



PROYECTO DL

Fredy Velasquez
Angel Higueros

DESCRIPCIÓN DEL PROBLEMA

El acceso al contenido multimedia es fundamental, pero la **comunidad con discapacidad auditiva aún enfrenta barreras.**

Aunque el lenguaje de señas ha sido esencial, su **implementación depende de intérpretes humanos**, limitando la autonomía.

Abordar esto es clave para **mejorar la accesibilidad y la disponibilidad de contenido inclusivo.**

PROPUESTA DE SOLUCIÓN

Desarrollo de un modelo de aprendizaje profundo especializado en procesar videos de habla clara para traducir movimientos labiales a texto con alta precisión en entornos controlados.

HERRAMIENTAS APLICADAS

1 TensorFlow



2 Numpy



3



4

Imageio

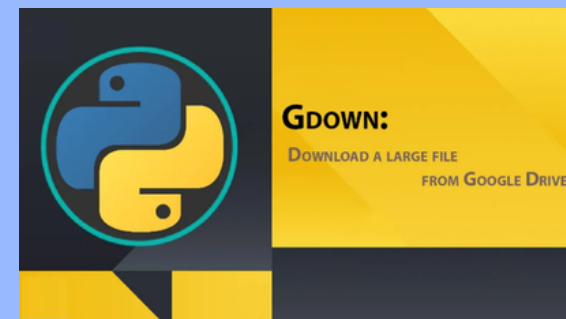


imageio

5 OpenCV



6 Gdown



7 Typing



ASPECTOS A TOMAR EN CONSIDERACIÓN

MODELO

1. **Sequential():** Se utiliza para definir un modelo secuencial en Keras, donde las capas se apilan en un orden específico.
2. **Capas Conv3D:** Estas capas utilizan filtros convolucionales tridimensionales para procesar los datos. Se define una capa con 128 filtros, tamaño de kernel de 3, y un shape de entrada específico, utilizando 'padding' para mantener el tamaño de salida.
3. **Activación 'relu':** Es una función de activación que introduce no linealidad, permitiendo que el modelo maneje mejor las complejidades de los datos.
4. **Capas MaxPool3D:** Estas capas reducen las dimensiones espaciales del volumen de entrada, disminuyendo la cantidad de parámetros y computación y ayudando a prevenir el sobreajuste.
5. **TimeDistributed(Flatten()):** Esta operación transforma los datos multidimensionales en un vector unidimensional para cada segmento temporal.
6. **Capas Bidirectional(LSTM):** Permiten que el modelo procese datos en ambas direcciones, lo que es útil para capturar patrones en secuencias temporales o series de imágenes.
7. **Dropout(.5):** Es una técnica de regularización que "apaga" unidades aleatorias durante el entrenamiento para evitar el sobreajuste.
8. **Dense:** Una capa densa o completamente conectada que, en este caso, se utiliza para la capa de salida con una activación 'softmax' para obtener probabilidades.

RESULTADOS DEL ENTRENAMIENTO



```
Epoch 1/100
1/1 [=====] - 2s 2s/step
Original: lay blue in x five soon
Prediction: le e e n no
~~~~~
1/1 [=====] - 2s 2s/step
Original: bin white in g two please
Prediction: le e e e n n
~~~~~
450/450 [=====] - 646s 1s/step - loss: 84.3118 - val_loss: 70.8771 - lr: 1.0e-04
```

```
Epoch 40/100
1/1 [=====] - 0s 82ms/step
Original: place red in v six please
Prediction: place red in six please
~~~~~
1/1 [=====] - 0s 82ms/step
Original: lay red in y three again
Prediction: lay red in thre again
~~~~~
450/450 [=====] - 451s 1s/step - loss: 7.6221 - val_loss: 4.7672 - lr: 3.7e-05
```

```
Epoch 80/100
1/1 [=====] - 0s 83ms/step
Original: bin white at t four please
Prediction: bin white at t four please
~~~~~
1/1 [=====] - 0s 83ms/step
Original: bin white in g three again
Prediction: bin white in g three again
~~~~~
450/450 [=====] - 450s 1s/step - loss: 7.9465 - val_loss: 5.549 - lr: 4.1e-05
```


Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]

~~~~~ REAL TEXT

[<tf.Tensor: shape=(), dtype=string, numpy=b'place red at c six now'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in

~~~~~ PREDICTIONS

[<tf.Tensor: shape=(), dtype=string, numpy=b'place red at c six now'>]
```


Texto real

```
print('~'*100, 'REAL TEXT')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in [sample[1]]]

~~~~~ REAL TEXT

[<tf.Tensor: shape=(), dtype=string, numpy=b'set blue in a two please'>]
```

Predicción

```
print('~'*100, 'PREDICTIONS')
[tf.strings.reduce_join([num_to_char(word) for word in sentence]) for sentence in

~~~~~ PREDICTIONS

[<tf.Tensor: shape=(), dtype=string, numpy=b'set blue in a two please'>]
```

CONCLUSIONES



1

El avance tecnológico, aplicado de manera adecuada, puede generar herramientas que mejoren la vida de personas con capacidades diferentes.

2

El modelo cumplió con las expectativas al "leer" los labios en videos, predecir mensajes y generar texto preciso.

3

Texto del párrafo

Métricas como pérdida, pérdida de evaluación y tasa de aprendizaje indican un desempeño excelente del modelo, validando su correcta construcción.

4

Con más recursos y un conjunto de datos extenso, el modelo podría mejorar su rendimiento para aplicaciones cotidianas.

5

Decisiones correctas en la construcción del modelo, evidenciadas por resultados esperados y un cumplimiento efectivo de la tarea de lectura labial.