In [1]:

```
## Import all important libraries
from tensorflow.keras.applications.imagenet_utils import preprocess_input
from tensorflow.keras.applications import ResNet50, imagenet_utils
from tensorflow.keras.preprocessing.image import img_to_array, load_img, ImageDataGenerat
or
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, Dropout, MaxPooling2D, Flatten, Dense, Batch
Normalization, Input, \
                                    LSTM, Embedding, Input, TimeDistributed, Bidirection
al, Activation, RepeatVector, Concatenate
from keras.preprocessing.sequence import pad_sequences


import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from glob import glob
import cv2
import os
```

In [2]:

```
## Link a colab with google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:

```
cd /content/drive/MyDrive/assignment
```

```
/content/drive/MyDrive/catalogue assignment
```

In [4]:

```
pwd
```

Out[4]:

```
'/content/drive/MyDrive/catalogue assignment'
```

In [5]:

```
## Read the file
df = pd.read_excel('dataset.xlsx')
df.head()
```

Out[5]:

	Title	Description	Material	Pattern	Neckline	Image_Path	Image_Path.1	Unnamed: 7
0	Peach Poly Crepe jumpsuit	This stylish foil print kurta from janasya is ...	Crepe	Printed	Round Neck	/images/pic_0.jpg	/images/pic_0.jpg	pic_0.jpg
1	Light Brown Bias Yoke Checks Top	This check pattern top by Work Label is crafte...	Cotton	Checks	Round Neck	/images/pic_1.jpg	/images/pic_1.jpg	pic_1.jpg
2	Off White Geometric Straight Cotton Dobby Top ...	Featuring elegant printed details, this off wh...	Viscose	Checks	Round Neck	/images/pic_2.jpg	/images/pic_2.jpg	NaN

	Title	Description	Material	Pattern	Neckline	Image_Path	Image_Path.1	Unnamed: 9
3	Blue Me Away Cape Top	Add an extra dose of style to your casual ward...	Polyester	Solid/Plain	V-Neck	/images/pic_3.jpg	/images/pic_3.jpg	NaN
4	Yellow On A High Gown	Yellow polyester georgette maxi dress. Polyest...	Polyester	Solid/Plain	V-Neck	/images/pic_4.jpg	/images/pic_4.jpg	NaN

In [6]:

```
## Store the image path and description in list
path = list(df['Image_Path'])
Iamge_Description = list(df["Description"])
```

Image Preprocessing

In [7]:

```
## Edit the image path
imagePath = []
for i in path:
  ip = "." + i
  imagePath.append(ip)
```

In [11]:

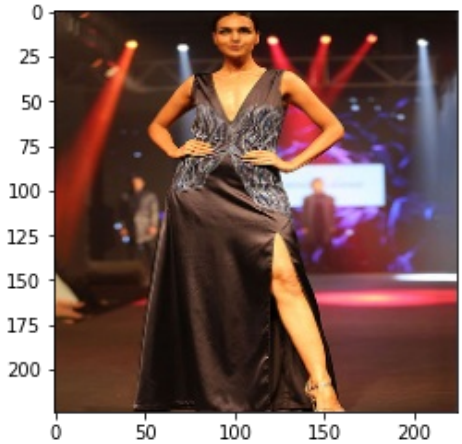
```

```

In [8]:

```
img = cv2.imread('./images/pic_82.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224,224))
plt.imshow(img)
```

Out[8]:

```
<matplotlib.image.AxesImage at 0x7ff1b78db0b8>
```



In [9]:

```
## Call ResNet model
ResNet = ResNet50(include_top=True)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet
/resnet50_weights_tf_dim_ordering_tf_kernels.h5
102973440/102967424 [==============================] - 4s 0us/step
```

In [10]:

```
## Apply ResNet model on all images
last = ResNet.layers[-2].output
modell = Model(inputs = ResNet.input,outputs = last)
modell.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)      0
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, 230, 230, 3)       0
_________________________________________________________________
conv1_conv (Conv2D)          (None, 112, 112, 64)      9472
_________________________________________________________________
conv1_bn (BatchNormalization (None, 112, 112, 64)      256
_________________________________________________________________
conv1_relu (Activation)      (None, 112, 112, 64)      0
_________________________________________________________________
pool1_pad (ZeroPadding2D)    (None, 114, 114, 64)      0
_________________________________________________________________
pool1_pool (MaxPooling2D)    (None, 56, 56, 64)        0
_________________________________________________________________
conv2_block1_1_conv (Conv2D) (None, 56, 56, 64)        4160
_________________________________________________________________
conv2_block1_1_bn (BatchNormali (None, 56, 56, 64)     256
_________________________________________________________________
conv2_block1_1_relu (Activation (None, 56, 56, 64)     0
_________________________________________________________________
conv2_block1_2_conv (Conv2D) (None, 56, 56, 64)        36928
_________________________________________________________________
conv2_block1_2_bn (BatchNormali (None, 56, 56, 64)     256
_________________________________________________________________
conv2_block1_2_relu (Activation (None, 56, 56, 64)     0
_________________________________________________________________
conv2_block1_0_conv (Conv2D) (None, 56, 56, 256)       16640
_________________________________________________________________
conv2_block1_3_conv (Conv2D) (None, 56, 56, 256)       16640
_________________________________________________________________
conv2_block1_0_bn (BatchNormali (None, 56, 56, 256)    1024
_________________________________________________________________
conv2_block1_3_bn (BatchNormali (None, 56, 56, 256)    1024
```

