

# TSIGN

Sign language tool





# ¿Por qué TSign?

TSIGN es una herramienta para las personas sordomudas quienes además de su lengua nativa, tienen que aprender otros idiomas, pues el lenguaje de las señas no es universal, sino que cambia dependiendo del idioma en el que las personas quieran comunicarse. Es por esto que esta aplicación ayudará a las instituciones en el proceso de capacitación de una forma interactiva.



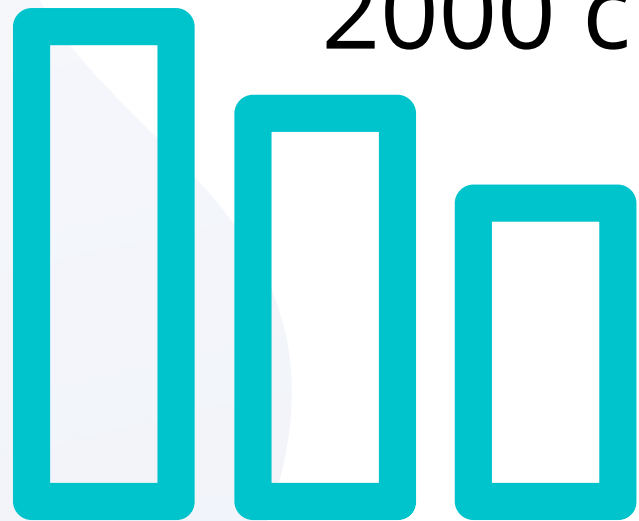


# DATASET

## WLASL

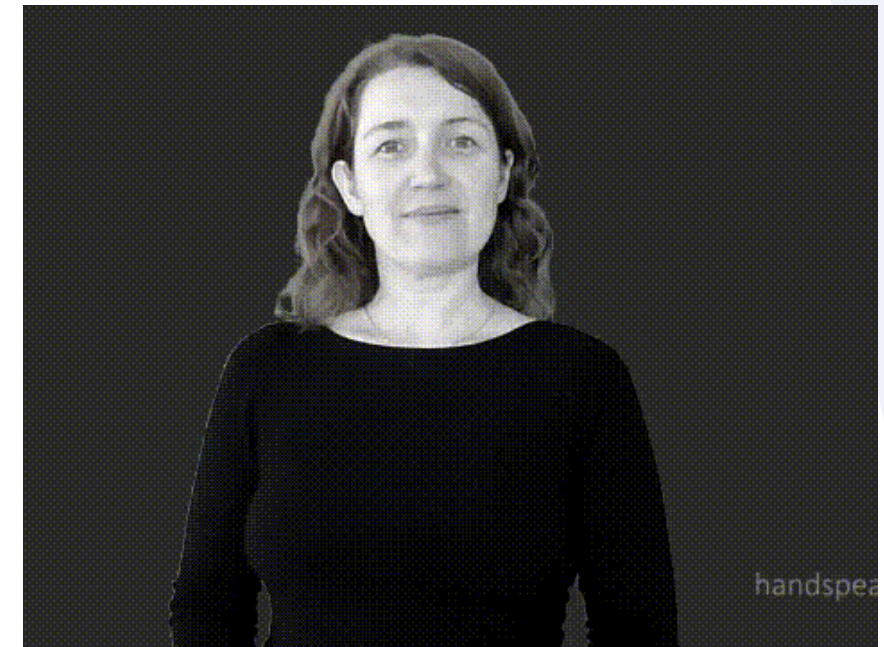
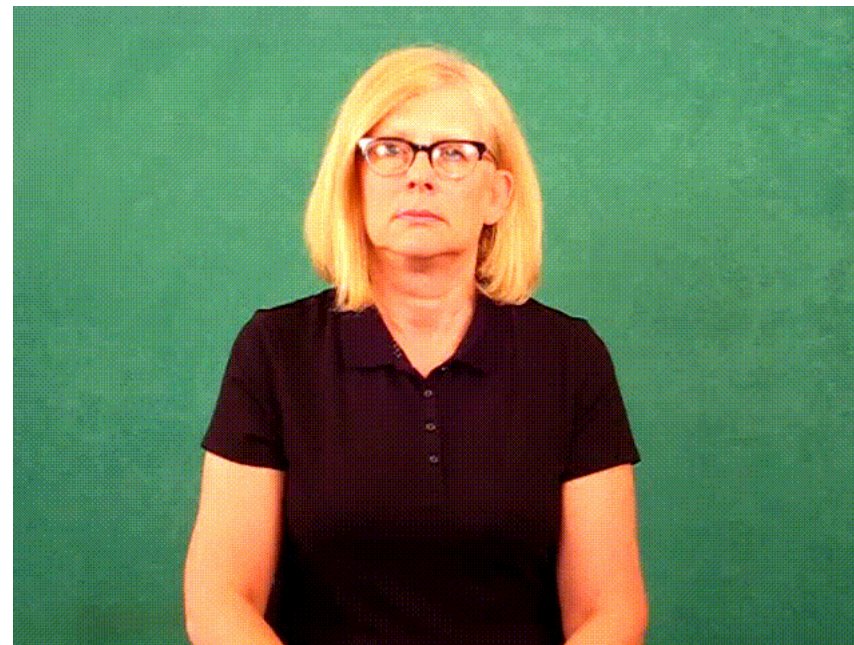
21083 vídeos

2000 clases



800 vídeos

10 clases





# PRE PROCESAMIENTO DE LOS VÍDEOS



Debido a que cada clase tiene un promedio de 10 a 12 vídeos se crearon más mediante el uso de métodos como lo son desplazamiento, espejo y cambio en el orden de los canales RGB, de esta forma logrando subir el numero de muestras aproximadamente a 80 por cada clase

