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

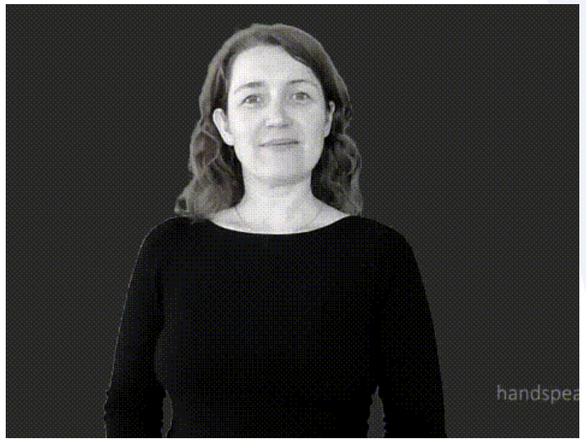
WLASL

21083 vídeos

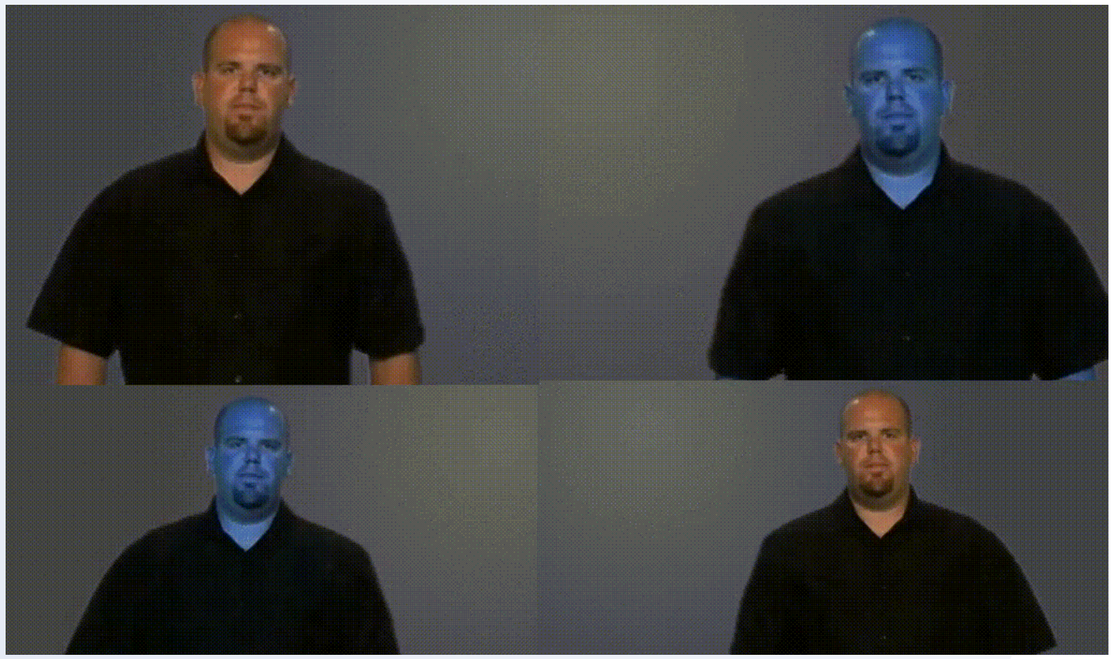
2000 clases

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

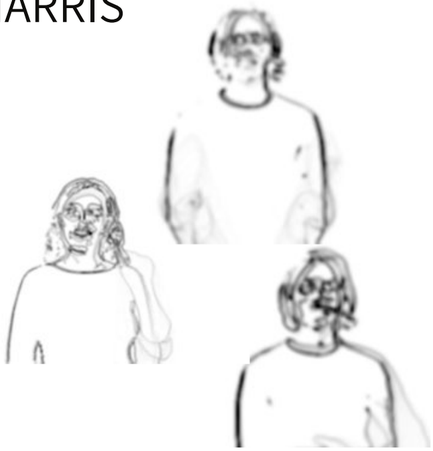
EDGES WITH OPTICAL FLOW

04

BAG OF WORDS

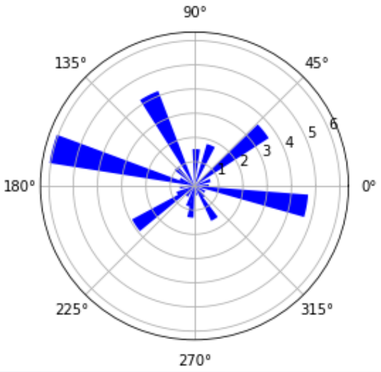
05

EDGES WITH CORNER HARRIS



06

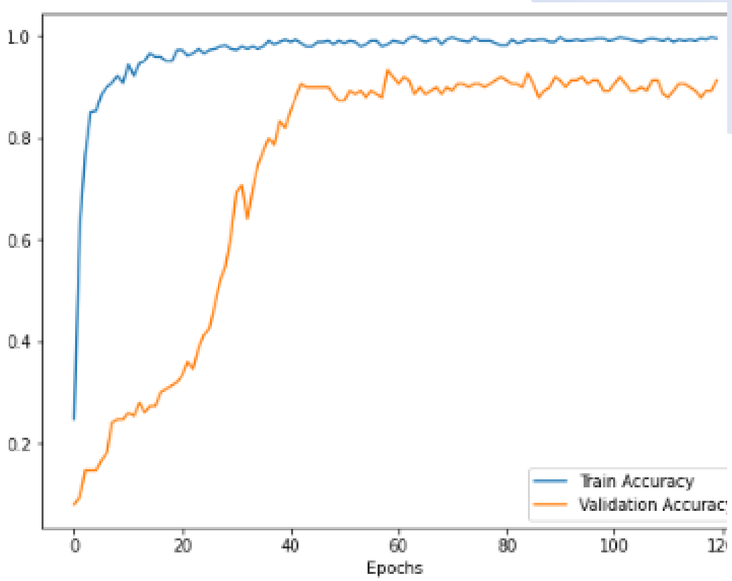
OPTICAL FLOW AND CART POLAR



MEJORES RESULTADOS

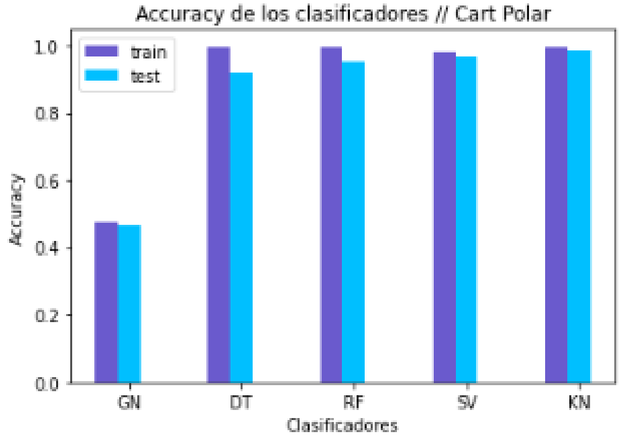