```
]
__________________________________________________________________________________________________
__________
conv2_block1_add (Add)          (None, 56, 56, 256)  0           conv2_block1_0_bn[0][0]
                                                                 conv2_block1_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv2_block1_out (Activation)   (None, 56, 56, 256)  0           conv2_block1_add[0][0]
__________________________________________________________________________________________________
__________
conv2_block2_1_conv (Conv2D)    (None, 56, 56, 64)   16448       conv2_block1_out[0][0]
__________________________________________________________________________________________________
__________
conv2_block2_1_bn (BatchNormali (None, 56, 56, 64)   256         conv2_block2_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_1_relu (Activation (None, 56, 56, 64)   0           conv2_block2_1_bn[0][0]
__________________________________________________________________________________________________
__________
conv2_block2_2_conv (Conv2D)    (None, 56, 56, 64)   36928       conv2_block2_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_2_bn (BatchNormali (None, 56, 56, 64)   256         conv2_block2_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_2_relu (Activation (None, 56, 56, 64)   0           conv2_block2_2_bn[0][0]
__________________________________________________________________________________________________
__________
conv2_block2_3_conv (Conv2D)    (None, 56, 56, 256)  16640       conv2_block2_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_3_bn (BatchNormali (None, 56, 56, 256)  1024        conv2_block2_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_add (Add)          (None, 56, 56, 256)  0           conv2_block1_out[0][0]
                                                                 conv2_block2_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv2_block2_out (Activation)   (None, 56, 56, 256)  0           conv2_block2_add[0][0]
__________________________________________________________________________________________________
__________
conv2_block3_1_conv (Conv2D)    (None, 56, 56, 64)   16448       conv2_block2_out[0][0]
__________________________________________________________________________________________________
__________
conv2_block3_1_bn (BatchNormali (None, 56, 56, 64)   256         conv2_block3_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv2_block3_1_relu (Activation (None, 56, 56, 64)   0           conv2_block3_1_bn[0][0]
__________________________________________________________________________________________________
__________
conv2_block3_2_conv (Conv2D)    (None, 56, 56, 64)   36928       conv2_block3_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv2_block3_2_bn (BatchNormali (None, 56, 56, 64)   256         conv2_block3_2_conv[0][0
```

```
]
__________________________________________________________________________________________________
__________
conv2_block3_2_relu (Activation (None, 56, 56, 64)   0           conv2_block3_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv2_block3_3_conv (Conv2D)    (None, 56, 56, 256)  16640       conv2_block3_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv2_block3_3_bn (BatchNormali (None, 56, 56, 256)  1024        conv2_block3_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv2_block3_add (Add)          (None, 56, 56, 256)  0           conv2_block2_out[0][0]

                                                                 conv2_block3_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv2_block3_out (Activation)   (None, 56, 56, 256)  0           conv2_block3_add[0][0]

__________________________________________________________________________________________________
__________
conv3_block1_1_conv (Conv2D)    (None, 28, 28, 128)  32896       conv2_block3_out[0][0]

__________________________________________________________________________________________________
__________
conv3_block1_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block1_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_1_relu (Activation (None, 28, 28, 128)  0           conv3_block1_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv3_block1_2_conv (Conv2D)    (None, 28, 28, 128)  147584      conv3_block1_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_2_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block1_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_2_relu (Activation (None, 28, 28, 128)  0           conv3_block1_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv3_block1_0_conv (Conv2D)    (None, 28, 28, 512)  131584      conv2_block3_out[0][0]

__________________________________________________________________________________________________
__________
conv3_block1_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block1_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_0_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block1_0_conv[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_3_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block1_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_add (Add)          (None, 28, 28, 512)  0           conv3_block1_0_bn[0][0]

                                                                 conv3_block1_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv3_block1_out (Activation)   (None, 28, 28, 512)  0           conv3_block1_add[0][0]
```

```
_____ _____                                                    _____ _____
_____
conv3_block2_1_conv (Conv2D)    (None, 28, 28, 128)  65664       conv3_block1_out[0][0]

_____
_____
conv3_block2_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block2_1_conv[0][0
]
_____
_____
conv3_block2_1_relu (Activation (None, 28, 28, 128)  0           conv3_block2_1_bn[0][0]

_____
_____
conv3_block2_2_conv (Conv2D)    (None, 28, 28, 128)  147584      conv3_block2_1_relu[0][0
]
_____
_____
conv3_block2_2_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block2_2_conv[0][0
]
_____
_____
conv3_block2_2_relu (Activation (None, 28, 28, 128)  0           conv3_block2_2_bn[0][0]

_____
_____
conv3_block2_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block2_2_relu[0][0
]
_____
_____
conv3_block2_3_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block2_3_conv[0][0
]
_____
_____
conv3_block2_add (Add)          (None, 28, 28, 512)  0           conv3_block1_out[0][0]

                                                                 conv3_block2_3_bn[0][0
]
_____
_____
conv3_block2_out (Activation)   (None, 28, 28, 512)  0           conv3_block2_add[0][0]

_____
_____
conv3_block3_1_conv (Conv2D)    (None, 28, 28, 128)  65664       conv3_block2_out[0][0]

_____
_____
conv3_block3_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block3_1_conv[0][0
]
_____
_____
conv3_block3_1_relu (Activation (None, 28, 28, 128)  0           conv3_block3_1_bn[0][0]

_____
_____
conv3_block3_2_conv (Conv2D)    (None, 28, 28, 128)  147584      conv3_block3_1_relu[0][0
]
_____
_____
conv3_block3_2_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block3_2_conv[0][0
]
_____
_____
conv3_block3_2_relu (Activation (None, 28, 28, 128)  0           conv3_block3_2_bn[0][0]

_____
_____
conv3_block3_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block3_2_relu[0][0
]
```

```
__________
conv3_block3_3_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block3_3_conv[0][0
]
__________
conv3_block3_add (Add)          (None, 28, 28, 512)  0           conv3_block2_out[0][0]
                                                                 conv3_block3_3_bn[0][0
]
__________
conv3_block3_out (Activation)   (None, 28, 28, 512)  0           conv3_block3_add[0][0]
__________
conv3_block4_1_conv (Conv2D)    (None, 28, 28, 128)  65664       conv3_block3_out[0][0]
__________
conv3_block4_1_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block4_1_conv[0][0
]
__________
conv3_block4_1_relu (Activation (None, 28, 28, 128)  0           conv3_block4_1_bn[0][0]
__________
conv3_block4_2_conv (Conv2D)    (None, 28, 28, 128)  147584      conv3_block4_1_relu[0][0
]
__________
conv3_block4_2_bn (BatchNormali (None, 28, 28, 128)  512         conv3_block4_2_conv[0][0
]
__________
conv3_block4_2_relu (Activation (None, 28, 28, 128)  0           conv3_block4_2_bn[0][0]
__________
conv3_block4_3_conv (Conv2D)    (None, 28, 28, 512)  66048       conv3_block4_2_relu[0][0
]
__________
conv3_block4_3_bn (BatchNormali (None, 28, 28, 512)  2048        conv3_block4_3_conv[0][0
]
__________
conv3_block4_add (Add)          (None, 28, 28, 512)  0           conv3_block3_out[0][0]
                                                                 conv3_block4_3_bn[0][0
]
__________
conv3_block4_out (Activation)   (None, 28, 28, 512)  0           conv3_block4_add[0][0]
__________
conv4_block1_1_conv (Conv2D)    (None, 14, 14, 256)  131328      conv3_block4_out[0][0]
__________
conv4_block1_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block1_1_conv[0][0
]
__________
conv4_block1_1_relu (Activation (None, 14, 14, 256)  0           conv4_block1_1_bn[0][0]
__________
conv4_block1_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block1_1_relu[0][0
]
```

