1. Initial bullet point ( “.” For pasting code) (make your paper in same format as ANN IEEE)
2. Managing a mamba environment that includes pip packages
   1. Unless there’s a specific reason not do so, always try to mamba install packages instead of using pip, as mamba can’t detect pip packages when checking for already installed packages. For example, if you pip install package “A==1.24”, then mamba install “A==1.21”, you’ll see that in both “mamba list” output and “environment.yml” exported file, only “A==1.24” is visible under “pypi” distribution. Moreover, “A==1.21” will not be visible until you pip uninstall “A==1.24”.
   2. However, the reverse is not true when installing via pip: when you pip install packages, pip can see the mamba packages already installed and thus skip them and only pip install packages not found in neither pip nor mamba.
   3. Therefore, the file “package\_dep\_info.py”, which requires “pipdeptree” library, is aimed at showing dependencies/required by packages of passed packages and also listing leaf packages. Demonstration of this python script is shown in the **figures below**.  
      A picture containing text, screenshot, font, software

      Description automatically generated  
      A screen shot of a computer

      Description automatically generated with medium confidence  
      A screenshot of a computer program

      Description automatically generated with medium confidence
   4. The aforementioned python script is related to clean uninstallation process, but regarding installation, our goal is to minimize the packages installed by pip, as mamba manages dependency tracking better than pip. Now, suppose you want to get library ‘a’, then the steps to get it:
      1. Check if it has suitable mamba install command
      2. If not, pip install but with dry run to see packages that would be installed, for example:  
         “pip install fastdup --dry-run”  
         “Would install fastdup-1.4 opencv-python-headless-4.7.0.72 requests-2.28.1 sentry-sdk-1.24.0”
      3. Then, try to get these dependencies using mamba first like this example commands:  
         “mamba install -c fastai opencv-python-headless==4.7.0.72”  
         “mamba install -c anaconda requests=2.28.1”  
         Now, “fastdup” and “sentry-sdk” with the updated version “1.24.0” are currently not found as conda packages, so either get them using pip, or get an older version of “sentry-sdk” (this option is possible in our specific example, as the “fastdup” states that any “sentry-sdk” version is allowed, as seen in **figure below**:
3. Dataset\_preprocessing\_part\_3.ipynb for : cleaning academicDigitan & fSocialMedia:
   1. Manually cleaning by reordering images based on ratio and number of words criteria
   2. Fastdup:
      1. Outliers
      2. Mean image (dark, bright metrics)
4. Steps to enhance time performance
   1. Pin memory, affected much (almost double speed)
   2. Workers, made it consistent (same speed across batches)
   3. Profiler 🡪 biggest problem in train
   4. Decreasing params from 19.8 million to around 9 🡪 didn’t affect much
5. Steps to prevent always classifying same label (label 0 for example instead of label 1)
   1. CrossEntropyLoss(y\_pred\_without\_softmax, y\_true) instead of CrossEntropyLoss(y\_pred, y\_true)
   2. Possibly (didn’t try it yet): change learning rate   
      source:  
      <https://discuss.pytorch.org/t/why-am-i-getting-same-output-values-for-every-single-data-in-my-ann-model-for-multi-class-classification/57760/2>