Layer (type)	Output Shape	Param #
dense_93 (Dense)	(None, 700)	14700
batch_normalization_40 (Batc	(None, 700)	2800
dense_94 (Dense)	(None, 400)	280400
batch_normalization_41 (Batc	(None, 400)	1600
dense_95 (Dense)	(None, 300)	120300
batch_normalization_42 (Batc	(None, 300)	1200
dense_96 (Dense)	(None, 100)	30100
batch_normalization_43 (Batc	(None, 100)	400
dense_97 (Dense)	(None, 8)	808

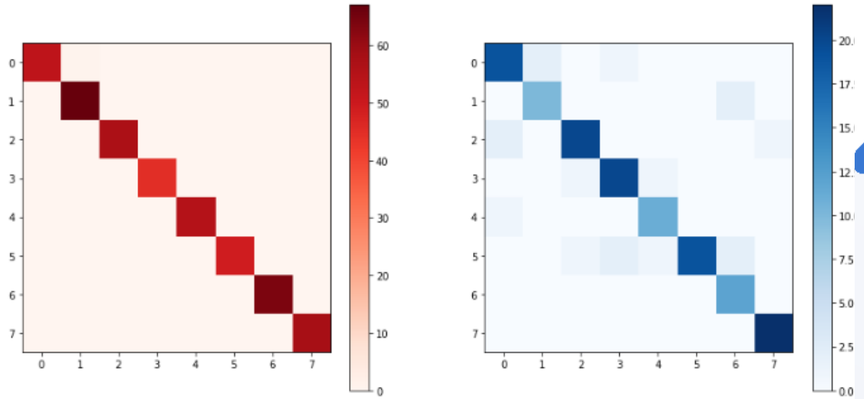
Total params: 452,308
Trainable params: 449,308
Non-trainable params: 3,000