```
__________
conv4_block1_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block1_2_conv[0][0
]
__________
__________
conv4_block1_2_relu (Activation (None, 14, 14, 256)  0           conv4_block1_2_bn[0][0]

__________
__________
conv4_block1_0_conv (Conv2D)    (None, 14, 14, 1024) 525312      conv3_block4_out[0][0]

__________
__________
conv4_block1_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block1_2_relu[0][0
]
__________
__________
conv4_block1_0_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block1_0_conv[0][0
]
__________
__________
conv4_block1_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block1_3_conv[0][0
]
__________
__________
conv4_block1_add (Add)          (None, 14, 14, 1024) 0           conv4_block1_0_bn[0][0]

                                                                 conv4_block1_3_bn[0][0
]
__________
__________
conv4_block1_out (Activation)   (None, 14, 14, 1024) 0           conv4_block1_add[0][0]

__________
__________
conv4_block2_1_conv (Conv2D)    (None, 14, 14, 256)  262400      conv4_block1_out[0][0]

__________
__________
conv4_block2_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block2_1_conv[0][0
]
__________
__________
conv4_block2_1_relu (Activation (None, 14, 14, 256)  0           conv4_block2_1_bn[0][0]

__________
__________
conv4_block2_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block2_1_relu[0][0
]
__________
__________
conv4_block2_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block2_2_conv[0][0
]
__________
__________
conv4_block2_2_relu (Activation (None, 14, 14, 256)  0           conv4_block2_2_bn[0][0]

__________
__________
conv4_block2_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block2_2_relu[0][0
]
__________
__________
conv4_block2_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block2_3_conv[0][0
]
__________
__________
conv4_block2_add (Add)          (None, 14, 14, 1024) 0           conv4_block1_out[0][0]

                                                                 conv4_block2_3_bn[0][0
]
```

```
__________
conv4_block2_out (Activation)   (None, 14, 14, 1024) 0           conv4_block2_add[0][0]

__________
conv4_block3_1_conv (Conv2D)    (None, 14, 14, 256)  262400      conv4_block2_out[0][0]

__________
conv4_block3_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block3_1_conv[0][0
]
__________
conv4_block3_1_relu (Activation (None, 14, 14, 256)  0           conv4_block3_1_bn[0][0]

__________
conv4_block3_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block3_1_relu[0][0
]
__________
conv4_block3_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block3_2_conv[0][0
]
__________
conv4_block3_2_relu (Activation (None, 14, 14, 256)  0           conv4_block3_2_bn[0][0]

__________
conv4_block3_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block3_2_relu[0][0
]
__________
conv4_block3_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block3_3_conv[0][0
]
__________
conv4_block3_add (Add)          (None, 14, 14, 1024) 0           conv4_block2_out[0][0]
                                                                 conv4_block3_3_bn[0][0
]
__________
conv4_block3_out (Activation)   (None, 14, 14, 1024) 0           conv4_block3_add[0][0]

__________
conv4_block4_1_conv (Conv2D)    (None, 14, 14, 256)  262400      conv4_block3_out[0][0]

__________
conv4_block4_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block4_1_conv[0][0
]
__________
conv4_block4_1_relu (Activation (None, 14, 14, 256)  0           conv4_block4_1_bn[0][0]

__________
conv4_block4_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block4_1_relu[0][0
]
__________
conv4_block4_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block4_2_conv[0][0
]
__________
conv4_block4_2_relu (Activation (None, 14, 14, 256)  0           conv4_block4_2_bn[0][0]

__________
conv4_block4_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block4_2_relu[0][0
```

```
]
__________________________________________________________________________________________________
__________
conv4_block4_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block4_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block4_add (Add)          (None, 14, 14, 1024) 0           conv4_block3_out[0][0]
                                                                 conv4_block4_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv4_block4_out (Activation)   (None, 14, 14, 1024) 0           conv4_block4_add[0][0]

__________________________________________________________________________________________________
__________
conv4_block5_1_conv (Conv2D)    (None, 14, 14, 256)  262400      conv4_block4_out[0][0]

__________________________________________________________________________________________________
__________
conv4_block5_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block5_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_1_relu (Activation (None, 14, 14, 256)  0           conv4_block5_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv4_block5_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block5_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block5_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_2_relu (Activation (None, 14, 14, 256)  0           conv4_block5_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv4_block5_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block5_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block5_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_add (Add)          (None, 14, 14, 1024) 0           conv4_block4_out[0][0]
                                                                 conv4_block5_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv4_block5_out (Activation)   (None, 14, 14, 1024) 0           conv4_block5_add[0][0]

__________________________________________________________________________________________________
__________
conv4_block6_1_conv (Conv2D)    (None, 14, 14, 256)  262400      conv4_block5_out[0][0]

__________________________________________________________________________________________________
__________
conv4_block6_1_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block6_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block6_1_relu (Activation (None, 14, 14, 256)  0           conv4_block6_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv4_block6_2_conv (Conv2D)    (None, 14, 14, 256)  590080      conv4_block6_1_relu[0][0
```

```
]
__________________________________________________________________________________________________
__________
conv4_block6_2_bn (BatchNormali (None, 14, 14, 256)  1024        conv4_block6_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block6_2_relu (Activation (None, 14, 14, 256)  0           conv4_block6_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv4_block6_3_conv (Conv2D)    (None, 14, 14, 1024) 263168      conv4_block6_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv4_block6_3_bn (BatchNormali (None, 14, 14, 1024) 4096        conv4_block6_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv4_block6_add (Add)          (None, 14, 14, 1024) 0           conv4_block5_out[0][0]

                                                                 conv4_block6_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv4_block6_out (Activation)   (None, 14, 14, 1024) 0           conv4_block6_add[0][0]

__________________________________________________________________________________________________
__________
conv5_block1_1_conv (Conv2D)    (None, 7, 7, 512)    524800      conv4_block6_out[0][0]

__________________________________________________________________________________________________
__________
conv5_block1_1_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block1_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_1_relu (Activation (None, 7, 7, 512)    0           conv5_block1_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv5_block1_2_conv (Conv2D)    (None, 7, 7, 512)    2359808     conv5_block1_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_2_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block1_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_2_relu (Activation (None, 7, 7, 512)    0           conv5_block1_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv5_block1_0_conv (Conv2D)    (None, 7, 7, 2048)   2099200     conv4_block6_out[0][0]