# TRATAMIENTO

01

BACKGROUND SUBTRACTION

02

OPTICAL FLOW

03

EDGES WITH OPTICAL  
FLOW

04

BAG OF WORDS

05

EDGES WITH CORNER  
HARRIS

06

CART POLAR



01

BACKGROUND  
SUBTRACTION



02

OPTICAL FLOW



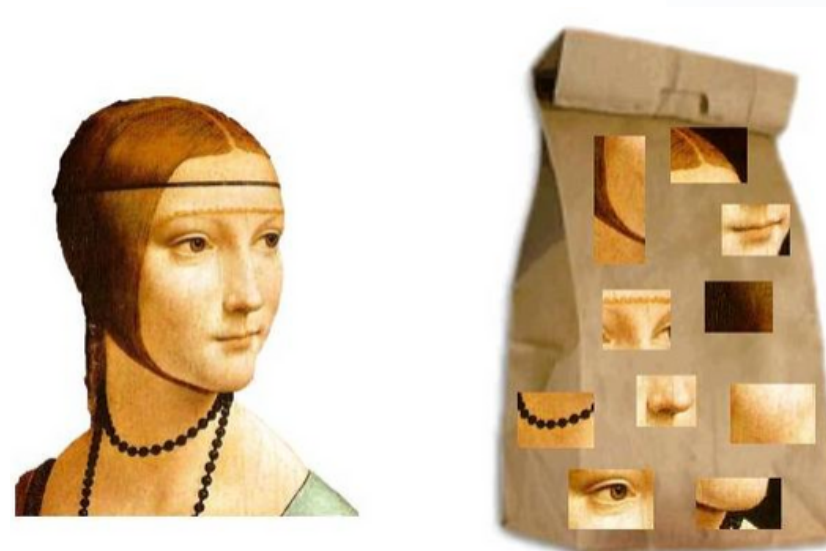
03

EDGES WITH  
OPTICAL FLOW



04

BAG OF WORDS



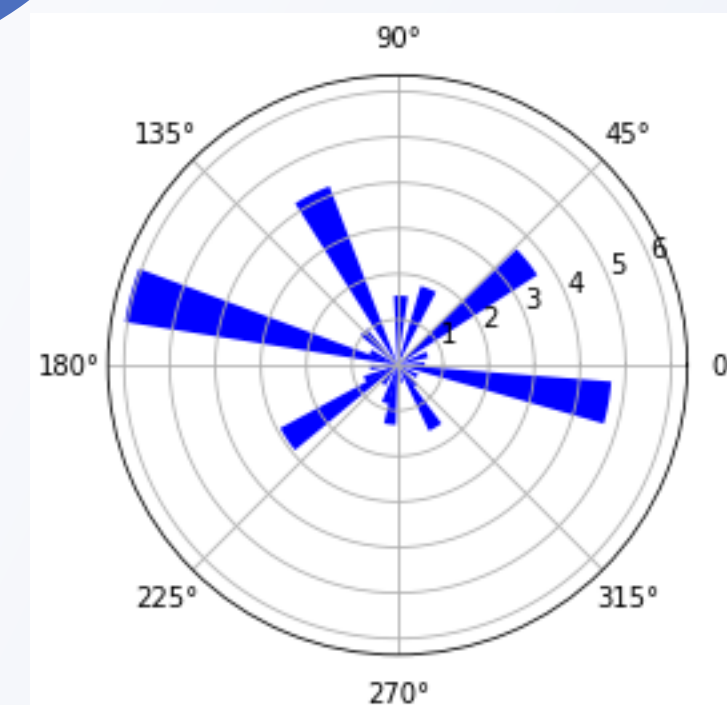
05

EDGES WITH CORNER  
HARRIS



06

CART POLAR



01

Naive Gaussian por defecto

02

Super Vector Machine, tipo poly con un grado de 7

03

Random Forest, número de arboles de 10 con una profundidad de 19

04

Decision Tree por defecto

05

K-NN, número de vecinos igual a 1

