

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingenierías**

**Un Nuevo Enfoque para el Reconocimiento de Gestos y  
Reconstrucción Facial usando Testores Típicos**

**Roberto Andrés Alvarado Moreira**

**Matemáticas**

Trabajo de fin de carrera presentado como requisito  
para la obtención del título de  
Matemático

Quito, 5 de junio de 2023

# UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

## HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

**Un Nuevo Enfoque para el Reconocimiento de Gestos y Reconstrucción Facial usando  
Testores Típicos**

**Roberto Andrés Alvarado Moreira**

Julio César Ibarra Fiallo, Ph.D. (c)



Antonio Di Teodoro, Ph.D.

.....

Quito, 5 de junio de 2023

**© DERECHOS DE AUTOR**

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Roberto Andrés Alvarado Moreira

Código: 00206411

C.I.: 1104718141

Fecha: Quito, 5 de junio de 2023

## ACLARACIÓN PARA PUBLICACIÓN

**Nota:** El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETheses>

## UNPUBLISHED DOCUMENT

**Note:** The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETheses>

## RESUMEN

Dentro del estudio de los testores en el mundo de procesos de clasificación, la limitante de conjuntos de datos discretos se presenta, haciendo que esta herramienta no tenga la extensibilidad deseada, en este trabajo se busca la generalización de los métodos para diferenciar dos objetos entre sí, con el objetivo que estas nuevas ideas se conviertan en la base para procesos generativos de datos. En este caso se toma como referencia un conjunto de puntos de referencias de rostros para un proceso sencillo de clasificación, se buscará encontrar lo que en este proyecto se conoce como matriz de disparidad, la generalización propuesta, para así conseguir el Testor según un threshold de similitud  $\alpha$ . De forma que se pueda conseguir un subconjunto de atributos que permitan que el conjunto de objetos aún sea discernible y utilizar estos valores para aplicarlos sobre un nuevo conjunto de datos generados por una Point-Cloud-GAN. Finalmente se logró encontrar un subconjunto que reduce el número de atributos a un 61.8 % del conjunto original y se consiguió plantear una arquitectura para una red generativa basada en el método de optimización para puntos, haciendo evidente que el concepto de testor puede ser extendido y a la vez su generalización puede tener aplicaciones en innumerables casos.

**Palabras clave:** Redes Neuronales, Modelos de Clasificación, Testores, Conjuntos Rugosos, Reductos, UMDA, Redes Generativas.

## ABSTRACT

Within the study of testors in the world of classification processes, the limitation of discrete data sets is presented, making this tool does not have the desired extensibility, this work seeks the generalization of the concept of testor and algorithms to find them, with the aim that these new ideas become the basis for generative data processes. In this case we take as reference a set of face reference points for a simple classification process, we will seek to find what in this project is known as  $\alpha$  testors, the proposed generalization. So that a subset of attributes can be obtained that allow the set of objects to still be discernible and use these values to apply them on a new set of data generated by an Point-Cloud-GAN. Finally, it was possible to find a subset that reduces the number of attributes to 61.8 % of the original set and to propose an architecture for a generative network based on a point cloud optimization method, making it evident that the concept of testor can be extended and at the same time its generalization can have applications in countless cases.

**Keywords:** Neural Networks, Classification Models, Testors, Rough Sets, Reduced Sets, UMDA, Generative Networks.

# ÍNDICE GENERAL

<b>1</b>	<b>Introducción</b>	<b>11</b>
<b>2</b>	<b>Definiciones y Métodos</b>	<b>13</b>
2.1	Testores . . . . .	13
2.2	UMDA . . . . .	14
2.3	Point-Cloud-GAN . . . . .	15
2.4	Base de datos: FACES . . . . .	15
2.5	Puntos de referencia . . . . .	15
2.6	Método Propuesto . . . . .	16
<b>3</b>	<b>Proceso y Resultados</b>	<b>18</b>
3.1	Análisis de imágenes . . . . .	18
3.2	Testores . . . . .	20
3.3	Point-Cloud-GAN para puntos de referencia y modelado 3D . . . . .	25
<b>4</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>27</b>
	<b>Bibliografía</b>	<b>27</b>

## ÍNDICE DE FIGURAS

3.1	Ejemplo de imágenes de la base de datos. Cara neutral (Izquierda), Cara sonriente (Derecha) . . . . .	18
3.2	Resultado del algoritmo MTCNN . . . . .	18
3.3	Rotación de imagen . . . . .	19
3.4	Imagen con 68 puntos de referencia del rostro . . . . .	19
3.5	Matriz de confusión para cromosoma seleccionado . . . . .	22
3.6	Resultados del algoritmo según la generación . . . . .	23
3.7	Generaciones según el número de iteración . . . . .	24
3.8	Rostros de la base de datos con el cromosoma seleccionado . . . . .	24
3.9	Generaciones según el número de iteración . . . . .	25
3.10	Clasificación de resultados de las GANs . . . . .	26

## ÍNDICE DE CUADROS

3.1	Arquitectura modelo de clasificación . . . . .	20
3.2	Elección de $\alpha$ a nivel experimental . . . . .	21
3.3	Resultados de proceso generativo . . . . .	23



## DEDICATORIA

Para mis padres, para mis hermanas, para mi mismo y para quién le sea útil.

## AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mi tutor de tesis, Julio Ibarra, por su invaluable orientación, apoyo incondicional y paciencia durante todo el proceso de investigación. Gracias a su experiencia y conocimientos, que han logrado formarme de la mejor manera posible. A David Hervas, mi primer profesor de la universidad, que con su manera tan especial de ver las matemáticas logró contagiarme en mí el amor por ellas.

Además quiero agradecer todas las personas que han sido parte de mi vida, mis padres, mis hermanas, mis abuelos, mi familia, mis amigos. Cada uno de ellos es y será parte de mi día a día como acompañantes lejanos de cada uno de mis pasos, estando siempre en mi memoria sus historias, sonrisas e ideas que han logrado convertirme en lo que soy.