__________________________________________________________________________________________________
__________
conv5_block1_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block1_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_0_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block1_0_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block1_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block1_add (Add)          (None, 7, 7, 2048)   0           conv5_block1_0_bn[0][0]

                                                                 conv5_block1_3_bn[0][0
```

```
]
__________________________________________________________________________________________________
__________
conv5_block1_out (Activation)   (None, 7, 7, 2048)   0           conv5_block1_add[0][0]

__________________________________________________________________________________________________
__________
conv5_block2_1_conv (Conv2D)    (None, 7, 7, 512)    1049088     conv5_block1_out[0][0]

__________________________________________________________________________________________________
__________
conv5_block2_1_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block2_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_1_relu (Activation (None, 7, 7, 512)    0           conv5_block2_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv5_block2_2_conv (Conv2D)    (None, 7, 7, 512)    2359808     conv5_block2_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_2_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block2_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_2_relu (Activation (None, 7, 7, 512)    0           conv5_block2_2_bn[0][0]

__________________________________________________________________________________________________
__________
conv5_block2_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block2_2_relu[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block2_3_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_add (Add)          (None, 7, 7, 2048)   0           conv5_block1_out[0][0]
                                                                 conv5_block2_3_bn[0][0
]
__________________________________________________________________________________________________
__________
conv5_block2_out (Activation)   (None, 7, 7, 2048)   0           conv5_block2_add[0][0]

__________________________________________________________________________________________________
__________
conv5_block3_1_conv (Conv2D)    (None, 7, 7, 512)    1049088     conv5_block2_out[0][0]

__________________________________________________________________________________________________
__________
conv5_block3_1_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block3_1_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block3_1_relu (Activation (None, 7, 7, 512)    0           conv5_block3_1_bn[0][0]

__________________________________________________________________________________________________
__________
conv5_block3_2_conv (Conv2D)    (None, 7, 7, 512)    2359808     conv5_block3_1_relu[0][0
]
__________________________________________________________________________________________________
__________
conv5_block3_2_bn (BatchNormali (None, 7, 7, 512)    2048        conv5_block3_2_conv[0][0
]
__________________________________________________________________________________________________
__________
conv5_block3_2_relu (Activation (None, 7, 7, 512)    0           conv5_block3_2_bn[0][0]
```

```
__________
conv5_block3_3_conv (Conv2D)    (None, 7, 7, 2048)   1050624     conv5_block3_2_relu[0][0
]
__________
conv5_block3_3_bn (BatchNormali (None, 7, 7, 2048)   8192        conv5_block3_3_conv[0][0
]
__________
conv5_block3_add (Add)          (None, 7, 7, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0
]
__________
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0           conv5_block3_add[0][0]
__________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           conv5_block3_out[0][0]
==================================================================================
=========
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
__________
__________
```

In [11]:

```
## Lets predict using ResNet model and add it in a dictionary
images_features = {}
count = 0
for i in imagePath:
    img = cv2.imread(i)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (224,224))

    img = img.reshape(1,224,224,3)
    pred = modell.predict(img).reshape(2048,)

    # img_name = i.split('/')[-1]

    images_features[i] = pred

    count += 1

    if count > 1499:
        break

    elif count % 50 == 0:
        print(count)
```

```
50
100
150
200
250
300
350
400
450
500
```

In [36]:

```
## Values stored in image features dictionary
 images_features['./images/pic_82.jpg']
```

Out[36]:

```
array([1.6811163 , 2.175395  , 0.06980729, ..., 0.03872671, 1.4715201 ,
       0.19152299], dtype=float32)
```

In [ ]:

```
```

Text Preprocessing

In [12]:

```
# Add Imagepath and Description in one dictionary i.e., Key as Image path and values as d
escription
dataa = {}
for i, j in zip(imagePath, Iamge_Description):
  dataa[i] = j
```

In [211]:

```
for i in range(5):
  plt.figure()
  img_name = imagePath[i]


  img = cv2.imread(img_name)

  img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
  plt.xlabel(dataa[img_name])
  plt.imshow(img)
```

This stylish foil print kurta from janasya is made of poly crepe and comes in an attractive peach color. It features 3/4 sleeve,round neck,a-line and it is calf length kurta that is suitable for casual occasions. Team it with matching leggings for a chic look.

This check pattern top by Work Label is crafted in cotton. Featuring a bias check at the yoke and straight check pattern in bottom half, a smart round Neckline, 3/4th sleeves, this mid hip length top offers a stylish & comfortable fit. Style this top with a trouser/skirt and medium high heels for chic look at work. This top can also be styled with a pair of stud earrings and a pair your regular sneakers to attain a casual look.

Featuring elegant printed details, this off white top and skirt set from Jaipur Kurti makes a statement addition to your casual wardrobe. Style this set with a pair of high heels and statement accessories to complete the look.

Add an extra dose of style to your casual wardrobe with this elegant blue cape top from Twenty Dresses. Style With: A pair of black denims, colour block heels and statement earrings will complete this look!

Yellow polyester georgette maxi dress. Polyester knit lining inside for comfort. Overlapping v neckline with gathers at bust. Flared sleeves with attached belt at the waist. Concealed zipper on the left.

In [ ]:

```
```

In [13]:

```
## Preprocess the description
def preprocessed(txt):
    modified = txt.lower()
    ## Add start and end element in description
    modified = 'startofseq ' + modified + ' endofseq'
    return modified
```

In [14]:

```
## Apply the preprocessing on our dict
for k,v in dataa.items():
      dataa[k] = preprocessed(v)
```

In [15]:

```
## After preprocessing
dataa['./images/pic_199.jpg']
```

Out[15]:

```
'startofseq olive polyester crepe dress. wrap style v neckline with three fourth sleeves.
button detail on the left thigh. elasticated waist with an attached tie up belt. ruching
on the sleeve sides with tie up detailing. fit and flare silhouette. endofseq'
```

In [ ]:

```
```

Create Vocabulary

In [16]:

```
## Create a Vocabulary or Vocabulary of dictionary
count_words = {}
for k,v in dataa.items():

  for word in v.split():
    if word not in count_words:

      count_words[word] = 0

    else:
      count_words[word] += 1
```

In [17]:

```
## total number of style word present in our description
count_words['style']
```

Out[17]:

```
231
```

In [38]:

```
## Total words present
len(count_words)
```

Out[38]:

```
1939
```

In [ ]:

```

```

In [18]:

```
## concert a caption dict in the interger words i.e., machine understandable form
## Key as imagePath and description converted to integers
for k, v in dataa.items():

  encoded = []
  for word in v.split():
    encoded.append(count_words[word])

  dataa[k] = encoded
```

In [129]:

```
### decription in integer format
dataa['./images/pic_199.jpg']
```

Out[129]:

```
[499,
 5,
 119,
 45,
 46,
 23,
 231,
 40,
 59,
 713,
 14,
 12,
 34,
 44,
 19,
 136,
 472,
 14,
 0,
 24,
 56,
 713,
 80,
 8,
 26,
 48,
 4,
 6,
 136,
 472,
 15,
 5,
 713,
 26,
 48,
 9,
 54,
 921,
```

```
 34,
 40,
 499]
```

In [ ]:

```
```

Built Generator Function

Splitting each word and setting next word to repredicted as our output.

In [19]:

```
max_len = 0
for k, v in dataa.items():

  if len(v) > max_len:
    max_len = len(v)
    print(v)
```

```
[499, 562, 18, 2, 28, 7, 166, 1, 238, 25, 200, 1, 45, 921, 18, 226, 80, 1, 7, 2, 175, 128,
2, 0, 0, 921, 175, 238, 0, 14, 7, 45, 238, 1, 308, 61, 5, 35, 175, 713, 6, 4, 308, 876, 32
, 167, 499]
[499, 562, 7, 6, 197, 114, 13, 5, 238, 126, 226, 0, 40, 876, 1, 7, 137, 472, 12, 921, 29,
7, 6, 226, 14, 0, 876, 7, 46, 12, 29, 56, 562, 1, 3, 14, 197, 0, 876, 18, 44, 13, 66, 231,
562, 197, 713, 876, 0, 921, 0, 80, 151, 308, 32, 37, 137, 0, 562, 197, 21, 4, 39, 4, 713,
876, 183, 200, 4, 29, 921, 876, 183, 126, 77, 19, 326, 3, 876, 61, 167, 499]
[499, 66, 64, 76, 5, 46, 26, 48, 59, 19, 713, 876, 11, 3, 30, 10, 137, 472, 69, 12, 921,
44, 13, 69, 63, 308, 51, 921, 53, 5, 1, 137, 472, 11, 308, 45, 64, 23, 10, 20, 69, 713, 87
6, 5, 2, 308, 472, 5, 126, 2, 23, 238, 3, 226, 308, 876, 3, 19, 45, 238, 876, 3, 200, 9,
16, 44, 45, 10, 36, 28, 151, 876, 12, 24, 921, 7, 16, 57, 9, 49, 7, 7, 226, 2, 1, 499]
[499, 562, 66, 64, 76, 214, 114, 2, 238, 126, 226, 45, 921, 128, 876, 22, 66, 0, 59, 12,
52, 0, 118, 166, 472, 12, 876, 69, 11, 51, 19, 921, 876, 77, 66, 231, 562, 214, 713, 876,
183, 200, 80, 151, 921, 68, 55, 326, 152, 472, 167, 50, 562, 214, 17, 7, 7, 562, 19, 238,
8, 713, 13, 921, 7, 114, 8, 7, 921, 7, 8, 7, 200, 14, 921, 18, 10, 7, 166, 7, 9, 7, 308, 8
76, 10, 7, 7, 499]
```

In [21]:

```
## Maximum length array
max_len
```

Out[21]:

```
88
```

In [20]:

```
## Generate a X, y_in and y_out taking photo dict(i.e., contain ResNet prediction) and ca
ption dict(i.e., contain description in integer)
VOCAB_SIZE = len(count_words)

def generator(photo, caption):

  X = []
  y_in = []
  y_out = []

  for k, v in caption.items():

    for i in range(1, len(v)):
      X.append(photo[k])          ## Appending image featurees to X var

      in_seq= [v[:i]]             ## splitting the description in a sequence i.e., 0-1, 0
-1-2, 0-1-2-3, ...........
      out_seq = v[i]              ## Next word

      ## use pad sequence to convert the variable length to max_len i.e., 88
      in_seq = pad_sequences(in_seq, maxlen=max_len, padding='post', truncating='post')[
```

```
0]
      out_seq = to_categorical([out_seq], num_classes=VOCAB_SIZE)[0]

      y_in.append(in_seq)
      y_out.append(out_seq)

  return X, y_in, y_out
```

In [22]:

```
## run the generator func and add values in Variables
X, y_in, y_out = generator(images_features, dataa)
```

In [23]:

```
## length of all variables
len(X), len(y_in), len(y_out)
```

Out[23]:

```
(18526, 18526, 18526)
```

In [24]:

```
## convert them  in array
X = np.array(X)
y_in = np.array(y_in, dtype='float64')
y_out = np.array(y_out, dtype='float64')
```

In [25]:

```
## shapes
X.shape, y_in.shape, y_out.shape
```

Out[25]:

```
((18526, 2048), (18526, 88), (18526, 1939))
```

In [ ]:

```

```

In [26]:

```
## Built a model
embedding_size = 128
max_len = max_len
vocab_size = len(count_words)

image_model = Sequential()

image_model.add(Dense(embedding_size, input_shape=(2048,), activation='relu'))
image_model.add(RepeatVector(max_len))

image_model.summary()

language_model = Sequential()

language_model.add(Embedding(input_dim=vocab_size, output_dim=embedding_size, input_lengt
h=max_len))
language_model.add(LSTM(256, return_sequences=True))
language_model.add(TimeDistributed(Dense(embedding_size)))

language_model.summary()

conca = Concatenate()([image_model.output, language_model.output])
x = LSTM(128, return_sequences=True)(conca)
x = LSTM(512, return_sequences=False)(x)
x = Dense(vocab_size)(x)
out = Activation('softmax')(x)
model = Model(inputs=[image_model.input, language_model.input], outputs = out)
```