06

Deep neural Network

07

Convulotional Neural Network

# CLASIFICADORES

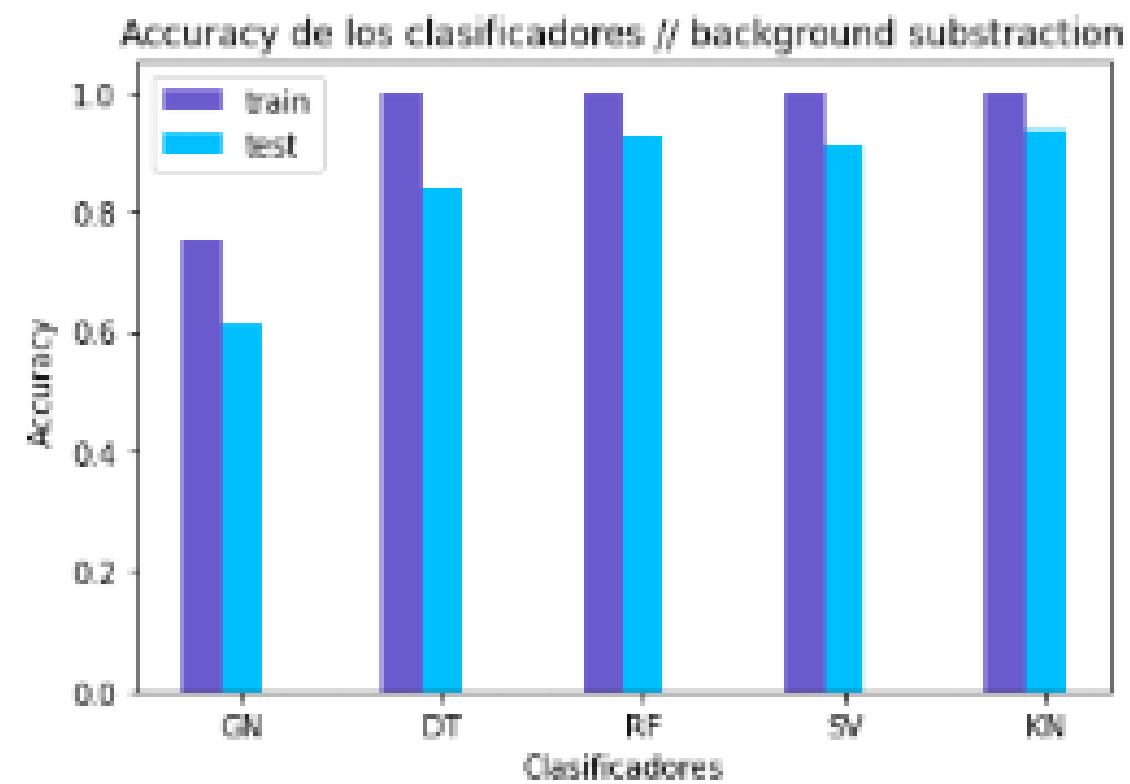


# RESULTADOS CLASIFICADORES

01

Background Substraction

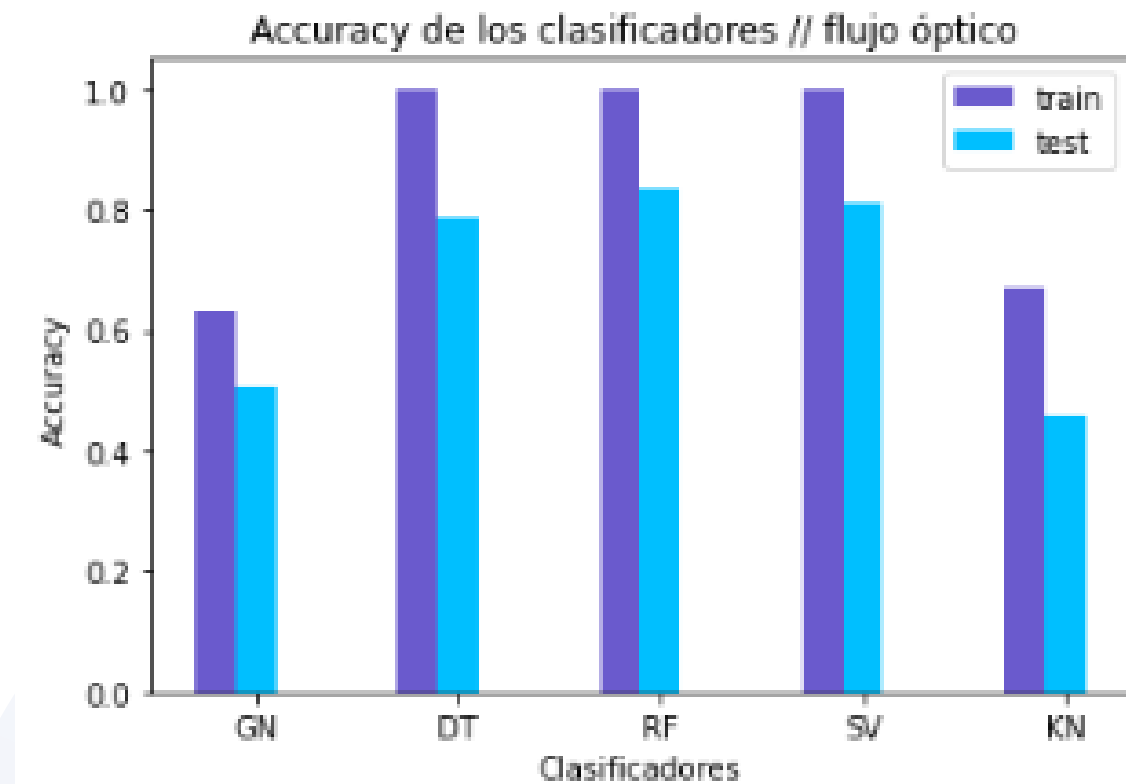
accuracy NB 0.581 (+/- 0.06281)  
accuracy DT 0.812 (+/- 0.03260)  
accuracy RF 0.885 (+/- 0.02610)  
accuracy SVC 0.879 (+/- 0.03064)  
accuracy KNeighbors 0.883 (+/- 0.04301)



02

Optical Flow

accuracy NB 0.480 (+/- 0.06585)  
accuracy DT 0.778 (+/- 0.04361)  
accuracy RF 0.872 (+/- 0.02534)  
accuracy SVC 0.836 (+/- 0.04239)  
accuracy KNeighbors 0.436 (+/- 0.06564)



