

# Desarrollo e Implementación de un Software de Reconocimiento de Dólares Americanos Dirigido a Personas con Discapacidad Visual Utilizando Teléfonos Móviles Inteligentes con Sistema Operativo Android

Andrés Guaita  
Christian Catacta  
Fabián Sáenz  
Paúl Bernal

Departamento de Eléctrica y Electrónica, Escuela Politécnica del Ejército (ESPE)  
Sangolquí, Ecuador

**Resumen-** El presente artículo trata sobre el desarrollo e implementación de un software de Reconocimiento de la Denominación de Dólares Americanos dirigido a personas con discapacidad visual, el cual trabajará en Teléfonos Móviles Inteligentes que posean el Sistema Operativo Android de Google, para la concepción de dicho software se consideró la técnica de Análisis de Componentes Principales – PCA mediante la obtención de *Eigenfaces* y técnicas de Procesamiento Digital de Imágenes – PDI.

propietario, lo cual simplifica e incentiva el desarrollo del software, ya que no hay que pagar licencia alguna para crear aplicaciones para esta plataforma.

El software de reconocimiento de dólares americanos será desarrollado en dos etapas ya que al utilizarse el algoritmo del PCA es necesario tener una Etapa de Aprendizaje y una Etapa de Reconocimiento.

## I. INTRODUCCIÓN

Las personas no videntes según el Concejo Nacional de Discapacidades – CONADIS abarcan el 11% del total de personas con discapacidades en el Ecuador, dichas personas salen adelante a pesar de su discapacidad, sin embargo, hay ciertas actividades que les cuesta mucho realizar, una de ellas es el identificar la denominación de los billetes de dólar, por lo cual una de las maneras en las que se puede ayudar a estas personas es ideando alternativas tecnológicas que solucionen este tipo de problemas.

La idea de desarrollar aplicaciones para teléfonos móviles las cuales ayuden a las personas con discapacidad visual no es nueva, pero sí lo es, el llevar todas esas ideas al campo de los dispositivos móviles inteligentes, esto debido a que la tendencia actual de teléfonos móviles apunta hacia dispositivos más potentes y con sistema operativo.

Para el diseño de un software de reconocimiento de denominación de dólares americanos se debe tomar en cuenta los algoritmos a utilizarse y el dispositivo que lo va a ejecutar, por lo cual se ha considerado el utilizar la técnica de Análisis de Componentes Principales – PCA a través de la utilización de *Eigenfaces*, ya que la misma consume poca capacidad computacional a la vez que es muy eficaz, y dispositivos con sistema operativo Android ya que al ser basado en Linux no es

## II. ANALISIS DE COMPONENTES PRINCIPALES

El Análisis de las Componentes Principales - PCA, del inglés *Principal Component Analysis* es una técnica estadística que realiza una transformación lineal y ortogonal de los datos a un nuevo sistema de coordenadas tal que la máxima varianza queda proyectada sobre la primer coordenada, la segunda mayor varianza sobre la segunda coordenada, y así sucesivamente. Transforma un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas, y simplifica la transformación encontrando las componentes más cercanas a las variables originales pero ordenadas en forma decreciente al orden de su varianza [1].

EL funcionamiento del PCA al utilizarse en identificación de dólares americanos consiste en extraer las características más relevantes.

En términos matemáticos, se quiere encontrar las componentes principales de la distribución de caras, o lo que es lo mismo, los *eigenvectors* de la matriz de covarianza del conjunto de caras

### A. CALCULO DE EIGENFACES

Sea  $I(x,y)$  una imagen facial o de dólares americanos en escala de grises de  $N \times N$ . Dicha imagen puede ser

representada como un vector de dimensión  $N^2$  o equivalentemente, un punto en el espacio  $N^2$  - dimensional, por lo tanto, un conjunto de imágenes es mapeado a una colección de puntos en un espacio inmenso, considerando el tamaño de las imágenes con las que se suele trabajar. De esta manera, un conjunto de imágenes, sean estas faciales o de dólares americanos, dado que todas tienen un aspecto similar, se mapean en un sub espacio del espacio de imágenes y así pueden ser descritas por un sub espacio de menor dimensión. La idea es encontrar los vectores que mejor describan la distribución de caras dentro del espacio completo de imágenes. Estos vectores de longitud  $N^2$ , llamados *eigenfaces*, definen el sub espacio de imágenes, el cual se denomina espacio de caras

Sea  $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M$  un conjunto de imágenes faciales o de dólares americanos. La cara media o meanface  $\Psi$  del conjunto es definida por:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n. \quad (1)$$