```
# model.load_weights("../input/model_weights.h5")
model.compile(loss='categorical_crossentropy', optimizer='RMSprop', metrics=['accuracy']
)
model.summary()
```

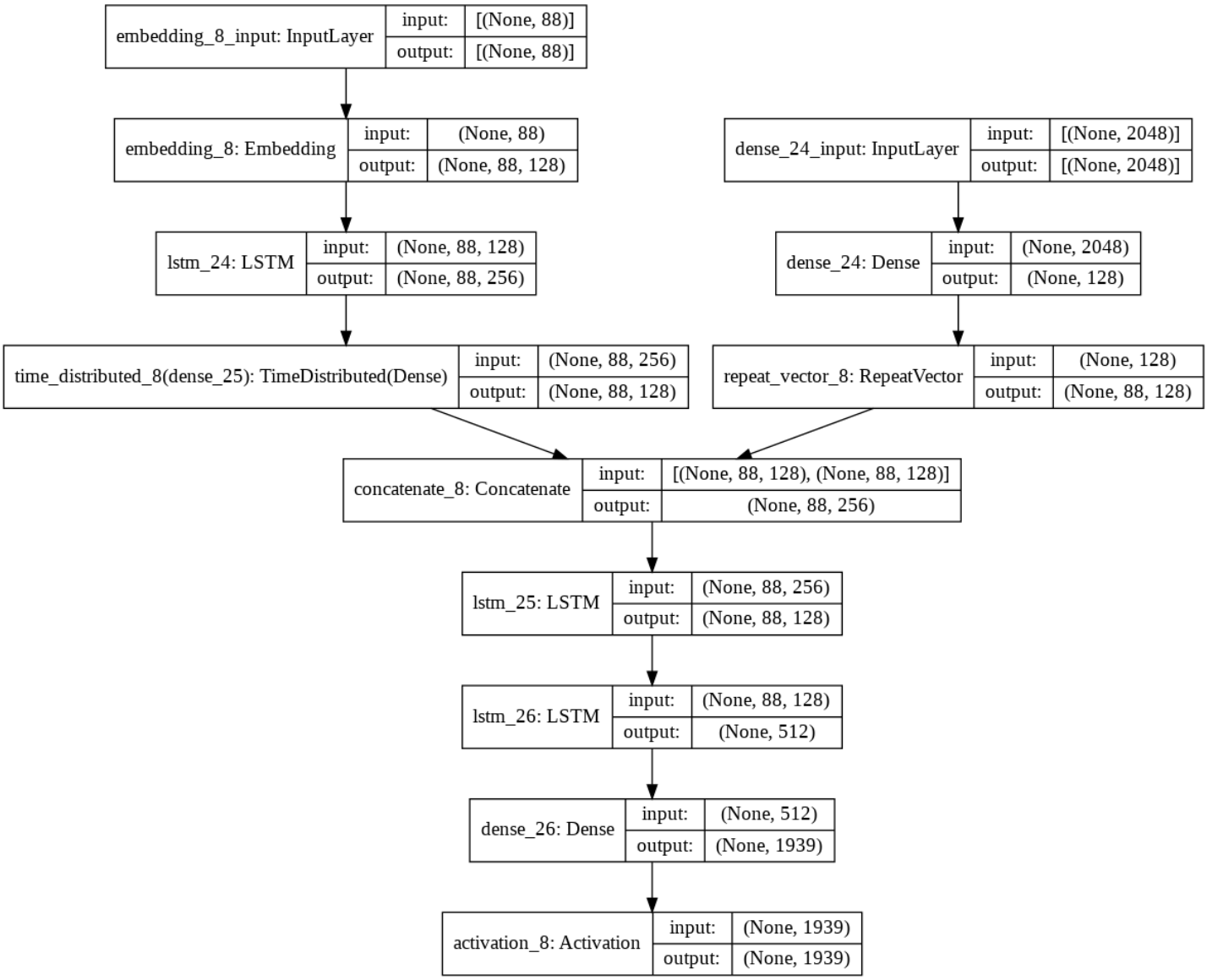
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 128)               262272
_________________________________________________________________
repeat_vector (RepeatVector) (None, 88, 128)           0
=================================================================
Total params: 262,272
Trainable params: 262,272
Non-trainable params: 0
_________________________________________________________________
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 88, 128)           248192
_________________________________________________________________
lstm (LSTM)                  (None, 88, 256)           394240
_________________________________________________________________
time_distributed (TimeDistri (None, 88, 128)           32896
=================================================================
Total params: 675,328
Trainable params: 675,328
Non-trainable params: 0
_________________________________________________________________
Model: "model_1"
_____________________________________________________________________________________________
_________
Layer (type)                    Output Shape         Param #     Connected to

=============================================================================================
=========
embedding_input (InputLayer)    [(None, 88)]         0

_____________________________________________________________________________________________
_________
dense_input (InputLayer)        [(None, 2048)]       0

_____________________________________________________________________________________________
_________
embedding (Embedding)           (None, 88, 128)      248192      embedding_input[0][0]

_____________________________________________________________________________________________
_________
dense (Dense)                   (None, 128)          262272      dense_input[0][0]

_____________________________________________________________________________________________
_________
lstm (LSTM)                     (None, 88, 256)      394240      embedding[0][0]

_____________________________________________________________________________________________
_________
repeat_vector (RepeatVector)    (None, 88, 128)      0           dense[0][0]

_____________________________________________________________________________________________
_________
time_distributed (TimeDistribut (None, 88, 128)      32896       lstm[0][0]

_____________________________________________________________________________________________
_________
concatenate (Concatenate)       (None, 88, 256)      0           repeat_vector[0][0]
                                                                 time_distributed[0][0]

_____________________________________________________________________________________________
```

```
_________
lstm_1 (LSTM)                   (None, 88, 128)       197120      concatenate[0][0]

_____________________________________________________________________________________
_________
lstm_2 (LSTM)                   (None, 512)           1312768     lstm_1[0][0]

_____________________________________________________________________________________
_________
dense_2 (Dense)                 (None, 1939)          994707      lstm_2[0][0]

_____________________________________________________________________________________
_________
activation (Activation)         (None, 1939)          0           dense_2[0][0]

==========================================================================================
=========
Total params: 3,442,195
Trainable params: 3,442,195
Non-trainable params: 0
_____________________________________________________________________________________
_________
```

In [231]:

```
## plot the model
from keras.utils import plot_model
plot_model(model, show_shapes=True)
```

Out[231]:



In [39]:

```
## fit the model
```

```
## Fit the model
model.fit([X, y_in], y_out, batch_size=512, epochs=100)
```

```
Epoch 1/100
37/37 [==============================] - 5s 122ms/step - loss: 1.4714 - accuracy: 0.4606
Epoch 2/100
37/37 [==============================] - 5s 122ms/step - loss: 1.4515 - accuracy: 0.4631
Epoch 3/100
37/37 [==============================] - 5s 122ms/step - loss: 1.4225 - accuracy: 0.4693
Epoch 4/100
37/37 [==============================] - 5s 122ms/step - loss: 1.4209 - accuracy: 0.4718
Epoch 5/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3986 - accuracy: 0.4790
Epoch 6/100
37/37 [==============================] - 5s 122ms/step - loss: 1.4113 - accuracy: 0.4777
Epoch 7/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3611 - accuracy: 0.4937
Epoch 8/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3763 - accuracy: 0.4870
Epoch 9/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3655 - accuracy: 0.4924
Epoch 10/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3272 - accuracy: 0.5000
Epoch 11/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3415 - accuracy: 0.4927
Epoch 12/100
37/37 [==============================] - 5s 122ms/step - loss: 1.3139 - accuracy: 0.5032
Epoch 13/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2895 - accuracy: 0.5129
Epoch 14/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2911 - accuracy: 0.5095
Epoch 15/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2813 - accuracy: 0.5110
Epoch 16/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2531 - accuracy: 0.5260
Epoch 17/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2794 - accuracy: 0.5088
Epoch 18/100
37/37 [==============================] - 5s 123ms/step - loss: 1.2497 - accuracy: 0.5239
Epoch 19/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2194 - accuracy: 0.5336
Epoch 20/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1858 - accuracy: 0.5460
Epoch 21/100
37/37 [==============================] - 5s 122ms/step - loss: 1.2254 - accuracy: 0.5234
Epoch 22/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1997 - accuracy: 0.5367
Epoch 23/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1962 - accuracy: 0.5363
Epoch 24/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1838 - accuracy: 0.5397
Epoch 25/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1940 - accuracy: 0.5369
Epoch 26/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1538 - accuracy: 0.5476
Epoch 27/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1542 - accuracy: 0.5473
Epoch 28/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1371 - accuracy: 0.5533
Epoch 29/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1375 - accuracy: 0.5547
Epoch 30/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1251 - accuracy: 0.5603
Epoch 31/100
37/37 [==============================] - 5s 123ms/step - loss: 1.1008 - accuracy: 0.5701
Epoch 32/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1123 - accuracy: 0.5619
Epoch 33/100
37/37 [==============================] - 5s 122ms/step - loss: 1.0945 - accuracy: 0.5764
Epoch 34/100
37/37 [==============================] - 5s 122ms/step - loss: 1.1006 - accuracy: 0.5672
Epoch 35/100
37/37 [==============================] - 5s 122ms/step - loss: 1.0747 - accuracy: 0.5790
```

```
Epoch 36/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0785 - accuracy: 0.5794
Epoch 37/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0668 - accuracy: 0.5803
Epoch 38/100
37/37 [==============================] - 5s 122ms/step - loss: 1.0536 - accuracy: 0.5852
Epoch 39/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0509 - accuracy: 0.5840
Epoch 40/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0445 - accuracy: 0.5878
Epoch 41/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0306 - accuracy: 0.5897
Epoch 42/100
37/37 [==============================] - 5s 122ms/step - loss: 1.0072 - accuracy: 0.6037
Epoch 43/100
37/37 [==============================] - 5s 122ms/step - loss: 1.0295 - accuracy: 0.5896
Epoch 44/100
37/37 [==============================] - 4s 121ms/step - loss: 1.0032 - accuracy: 0.6005
Epoch 45/100
37/37 [==============================] - 4s 121ms/step - loss: 0.9969 - accuracy: 0.6082
Epoch 46/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9912 - accuracy: 0.6013
Epoch 47/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9955 - accuracy: 0.6054
Epoch 48/100
37/37 [==============================] - 4s 121ms/step - loss: 0.9620 - accuracy: 0.6213
Epoch 49/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9702 - accuracy: 0.6109
Epoch 50/100
37/37 [==============================] - 4s 122ms/step - loss: 0.9627 - accuracy: 0.6186
Epoch 51/100
37/37 [==============================] - 4s 122ms/step - loss: 0.9370 - accuracy: 0.6305
Epoch 52/100
37/37 [==============================] - 4s 121ms/step - loss: 0.9604 - accuracy: 0.6214
Epoch 53/100
37/37 [==============================] - 4s 121ms/step - loss: 0.9364 - accuracy: 0.6233
Epoch 54/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9293 - accuracy: 0.6257
Epoch 55/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9374 - accuracy: 0.6290
Epoch 56/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8935 - accuracy: 0.6423
Epoch 57/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9190 - accuracy: 0.6292
Epoch 58/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9076 - accuracy: 0.6348
Epoch 59/100
37/37 [==============================] - 5s 122ms/step - loss: 0.9192 - accuracy: 0.6288
Epoch 60/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8836 - accuracy: 0.6448
Epoch 61/100
37/37 [==============================] - 4s 121ms/step - loss: 0.9071 - accuracy: 0.6333
Epoch 62/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8794 - accuracy: 0.6436
Epoch 63/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8736 - accuracy: 0.6460
Epoch 64/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8650 - accuracy: 0.6546
Epoch 65/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8506 - accuracy: 0.6574
Epoch 66/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8570 - accuracy: 0.6542
Epoch 67/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8744 - accuracy: 0.6498
Epoch 68/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8098 - accuracy: 0.6746
Epoch 69/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8531 - accuracy: 0.6551
Epoch 70/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8211 - accuracy: 0.6654
Epoch 71/100
37/37 [==============================] - 4s 122ms/step - loss: 0.8292 - accuracy: 0.6617
```

```
Epoch 72/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8033 - accuracy: 0.6729
Epoch 73/100
37/37 [==============================] - 5s 122ms/step - loss: 0.8163 - accuracy: 0.6654
Epoch 74/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8328 - accuracy: 0.6651
Epoch 75/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8084 - accuracy: 0.6743
Epoch 76/100
37/37 [==============================] - 4s 122ms/step - loss: 0.7946 - accuracy: 0.6769
Epoch 77/100
37/37 [==============================] - 4s 121ms/step - loss: 0.8199 - accuracy: 0.6652
Epoch 78/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7744 - accuracy: 0.6887
Epoch 79/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7978 - accuracy: 0.6704
Epoch 80/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7784 - accuracy: 0.6856
Epoch 81/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7526 - accuracy: 0.6946
Epoch 82/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7907 - accuracy: 0.6760
Epoch 83/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7506 - accuracy: 0.6970
Epoch 84/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7514 - accuracy: 0.6943
Epoch 85/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7391 - accuracy: 0.6976
Epoch 86/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7358 - accuracy: 0.7040
Epoch 87/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7416 - accuracy: 0.6930
Epoch 88/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7407 - accuracy: 0.6992
Epoch 89/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7381 - accuracy: 0.6982
Epoch 90/100
37/37 [==============================] - 5s 122ms/step - loss: 0.7355 - accuracy: 0.6971
Epoch 91/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7219 - accuracy: 0.7032
Epoch 92/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7267 - accuracy: 0.7031
Epoch 93/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7046 - accuracy: 0.7103
Epoch 94/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7373 - accuracy: 0.7016
Epoch 95/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7072 - accuracy: 0.7162
Epoch 96/100
37/37 [==============================] - 4s 121ms/step - loss: 0.7127 - accuracy: 0.7124
Epoch 97/100
37/37 [==============================] - 4s 121ms/step - loss: 0.6965 - accuracy: 0.7169
Epoch 98/100
37/37 [==============================] - 4s 121ms/step - loss: 0.6849 - accuracy: 0.7186
Epoch 99/100
37/37 [==============================] - 4s 121ms/step - loss: 0.6938 - accuracy: 0.7154
Epoch 100/100
37/37 [==============================] - 4s 121ms/step - loss: 0.6948 - accuracy: 0.7156
```

