

---

**Algorithm 1** Pseudocode of USL Person ReId Algorithm

---

**Require:** Unlabeled training data  $X$ ;

**Require:** Initialize the backbone encoder  $f_\theta$  with  
ImageNet-pretrained ResNet-50;

- 1: Set *random.seed()* etc. to make random number unchangeable for reproducing results;
  - 2: Create *log.txt* and prepare to print in it through function about *stdout*;
  - 3: **Create dataset**
  - 4: Get (*fname,pid,camid*) from image file path by using function *re.compile()*
  - 5: Inherit *Dataset class* and rewrite function *getitem* to pack data and return (*img,fname,pid,camid,index*)
  - 6: Get data in each batch from *Dataloader(transformed dataset,batchsize etc.)*
  - 7: **Create model**
  - 8: Rewrite function *init()* to show structure of model and function *forward()* in class *ReidModel*;
  - 9: Create model instance through calling *factory ['name'](\*args,\*\*kwargs)* e.g: *ReidModel*;
  - 10: **Optimizer**
  - 11: *Optimizer*(params requiring grad, learning rate, etc.) e.g:Adma;
  - 12: Set learning rate scheduler or how to change lr;
  - 13: **Train and evaluate**
  - 14: **for** *epoch* in [1,num\_epochs] **do**
  - 15:     Create cluster loader;
  - 16:     Extract feature vectors  $X^{key}$  from  $X$  by  $f_\theta$ ;
  - 17:     Compute distance among features;
  - 18:     Clustering  $X^{key}$  into  $N$  clusters with DBScan;
  - 19:     Create hybrid memory;
  - 20:     **Create class Memory**;
  - 21:     forward: memory dict forwards by computing similarity e.g:*mm*;
  - 22:     backward: return grad output and show how to update memory dict;
  - 23:     Initialize memory dictionary;
  - 24:     **Trainer**
  - 25:     **for**  $i$  in [1,num\_iterations] **do**
  - 26:         Sample  $P \times K$  query images from  $X$ ;
  - 27:         Compute loss;
  - 28:         Update memory with cluster feature;
  - 29:     **end for**
  - 30:     **Evaluate and compute mAP**
  - 31:     **Change learning rate**
  - 32: **end for**
-