Cada *cara* difiere de ésta por el vector  $\Phi_i = \Gamma_i - \Psi$ . A partir de este conjunto de vectores, se busca un conjunto de  $M$  vectores ortonormales,  $u_n$ , los cuales mejor describan los datos. El  $k$ -ésimo vector,  $u_k$ , es seleccionado tal que

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (u_k^T \Phi_n)^2 \quad (2)$$

Es un máximo, sujeto a que los vectores  $u_k$  sean ortonormales. Los vectores  $u_k$  y los escalares  $\lambda_k$  son los *eigenvectors* y *eigenvalues*, respectivamente, de la matriz de covarianza:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T \quad (3)$$

$$C = AA^T,$$

Donde  $A = [\Phi_1 \Phi_2 \Phi_3 \dots \Phi_M]$ . La matriz  $C$  resultante es de  $N^2 \times N^2$  y obtener  $N^2$  *eigenvectors* y *eigenvalues* es una tarea costosa para el tamaño de imágenes comúnmente procesadas.

Una consideración a tener en cuenta es que si el número de puntos en el espacio de imágenes es menor que la dimensión del mismo  $M < N^2$ , habrá solamente  $M - 1$  *eigenvectors* significantes, el resto tendrá asociados *eigenvalues* iguales a cero. En este caso, se tienen  $M$  imágenes caras de dimensión  $N^2$  y se pueden encontrar los *eigenvectors*  $N^2$  - dimensionales resolviendo los *eigenvectors* para una matriz de  $M \times M$  y luego realizar una combinación lineal de las imágenes faciales  $\Phi_1$ .

Consideremos los *eigenvectors*  $v_l$  de  $A^T A$  tal que

$$A^T A v_l = \mu_l v_l. \quad (4)$$

Pre-multiplicando ambos lados por  $A$ , se tiene

$$AA^T A v_l = \mu_l A v_l. \quad (5)$$

Reemplazando la ecuación (4) en (5)

$$C A v_l = \mu_l A v_l. \quad (6)$$

De aquí se ve que  $A v_l$  son los *eigenvectors* de la matriz de covarianza  $C$ . Entonces, se construye la matriz de  $M \times M$ ,  $L = A^T A$ , donde  $L_{mn} = \Phi_m^T \Phi_n$ , y se encuentran los *Meigenvectors*  $v_l$ , de  $L$ . Estos vectores determinan la combinación lineal de las  $M$  imágenes faciales para obtener los *eigenvectors*

$$u_l = A v_l. \quad (7)$$

Es importante normalizar  $u_l$  tal que  $\|u_l\| = 1$  esto se lo puede hacer de la siguiente manera

$$u_l = A v_l D_l^{-1/2} \quad (8)$$

Donde  $D_l$  es una matriz diagonal con los *eigenvalues* de  $L$ .

De esta forma se reducen enormemente los cálculos del orden del número de píxeles en las imágenes ( $N^2$ ) al orden del número de imágenes en el conjunto ( $M$ ). Los *eigenvalues* asociados permiten ordenar los *eigenvectors* de acuerdo al aporte que hacen a la variación entre imágenes

#### B. PROYECCIÓN Y RECONSTRUCCIÓN DE UNA IMAGEN UTILIZANDO EIGENFACES.

Una imagen facial o de Dólares Americanos  $\Gamma$  es proyectada sobre el *espacio de caras* por medio de la operación:

$$w_k = u_k^T (\Gamma - \Psi), \quad \text{para } k = 1, \dots, M. \quad (9)$$

Los pesos  $w_k$  forman un vector  $\Omega^T = [w_1 w_2 w_3 \dots w_M]$  que describen el aporte de cada *eigenface* en representar la imagen cara [39], tratando a las *eigenfaces* como un conjunto base para las imágenes faciales o de dólares americanos.

La reconstrucción exacta de una cara se realiza a través de una combinación lineal de las *eigenfaces* ponderadas con los respectivos pesos  $w_k$

$$\Gamma = \sum_{k=1}^M w_k u_k \quad (10)$$

### C. RECONOCER UNA CARA

Un método para determinar si una determinada cara se encuentra en la base de entrenamiento, consiste en encontrar la clase de cara  $k$  que minimice la *distancia Euclideana*.

$$\varepsilon_k = |(\Omega - \Omega_k)| \quad (11)$$

$$\varepsilon_k < \text{umbral}$$

En donde  $\Omega_k$  es un vector que describe la  $k$  – ésima clase decara. Si  $\varepsilon_k$  es menor que un “umbral predefinido”, entonces, la cara se clasifica como perteneciente a la clase  $k$ . Lo cual implica, haber identificado la cara en la base de datos correspondiente

## III. PROCESAMIENTO DIGITAL DE IMAGENES

### A. SUBMUESTREO DE IMÁGENES POR MEDIA

Este proceso de submuestreo considera un ventaneado de la imagen original para submuestrearla, el número de ventanas dependerá del número de pixeles de la imagen que se desea obtener con el submuestreo ya que cada ventana se utilizará para obtener el valor de cada pixel de la imagen procesada, en la Figura 1 se muestra la forma en la que cada ventana de la imagen original aporta con un valor de pixel de la imagen resultante [2].