99% train

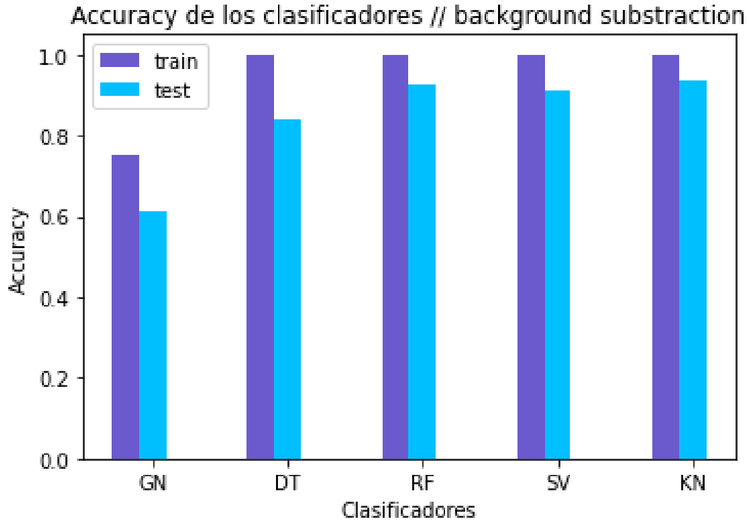
91% test

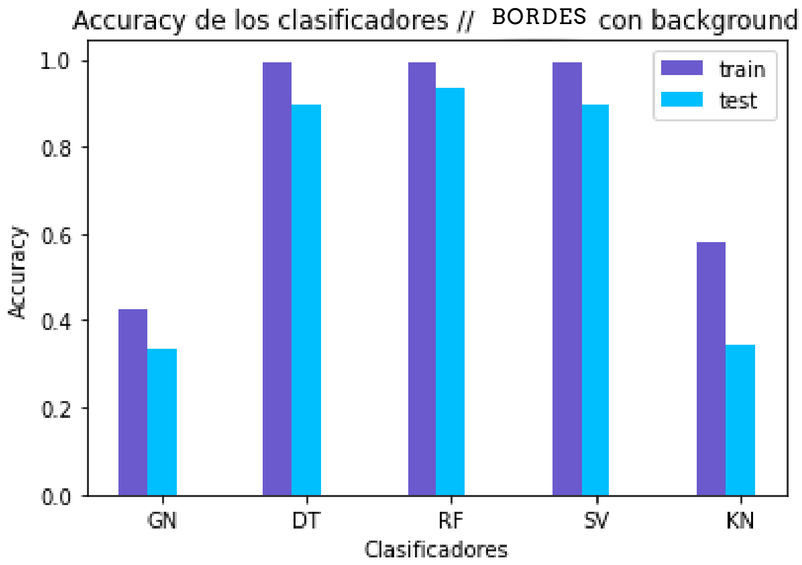


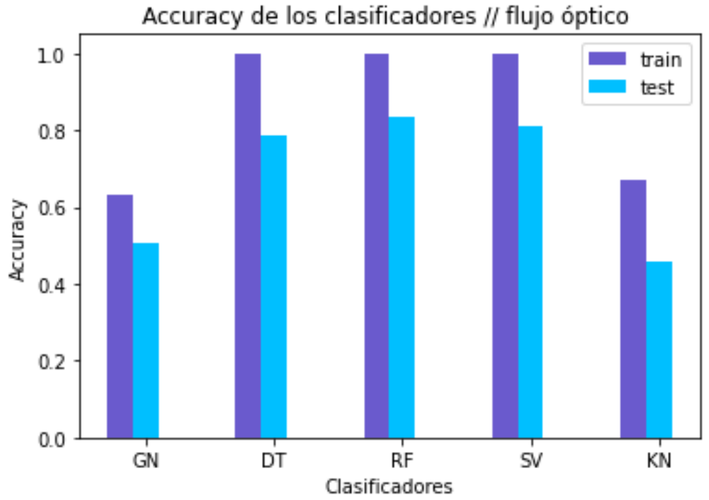


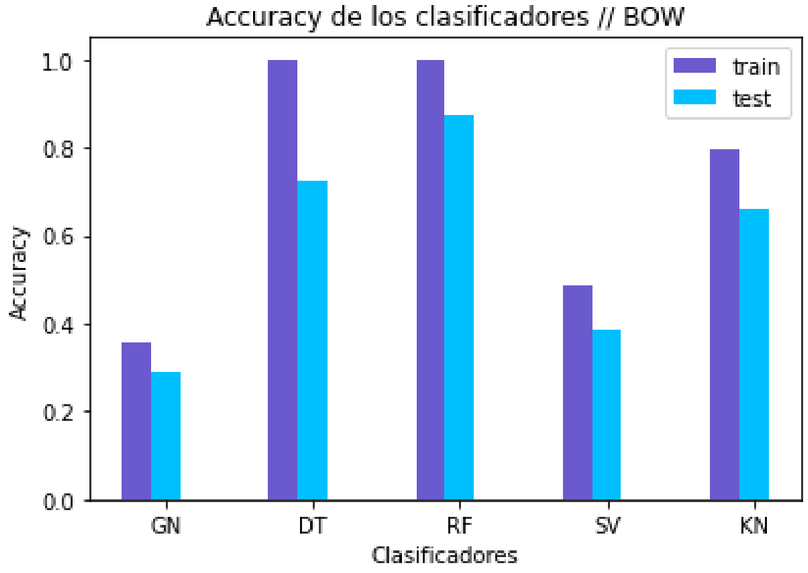
Matriz de confusión SVC Cart Polar

OTROS RESULTADOS



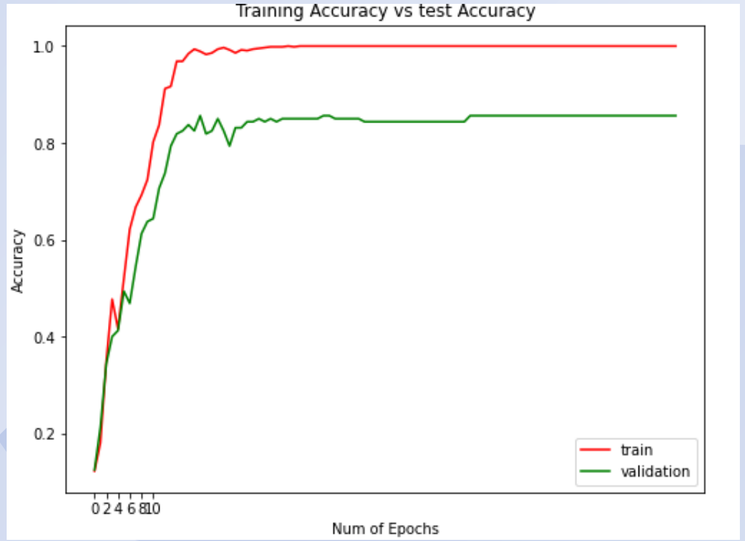




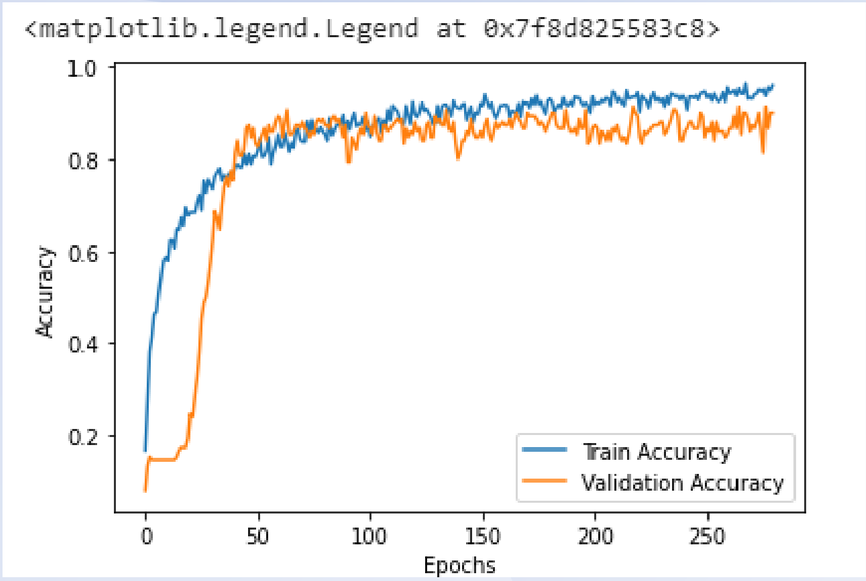


OTROS RESULTADOS

```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_45 (Conv2D)           (None, 96, 96, 64)        1664
_________________________________________________________________
conv2d_46 (Conv2D)           (None, 94, 94, 64)        36928
_________________________________________________________________
max_pooling2d_30 (MaxPooling (None, 47, 47, 64)        0
_________________________________________________________________
conv2d_47 (Conv2D)           (None, 45, 45, 64)        36928
_________________________________________________________________
max_pooling2d_31 (MaxPooling (None, 22, 22, 64)        0
_________________________________________________________________
flatten_15 (Flatten)         (None, 30976)             0
_________________________________________________________________
dense_30 (Dense)             (None, 128)               3965056
_________________________________________________________________
dense_31 (Dense)             (None, 11)                1419
=================================================================
Total params: 4,041,995
Trainable params: 4,041,995
Non-trainable params: 0
_________________________________________________________________
```

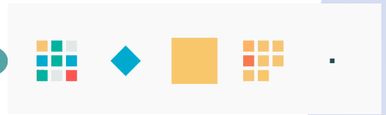


