

Monkeypox Clasificación

Gysselis Vásquez
César Hurtado
Wilmer Farfán



Contenido

01.

Introducción



02.

Objetivos



03.

Dataset



04.

Machine Learning



05.

Deep Learning



06.

Resultados y conclusiones



Introducción



Objetivos

- ✓ Experimentar con estimadores de Machine learning que sean capaces de clasificar imágenes de lesiones cutáneas en dos categorías: viruela del mono y otras lesiones cutáneas.
- ✓ Desarrollar modelos de aprendizaje profundo que puedan identificar con precisión las características distintivas de la viruela del mono para su clasificación.
- ✓ Realizar un análisis de los resultados e identificar cuál de todos estos algoritmos tiene el desempeño más relevante.





Dataset



Características

Cuenta con dos clases, las cuales se dividen en dos carpetas:

Monkeypox (1168 imágenes): casos correspondientes a viruela del mono.

Others (1439 imágenes): lesiones cutáneas relacionadas a otros tipos de viruela.

Formato: (224, 224, 3)



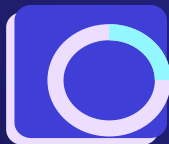
+ ▶ Pre procesamiento

- ✓ Crear dos arreglos, uno donde se guardan las imágenes y el otro donde se almacenan sus respectivas etiquetas.
- ✓ Redimensionar las imágenes para optimizar el consumo de recursos y mejores resultados:
 $(224,224,3) \rightarrow (128,128,3)$
- ✓ Normalizar los datos:
 $(dataset)/255$
- ✓ Permutar los datos
- ✓ Particionar los datos de entrenamiento y testeo

Machine Learning



Estimadores



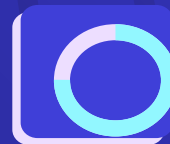
Gaussian NB

Parámetros por defecto



Decision Tree

max_depth: 20

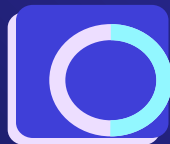


Random Forest

n_estimators: 100
max_depth: ninguno



SVC poly



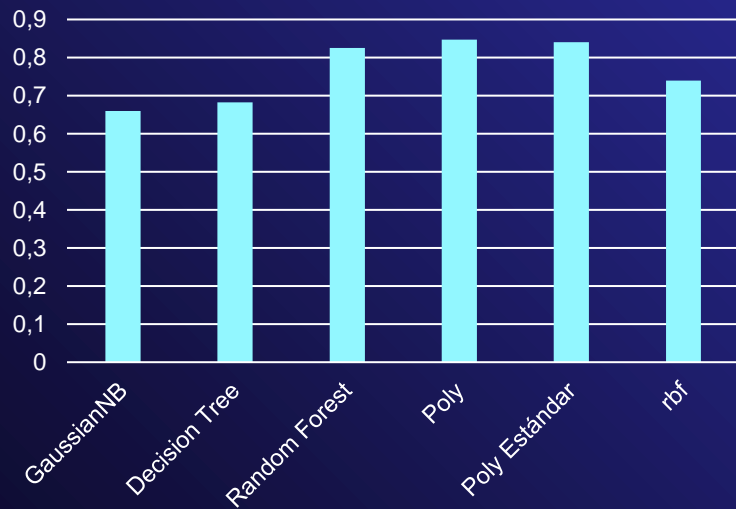
SVC poly standar



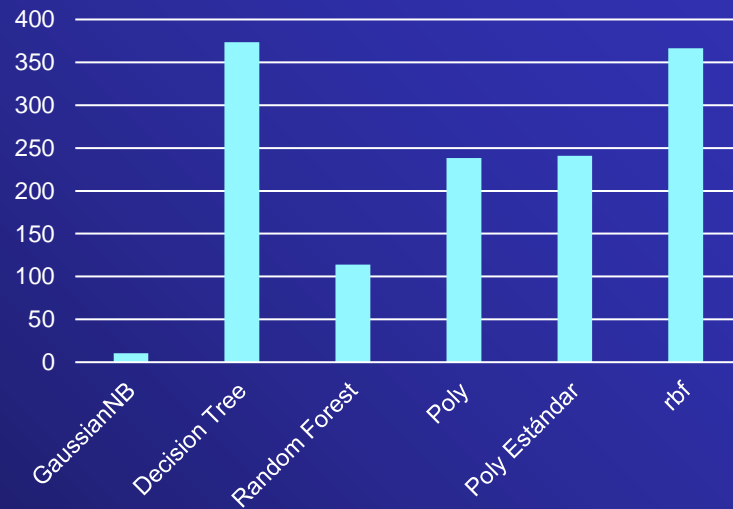
SVC rbf

Resultados

Accuracy



Tiempo de ejecución





Deep Learning

Modelos



DNN con

optimizador Adam

```
Layer input_shape=[128, 128, 3])  
Activación de capas ocultas: ReLu  
1:layers.Dense(512)  
2:layers.Dense(1024)  
3:layers.Dense(2048)  
4:layers.Dense(4096)
```

```
output layers.Dense(2, activation=tf.nn.sigmoid)
```