# CAPÍTULO 1

## INTRODUCCIÓN

Las redes neuronales artificiales, se han mostrado como el nuevo paradigma de investigación dentro del aprendizaje de máquina, desde las ideas clásicas de las redes neuronales recursivas (Hochreiter and Schmidhuber, 1997) hasta las nuevas ideas de las redes neuronales generativas (Goodfellow et al., 2014), cada día los avances demuestran la necesidad de nuevas ideas y de nuevas formas de mejorarlas.

En el actual trabajo se presentará un análisis sobre el concepto de testores (Alba and Santana, 2010) una nueva forma de utilizar los sistemas de información para mejorar la implementación de procesos de clasificación. El concepto de los mismos, de manera más específica, se presenta como un método de conseguir un subconjunto de los atributos originales de forma que el nuevo subconjunto de objetos, definido solamente por estos atributos aún mantenga la capacidad de discernibilidad en un proceso de clasificación (Lazo-Cortes et al., 2001).

La propuesta busca una generalización de diferenciar entre objetos para poder utilizar los testores (Torres-Constante et al., 2022), previamente definido simplemente sobre datos discretos, por medio de una función de igualdad, de forma que se logre conseguir una extensión para conjuntos de datos continuos, donde el uso de métricas nos permiten hacer comparaciones entre objetos donde el paradigma de igualdad no es necesario. Logrando que el enfoque de diferencia se vea extendido a similitud, este pequeño cambio conceptual puede ser la base para comprender varios sistemas de información que no estaban al alcance de nuestras herramientas.

Como es conocido los métodos para la obtención de testores típicos, sea el algoritmo YYC (Alba-Cabrera et al., 2014), Fast-BR (Rodríguez et al., 2017) o cualquier otro, se concentran en

la matriz de discernibilidad que utiliza valores binarios que representan la comparación entre atributos, sin embargo, en este caso se hará uso del método UMDA (Alba-Cabrera et al., 2000) ya que su base teórica puede ser fácilmente extendida a valores continuos. No es el único método y la conclusión del análisis invita al uso de estos métodos extendiendo su definición a valores continuos.

Uno de los objetivos planteados es que sea posible llegar a la definición clásica de lo que es la matriz de discernibilidad por medio de nuestra nuevas definiciones, en otras palabras, lograr que la definición clásica se vuelva un caso de nuestra generalización, esto se busca ya que la interpretación de igualdad puede ser redefinida a una comparación según una métrica (en específico la métrica discreta), lo que conllevaría a métodos más sencillos para encontrar generalizaciones de los algoritmos actuales.

Este trabajo es parte de un proyecto a futuro planteado con el objetivo principal de la reconstrucción facial 3D, las redes generativas o GANs (Goodfellow et al., 2014) nos dan una nueva perspectiva para este proyecto, pero estas requieren de un subproceso de clasificación por esta razón los testores podrían mejorar las mismas, sin embargo, en la mayoría de aplicaciones actuales los datos que se utilizan son continuos, por esta razón buscamos que esta nueva definición de testores pueda convertirse en la base teórica para lograr mejores métodos en los procesos generativos. En este caso se presentará una red generativa basada en el algoritmo planteado en (Mao et al., 2017) conocido como Point-Cloud-GAN(Li et al., 2018), una variación de la versión original de las GANs.

## CAPÍTULO 2

### DEFINICIONES Y MÉTODOS

#### 2.1. Testores

Sea  $C$  un conjunto de  $m$  objetos descritos por un grupo de  $n$  atributos y una clase, en este caso se puede definir a  $C$  como un sistema de información (Alba-Cabrera et al., 2000) en total dentro de todo el conjunto consideramos  $l$  clases. Sea  $J$  el conjunto de todos los atributos sobre un objeto tal que  $J = \{j_1, j_2, \dots, j_n\}$

##### Definición 2.1.1. Matriz de Discernibilidad

Consideramos la matriz  $M_{p \times n}$  binaria donde cada fila es una comparación entre una pareja de objetos  $O_1, O_2$  respectivamente de clases  $l_1, l_2$  donde  $l_1 \neq l_2$ . Sea  $O_{ij}$  el atributo  $j$  del objeto  $i$ . La fila esta definida para la comparación entre  $O_1, O_2$ , como  $[m_0, m_1, m_2, \dots, m_n]$  donde

$$m_i = \begin{cases} 1 & \text{si } O_1 = O_2 \\ 0 & \text{si } O_1 \neq O_2 \end{cases}$$

##### Definición 2.1.2. Orden de las filas (Alba and Santana, 2010)

Sea  $r, q$  dos filas dentro de una matriz de discernibilidad, decimos que  $r < q$  si  $\forall i, r_i \leq q_i$  y  $\exists j$  tal que  $r_j \neq q_j$

##### Definición 2.1.3. Fila básica (Alba and Santana, 2010)

Una fila se considera básica sobre una matriz  $M$  si es que no hay otra fila menor a ella dentro de la matriz

##### Definición 2.1.4. Matriz Básica (Alba and Santana, 2010)

Una matriz  $B \subset M$ , donde  $M$  es una matriz de discernibilidad, es básica si contiene todas las filas básicas únicas de la matriz  $M$

**Definición 2.1.5.** Fila de ceros (Torres-Constante et al., 2022)

Sea una fila  $f$  de una matriz  $M$ ,  $f$  se considera una matriz de ceros si es que todos los elementos de esta fila son cero

**Definición 2.1.6.** Testor (Alba and Santana, 2010)

Un conjunto  $T \subset J$  se le llama testor si dentro  $C_T$ , que es el conjunto  $C$  sin todos los atributos que no se encuentren en  $T$ , no existen filas de ceros.