120	126	120	121	126	132	142	123
130	132	132	134	142	145	142	152
150	157	148	142	132	135	152	142
153	164	165	176	142	123	132	132
153	145	145	122	153	154	152	153
156	145	145	154	153	164	152	146
132	165	123	132	123	164	132	164
132	145	134	142	145	142	145	154

Ventanas Imagen Original

➔

120	131
144	158

Resultado  
Sub-muestreo

Figura 1 Ventaneado de Imagen

El submuestreo por media utiliza la media de los valores de la ventana para obtener el valor del pixel de la imagen que se desea obtener.

La Figura 2 muestra el resultado de este tipo de submuestreo.



Figura 2 Submuestreo por media

### B. FILTRO MEDIA PONDERADA

Los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa

### C. FILTRO MENOS LAPLACIANO

Es un filtro que realiza rasgos lineales en las imágenes, restando la imagen original de la imagen obtenida con el Laplaciano utilizando la máscara de la Figura 3.

0	-1	0
-1	5	-1
0	-1	0

Div=9

Figura 3 Máscara filtro menos Laplaciano

### D. EXPANSIÓN DEL HISTOGRAMA

Ésta operación opuesta a la contracción del histograma. Su función es:

$$g(i, j) = \left[ \frac{f(i, j) - f(i, j)_{MIN}}{f(i, j)_{MAX} - f(i, j)_{MIN}} \right] [MAX - MIN] + MIN, \quad (12)$$

Donde  $f(i, j)_{MAX}$  y  $f(i, j)_{MIN}$  son el máximo y mínimo nivel de gris respectivamente de la imagen de entrada, MAX y MIN son el máximo y mínimo nivel de gris respectivamente que se desea que tenga la imagen de salida.

Esta ecuación toma una imagen de entrada  $f(x, y)$  y expande el histograma a lo largo del rango de valores completo de los niveles de gris. Esto tiene el efecto de incrementar el contraste de una imagen de bajo contraste. Si se desea que la expansión no cubra el rango total posible de niveles de gris, se pueden especificar diferentes valores para MAX y MIN.

Si muchos valores de una imagen caen dentro de un pequeño rango y existe un cierto número de valores extremos, entonces el histograma abarca todo el rango de valores y una expansión pura del histograma no mejora la imagen como se mira en la Figura 4. En este caso suele ser una práctica habitual recortar los niveles de gris en los extremos a los valores de gris más bajo y más alto del rango (para una imagen de 8 bits serían 0 y 255).

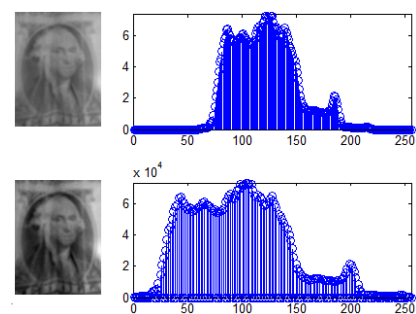


Figura 4 Expansión del Histograma

#### IV. DESARROLLO DE LA ETAPA DE APRENDIZAJE

Para el desarrollo de esta etapa se utilizará Matlab que un software desarrollado por *MathWorks™* especializado en el tratamiento de matrices, por ende es ideal para el procesamiento digital de imágenes.

La función de esta herramienta dentro del presente proyecto radica en la generación de los componentes principales *eigenvectors*, obtenidos del banco de imágenes de dólares americanos.

##### A. OBTENCIÓN DEL BANCO DE IMÁGENES DE DÓLARES AMERICANOS

Debido a que se utiliza el algoritmo de PCA, se tiene que tomar varias muestras de los billetes para almacenarlas en un Banco de Imágenes de Dólares Americanos y utilizarlas en el reconocimiento. El algoritmo de PCA considerará la imagen central de los billetes que posee el rostro de los presidentes norteamericanos en su parte frontal como se puede ver en la Figura 5.



Figura 5 Zona de interés del Billete.

Para la conformación del Banco de Imágenes se tomaran 24 muestras distribuidas de la siguiente manera vistas en la Tabla 1.