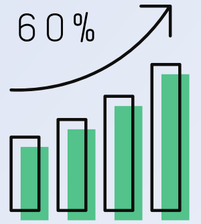

```
Model: "sequential_21"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_44 (Dense)             (None, 700)               14700
_________________________________________________________________
batch_normalization_8 (Batch (None, 700)               2800
_________________________________________________________________
dense_45 (Dense)             (None, 400)               280400
_________________________________________________________________
batch_normalization_9 (Batch (None, 400)               1600
_________________________________________________________________
dropout_6 (Dropout)          (None, 400)               0
_________________________________________________________________
dense_46 (Dense)             (None, 300)               120300
_________________________________________________________________
batch_normalization_10 (Batc (None, 300)               1200
_________________________________________________________________
dropout_7 (Dropout)          (None, 300)               0
_________________________________________________________________
dense_47 (Dense)             (None, 100)               30100
_________________________________________________________________
batch_normalization_11 (Batc (None, 100)               400
_________________________________________________________________
dropout_8 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_48 (Dense)             (None, 8)                 808
=================================================================
Total params: 452,308
Trainable params: 449,308
Non-trainable params: 3,000
_________________________________________________________________
```



PRUEBA FINAL

PROCESAMIENTO
CON RANDOM FOREST





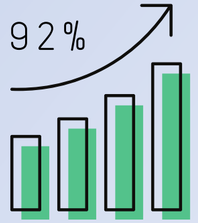

```
1 neigh.predict(Xtest)
```

```
array(['accomplish'], dtype=object)
```

```
1 neigh.predict_proba(Xtest)
```

```
array([[0.00109297, 0.00337066, 0.093046  , 0.02047726, 0.10234718
        0.16551928, 0.02163382, 0.59251283]])
```

PROCESAMIENTO
CON LA RED NEURONAL




```
1 a = np.argmax(model.predict(Xtest),axis=-1)
2 b = a[0]
3 print(elimanteDuplicated(y2)[b])
```

```
accomplish
```

```
1 model.predict(Xtest)
```

```
array([[7.3256786e-05, 4.1758353e-06, 1.9178513e-02, 3.3226101e-05,
        4.9589198e-05, 4.7792503e-07, 5.8723256e-02, 9.2193753e-01]],
      dtype=float32)
```

CONCLUSIONES

- El lenguaje de señas requiere de varías muestras por clase debido a que muchas de sus palabras tienen movimientos similares
- El background da un mejor resultado del movimiento, debido a que las palabras a clasificar tienen un gesto muy similar entre ellas.
- Los histogramas polares son un muy buen descriptor y reduce cada vídeo a un mínimo de valores.
- Se requirió calcular el flujo óptico con la función canny de opencv para cada imagen a pasar y luego normalizar cada histograma además finalmente normalizar la suma de todos por vídeo, para obtener unos buenos resultados con el método de histogramas.

REFERENCIAS

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

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

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/