**Definición 2.1.7.** Atributo no removible (Torres-Constante et al., 2022)

Se considera un atributo  $j_{ik}$  del testor  $T$  no removible si existe una fila en  $B_T$  tal que si se elimina la fila de la matriz  $B_T$ , entonces la fila correspondiente al atributo  $j_{ik}$  dentro de  $B_T$ , se convertiría en una fila de ceros. Esto quiere decir que si tomamos la matriz  $B_{T-\{j_{ik}\}}$  entonces esta no cumpliría lo necesario para considerar el conjunto como testor

**Definición 2.1.8.** Testor típico (Torres-Constante et al., 2022)

Sea  $T = \{j_{i1}, j_{i2}, \dots, j_{ik}\}$  un testor sobre la matriz de discernibilidad  $M$ , consideramos  $T$  típico si es que todos los  $j_{i\alpha} \in T$  son atributos no removibles sobre  $T$

## 2.2. UMDA

Para encontrar los testores vamos a hacer uso del método del UMDA, planteado en (Alba-Cabrera et al., 2000),

**Definición 2.2.1.** Población inicial y Cromosoma (Alba-Cabrera et al., 2000)

Una población inicial se le llama a una familia de subconjuntos de  $N$  atributos elegidos aleatoriamente del conjunto  $J$ . Llamamos a un cromosoma a  $x_k = (x_{k1}, \dots, x_{kn})$ ;  $k = 1, \dots, N$ . Y definimos a  $T \subset J$  eliminando todos los atributos de  $J$  donde  $x_{ki} = 0$

**Definición 2.2.2.** Función Objetivo (Alba-Cabrera et al., 2000)

Definimos la función objetivo que se utilizará para evaluar los cromosomas como

$$f(x_k) = \alpha \frac{t(x_k)}{n} + (1 - \alpha) \frac{p(x_k)}{\sum_{v=1, \dots, n} x_{kv}} \quad (2.1)$$

donde  $t(x_k)$  es el número de filas de no ceros en la matriz  $C_T$  (Siendo  $C_T$  como en 2.1.6). Y  $p(x_k)$  el numero de atributos típicos en  $C_T$ .  $\alpha$  se utiliza como un coeficiente de regulación.

### **2.3. Point-Cloud-GAN**

La nube de puntos es una técnica de adquisición de datos que permite la captura tridimensional de un objeto o entorno. El uso de la nube de puntos se ha popularizado en diferentes campos de investigación, incluyendo la robótica (Li et al., 2016), la realidad virtual (Chen et al., 2020), la inspección de calidad (Eslami et al., 2021) y la ingeniería civil (Xu et al., 2020). La precisión y resolución de los datos de la nube de puntos dependen de la tecnología utilizada para la captura y procesamiento de los datos. Por lo tanto, es importante realizar una revisión exhaustiva de las diferentes técnicas y herramientas disponibles para obtener datos de alta calidad. En la tesis de García (García, 2021), se presenta una revisión sistemática de las diferentes técnicas de adquisición y procesamiento de nubes de puntos, así como la evaluación de la calidad de los datos obtenidos a través de estas técnicas. La tesis también describe la generación de modelos 3D a partir de los datos de la nube de puntos, utilizando diferentes software y técnicas de procesamiento de datos.

### **2.4. Base de datos: FACES**

La base de datos FACES (Max Planck Institute for Human Development, 2005) es una herramienta útil para la investigación en el campo de la visión por computadora y la inteligencia artificial. Contiene más de 1,300 imágenes faciales de alta resolución de personas de diferentes edades, razas y géneros, capturadas en diferentes condiciones de iluminación y con distintas expresiones faciales. Además, esta base de datos cuenta con anotaciones precisas de las posiciones de los ojos, la nariz y la boca de cada persona, lo que facilita el desarrollo de algoritmos de reconocimiento facial y de expresiones faciales. Esta base de datos ha sido utilizada en numerosas investigaciones, como en la detección de emociones, el reconocimiento de género y edad, y en el desarrollo de sistemas de seguridad basados en el reconocimiento facial. Además, FACES ha sido ampliamente reconocida como una de las bases de datos más completas y relevantes en el campo de visión computacional.

### **2.5. Puntos de referencia**

En la actualidad, el reconocimiento facial se ha convertido en un área de investigación importante, especialmente en la clasificación de emociones a partir de imágenes faciales. La idea de utilizar

los 68 puntos faciales para el reconocimiento facial se basa en el trabajo pionero de Paul Ekman y Wallace Friesen (Ekman and Friesen, 1978), quienes identificaron estos puntos como claves para medir las expresiones faciales en la década de 1970. Desde entonces, la configuración de 68 puntos ha sido ampliamente utilizada en la investigación de reconocimiento facial y es comúnmente conocida como el sistema de "Facial Action Coding System"(FACS). Sin embargo, es importante tener en cuenta que el número de puntos faciales necesarios para el reconocimiento facial puede variar según la aplicación y la complejidad de las imágenes utilizadas. Algunos estudios han demostrado que se pueden utilizar tan solo 10 puntos para obtener un reconocimiento facial preciso, mientras que otros estudios han mostrado que el uso de un mayor número de puntos puede mejorar la precisión en la clasificación de emociones.

## 2.6. Método Propuesto

El método propuesto se basará principalmente en la generalización de la obtención de la matriz básica sobre un conjunto de datos que no necesariamente necesita ser discreto, para este caso se definen y redefinen nuevas estructuras, y como base vamos a considerar a un conjunto  $C$ , donde se encuentran  $m$  objetos descritos por una clase, en total dentro de todo el conjunto  $l$  clases.

