Fashion Mnist

Pasos para su despliegue:

-*Tener python y django instalados*: **py -m pip install Django**

-*Instalar los paquetes necesarios*: **pip install tensorflow**
En teoría este instala todas las dependencias que necesita y se usa en el proyecto.

-*Clonar el respositorio desde la cmd*: **git clone**
https://github.com/macallejonieszaidinvergeles/keras-fashion.git

-*Acceder a esta carpeta*: **cd keras-fashion**

-*Acceder a la carpeta del proyecto*: **cd webFashion**

-*Activar entorno de django*: **py manage.py runserver**

-*Acceder a la siguiente url*: **http://127.0.0.1:8000/inicio/**

-Subir una foto, y visualizar la predicción realizada, que se mostrará en la misma página, al final, con el nombre de lo que corresponde este.

Pasos del proyecto

Una vez cargada la foto, y enviada por el formulario, esta se guarda localmente en una carpeta llamada media, que hemos creado y definido en los ajustes de django

```
128
129  MEDIA_URL = '/media/'
130
131  MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

Después es procesada, donde se convierte a 28x28, que es el formato con el que trabaja el dataset. Continuamos con conversión a un array de 1D y su normalización, dividiendo estos valores entre 255.Una vez que nuestra imagen esté lista para el proceso, se continúa con el proceso de predicción.

Se carga el modelo guardado localmente, descargado de colab. Haremos un predict en base a esta imagen.

```
print(y_pred)
prediction = tf.argmax(y_pred, axis=-1).numpy()
print("1a prediccion es:",prediction)
print("0: T-shirt/top 1: Trouser 2: Pullover 3: Dress 4: Coat 5: Sandal 6: Shirt 7: Sneaker 8: Bag 9: \n Ankle boot /n0: camiseta/top 1: pantalón 2: jersey 3: vestido

dict_items = {0: "camiseta/top" ,1: "pantalón", 2: "jersey", 3: "vestido", 4: "abrigo", 5: "sandalia",
 6: "camisa", 7: "tenis", 8: "bolso", 9: "botín"}

prediction = str(prediction)
prediction = int(prediction[1:2])

for item in dict_items:
    if prediction == item:
        result = dict_items[item]
print("result:",result)
```

Una vez que hagamos la predicción nos devolverá un array numérico. Tras un proceso, obtenemos la cadena a la que corresponde este valor, que será lo que devolveremos a la vista. Junto a la imagen.

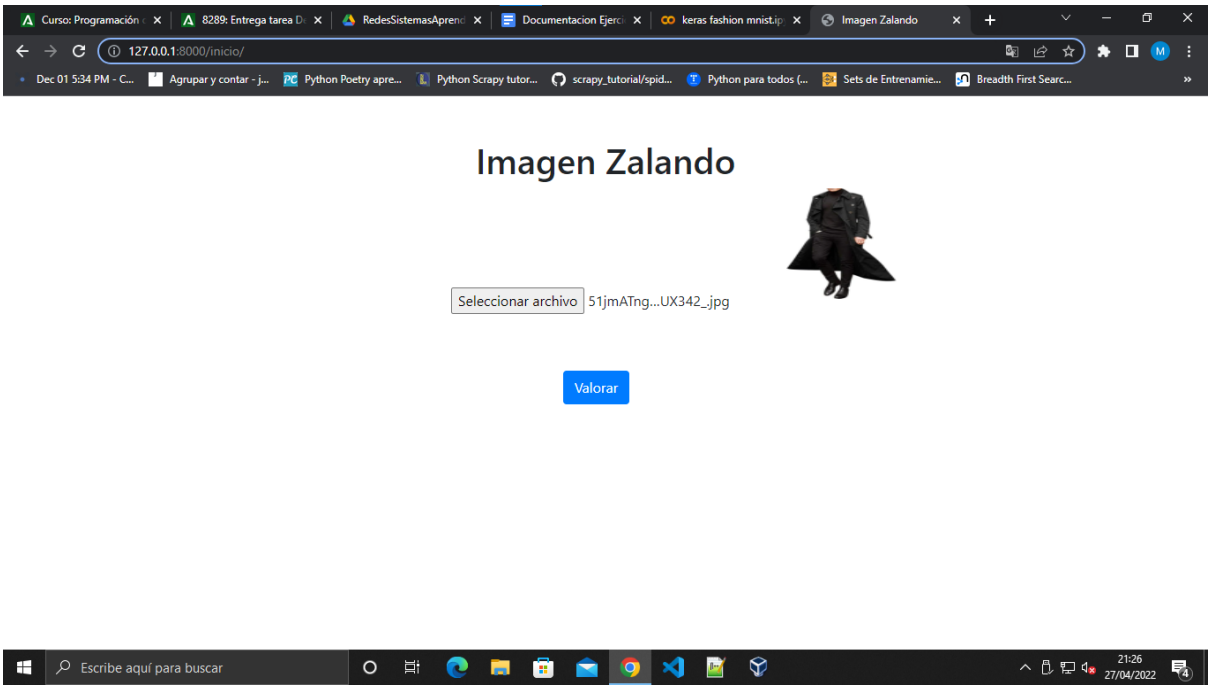
```
return render(request, "inicio.html",{"prediction":result,'file_url': file_url[1:]})
```

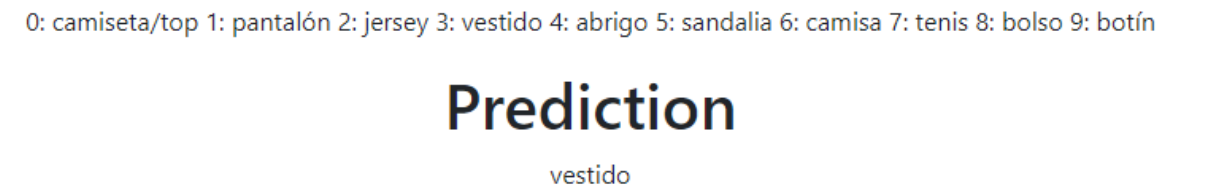
Evaluación para 4 casos:

Inicializ. de pesos	Nº de neuronas/ capa intermedia	Función de activación por cada capa	Optimizador empleado	Función de pérdida utilizada	Precisión alcanzada de 0 a 1
inic. de Xavier	64	keras.layers.Dense(64, activation='relu'), keras.layers.Dense(10, activation='softmax')	rmsprop	sparse_categorical_crossentropy	0.87
inic. de Xavier	128	keras.layers.Dense(128, activation='relu'), keras.layers.Dense(10, activation='softmax')	adam	sparse_categorical_crossentropy	0.88
inic. de Xavier	256	keras.layers.Dense(256, activation='sigmoid'), keras.layers.Dense(10, activation='softmax')	rmsprop	sparse_categorical_crossentropy	0.87

inic. de Xavier	512	ras.layers.Dense(512, activation='sigmoid'), keras.layers.Dense(10, activation='softmax')	adam	sparse_categorical_crossentropy	0.88

Pruebas





Seleccionar archivo chanclas-...-35543.jpg

Valorar

Valorar

0: camiseta/top 1: pantalón 2: jersey 3: vestido 4: abrigo 5: sandalia 6: camisa 7: tenis 8: bolso 9: botín

Prediction

sandalia

Conclusión

Es una buena herramienta para clasificar imágenes, ya que tiene mucha potencia, pero siempre que no se cambie mucho el patrón de imagen, ya que si ponemos por ejemplo una foto de una persona, o que salgan partes del cuerpo, puede fallar y detectar otros valores.

Como podría ser unas chanclas con pies, donde se ve mucho los pies, que detectaría como un bolso en este caso.