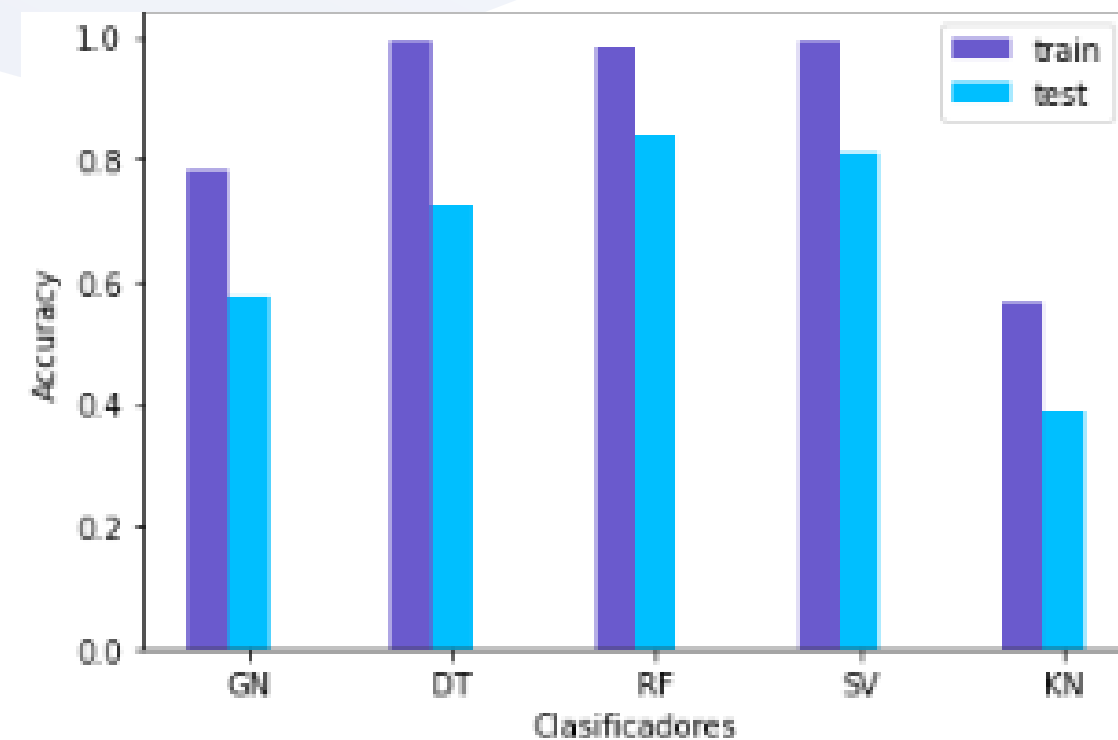


# RESULTADOS CLASIFICADORES

03

EDGES WITH OPTICAL FLOW

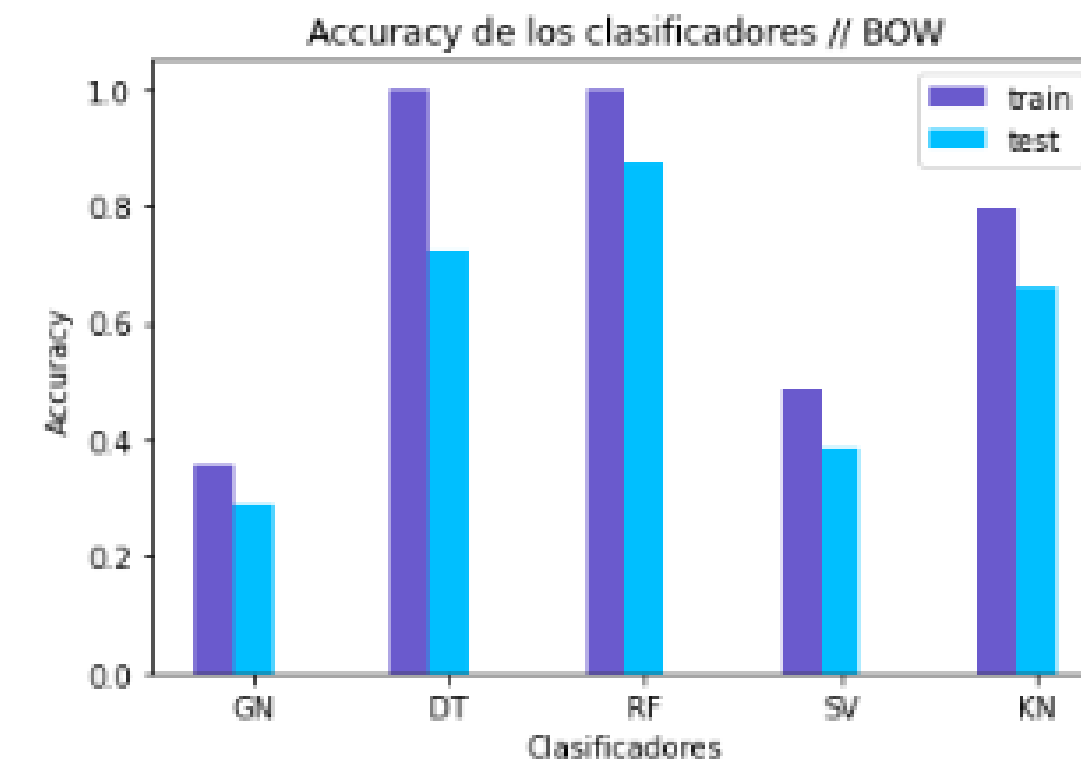
```
accuracy NB 0.603 (+/- 0.04519)
accuracy DT 0.761 (+/- 0.02972)
accuracy RF 0.815 (+/- 0.04474)
accuracy SVC 0.832 (+/- 0.04426)
accuracy KNeighbors 0.323 (+/- 0.04673)
```



04

BAG OF WORDS

```
accuracy NB 0.313 (+/- 0.03991)
accuracy DT 0.758 (+/- 0.04964)
accuracy RF 0.846 (+/- 0.01997)
accuracy SVC 0.402 (+/- 0.02686)
accuracy KNeighbors 0.597 (+/- 0.07151)
```