##### B. PROCESAMIENTO DE LAS IMÁGENES DEL BANCO DE IMÁGENES

Una vez obtenidas las muestras se procederá a realizar un proceso de submuestreo por media, ya que trabajar con la resolución original resultaría costoso computacionalmente.

A continuación las imágenes serán pasadas por un filtro de media ponderada el cual suavizará las imágenes evitando que parezca borrosa después se aplicará un filtro menos laplaciano el cual resaltará los detalles, este proceso se muestra en la Figura 6.

Con las imágenes ya mejoradas mediante técnicas de PDI se procederá a extraer los *eigenvectors* mediante Matlab.

##### C. ALMACENAMIENTO DE EIGENVECTORES

El almacenamiento de los *eigenvectors* con Matlab se lo realizará en archivos (.xml), aprovechando la capacidad que nos brinda Matlab para crear y editar archivos de texto, de tal forma que la aplicación de Android, sea capaz de leer los datos como un recurso de la aplicación a través de la clase R.java.

Dichos archivos serán guardados en la carpeta “/res/values/” de la aplicación de Android ya que responden a un tipo de recurso llamado arrays.xml que sirven para definir arreglos de vectores de tipo String o enteros (int).

Tabla 1 Obtención del Banco de Imágenes de dólares Americanos

Cara	No. De Muestras	Ángulos de Inclinación				Detalles
		0°	90°	180°	270°	
Frontal	24	±1,	±1,	±1,	±1,	Cada muestra se tomó con cierto grado de inclinación debido a que las personas no videntes no siempre podrán colocar los billetes en una posición exacta.
		±5,	±5,	±5,	±5,	
		±10	±10	±10	±10	

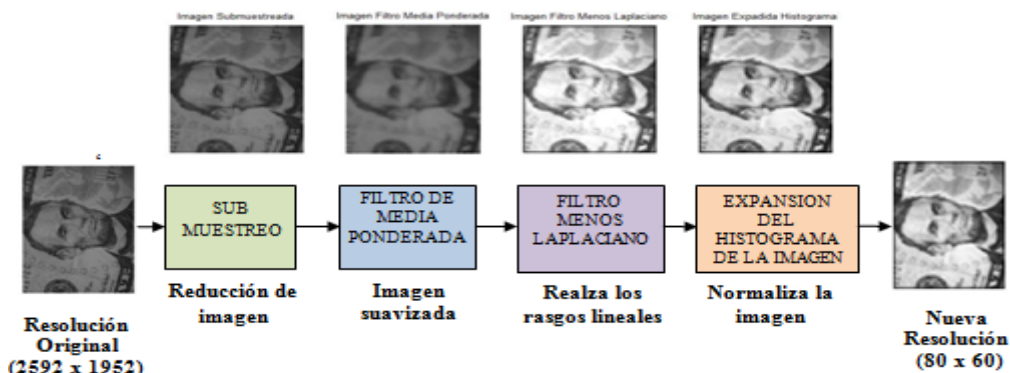


Figura 6 Pre procesamiento de Imagen de muestra

## V. DESARROLLO DE LA ETAPA DE RECONOCIMIENTO

Para el desarrollo de esta etapa se utilizará la herramienta Eclipse IDE y el lenguaje de programación Java el cual es el oficialmente soportado para el desarrollo de aplicaciones en Android.

### A. OBTENCIÓN DE UNA IMAGEN UTILIZANDO LA CÁMARA DE UN DISPOSITIVO MÓVIL INTELIGENTE CON ANDROID SO

Para la obtención de la imagen por medio de la cámara del teléfono móvil se deberá utilizar las siguientes clases *Camera* y *SurfaceView* [3].

Dichas clases proveerán todos los métodos necesarios para capturar imágenes con la cámara del dispositivo.

### B. PRE PROCESAMIENTO DE LA IMAGEN CAPTURADA

Posteriormente de capturar la imagen se procederá a pre-procesarla, debido a que, al tratarse de una imagen obtenida con un dispositivo móvil la misma poseerá una contaminación gracias a las diferentes condiciones de iluminación, así también, al ser uno de los requerimientos la identificación en tiempo real la imagen deberá ser sub-muestreada para reducir el tiempo de procesamiento.

Ya que el algoritmo del PCA solo requiere de una imagen en escala de grises, se procederá a trabajar solo con los datos de *luminancia* - *Y* provenientes de la cámara.

Los procesos por los cuales atravesará la imagen en escala de grises, serán los mismos por los cuales pasaron las imágenes pertenecientes al banco de imágenes de aprendizaje y mostrado en la Figura 6.

