# **Task 1**

# **shoes sorting algorithm**

# **Steps followed:**

Step wise explanation of my shoe sorting task.

1. Loading a Pre-trained Model:

I used a pre-trained model called "ResNet50." This model is a type of deep learning neural network that is already trained on a large dataset of images to recognize different objects. We will use this model as a feature extractor.

2. Adding a Custom Classification Head (Important step)

After loading the ResNet50 model, I modify it by adding a new layer on top. This new layer will be used to classify the images into three classes: "boot," "sandal," and "shoe." If we don’t add this extra layer we will encounter an error called index level missing because that pre-trained model contains 1000 classes and we are working with 3 classes. Finally, we add an output layer with a Softmax activation function, which will give us the probabilities of the image belonging to each class.

3. Predicting Class Labels and Sorting Images:

Then my code goes through the images in the dataset, one by one. It loads each image and preprocesses. It then makes predictions using the modified model. The image is then moved to a corresponding class folder based on its predicted class. For example, if the model predicts the image as a "boot," it will be moved to the "boot" folder, and similarly for "sandal" and "shoe."

4. Handling Invalid Images:

I also added code to take care of handling invalid image files. If an image file cannot be identified as valid or incompatible with the ResNet50 model's input size, it will be skipped, and a message will be printed.

5. Cleaning Up:

After processing all the images, I removes the temporary extracted directory to clean up the workspace.

# **Learning points:**

* Unsupervised learning
* K means clustering
* Imagenet pre-trained models
* Modifications of the model according to requirements

### Output Directory:

<https://drive.google.com/drive/folders/1GFPkiSzH6GYa2K5PDwVN_jwAYdTLAK_Y?usp=sharing>