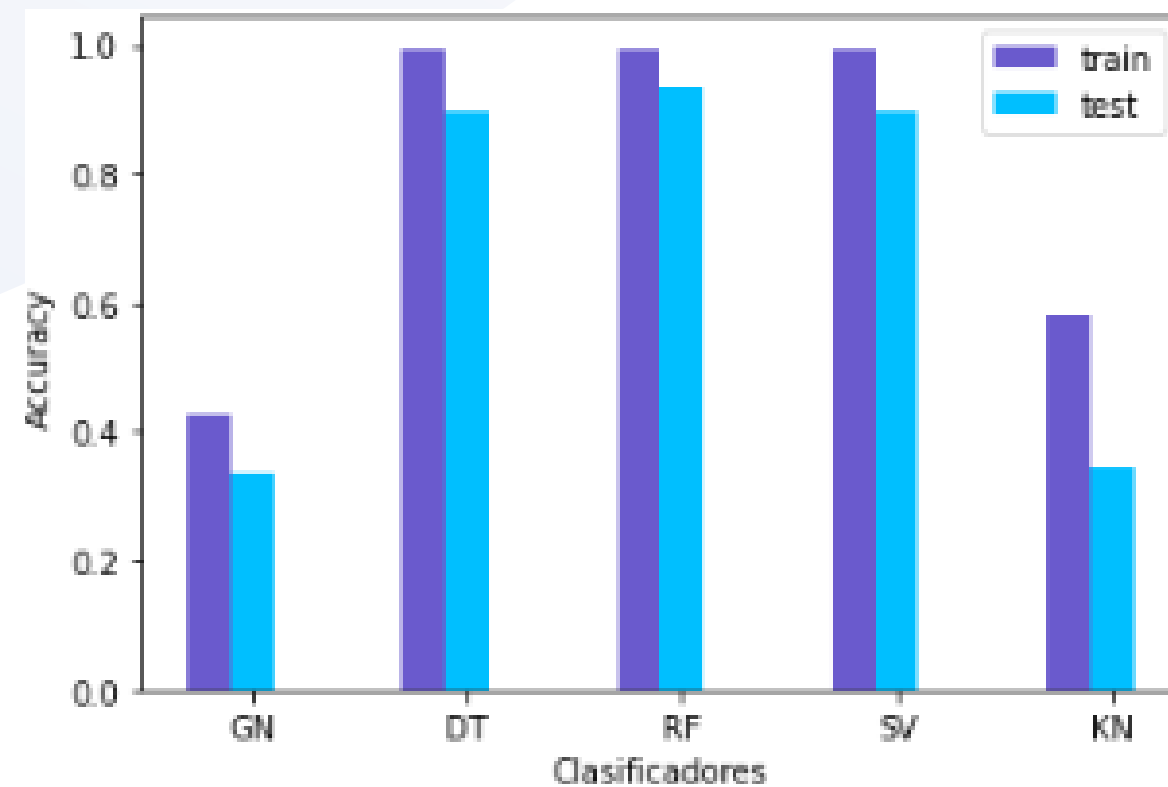


# RESULTADOS CLASIFICADORES

05

EDGES WITH CORNER HARRIS

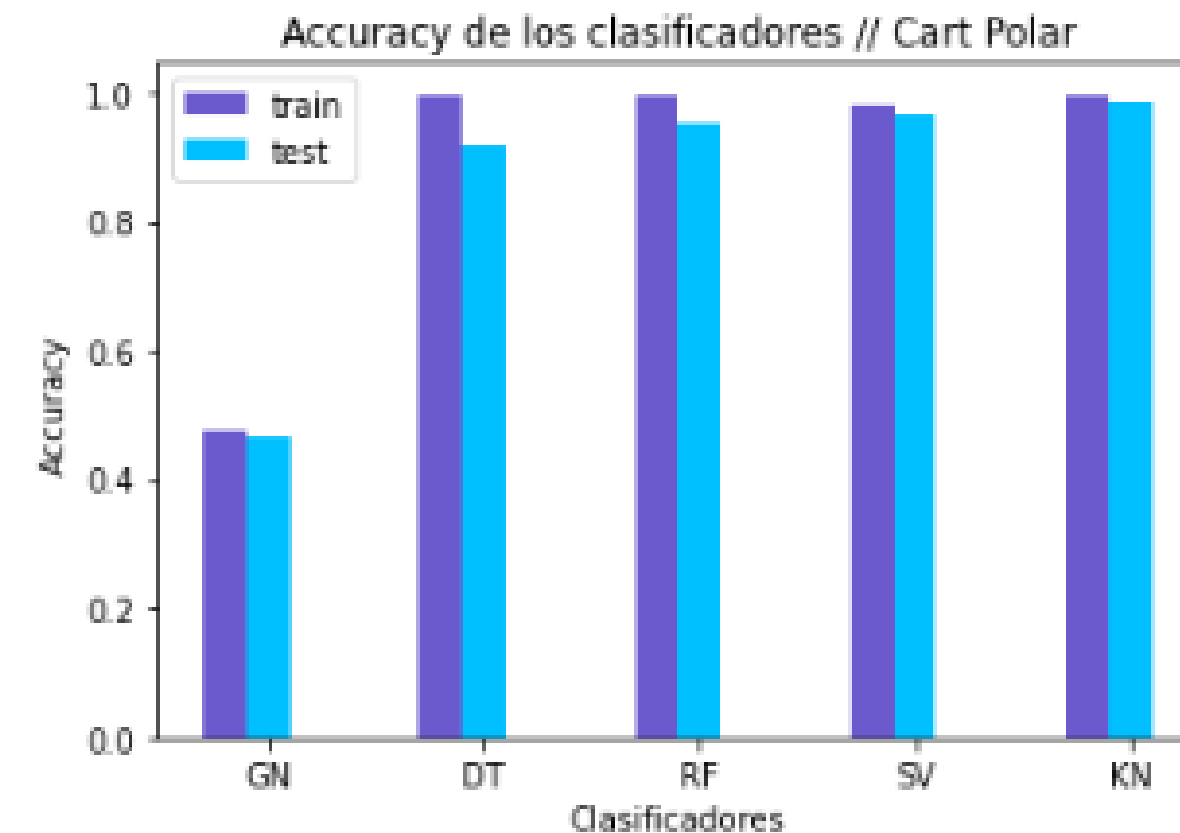
```
accuracy NB 0.332 (+/- 0.04842)
accuracy DT 0.801 (+/- 0.04713)
accuracy RF 0.871 (+/- 0.02269)
accuracy SVC 0.856 (+/- 0.04572)
accuracy KNeighbors 0.384 (+/- 0.03014)
```



06

CART POLAR

```
accuracy NB 0.452 (+/- 0.04924)
accuracy DT 0.835 (+/- 0.05306)
accuracy RF 0.906 (+/- 0.03828)
accuracy SVC 0.883 (+/- 0.03317)
accuracy KNeighbors 0.911 (+/- 0.04032)
```