**Definición 2.6.1.** Feature, Atributo y Objeto (Alba and Santana, 2010)

Sea un conjunto que se conocerá como feature,  $J_i$ , tal que  $(J_i, d_i)$  es un espacio métrico. Sea  $O$  un objeto  $O = [j_0, j_1, \dots, j_n]$ , tal que  $j_k \in J_k$ , vamos a llamar  $O_i$  como un objeto dentro de  $T$  si esta definido de la siguiente forma,  $O_i \in T$  tal que  $O_i = [j_{i0}, j_{i1}, \dots, j_{in}]$  tal que  $j_{ik} \in J_k$ . Se le llama atributo a cada elemento de  $O_i$ , y cada atributo es un elemento de un feature.

**Definición 2.6.2.** Disparidad entre objetos

Sean dos objetos de un mismo conjunto de entrada,  $O_i, O_j$ , consideramos la disparidad entre los objetos y lo denotamos como  $O_i O_j$  donde

$$O_i O_j = [d_1(O_{i1}, O_{j1}), d_2(O_{i2}, O_{j2}), \dots, d_n(O_{in}, O_{jn})]$$

**Definición 2.6.3.** Matriz de Disparidad

Sea la matriz  $\Gamma_{p \times n}$  donde cada fila es la disparidad entre dos objetos  $O_1, O_2 \in T$  respectivamente de clases  $l_1, l_2$  donde  $l_1 \neq l_2$ .



**Definición 2.6.4.** Disparidad prescindible

Sea un conjunto  $T$ , y sea una secuencia finita  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ , donde  $\alpha_i$  es definido como un threshold de similitud entre objetos del feature  $J_i$ , consideramos  $O_m O_n$  como una disparidad prescindible si es que

$$\forall k \in \{1, \dots, n\}, d_k(j_{nk}, j_{mk}) < \alpha_k$$

**Definición 2.6.5.** Testor según  $\alpha$ 

Llamamos a un conjunto  $T[\alpha] \subset J$ , Testor según  $\alpha$ , si es que  $C_{T[\alpha]}$  (similar a la definición **2.1.6**) no tiene disparidades prescindibles para la secuencia  $[\alpha_k]$ .

A parte de esto tenemos que extender ciertas definiciones previas para poder utilizar las herramientas que se plantearon. En específico la definición **2.1.7**, tenemos que considerar que un feature no es removible dentro de un Testor según  $\alpha$  si es que existe una disparidad de la matriz  $C_{T[\alpha]}$  tal que si se elimina este feature, entonces esta disparidad se convierte en prescindible. Y finalmente tenemos que alterar la definición **2.2.2** donde simplemente tenemos que especificar que la función  $t(x_k)$  se refiere al número de disparidades no prescindibles en la matriz  $C_{T[\alpha]}$

**Definición 2.6.6.** Testor Típico según  $\alpha$ 

Llamamos a un Testor según  $\alpha$  típico si es que todos sus features no son removibles

Esta investigación se basará en la obtención de Testores para la base de datos FACES, obtención de puntos de referencia, seguido de un análisis para la clasificación de los rostros mostrando un gesto de sonrisa o neutral, y a la vez encontrar los posibles testores con nuestras nuevas definiciones de matriz de disparidades con el método UMDA, finalmente un proceso de una red generativa con los puntos obtenidos para la clasificación de las mismas.

# CAPÍTULO 3

## PROCESO Y RESULTADOS

### 3.1. Análisis de imágenes

Como primer punto tenemos la base de datos FACES (Max Planck Institute for Human Development, 2005), en este caso se planteará simplemente con los grupos de imágenes que representan caras sonrientes y caras neutrales

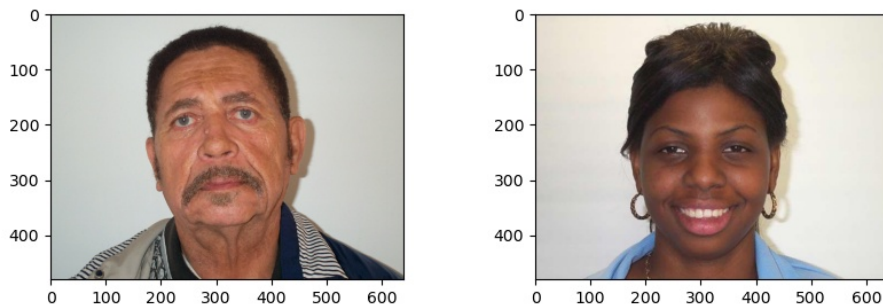


Figura 3.1: Ejemplo de imágenes de la base de datos. Cara neutral (Izquierda), Cara sonriente (Derecha)

Tenemos en total un total 1991 de imágenes, a este conjunto de objetos se denotará como  $I$ , 1447 de la categoría neutrales ( $l_1$ ) y 517 de la categoría sonrientes ( $l_2$ ). En este caso, primeramente vamos a hacer uso de la librería Deepface de python para centrar y alienar los rostros de las imágenes, en este caso utiliza el algoritmo MTCNN (Zhang et al., 2016).

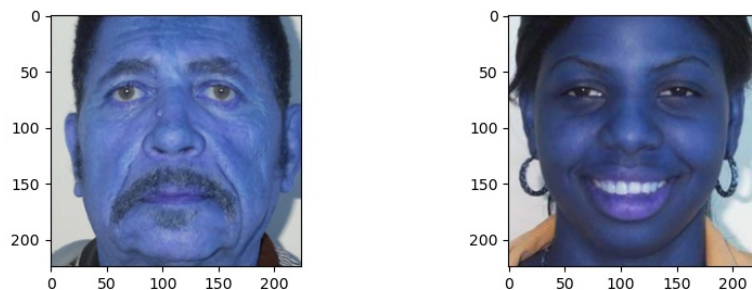


Figura 3.2: Resultado del algoritmo MTCNN

Tenemos las imágenes niveladas y simplificadas por la rotación de los rostros según  $\theta$  tal que

$$\theta = \text{atan2}(y_2 - y_1, x_2 - x_1)$$

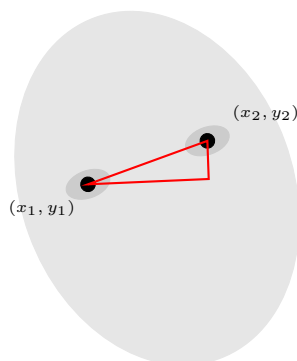


Figura 3.3: Rotación de imagen

vamos a hacer uso del algoritmo para encontrar los puntos de referencia dentro de la herramienta de mediapipe de Google (Kartynnik et al., 2019). Esta herramienta da como resultado un total de 468 puntos, sin embargo, como fue explicado previamente, se hará uso de un los 68 puntos representativos para la clasificación de gestos faciales (Ekman and Friesen, 1978). Obteniendo como resultado

