

HeadMouse

Tique Juan(2420151028), Sierra Juan(2420151038)
2420151028@estudiantesunibague.edu.co, 2420151038@estudiantesunibague.edu.co
Universidad de Ibagué
Ibagué-Tolima
Carrera 22 Calle 67 B/Ambalá
Presentado a Ing Harold F. Moreno

Abstract

This project is developed through a supervised learning algorithm that allows to identify in real time the gestures made by the user using the random forest classifier, each gesture is assigned a mouse action which is activated by a time counter in each gesture, the algorithm trains with a database created from the user's gestures.

1. Objetivos

Objetivo general

Desarrollar un algoritmo capaz de manipular las acciones principales de un mouse mediante los gestos y expresiones faciales.

Objetivos específicos

- Identificar los posibles gestos de la cara para emular las acciones del mouse.
- Realizar una base de datos que contenga el valor de la magnitud generada por el gesto realizado, por una muestra de personas y seleccionar los mejores descriptores.
- Entrenar el algoritmo de reconocimiento de gestos con la base de datos anteriormente mencionada mediante aprendizaje supervisado y evaluar su desempeño en tiempo real.

2. Introducción

El proyecto HeadMouse se desarrollado con el fin de mejorar la calidad de vida de las perso-

nas con dificultad de movimiento o ausencia de extremidades superiores. Mediante machin learning se logró implementar un algoritmo en python que controla las acciones del mouse mediante el reconocimiento de gestos faciales en este caso 5 gestos diferentes, de esta manera es posible controlar el mouse en su totalidad solo con el movimiento de la cara, con esto se espera que cualquier persona sin importar su condición pueda manejar el mouse por cuenta propia.

3. Metodología

Elección y analisis de descriptores

El primer paso para el desarrollo de este proyecto es la elección de los descriptores que son la base para el entrenamiento del algoritmo, en este caso los descriptores son los gestos faciales, como primera instancia se eligieron seis gestos los cuales son abrir la boca, silbar, sonreír, subir las cejas, fruncir el ceño y cerrar los ojos. De estos seis se descartó el gesto de cerrar los ojos ya que es un descriptor que no genera buenas condiciones para el entrenamiento, interfiriendo con los descriptores restantes. A cada descriptor se le asigna una clase del 0 al 5 dejando la clase cinco

para los datos sin gestos de la cara del usuario, esto con el fin de que el algoritmo tenga una clase en la cual no tenga ningún descriptor ni acción de mouse asignada, de esta forma se minimiza el error que generaría el algoritmo al asignar los datos de la cara sin gestos al descriptor que más se les acerque a estos datos.

