

MANTA Mainnet Miner Deployment Guide

1. Hardware Requirement

CPU: Intel Xeon series with a base frequency of no less than 2.0 GHz and at least 8 cores

GPU: Nvidia Pascal or higher, video memory of no less than 12 GB

Internet: Internal bandwidth over 10 Gbps, external bandwidth over 10 Gbps

Harddrive: At least 700G SSD storage (500G for storing Matrix Mainnet data and 200G for storing models and training logs)

Storage: At least 32 GB

Network Configuration: A server that can be accessed through public network with an address in the format of <http://{IP}>. (The IP part should be accessible through public network.)

2. Distributed Auto-ML Web Service Configuration

A. Training Data Collection

The project will utilise two datasets for image categorisation and the training of parallax estimation models.

Image categorisation: ImageNet

Link: <https://www.image-net.org/challenges/LSVRC/2012/>

For more details, visit

https://blog.csdn.net/Yuan_mingyu/article/details/123940228

a. Dataset Download

Download the datasets at <https://www.image-net.org/challenges/LSVRC/index.php>.

ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)

[Back to Main page](#)

Citation^{NEW}

When using the dataset, please cite:

Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) **ImageNet Large Scale Visual Recognition Challenge**. *arXiv:1409.0575*, 2014. [paper](#) | [bibtex](#)

Development Kit

The development kit includes

- Meta data for the competition categories.
- Matlab routines for evaluating submissions.

Please be sure to consult the readme file included in the development kit.

[Development kit \(Task 1 & 2\)](#). 2.5MB.

[Development kit \(Task 3\)](#). 22MB.

Images

[Training images \(Task 1 & 2\)](#). 138GB. MD5: 1d675b47d978889d74fa0da5fadfb00e

[Training images \(Task 3\)](#). 728MB. MD5: ccaf1013018ac1037801578038d370da

[Validation images \(all tasks\)](#). 6.3GB. MD5: 29b22e2961454d5413ddabcf34fc5622

[Test images \(all tasks\)](#). 13GB. MD5: e1b8681fff3d63731c599df9b4b6fc02

If you downloaded ILSVRC 2012 test images on or before 10/10/2019, please apply [this patch](#) to replace a subset of images (a total of 2 images are replaced). Note that training and validation images are not affected.

Terms of use: by downloading the image data from the above URLs, you agree to the following terms:

1. You will use the data only for non-commercial research and educational purposes.
2. You will NOT distribute the above URL(s).
3. Stanford University and Princeton University make no representations or warranties regarding the data, including but not limited to warranties of non-infringement or fitness for a particular purpose.
4. You accept full responsibility for your use of the data and shall defend and indemnify Stanford University and Princeton University, including their employees, officers and agents, against any and all claims arising from your use of the data, including but not limited to your use of any copies of copyrighted images that you may create from the data.

CSDN @Yuan_mingyu

For image recognition, download the two files in red brackets

Training images (Task 1 & 2)

(https://image-net.org/data/ILSVRC/2012/ILSVRC2012_img_train.tar) Validation images (all tasks)

(https://image-net.org/data/ILSVRC/2012/ILSVRC2012_img_val.tar)

b. Dataset processing

Having downloaded the training and verification datasets, now we need to convert these datasets into a format that can be directly loaded by models.

First, decompress **ILSVRC2012_img_train.tar** to **train**. The decompressed folder should contain 1,000 tar files, each representing one category of images. The files are named accordingly, so don't rename the files. Simply decompress these **tar** files.

Decompress.

```
mkdir train tar xvf ILSVRC2012_img_train.tar -C ./train
```

As there are too many to decompress, write a script **unzip.sh** as below:

```
1 | #!/bin/bash
2 | dir=./train
3 | for x in `ls $dir/*.tar`
4 | do
5 |     filename=`basename $x .tar`
6 |     mkdir $dir/$filename
7 |     tar -xvf $x -C $dir/$filename
8 | done
9 | rm *.tar
```

Give the script execution permissions.

```
chmod +x ./unzip.sh
```

./unzip.sh

As **ILSVRC2012_img_train.tar** may be a bit large, you may delete it afterwards. Move **train.tar** to **train** first, before proceeding.

