

Clasificado de imágenes de ropa basado en IA

1st Alejandro Giraldo Cardona
Facultad de Ingeniería
Universidad Jorge Tadeo Lozano
Bogotá, Colombia
alejandro.giraldoc@utadeo.edu.co

2nd Johan Sebastián Parra Maldonado
Facultad de Ingeniería
Universidad Jorge Tadeo Lozano
Bogotá, Colombia
sebastian.parram@utadeo.edu.co

I. INTRODUCCIÓN

De acuerdo a las clases vistas, se elaborará un proyecto aplicando Machine Learning. Este proyecto tiene como objetivo implementar algoritmos para la construcción de un clasificador de imágenes de ropa basado en la extracción de las características de la imagen (Píxeles). El conjunto de datos utilizado consiste en: más de sesenta mil imágenes, las cuales tienen como datos los píxeles de cada una. Se hará un estudio de estos y se aplicará un modelo, el cual nos dará como resultado información de interés. Se aplicarán conocimientos aprendidos en los talleres realizados, como NumPy, Matplotlib para visualización de gráficas y Pandas. Como ya sabemos, Machine Learning es una rama de la inteligencia artificial que permite que las máquinas aprendan sin ser expresamente programadas para ello. Una habilidad indispensable para hacer sistemas capaces de identificar patrones entre los datos para hacer predicciones.

En esta segunda entrega se desarrollará el pre-procesamiento de los datos y el entrenamiento del mismo con los resultados correspondientes. Se realizará la carga del archivo que contiene los datos al colab, luego, se graficará y se estudiará sobre los tipos de datos que se obtienen y se quieren estudiar. Se creará una variable que contendrá el modelo de datos que se analizarán y finalmente se desarrolla un entrenamiento del modelo, el tipo de entrenamiento se escoge de acuerdo a la exigencia y atributos del modelo.

Palabras clave:

Machine Learning, fit, algoritmos, píxeles, Pandas, Numpy y fit.

II. DESCRIPCIÓN DEL MÉTODO

Se utilizará Keras de TensorFlow para realizar el modelo de los datos. Debido a que mediante la construcción del modelo con Keras, permitirá el uso de un set de datos de tres dimensiones, lo cual es lo requerido para las imágenes. Keras es una biblioteca que funciona a nivel de modelo: proporciona bloques modulares sobre los que se pueden desarrollar modelos complejos de aprendizaje profundo. A diferencia de los frameworks, este software de código abierto no se utiliza para operaciones sencillas de bajo nivel, sino que utiliza las bibliotecas de los frameworks de aprendizaje automático vinculadas, que en cierto modo actúan como un motor de backend para Keras.

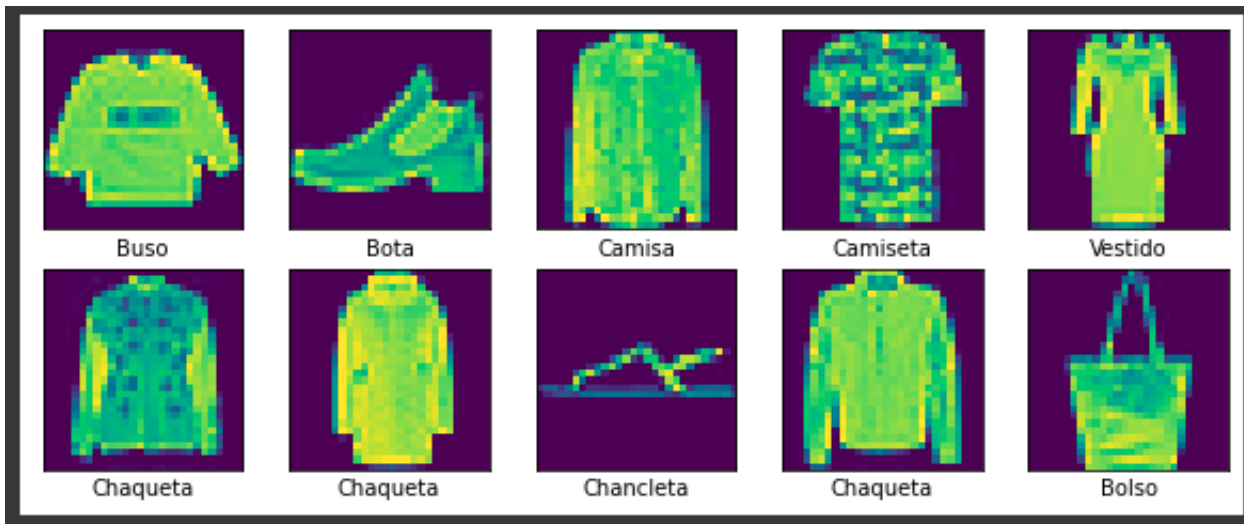
III. VISUALIZACIÓN Y ANÁLISIS DE LOS DATOS.