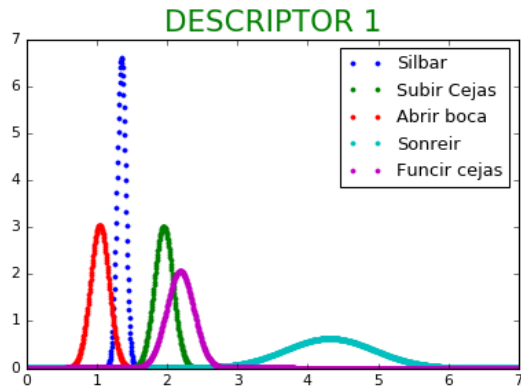


Figura 1: Campanas de Gauss para el descriptor 1

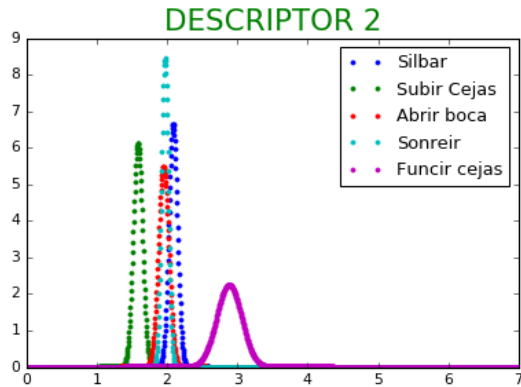


Figura 2: Campanas de Gauss para el descriptor 2

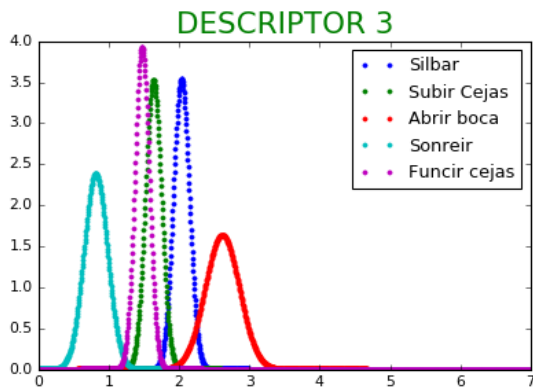


Figura 3: Campanas de Gauss para el descriptor 3

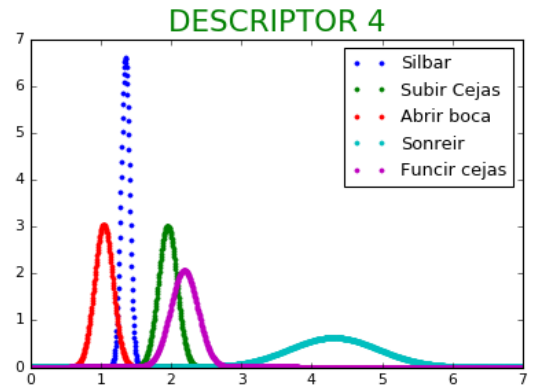


Figura 4: Campanas de Gauss para el descriptor 4

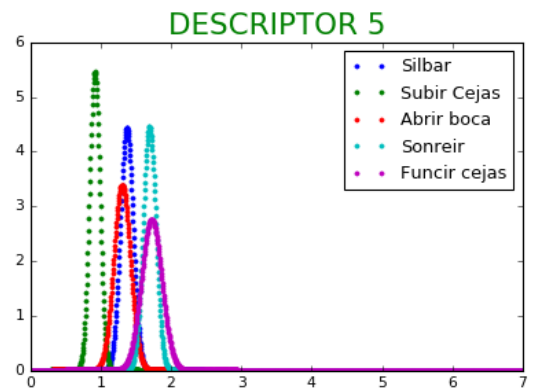


Figura 5: Campanas de Gauss para el descriptor 5

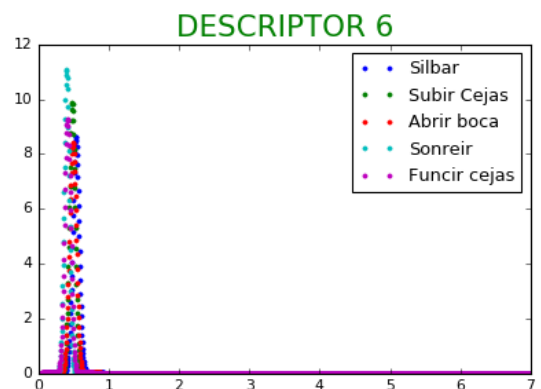


Figura 6: Campanas de Gauss para el descriptor 6

Después del proceso de selección de los descriptores, se observa que las clases son muy parecidas al momento de evaluarlas con el descriptor 6, debido a esto el descriptor se descarta ya que puede enviar datos falsos al predictor. Después se procede a realizar un trabajo matemático con cada uno de los gestos que se van a usar, esto con el fin de calcular el valor que diferencia el gesto de los otros, este cálculo se realiza con los puntos que se generan en el contorno de la cara, cada punto tiene un valor en píxeles y al realizar un gesto en alguna parte específica de la cara los valores de estos puntos cambian lo cual genera un cambio de valores entre los puntos, es por este medio que se puede conocer el valor específico que arroja cada gesto. Estos valores únicos de cada gesto son los que se muestrean para realizar la base de datos para el entrenamiento del algoritmo principal.

Silbar [0]

El descriptor silbar al cual se le ha asignado la clase “0”, genera un valor diferencial del resto al juntar los puntos del contorno de la boca.

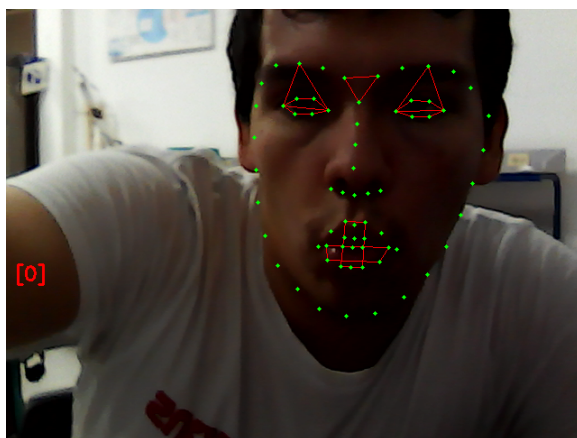


Figura 7: Silbar [0]

Subir las cejas [1]

El descriptor subir las cejas al cual se le ha asignado la clase “1”, genera un valor diferencial del resto al crear un aumento en la distancia entre los puntos extremos del contorno de cada uno de los ojos y el punto central del contorno de cada ceja.