The final training dataset should be something like the following.

```
yuanmy@lab3090:/data/ImageNet/train$ ls
n01440764 n01755581 n02012849 n02097658 n02113978 n02356798 n02676566 n02939185 n03216828 n03538406 n03791053 n03983396 n04254120 n04483307 n07715103
n01443537 n01756291 n02013706 n02098105 n02114367 n02361337 n02687172 n02948072 n03218198 n03544143 n03792782 n03991062 n04254680 n04485082 n07716358
n01484850 n01768244 n02017213 n02098286 n02114548 n02363005 n02690373 n02950826 n03220513 n03584254 n03792972 n03992509 n04254777 n04486054 n07716906
n01491361 n01770081 n02018207 n02098413 n02114712 n02364673 n02692877 n02951358 n03223299 n03584829 n03793489 n03995372 n04258138 n04487081 n07717410
n01494475 n01770393 n02018795 n02099267 n02114855 n02389026 n02699494 n02951585 n03240683 n03590841 n03794056 n03998194 n04259630 n04487394 n07717556
n01496331 n01773157 n02025239 n02099429 n02115641 n02391049 n02701002 n02963159 n03249569 n03594734 n03796401 n04004767 n04263257 n04493381 n07718472
n01498041 n01773549 n02027492 n02099601 n02115913 n02395406 n02704792 n02965783 n03250847 n03594945 n03803284 n04005630 n04264628 n04501370 n07718747
n01514668 n01773797 n02028035 n02099712 n02116738 n02396427 n02708093 n02966193 n03255030 n03595614 n03804744 n04008634 n04265275 n04505470 n07720875
n01514859 n01774384 n02033041 n02099849 n02117135 n02397096 n02727426 n02966687 n03259280 n03598930 n03814639 n04009552 n04266014 n04507155 n07730033
n01518878 n01774750 n02037110 n02100236 n02119022 n02398521 n02730930 n02971356 n03271574 n03599486 n03814906 n04019541 n04270147 n04509417 n07734744
n01530575 n01775062 n02051845 n02100583 n02119789 n02403003 n02747177 n02974003 n03272010 n03602883 n03825788 n04023962 n04273569 n04515003 n07742313
n01531178 n01776313 n02056570 n02100735 n02120079 n02408429 n02749479 n02977058 n03272562 n03617480 n03832673 n04026417 n04275548 n04517823 n07745940
n01532829 n01784675 n02058221 n02100877 n02120505 n02410509 n02769748 n02978881 n03290653 n03623198 n03837869 n04033901 n04277352 n04522168 n07747607
n01534433 n01795545 n02066245 n02101006 n02123045 n02412080 n02776631 n02979186 n03291819 n03627232 n03838899 n04033995 n04285008 n04523525 n07749582
n01537544 n01796340 n02071294 n02101388 n02123159 n02415577 n02777292 n02980441 n03297495 n03630383 n03840681 n04037443 n04286575 n04525938 n07753113
n01558993 n01797886 n02074367 n02101556 n02123394 n02417914 n02782093 n02981702 n03314780 n03633091 n03841143 n04039381 n04296562 n04525305 n07753275
n01560419 n01798484 n02077923 n02102040 n02123597 n02422106 n02783161 n02988304 n03325584 n03637318 n03843555 n04040759 n04310018 n04532106 n07753592
n01580077 n01806143 n02085620 n02102177 n02124075 n02422699 n02786058 n02992211 n03337140 n03642806 n03854065 n04041544 n04311004 n04532670 n07754684
n01582220 n01806567 n02085782 n02102318 n02125311 n02423022 n02787622 n02992529 n03344393 n03649909 n03857828 n04044716 n04311174 n04536866 n07760859
n01592084 n01807496 n02085936 n02102480 n02127052 n02437312 n02788148 n02999410 n03345487 n03657121 n03866082 n04049303 n04311715 n04540053 n07768694
n01601694 n01817953 n02086079 n02102973 n02128385 n02437616 n02790996 n03000134 n03347037 n03658185 n03868242 n04065272 n04325704 n04542943 n07802026
n01608432 n01818515 n02086240 n02104029 n02128757 n02441942 n02791124 n03000247 n03355925 n03661043 n03868863 n04067472 n04326547 n04548280 n07831146
n01614925 n01819313 n02086646 n02104365 n02128925 n02442845 n02791270 n03000684 n03372029 n03662601 n03871628 n04069434 n04328186 n04548362 n07836838
n01616318 n01820546 n02086910 n02105056 n02129165 n02443114 n02793495 n03014705 n03376595 n03666591 n03873416 n04070727 n04330267 n04550184 n07860988
n01622779 n01824575 n02087046 n02105162 n02129604 n02443484 n02794156 n03016953 n03379051 n03670208 n03874293 n04074963 n04332243 n04552348 n07871810
n01629819 n01828970 n02087394 n02105251 n02130308 n02444819 n02795169 n03017168 n03384352 n03673027 n03874599 n04081281 n04335435 n04553703 n07873807
n01630670 n01829413 n02088094 n02105412 n02132136 n02445715 n02797295 n03018349 n03388043 n03676483 n03876231 n04086273 n04336792 n04554684 n07875152
n01631663 n01833805 n02088238 n02105505 n02133161 n02447366 n02799071 n03026506 n03388183 n03680355 n03877472 n04090263 n04344873 n04555703 n07878068
```

The verification dataset is relatively simple. It only contains 50,000 images. We could simply decompress **ILSVRC2012_img_val.tar**, but for ease of use afterwards, we should divide these images into 1,000 categories. (Just like what we did with the training dataset, create 1,000 folders and put the images in their corresponding categories.)