### C. PROYECCIÓN DE LA IMAGEN AL ESPACIO DE CARAS

Una vez pre-procesada la imagen se procederá a proyectarla utilizando los *eigenvector* obtenidos a partir de la Etapa de Aprendizaje y almacenados en la carpeta “/res/values/” de la aplicación.

La proyección de la imagen obtenida con la cámara del teléfono móvil se la realizará aplicando la Fórmula 9, con este procedimiento se obtendrá los pesos *w* correspondientes a esta imagen, con lo cual se podrá realizar la comparación e identificación de la misma.

## D. COMPARACIÓN E IDENTIFICACIÓN DE LA IMAGEN CAPTURADA

Para el procedimiento de comparación, se obtendrá la distancia Euclideana con la Fórmula 11 entre los pesos *w* de la imagen obtenida por la cámara y los pesos de las imágenes del banco de muestras.

El umbral descrito en la Fórmula 11 será obtenido a partir de pruebas en varios tipos de iluminación.

## VI. RESULTADOS

El software de reconocimiento se probó en dos dispositivos los cuales varían en la velocidad del procesador y tipo de software:

Dispositivo 1.

Hardware:

- Procesador de 600 MHz.

Software

- Android 2.1 (Eclair).

Dispositivo 2

Hardware:

- Procesador de 540 MHz.

Software

- Android 2.2 (Froyo).

Las pruebas se hicieron variando la resolución de submuestreo de las imágenes capturadas con lo cual se obtuvo los siguientes resultados:

Resolución	Efectividad 540 MHz.	Efectividad 600 MHz.
80x60	48%	99%
160x60	48%	97%
80x120	32%	92%
160x120	8%	75%

Según los datos que revelan la Tabla anterior, el tamaño al cual se sub-muestra la imagen no es relevante en el dispositivo con el procesador de 600MHz, sin embargo, si afecta en la lectura del dispositivo con procesador de 540MHz, a mas de esto se comprueba que la velocidad a la que el dispositivo da el resultado depende también de de la duración de clip de sonido que describe el tipo de billete, ya que a si el dispositivo identifica el billete pero el clip de sonido no termina de reproducirse no podrá reproducir la el resultado de la nueva lectura.

También se pudo constatar que la resolución a la que el sistema tiene una mejor eficiencia es la de 160x60 ya que con

esta resolución se obtiene una mayor cantidad de aciertos en el dispositivo de 540MHz y 600Mhz.

## VII. CONCLUSIONES

El algoritmo de Análisis de Componentes Principales – PCA es muy adecuado para el reconocimiento de patrones, sin embargo, depende mucho de las condiciones de luz y la posición de la imagen a reconocer, por lo cual su eficiencia se ve comprometida al aplicarse sobre un sistema que se traslada como es el caso de un teléfono celular, por lo tanto si se lo desea aplicar en dichos dispositivos se lo debe hacer utilizando filtros y técnicas de Procesamiento Digital de Imágenes que aumenten la calidad de la imagen a reconocer.

Al implementarse algoritmos que involucran Procesamiento Digital de Imágenes dentro de dispositivos móviles se sobrecarga al procesador con cálculos extensos por lo cual la respuesta de dichos algoritmos podría tardar varios segundos.

## VIII. BIOGRAFIAS

**Carlos Andrés Guaita Ayala**, nace el 30 de Octubre de



1987 en Quito, Ecuador. Su educación primaria y secundaria la realiza en la Unidad Educativa Franciscana San Andrés actualmente se encuentra cursando el ultimo nivel de la carrera de Ingeniería Electrónica y Telecomunicaciones en la Escuela Politécnica del Ejército.

**Christian Roberto Catacta LLive**, nace el 08 de Febrero de 1986 en la ciudad de Quito,



Ecuador. Su educación primaria la realizo en la Escuela de Prácticas “Simón Bolívar” y la educación secundaria en el Instituto Tecnológico Superior “Central Técnico”. Actualmente se encuentra en calidad de egresado obteniendo el título de Ingeniería en Electrónica y Telecomunicaciones en la Escuela Politécnica del Ejército.

## IX. REFERENCIAS

- [1] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3:71–86, MIT Press, 1991.
- [2] Procesamiento de imágenes, “Técnicas de realce de imágenes”, [http://arantxa.ii.uam.es/~tacc1/mm\\_05/Slides/2per\\_page\\_pdf/J/07.pdf](http://arantxa.ii.uam.es/~tacc1/mm_05/Slides/2per_page_pdf/J/07.pdf), consultado el 04 de Agosto de 2011.
- [3] Android, Developers, Android 2.1 Platform, <http://developer.android.com/sdk/android-2.1.html>, 2011, consultado el 18 de Abril de 2011.