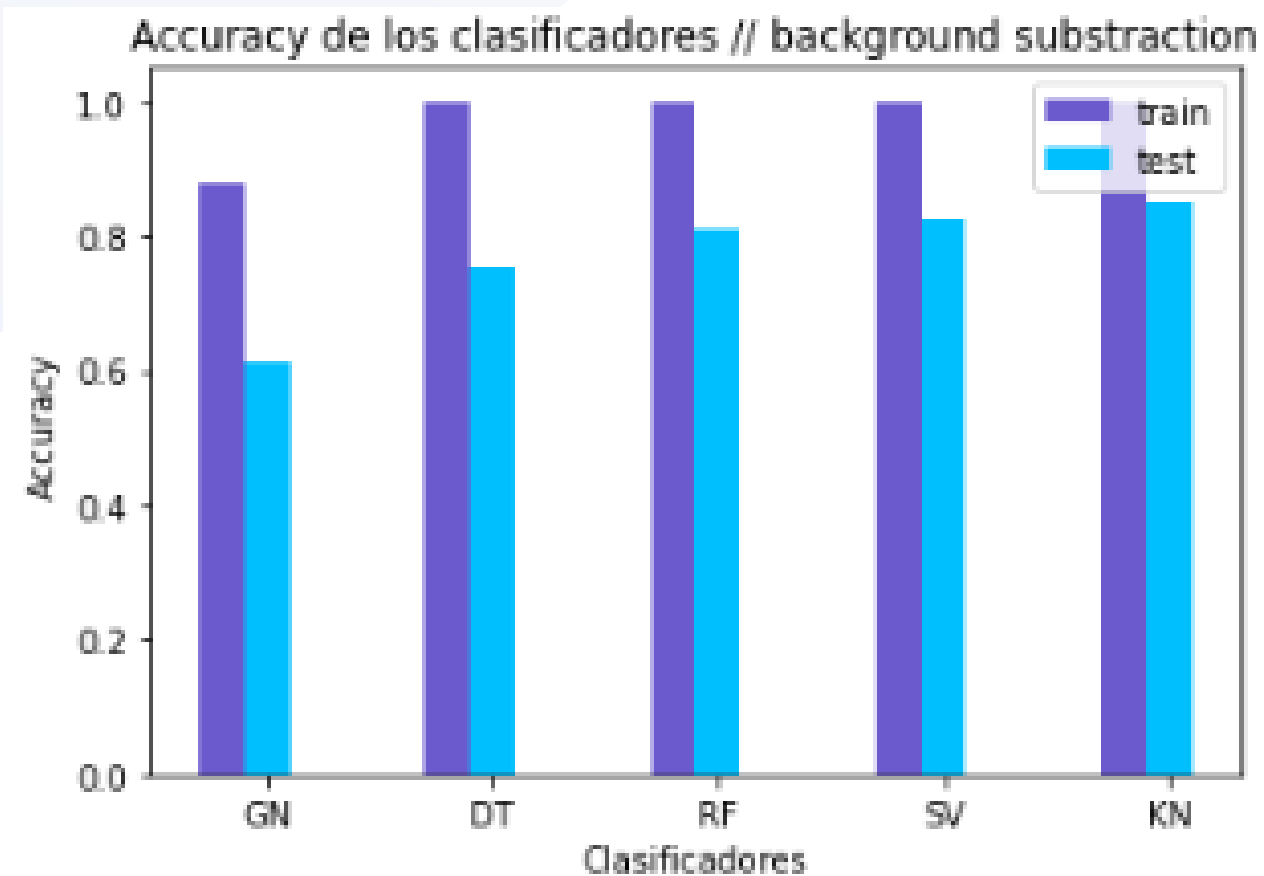


# RESULTADOS CLASIFICADORES

07

Uso del PCA en Background Substracion  
con un valor de 150 componentes

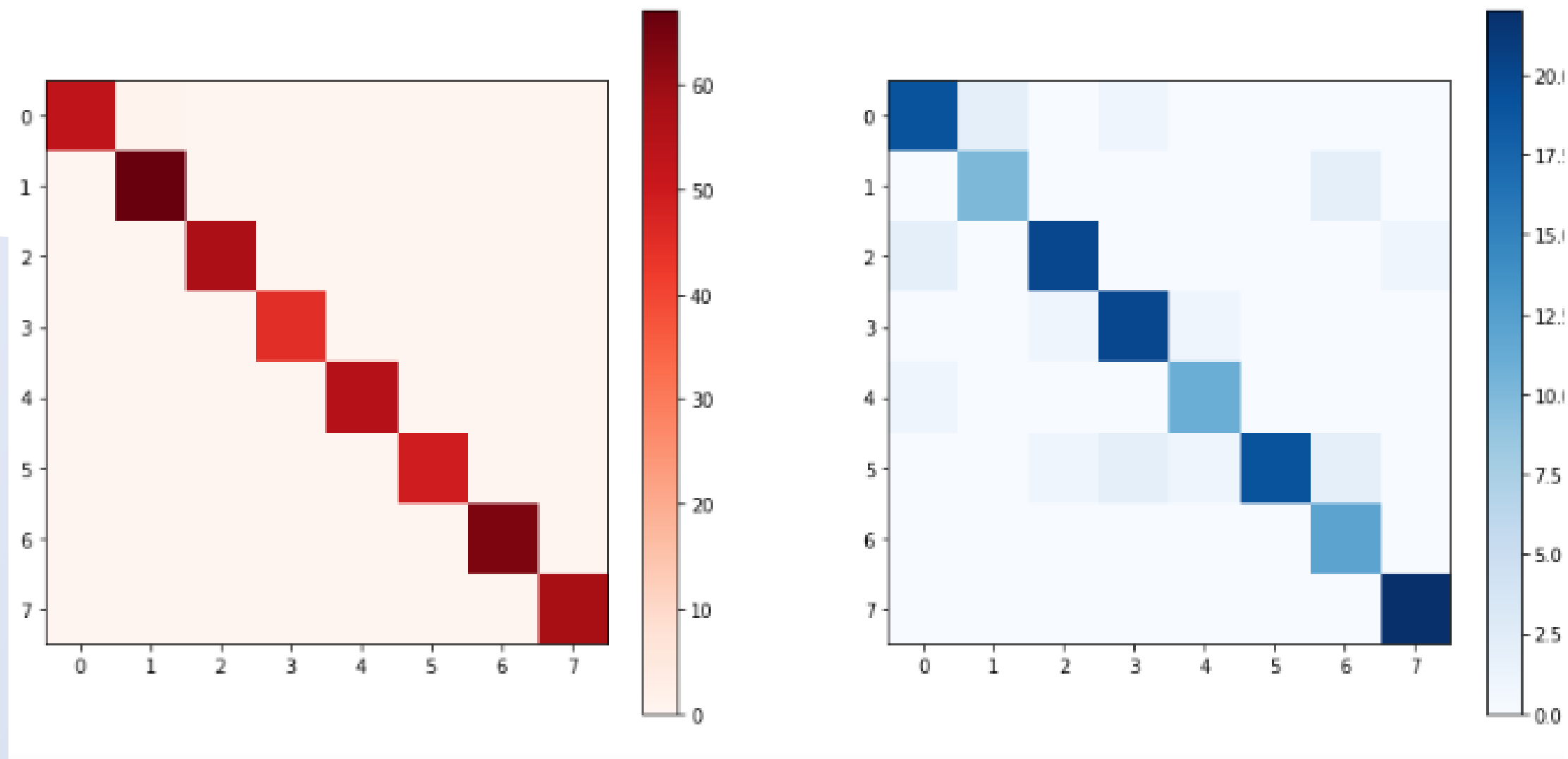
```
accuracy NB 0.730 (+/- 0.05280)  
accuracy DT 0.841 (+/- 0.04577)  
accuracy RF 0.895 (+/- 0.03758)  
accuracy SVC 0.895 (+/- 0.03021)  
accuracy KNeighbors 0.863 (+/- 0.04031)
```