First decompress.

```
mkdir val tar xvf ILSVRC2012_img_val.tar -C ./val
```

Enter **val**. Download the script and execute.


```
cd val
wget -qO-
```

```
https://raw.githubusercontent.com/soumith/imagenetloader.torch/master/valprep.sh
chmod +x ./valprep.sh
./valprep.sh rm valprep.sh
```

The final form of the verification dataset should look like the following screenshot.

```
yuanmy@lab3090:~/data/ImageNet/val$ ls
n01440764 n01755581 n02012849 n02097658 n02113978 n02356798 n02676566 n02939185 n03216828 n03538406 n03791053 n03983396 n04254120 n04483307 n07715103
n01443537 n01756291 n02013706 n02098105 n02114367 n02361337 n02687172 n02948072 n03218198 n03544143 n03792782 n03991062 n04254680 n04485082 n07716358
n01484850 n01768244 n02017213 n02098286 n02114548 n02363005 n02690373 n02950826 n03220513 n03584254 n03792972 n03992509 n04254777 n04486054 n07716906
n01491361 n01770081 n02018207 n02098413 n02114712 n02364673 n02692877 n02951358 n03223299 n03584829 n03793489 n03995372 n04258138 n04487081 n07717410
n01494475 n01770393 n02018795 n02099267 n02114855 n02389026 n02699494 n02951585 n03240683 n03590841 n03794056 n03998194 n04259630 n04487394 n07717556
n01496331 n01773157 n02025239 n02099429 n02115641 n02391049 n02701002 n02963159 n03249569 n03594734 n03796401 n04004767 n04263257 n04493381 n07718472
n01498041 n01773549 n02027492 n02099601 n02115913 n02395406 n02704792 n02965783 n03250847 n03594945 n03803284 n04005630 n04264628 n04501370 n07718747
n01514668 n01773797 n02028035 n02099712 n02116738 n02396427 n02708093 n02966193 n03255030 n03595614 n03804744 n04008634 n04265275 n04505470 n07720875
n01514859 n01774384 n02033041 n02099849 n02117135 n02397096 n02727426 n02966687 n03259280 n03598930 n03814639 n04009552 n04266014 n04507155 n07730033
n01518878 n01774750 n02037110 n02100236 n02119022 n02398521 n02730930 n02971356 n03271574 n03599486 n03814906 n04019541 n04270147 n04509417 n07734744
n01530575 n01775062 n02051845 n02100588 n02119789 n02403003 n02747177 n02974003 n03272010 n03602883 n03825788 n04023962 n04273569 n04515003 n07742313
n01531178 n01776313 n02056570 n02100735 n02120079 n02408429 n02749479 n02977058 n03272562 n03617480 n03832673 n04026417 n04275548 n04517823 n07745940
n01532829 n01784675 n02058221 n02100877 n02120505 n02410509 n02769748 n02978881 n03290653 n03623198 n03837869 n04033901 n04277352 n04522168 n07747607
n01534433 n01795545 n02066245 n02101006 n02123045 n02412080 n02776631 n02979186 n03291819 n03627232 n03838809 n04033995 n04285008 n04523525 n07749582
n01537544 n01796340 n02071294 n02101388 n02123159 n02415577 n02777292 n02980441 n03297495 n03630383 n03840681 n04037443 n04286575 n04525038 n07753113
n01558993 n01797886 n02074367 n02101556 n02123394 n02417914 n02782093 n02981792 n03314780 n03633091 n03841143 n04039381 n04296562 n04525305 n07753275
n01560419 n01798484 n02077923 n02102040 n02123597 n02422106 n02783161 n02988304 n03325584 n03637318 n03843555 n04040759 n04310018 n04532106 n07753592
n01580077 n01806143 n02085620 n02102177 n02124075 n02422699 n02786608 n02992211 n03337140 n03642806 n03854065 n04041544 n04311004 n04532670 n07754684
n01582220 n01806567 n02085782 n02102318 n02125311 n02423022 n02787622 n02992529 n03344393 n03649909 n03857828 n04044716 n04311174 n04536866 n07760859
n01592084 n01807496 n02085936 n02102480 n02127052 n02437312 n02788148 n02999410 n03345487 n03657121 n03866082 n04049303 n04317175 n04540053 n07768694
n01601694 n01817953 n02086079 n02102973 n02128385 n02437616 n02790996 n03000134 n03347037 n03658185 n03868242 n04065272 n04325704 n04542943 n07802026
n01608432 n01818515 n02086240 n02104029 n02128757 n02441942 n02791124 n03000247 n03355925 n03661043 n03868863 n04067472 n04326547 n04548280 n07831146
n01614925 n01819313 n02086646 n02104365 n02128925 n02442845 n02791270 n03000684 n03372029 n03662601 n03871628 n04069434 n04328186 n04548362 n07836838
n01616318 n01820546 n02086910 n02105956 n02129165 n02443114 n02793495 n03014705 n03376595 n03666591 n03872416 n04070727 n04330267 n04559184 n07869988
n01622779 n01824575 n02087046 n02105162 n02129604 n02443484 n02794156 n03016953 n03379051 n03670208 n03874293 n04074963 n04332243 n04552348 n07871810
n01629819 n01828970 n02087394 n02105251 n02130308 n02444819 n02795169 n03017168 n03384352 n03673027 n03874599 n04081281 n04335435 n04553703 n07873807
n01630670 n01829413 n02088094 n02105412 n02132136 n02445715 n02797295 n03018349 n03388043 n03676483 n03876231 n04086273 n04336792 n04554684 n07875152
n01631663 n01833805 n02088238 n02105505 n02133161 n02447366 n02799071 n03026506 n03388183 n03680355 n03877472 n04090263 n04344873 n04557048 n07880948
```