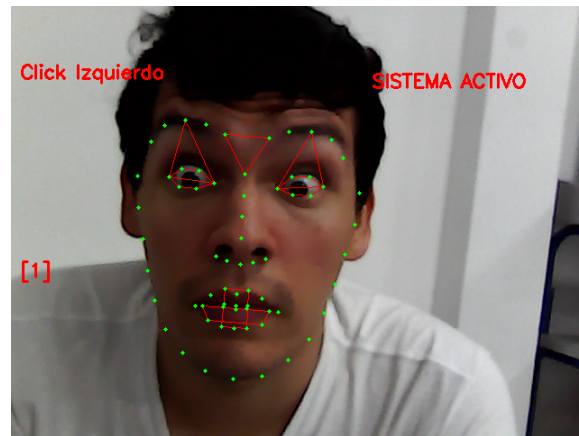


Figura 8: Subir cejas [1]

Abrir boca [2]

El descriptor abrir boca al cual se le ha asignado la clase “2”, genera un valor que lo diferencia del resto con la diferencia que se produce entre los puntos superiores e inferiores del contorno de la boca.



Figura 9: Abrir boca [2]

Sonrisa [3]

A el descriptor sonrisa se le ha asignado la clase “3”, este genera un valor diferencial del resto en el momento de alargar los puntos extremos del contorno de la boca.

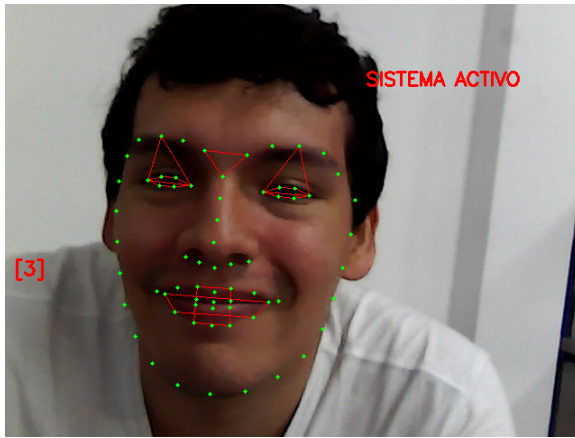


Figura 10: Sonrisa [3]

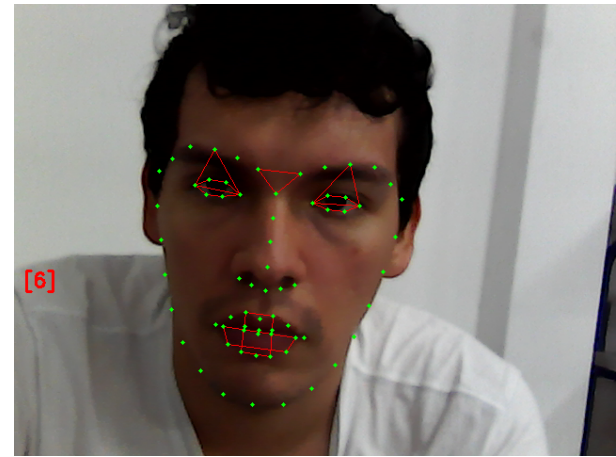


Figura 12: Cara normal [5]

Fruncir el ceño [4]

A este descriptor se le ha asignado la clase “4”, fruncir el ceño crea el valor que lo diferencia de los demás gestos al disminuir el área marcada por el punto superior del contorno de la nariz y los puntos iniciales generados por cada ceja.

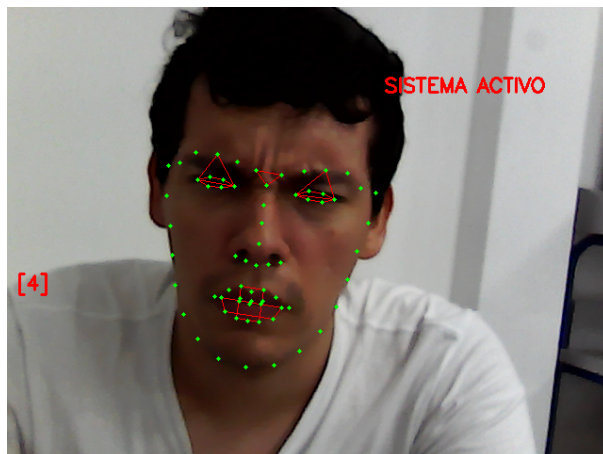


Figura 11: Fruncir el ceño [4]

Cara normal o sin gestos [5]

Cara normal o sin gestos es el nombre que se le ha implementado a los datos que generan cada uno de los descriptores en su estado de normalidad o relajación, este estado se produce cuando el usuario no está haciendo ningún gesto voluntariamente. A estos datos se les ha asignado la clase “5” y de esta forma el predictor tiene una clase a la cual puede asignar este estado de normalidad.