# MEJOR RESULTADO CLASIFICADOR

Obtenido con el tratamiento de Cart Polar

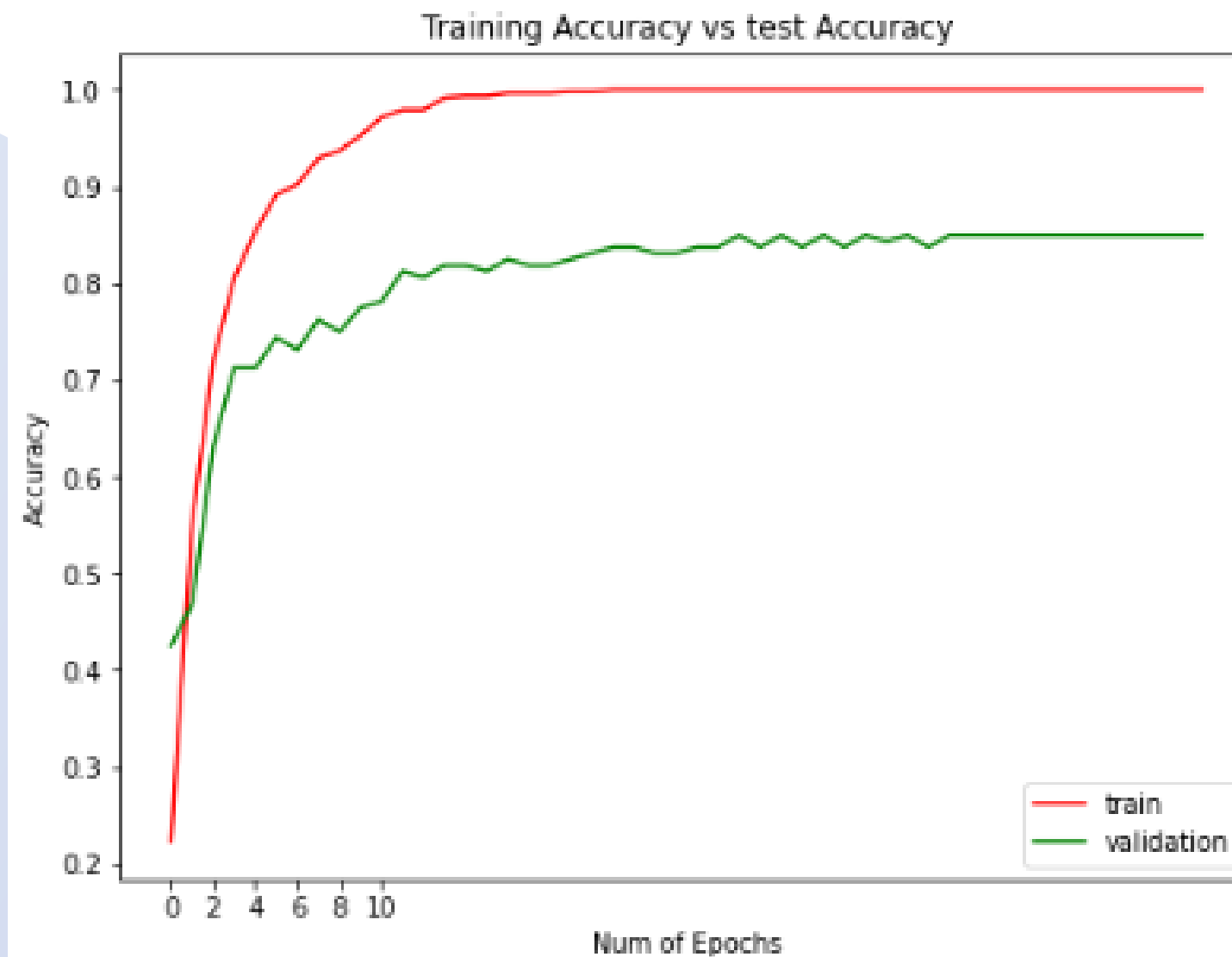


**Matriz de confusión SVC Cart Polar**  
**train 99% y test de 89%**

# MEJOR RESULTADO DNN

Obtenido con el tratamiento de background subtraction

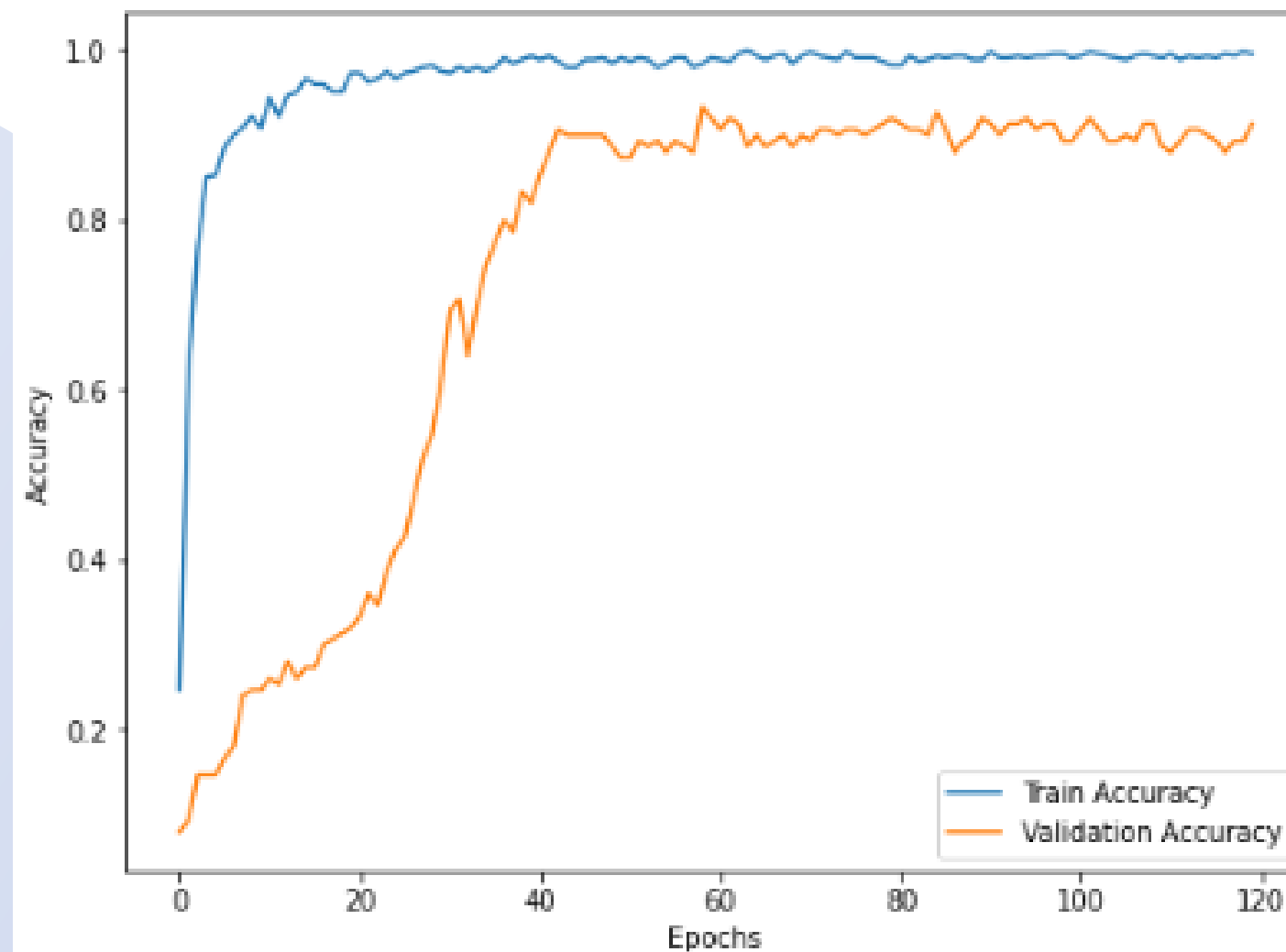
```
Epoch 48/50  
20/20 [=====] - 0s 5ms/step - loss: 0.0291 - accuracy: 1.0000 - val_loss: 0.5244 - val_accuracy: 0.8500  
Epoch 49/50  
20/20 [=====] - 0s 5ms/step - loss: 0.0284 - accuracy: 1.0000 - val_loss: 0.5287 - val_accuracy: 0.8500  
Epoch 50/50  
20/20 [=====] - 0s 5ms/step - loss: 0.0276 - accuracy: 1.0000 - val_loss: 0.5258 - val_accuracy: 0.8500
```



# MEJOR RESULTADO CNN

Obtenido con el Dataset Cart Polar

```
15/15 [=====] - 0s 8ms/step - loss: 0.1663 - accuracy: 0.9933 - val_loss: 0.4097 - val_accuracy: 0.8933  
Epoch 119/120  
15/15 [=====] - 0s 8ms/step - loss: 0.1459 - accuracy: 0.9978 - val_loss: 0.4144 - val_accuracy: 0.8933  
Epoch 120/120  
15/15 [=====] - 0s 8ms/step - loss: 0.1411 - accuracy: 0.9955 - val_loss: 0.4023 - val_accuracy: 0.9133
```





# CONCLUSIONES



Para este Dataset como se esperaba una red convolucional dio la mejor precisión debido a su proceso para hallar las mejores características de la imágenes



Los mejores resultados por parte de los tratamiento de datos fue Cart Polar por su gran clasificación con un array de pocos datos



Se evidencia que el clasificador con peor desempeño fue el Naive Gaussian debido a que los datos no tienen una distribución normal o semejante

# REFERENCIAS



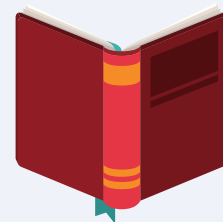
<https://hal.archives-ouvertes.fr/hal-01119640/document>



<https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>



[https://www.researchgate.net/publication/336797133\\_Word-level\\_Deep\\_Sign\\_Language\\_Recognition\\_from\\_Video\\_A\\_New\\_Large-scale\\_Dataset\\_and\\_Methods\\_Comparison](https://www.researchgate.net/publication/336797133_Word-level_Deep_Sign_Language_Recognition_from_Video_A_New_Large-scale_Dataset_and_Methods_Comparison)



<https://likegeeks.com/es/procesar-de-imagenes-en-python/>