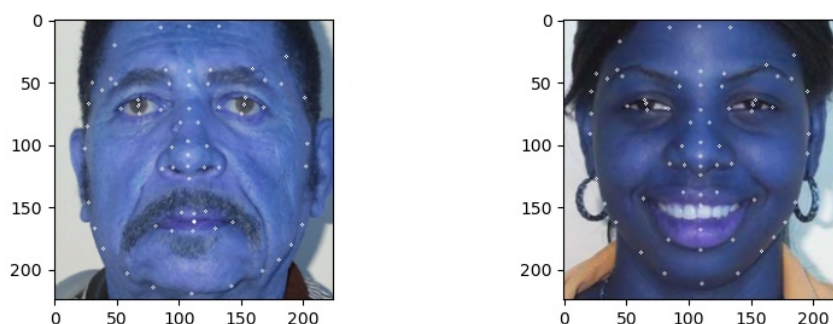


Figura 3.4: Imágen con 68 puntos de referencia del rostro

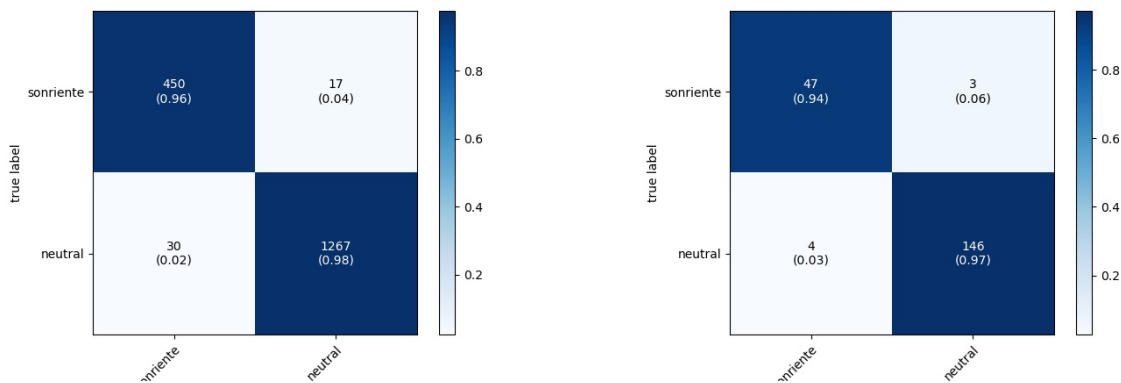
Teniendo todos los puntos de referencia, cada uno expresado en un triplete  $(x, y, z)$  que demuestran las coordenadas del punto, vamos a particionarlo para tener un conjunto de entrenamiento y un conjunto de validación, tenemos una matriz de validación con 200 objetos tal que 50 son de la clase sonriente y 150 de la clase de la clase neutral, entonces  $V_{200 \times 205}$  (205 columnas tal que

el 204 referen a 68\*3 puntos y 1 columna de clase) y a la vez una matriz de entrenamiento tal que  $T_{1791 \times 205}$ . Se hará uso de la siguiente arquitectura de clasificación

Convolution1D (128), kernel_size=3, padding=same
LeakyReLU (0.2)
Dropout (0.3)
Dense Layer (64)
LeakyReLU (0.2)
Dropout (0.3)
Dense Layer (32)
LeakyReLU (0.2)
Dropout (0.3)
Flatten
Dense Layer (1), activation = "sigmoid"

Cuadro 3.1: Arquitectura modelo de clasificación

Obtuyendo los siguientes resultados demostrados en la matriz de confusión



(a) Matriz de confusión para datos de entrenamiento

(b) Matriz de confusión para datos de validación

Obteniendo una precisión de 0,9586 y perdida de 0,102 para datos de entrenamiento. Y una precisión de 0,965 y perdida de 0,1285 para datos de validación.

### 3.2. Testores

Para este punto vamos hacer uso de las nuevas definiciones que planteamos en 2.6. Tomamos una muestra del conjunto  $I$  de 200 imágenes, con 100 de clase sonriente y 100 de clases neutrales. Para analizar estos datos definimos un feature del rostro al punto de referencia y tomamos el conjunto como  $J = \mathbb{R}^3$  como su representante.

Vamos a hacer uso de la métrica  $l^2$  o distancia euclideana. Sea  $x, y \in J$  tal que  $\mathbf{x} = (x_1, x_2, x_3)$  and  $\mathbf{y} = (y_1, y_2, y_3)$

$$d_{l^2}(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}.$$

Para este caso, todos los puntos de referencia recaen sobre este feature, entonces sabemos que sea  $O_i \in I$ ,

$$O_i = [j_{0i}, j_{1i}, \dots, j_{68i}] \quad \forall i \in \{1, 2, \dots, 68\}, \quad j_i \in J$$

Entonces en este caso vamos a conseguir la matriz de disparidad entre objetos de diferentes clases, como resultado tenemos una matriz  $\Gamma_{10000 \times 68}$ , donde cada fila es la disparidad entre dos objetos, siendo  $O_i O_j$  una fila en  $\Gamma_{10000 \times 68}$  tal que

$$O_i O_j = [d_{l^2}(j_{0i}, j_{0j}), \dots, d_{l^2}(j_{68i}, j_{68j})]$$

Tenemos entonces una matriz de disparidad completa y nuestro primer paso es encontrar el Testor según  $\alpha$  relacionado a la matriz, para esto encontramos todas las disparidades afines según una secuencia de  $(\alpha_n)$ , en este caso como los features que utilizamos pertenecen al mismo espacio métrico, vamos a hacer uso de un  $\alpha = (\alpha, \alpha, \dots, \alpha)$  donde todos sus elementos son iguales. En este caso se hace experimentalmente para un rango de valores definidos, y se encuentra el número de disparidades en el Testor según  $\alpha$  final.