Out[39]:

```
<tensorflow.python.keras.callbacks.History at 0x7fef61ddcac8>
```

In [40]:

```
## Inverse the dictionary
inv_dict = {v:k for k, v in count_words.items()}
```

In [45]:

```
## save the model
model.save('image_desc.h5')
```

In [ ]:

```
```

Predictions

In [69]:

```
## function to convert the image in machine readable format
def getImage(x):

  # test_img_path = imagePath[x]

  test_img = cv2.imread(x)
  test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)

  test_img = cv2.resize(test_img, (224,224))

  test_img = np.reshape(test_img, (1,224,224,3))

  return test_img
```

In [70]:

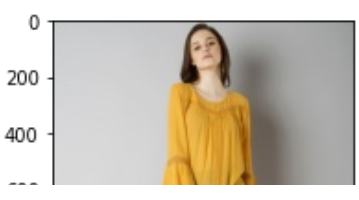
```
test_feature = modell.predict(getImage('newimg2.jpg')).reshape(1,2048)
```

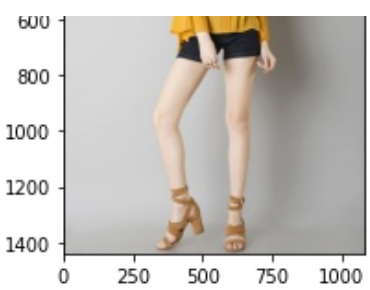
In [71]:

```
text_inp = ['startofseq']

count = 0
desc = ''
while count < 25:
  count += 1

  encoded = []
  for i in text_inp:
    encoded.append(count_words[i])

  encoded = [encoded]

  encoded = pad_sequences(encoded, padding='post', truncating='post', maxlen=max_len)


  prediction = np.argmax(model.predict([test_feature, encoded]))            ## Returns
a highest prob word

  sampled_word = inv_dict[prediction]

  desc = desc + ' ' + sampled_word

  if sampled_word == 'endofseq':
    break

  text_inp.append(sampled_word)
```

In [73]:

```
imag = cv2.imread('newimg2.jpg')
imag = cv2.cvtColor(imag, cv2.COLOR_BGR2RGB)
plt.imshow(imag)
plt.xlabel(desc);
```





this strappy tie-up top by is is your in left and a box neck, rib-knit fabric, fabric, rayon and and the the sling left left

Predict the decription of dataset images

In [ ]:

```
```

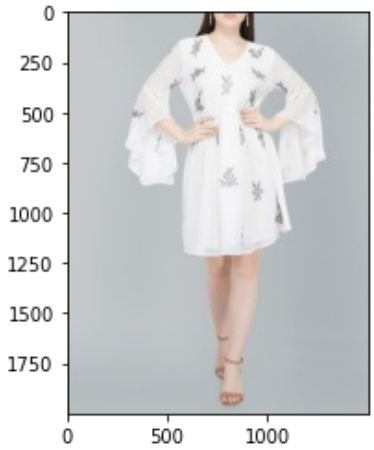
In [ ]:

```
## function to convert the image in machine readable format
def getImage(x):

  test_img_path = imagePath[x]

  test_img = cv2.imread(test_img_path)
  test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)

  test_img = cv2.resize(test_img, (224,224))

  test_img = np.reshape(test_img, (1,224,224,3))

  return test_img
```

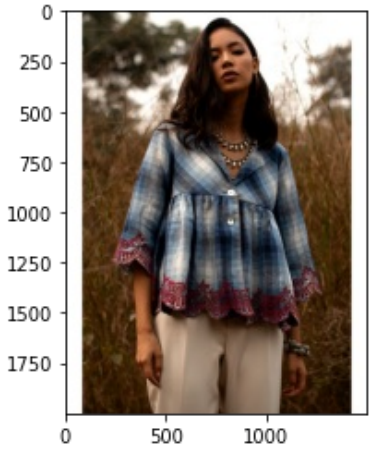
In [44]:

```
for i in range(5):

  no = np.random.randint(100,500,(1,1))[0,0]
  test_feature = modell.predict(getImage(no)).reshape(1,2048)

  test_img_path = imagePath[no]
  test_img = cv2.imread(test_img_path)
  test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)


  text_inp = ['startofseq']

  count = 0
  desc = ''
  while count < 25:
    count += 1

    encoded = []
    for i in text_inp:
      encoded.append(count_words[i])

    encoded = [encoded]

    encoded = pad_sequences(encoded, padding='post', truncating='post', maxlen=max_len)


    prediction = np.argmax(model.predict([test_feature, encoded]))

    sampled_word = inv_dict[prediction]

    desc = desc + ' ' + sampled_word

    if sampled_word == 'endofseq':
      break
```

```
    text_inp.append(sampled_word)

    plt.figure()
    plt.imshow(test_img)
    plt.xlabel(desc)
```



this white dress by platform platform is your in polyester and features a side neck, long half sleeves and kazo denims fun fun buy this



this neck, fun left from half rib-knit will be a buttoned tie to your placket you collection. side a half and rayon waist pair this

this black jumpsuit by fun is your in cotton and features a half rayon waist a fine style style tie-up and neck, neck, a a

cuffs half printed polyester fabric, style. only. sleeveless box tie with button fabric, the kazo front front front and and cuffs. pants neck waist a

this black placket jumpsuit from kazo will be a you tie to your placket you collection. side a tie neck and short waist pair this