Calibración del sistema

Para generar una base de datos del usuario lo suficientemente robusta para el entrenamiento del algoritmo principal, se diseñó un código capaz de hacer la toma automática de mil datos de cada gesto en un vídeo de aproximadamente 20 segundos, el trabajo del usuario es asignarle a cada gesto que esté haciendo en el momento la clase predeterminada para este, con esto se genera una base de 6000 datos la cual se ingresa en el algoritmo principal para su entrenamiento.

```
juan@juan-GL552VW:~/Escritorio/mousehead (2)$ python main.py
ingrese target
indique el target del gesto: 0
```

Figura 13: Asignación de clase

Algoritmo principal

Como etapa final se diseñó un algoritmo capaz de reconocer cada uno de los gestos predefinidos mediante el entrenamiento que se genera con random forest de la librería sklearn, este algoritmo se entrena a partir de la base de datos del usuario generada por el código de calibración, de esta forma genera en tiempo real una predicción de la clase que está haciendo el usuario. A cada clase se le ha implementado una acción del mouse por lo tanto cada vez que el usuario hace un gesto el predictor arroja su clase correspondiente la cual genera la acción designada del mouse, para que el sistema no active repetitivamente la acción del mouse en el momento de hacer un gesto prolongado se diseñó un contador de tiempo

para cada gesto, de esta forma ejecutar la acción del mouse una sola vez después de realizar el mismo gesto por un tiempo predefinido.

Cuando se desea realizar las acciones de mover el puntero o el scroll del mouse se ha implementado un sistema de habilitación y deshabilitación de estas acciones mediante un contador de gestos, al momento de hacer un gesto habilita la acción y al tener activa esta acción y repetir el gesto se desactiva la función. Por último, se ha designado un gesto para la activación y desactivación de todo el sistema siendo este gesto el que permite o niega la lectura del resto. Para facilitar el uso del sistema se ha puesto en pantalla cada una de las acciones y habilitaciones para que el usuario tenga en todo instante conocimiento de que acción puede o no realizar.

4. Conclusiones

- Seleccionar el algoritmo random forest para el entrenamiento ofreció una gran ventaja gracias a su facilidad y eficiencia para manejar bases de datos extensas y su gran tiempo de respuesta al momento de generar las predicciones acertadamente.
- Entrenar el algoritmo es un proceso determinante en el desempeño del clasificador, puesto que poner los datos de entrada del entrenamiento correctamente inciden en el desempeño final del clasificador.
- Descartar el descriptor de cerrar los ojos mejora los resultados de las predicciones ya que este proporcionaba al algoritmo datos erróneos para su entrenamiento.
- Tras analizar diferentes descriptores se encontró que los descriptores que reflejan una relación entre landmarks del rostro permite que haya una mayor robustez para la distinción de las clases.
- Se encontró que teniendo el usuario dentro de la base de datos se mejora el desempeño final del dispositivo.

- La base de datos debe contener personas con diferentes expresiones faciales, de manera que contenga diferentes posibilidades y hacer el algoritmo de predicción más robusto
- El uso de una cámara con buena definición es de gran ayuda para el proceso del algoritmo ya que ayuda a una mejor recolección de datos tanto para el entrenamiento como para las pruebas en tiempo real.
- La posición e iluminación de la cámara influye en gran medida el buen resultado del clasificador ya que la toma de datos falsos afecta directamente el entrenamiento de este y por ende su respuesta en las pruebas en tiempo real.

Referencias

- [1] online, title = "Mouse Cursor Control Using Facial Movements", url = "https://towardsdatascience.com/mouse-control-facial-movements-hci-app-c16b0494a971", year = "2019",
- [2] online, author = "M. Cardenas, R. Medel, J. Castillo, J. Vázquez and O. Casco", title = "Mouse Cursor Control Using Facial Movements", url = "http://sedici.unlp.edu.ar/bitstream/handle/10915/45467",
- [3] online, author = "sklearn.metrics", title = "f1_score", url = "https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html",
- [4] online, author = "A. Chandra Lagandula", title = "Mouse Cursor Control Using Facial Movements-An HCI Application", url = "https://towardsdatascience.com/mouse-control-facial-movements-hci-app-c16b0494a971",
- [5] Muarcia Harold, "Notas de clase de Inteligencia Artificial 2019A", Universidad de Ibagué, 2019