$\alpha$	Número de disparidas en Testor $\alpha$
0.01	9744
0.05	9684
0.1	5185
0.15	746
0.2	91

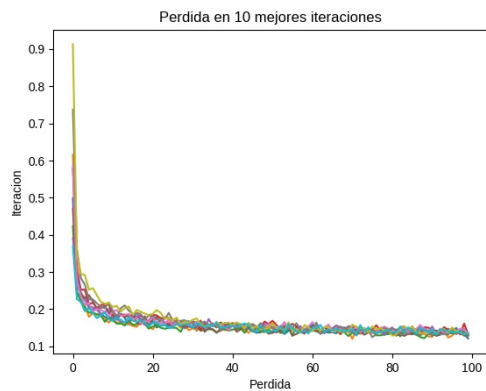
Cuadro 3.2: Elección de  $\alpha$  a nivel experimental

Se eligió entre los candidatos  $\alpha = 0,1$  y se tomo el Testor según este valor para el cálculo de testores típicos.

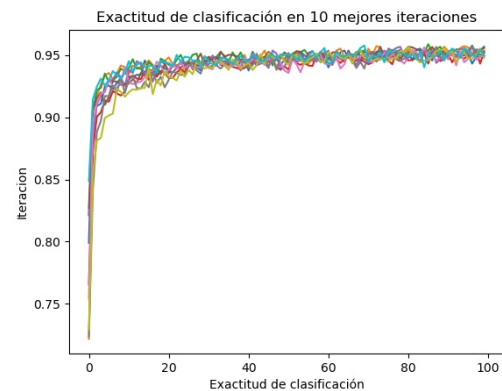
En este caso como se hará uso del método UMDA para conseguir un testor, para esto se repetirá todo el proceso un total de 20 veces. Definimos el coeficiente de regulación (el valor  $\alpha$  definido para la función objetivo) a ser 0,2, como es sugerido en (Alba-Cabrera et al., 2000) y se consiguen los siguientes resultados según el número de generaciones que se requiere, se tiene un threshold

de 0.98 similar a (Torres-Constante et al., 2022). Además como se busca encontrar el conjunto con mejor porcentaje de clasificación a la vez para cada una de estas iteraciones vamos a tener un registro de los valores de pérdida de clasificación y de exactitud de clasificación en el conjunto de validación, esto basado en el algoritmo propuesto en (Torres-Constante et al., 2022), vamos a elegir a la que tenga los mejores parámetros para continuar.

A continuación presentamos los resultados de pérdida y de exactitud para las iteraciones que presentaron los mejores resultados



(c) Pérdida para 10 mejores iteraciones



(d) Exactitud de clasificación para 10 mejores iteraciones

Finalmente elegimos el cromosoma que cuenta con un total de 42 features, significando un discriminador con 61.8 % de los features iniciales, este consiguió una pérdida del 0.12 y una precisión del 0.947

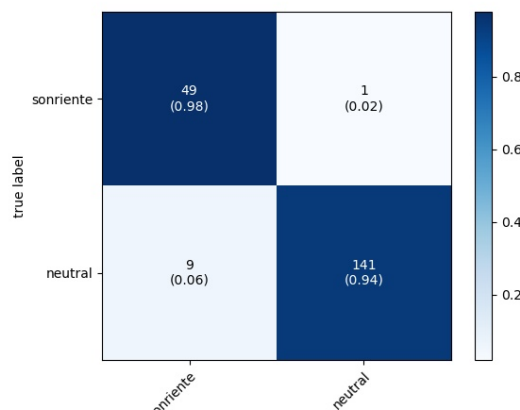
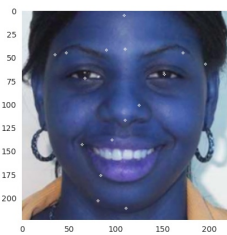


Figura 3.5: Matriz de confusión para cromosoma seleccionado

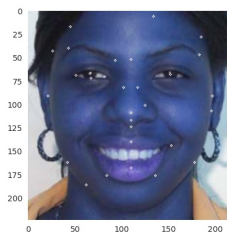
Podemos ver la evolución de este cromosoma según las generaciones en la siguiente tabla y resultados

Generación	# de no Ceros en Población	Función Objetivo
10	12	0.292
20	16	0.262
30	29	0.243
40	35	0.695
48	42	0.981

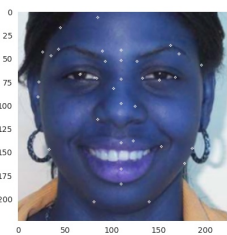
Cuadro 3.3: Resultados de proceso generativo