La visualización de datos es la representación gráfica de información y datos. Al utilizar elementos visuales como en este caso gráficos, las herramientas de visualización de datos proporcionan una manera accesible de ver y comprender tendencias, valores atípicos y patrones en los datos.

Para ello, obtuvimos las características que nos permiten diferenciar entre las distintas clases de imágenes almacenadas en el DataFrame con más de 60.000 imágenes. Las imágenes fueron convertidas a 28x28 en un arreglo de NumPy, con valores de píxel que varían de 0 a 255, por lo que permitió la forma de la imagen de cada una de las prendas.

Luego de implementar el modelo de los datos a estudiar, por medio de Keras de TensorFlow realizaremos el entrenamiento de los datos. Para comenzar a entrenar, se utilizó el método `model.fit` el cual ajusta el modelo a el set de datos de entrenamiento. Finalmente, con `score()` podremos obtener una estimación de la capacidad de acierto de nuestro modelo sobre los datos de trabajo.

Entrenar el modelo de red neuronal requiere de una serie de pasos, los cuales son: definir los datos, por consiguiente, el modelo aprende a asociar imágenes y etiquetas, por último, se le pide al modelo que haga predicciones a partir de los datos entregados. Como se aprecia en la **figura 2**, a medida que el modelo entrena, la pérdida y exactitud son desplegadas. Este modelo alcanza una exactitud de 0.83 (o 83 %) sobre el set de datos de entrenamiento.



(a) Prendas de ropa

```
[13] #Entrenamiento de los datos
      model.fit(image, labels, epochs=10)
```

```
Epoch 1/10
1875/1875 [=====] - 4s 2ms/step - loss: 10.8275 - accuracy: 0.6677
Epoch 2/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.7076 - accuracy: 0.7573
Epoch 3/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.6269 - accuracy: 0.7729
Epoch 4/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.5673 - accuracy: 0.8020
Epoch 5/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.5263 - accuracy: 0.8143
Epoch 6/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.5140 - accuracy: 0.8211
Epoch 7/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.5001 - accuracy: 0.8229
Epoch 8/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.4848 - accuracy: 0.8311
Epoch 9/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.4821 - accuracy: 0.8341
Epoch 10/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.4735 - accuracy: 0.8388
<tensorflow.python.keras.callbacks.History at 0x7f922922fa90>
```

(b) Entrenamiento del set de datos

IV. CONCLUSIONES

1) De acuerdo a los datos consultados y manipulados, se evidenció que existen varios tipos de prendas de vestir. Este vestuario está dividido en 9 tipos de prendas diferentes, como vestidos, tenis deportivos, camisetas, pantalones etc.

2) Se tuvo que crear un arreglo independiente para almacenar cada fila del archivo, este arreglo representaría cada prenda de vestir. Como los datos están dados en píxeles, el arreglo que se creó para representar cada prenda, se convierte en una matriz de 28x28.

3) Para representar los píxeles y mostrar las figuras de las prendas de vestir, se utilizó el método figure de matplotlib, este nos permite visualizar los píxeles en una imagen, ayudándonos a identificar el tipo de figura que se quiere representar.

4) Se observó que, en el entrenamiento del modelo, tuvo una eficacia de hasta el 83 %, siendo así un dato significativo en el momento que el modelo haga sus predicciones ya que le permitirá ser más preciso y obtener el resultado esperado.

5) Con este tipo de datos se puede utilizar keras, que nos permite encontrar un mododelo de datos por medio de TensorFlow y Machine learning, con esto se logrará identificar que tipo de modelo se adapta mejor a nuestro entrenamiento y y así lograr mas precisión a la hora de realizar predicciones a nuestro archivo de datos.

Repositorio de git

<https://github.com/sebastianparra0205/Proyecto-de-curso-IA>

REFERENCIAS

- [1] "Te contamos qué es el 'machine learning' y cómo funciona", BBVA NOTICIAS, 2021. [Online]. Available: <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/>. [Accessed: 16- Mar- 2021].
- [2] Luis Llamas. 2021. Machine learning con TensorFlow y Keras en Python. [online] Available at: <https://www.luisllamas.es/machine-learning-con-tensorflow-y-keras-en-python/>. [Accessed 4 April 2021].
- [3] J. Heras, Regresión Lineal: teoría y ejemplos en Python - IArtificial.net", IArtificial.net, 2021. [Online]. Available: <https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python>. [Accessed: 04- Apr- 2021].