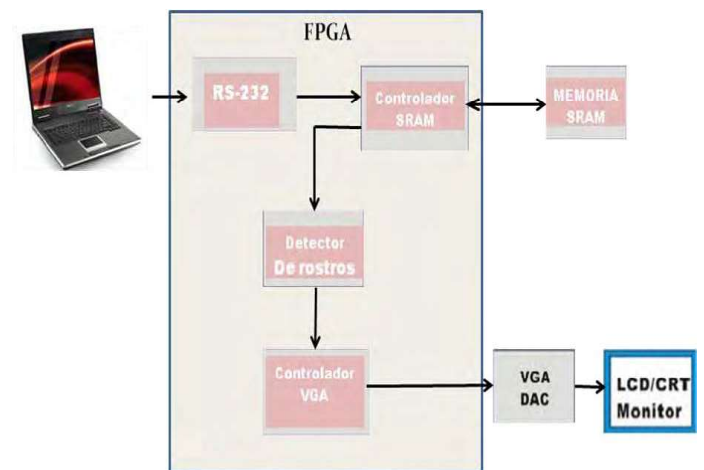


Figura 6: Bloque del Sistema en la FPGA

Se envía una imagen por el puerto serial conectado al **controlador RS-232** que a la vez está conectado a un **controlador SRAM**, este controlador está encargado de las operaciones de escritura y lectura de datos (píxeles) guardados en la memoria **SRAM** externa al FPGA.

Una vez guardada la imagen en la memoria SRAM, se procede a procesar los píxeles y detectar los rostros, para luego enviar los píxeles al **Controlador**

VGA, el cual enviará los píxeles al **VGA DAC** y mostrará la imagen ya procesada en la **pantalla LCD**.

RESULTADOS Y DISCUSIÓN

Base de datos

Gran parte de la base de datos (rostros) pertenece a la Universidad Técnica de Munich (Alemania), esta base de datos fue generada como parte del proyecto FG-Net (Face and Gesture Recognition Research Network) [5].

La otra parte de rostros y el total de no-rostros fueron conseguidos de internet.

Entrenamiento

El entrenamiento se realizó con 572 imágenes de rostros y con 760 imágenes no-rostros (paisajes, partes del cuerpo, objetos, etc.), sumando una total de 1332 imágenes (cada una de 24x24 píxeles).

Haciendo un total de $1332 \times 134736 = 179468352$ datos que son procesados durante el entrenamiento. El tiempo de entrenamiento fue de 5 días aproximadamente.

Se configuró el software para el cálculo de 100 parámetros (iteraciones), se obtuvo un error del 2% aproximadamente (con datos de entrenamiento).

El incremento de parámetros tiene una relación directa con el tiempo de detección de rostros ya que el programa debe analizar más parámetros sobre una imagen.

El entrenamiento se realizó para una estructura como en [3] adicionando la imagen integral y la extracción de características con base Haar presentadas en [4].

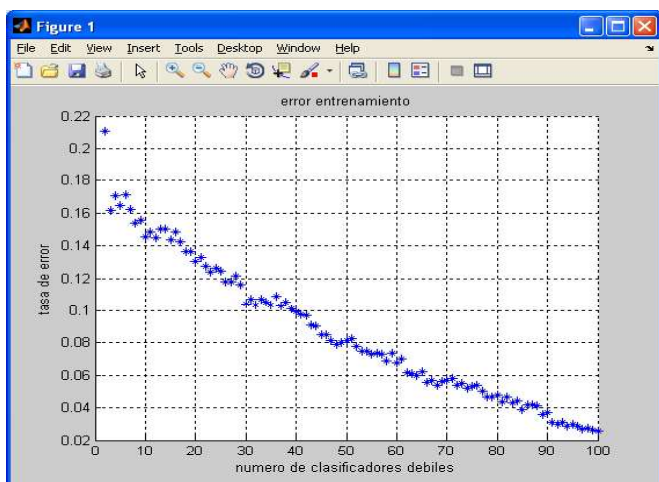


Figura 7: Tasa de Error vs Número de iteraciones

Optimización

Luego del entrenamiento del algoritmo, el sistema es optimizado con modificaciones del umbral [4] que determinan a un “rostro” y un “no-rostro”.

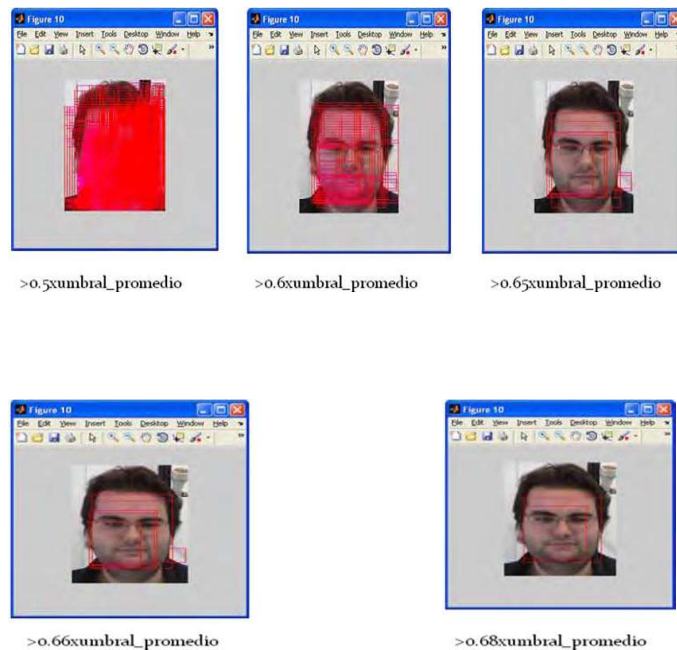


Figura 8: Optimización del sistema con una imagen de la base de datos de entrenamiento

CONCLUSIONES

- Se concluye que al incrementar el número de iteraciones el error disminuye, para 100 iteraciones, el error de clasificación es 2% y tiende a cero conforme se incrementa el número de iteraciones (figura 7).
- Se concluye que la eficiencia del Detector de Rostro depende de la cantidad de datos en la etapa de entrenamiento
- La optimización es realizado en un inicio con imágenes que conforman la base de datos y posteriormente con imágenes que no conforman la base de datos, además de varios test de prueba y error, incrementando la eficiencia (figura 8).

AGRADECIMIENTOS

Quiero agradecer a todas las personas que hicieron posible la realización de este trabajo y en forma particular al Centro de Investigación y Desarrollo en Ingeniería (CIDI) de la Facultad de Ingeniería Electrónica de la Universidad Tecnológica del Perú

por haber facilitado el local para la realización experimental de este trabajo y material de desarrollo. También al grupo de trabajo del proyecto FG-NET (Face and Gesture Recognition Research Network) de la Universidad Técnica de Munich (Alemania) por compartir su base de datos de imágenes.

REFERENCIAS

- [1] HENRY A. Rowley, SHUMEET Baluja y TAKEO Kanade. **Neuronal network-Based Face Detection**. 1998.
- [2] PEER Peter, SOLINA Franc. Automatic Detection of Human Faces in Images. (Univ. Ljubljana, Fac de Computación y Ciencias de la Información).
- [3] YOAV Freund, ROBERT E. Schapire. A

- Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. 1977.
- [4] PAUL Viola, MICHAEL Jones. Robust Real-Time Object Detection. 2001.
- [5] MARTA Lucia Guevara, JULIAN David Echeverry, WILLIAM Ardilla Urueña. Detección de rostros en Imágenes Digitales usando Clasificadores en Cascada. 2008.

E-mail: molina.saqui@gmail.com