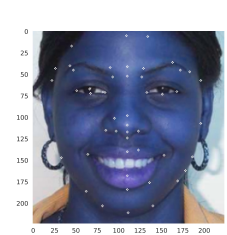
(a) Cromosoma en generación 20



(b) Cromosoma en generación 30



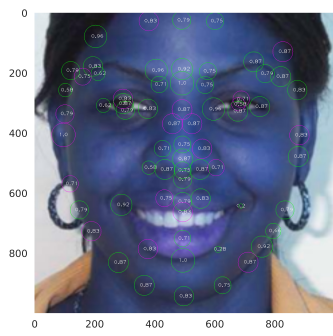
(c) Cromosoma en generación 40



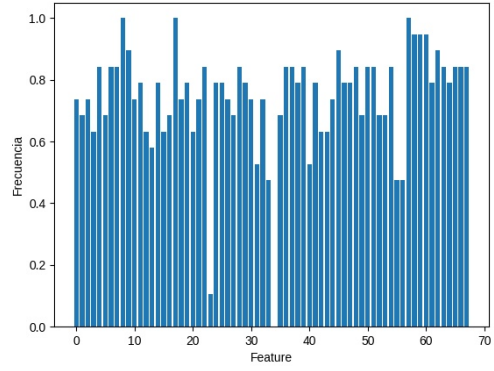
(d) Cromosoma Final

Figura 3.6: Resultados del algoritmo según la generación

Entonces fue posible conseguir un subconjunto de los features que que permite aún mantener la clasificación de las imágenes. Con la finalidad de extender el entendimiento del proceso, vamos a analizar los resultados de las diferentes iteraciones del algoritmo UMDA, para esto vamos a encontrar la frecuencia marginal de cada uno de los atributos dentro del conjunto de todos los cromosomas obtenidos en cada iteración, además podemos visualizar la distribución, donde el tamaño del círculo alrededor del punto de referencia muestra la frecuencia, y los que están en color verde son aquellos que forman parte de nuestro conjunto



(a) Frecuencia marginal en los rostros



(b) Frecuencia marginal por feature

Figura 3.7: Generaciones según el número de iteración

Podemos ver mas rostros con el cromosoma seleccionado



Figura 3.8: Rostros de la base de datos con el cromosoma seleccionado



### 3.3. Point-Cloud-GAN para puntos de referencia y modelado

#### 3D

Como última parte de nuestro proceso se define la red generativa, para esto se hace uso del discriminador y generador definido dentro del algoritmo Point-Cloud-GAN (Li et al., 2018), se podría creer que es posible hacer uso del mismo discriminador previo, sin embargo es importante reconocer la diferencia, ya que el previo define una categorización entre neutral y sonriente, mientras que el definido en el algoritmo funciona para predecir si un conjunto de puntos es verdadero o falso. En si el nuevo discriminador y generador está definido para las siguientes funciones de pérdida.

$$L_D = -[\log(D(x)) + \log(1 - D(G(z)))]$$

$$L_G = -\log(D(G(z)))$$

Esta red generativa dio como resultados los siguientes conjuntos para varias de las iteraciones

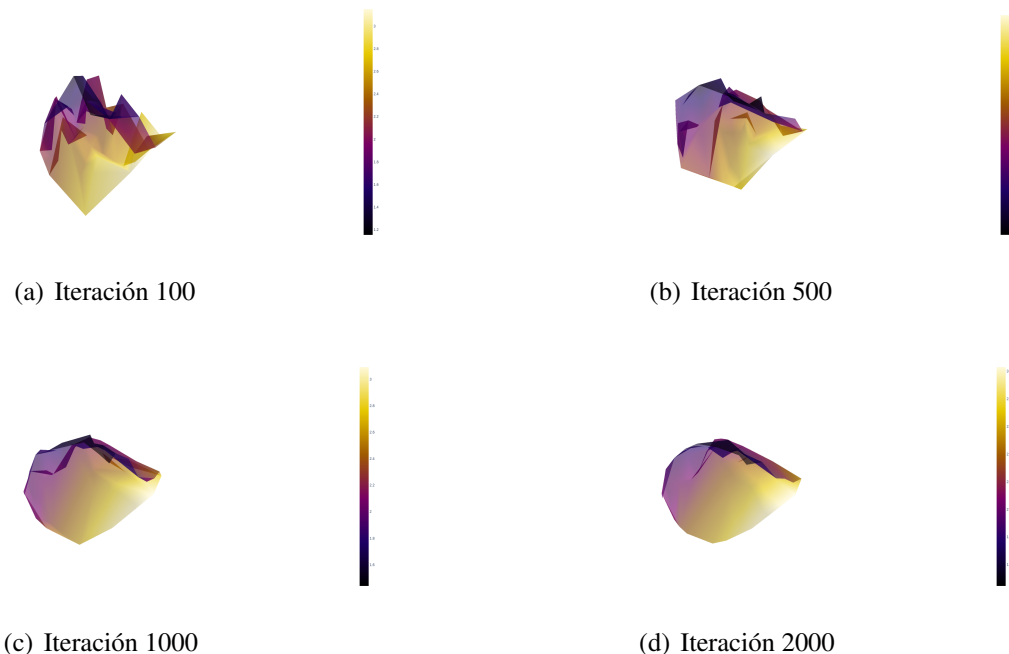


Figura 3.9: Generaciones según el número de iteración

En este caso se utilizo nuestro clasificador de rostros sonrientes y neutrales, se obtuvo un total de

1000 rostros generados, y se clasificó un total del 63.4 % de rostros sonrientes y el resto rostros los clasificó como neutrales Podemos visualizar más rostros generados y su clasificación