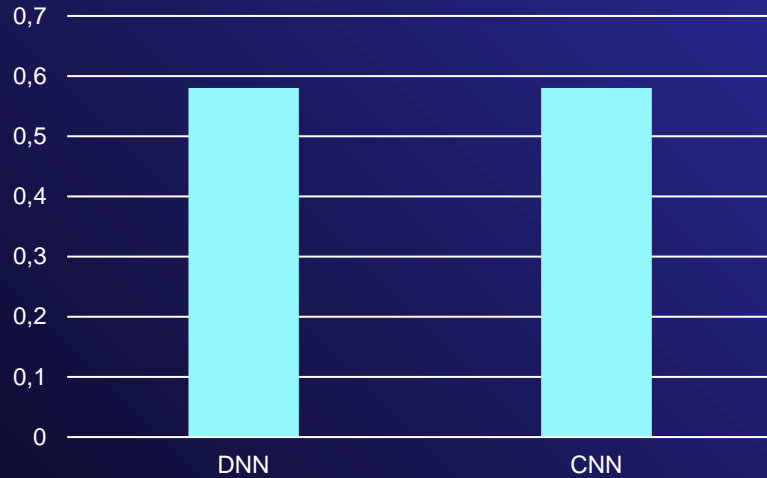


CNN

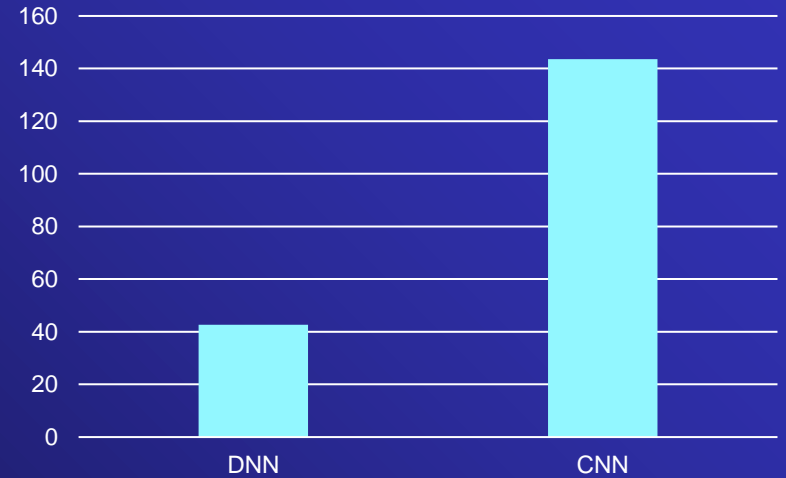
```
Layers input_shape=(128,128,3),  
      layers.AveragePooling2D(2, 2),  
layers.Conv2D(256, (3,3), activation='relu'),  
      layers.AveragePooling2D(2,2),  
layers.Conv2D(512, (3,3), activation='relu'),  
      layers.AveragePooling2D(2,2),  
      layers.Dropout(0.4),  
      layers.Flatten(),  
layers.Dense(1024, activation='relu'),  
      Dense(2, activation='sigmoid')
```

Resultados

Accuracy



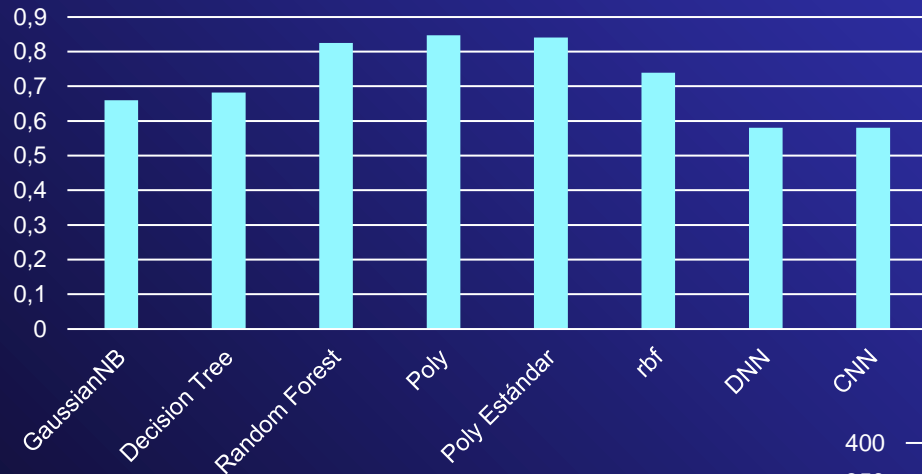
Tiempo de ejecución



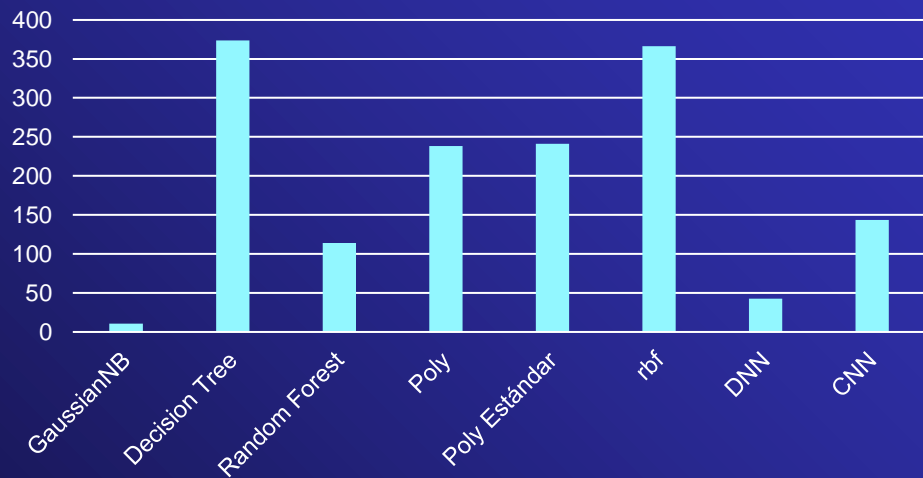
Resultados finales



Accuracy



Tiempo de ejecución





Notebook



Conclusiones



iGracias!