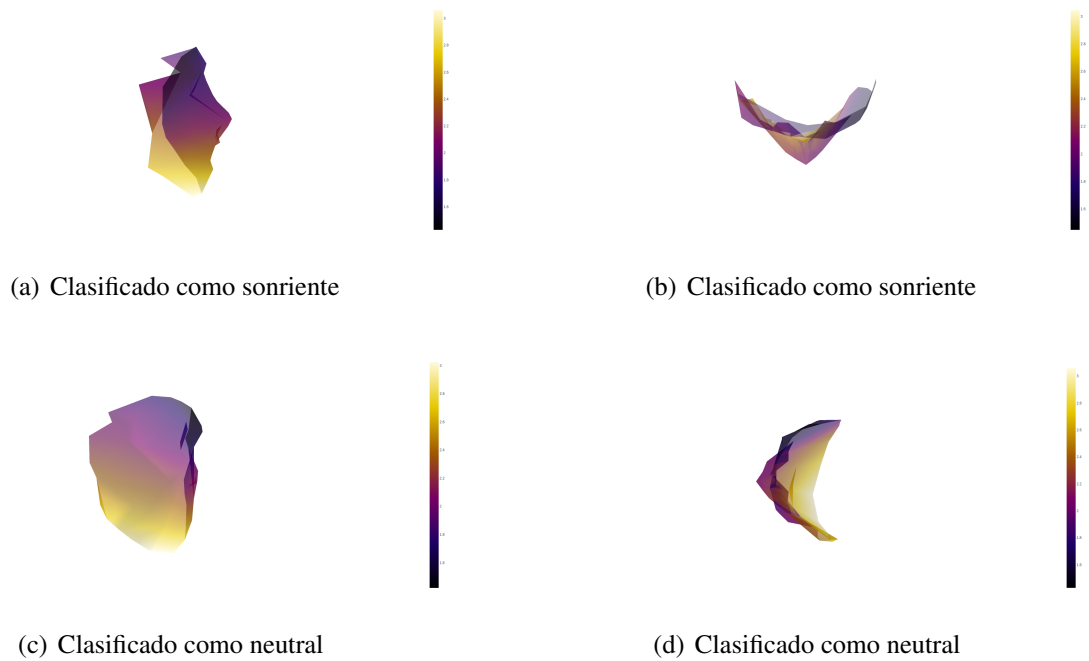


Figura 3.10: Clasificación de resultados de las GANs

## CAPÍTULO 4

### CONCLUSIONES Y TRABAJO FUTURO

La presentación de este nuevo método de análisis de sistemas de información permite la extensión de la idea de diferencia para un conjunto que no necesariamente debe ser discreto. En este trabajo con el método presentado se logró encontrar un discriminador que reduce a un 61.8 % de los features iniciales, manteniendo aún una exactitud de clasificación de 0.9481, lo que demuestra que aún tiene la capacidad de distinguir entre clases, podemos considerar que en este caso con las base de datos que se utilizó el método y el análisis funciona.

La propuesta tiene como finalidad exponer esta generalización para poder entender de forma más concreta que es lo que nos demuestra o detalla un testor. Mantener un seguimiento de esta idea permitirá solventar ciertas limitaciones o prácticas experimentales de este proyecto, específicamente hay dos que se muestran como el siguiente objetivo, primero entender cuál es la acción del  $\alpha$  o threshold de similitud dentro del Testor  $\alpha$  y por otro lado, entender y aplicar los algoritmos actuales con esta generalización, ya que por ejemplo, el algoritmo LEX (Alganza and Porrata, 2003) o el algoritmo YYC (Alba-Cabrera et al., 2014), presentan dentro de su estructura definiciones que se pueden extender a las propuestas.

## BIBLIOGRAFÍA

- Alba, E. and Santana, R. (2010). Generación de matrices para evaluar el desempeño de estrategias de búsqueda de testores típicos. *ACI Avances en Ciencias e Ingenierías*, 2(2).
- Alba-Cabrera, E., Ibarra-Fiallo, J., Godoy Calderón, S., Cervantes-Alonso, F., Robles, D., Interoceánica, V., and Quito (2014). Yyc: A fast performance incremental algorithm for finding typical testors.
- Alba-Cabrera, E., Santana, R., OCHOA-RODRIGUEZ, A., and Lazo-Cortés, M. (2000). Finding typical testors by using an evolutionary strategy.
- Alganza, Y. S. and Porrata, A. P. (2003). Lex: un nuevo algoritmo para el cálculo de los testores típicos.
- Chen, Y., Shen, Y., Shi, C., Liu, M., and Zhang, X. (2020). A novel real-time 3d reconstruction method for virtual reality. *IEEE Access*, 8:171023–171032.
- Ekman, P. and Friesen, W. V. (1978). *Facial Action Coding System (FACS) [Database record]*. APA PsycTests.
- Eslami, S., Karimaei, M., and Pashaei, M. (2021). Quality control of concrete surfaces based on photometric stereo and point cloud processing. *Automation in Construction*, 123:103494.
- García, J. (2021). Adquisición y procesamiento de nubes de puntos para la generación de modelos 3d. Master's thesis, Universidad Nacional de Colombia, Bogotá.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

- Kartynnik, Y., Ablavatski, A., Grishchenko, I., and Grundmann, M. (2019). Real-time facial surface geometry from monocular video on mobile gpus. *CoRR*, abs/1907.06724.
- Lazo-Cortes, M., Ruiz-Shulcloper, J., and Alba-Cabrera, E. (2001). An overview of the evolution of the concept of testor. *Pattern Recognition*, 34(4):753–762.
- Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., and Salakhutdinov, R. (2018). Point cloud gan. *arXiv preprint arXiv:1810.05795*.
- Li, H., Huang, Q., and Tan, T. (2016). Large-scale outdoor scene analysis from mobile laser scanning data with deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 118:27–42.
- Mao, X., Li, Q., Xie, H., Lau, R. Y., and Wang, Z. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.
- Max Planck Institute for Human Development, C. f. L. P. (2005). FACES Database. <https://faces.mpg.de/imeji/>. Accessed: 2023-01-31.
- Rodríguez, V., Martínez-Trinidad, J. F., Carrasco-Ochoa, J., and Lazo-Cortés, M. (2017). Fast-br vs. fast-ct ext: An empirical performance study.
- Torres-Constante, E., Ibarra-Fiallo, J., and Intriago-Pazmiño, M. (2022). *A New Handwritten Number Recognition Approach Using Typical Testors, Genetic Algorithms, and Neural Networks*, pages 291–305. Communications in Computer and Information Science book series (CCIS, volume 1532).
- Xu, Z., Zhang, J., and Huang, J. (2020). Use of point cloud data for asphalt pavement evaluation. *Journal of Computing in Civil Engineering*, 34(2):04019065.
- Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.