After processing, the dataset should be in the following format.

```
1 | -imagenet
2 |   -train
3 |   -val
```

Parallax estimation: SceneFlow

Link: <https://lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html>

| | FlyingThings3D  | Driving  | Monkaa  |
|------------------------|--|--|--|
| Raw data | | | |
| RGB images (cleanpass) | PNG: .torrent (37GB) WebP: .torrent (7.4GB) | PNG: .torrent (6.3GB) WebP: .torrent (1.5GB) | PNG: .tar (9.1GB) WebP: .tar (1.8GB) |
| RGB images (finalpass) | PNG: .torrent (43GB) WebP: .torrent (5.7GB) | PNG: .torrent (6.1GB) WebP: .torrent (926MB) | PNG: .tar (17GB) WebP: [disabled] |
| Camera data | .tar (15MB) | .tar (1.8MB) | .tar (3.7MB) |
| Object segmentation | .tar.bz2 (409MB, unzipped 104GB) | .tar.bz2 (78MB, unzipped 18GB) | .tar.bz2 (83MB, unzipped 34GB) |
| Material segmentation | .tar.bz2 (510MB, unzipped 104GB) | .tar.bz2 (170MB, unzipped 18GB) | .tar.bz2 (115MB, unzipped 34GB) |
| Derived data | | | |
| Disparity | .torrent (87GB, unzipped 104GB) | .torrent (9GB, unzipped 18GB) | .tar.bz2 (28GB, unzipped 34GB) |
| Disparity change | .torrent (116GB, unzipped 208GB) | .torrent (22GB, unzipped 35GB) | .tar.bz2 (35GB, unzipped 68GB) |
| Optical flow | .torrent (311GB, unzipped 621GB) | .torrent (50GB, unzipped 102GB) | .tar.bz2 (89GB, unzipped 201GB) |
| Motion boundaries | .tar.bz2 (615MB, unzipped 52GB) | .tar.bz2 (206MB, unzipped 8.6GB) | .tar.bz2 (106MB, unzipped 17GB) |

Bold sizes indicate that a compressed archive expands to a very much larger size (more than 100GB larger, or expansion factor > 10).

Load the six datasets in the following screenshot and decompress them.

After downloading is complete, modify the route through the following method for easy recognition by web services.

- **(1) ImageNet /mnt/imagenet**
| | --train | --val
- **(2) SceneFlow /mnt/SceneFlow**
| | --FlyingThings3D | --Monkaa | --Driving

B. Launching Web Service Training

(1) Enter **dist-automl** catalogue.

Command: **cd dist-automl**

(2) Install the python database required by the project.

Command: **pip install -r requirements.txt**

(3) Execute the “launch service” order.

Command: **streamlit run training_manager.py**

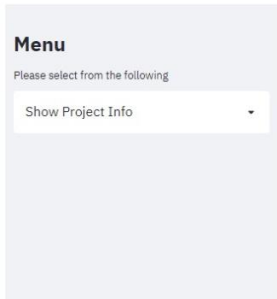
If the Terminal should show the following, it means the service has been successfully launched.

```
(base) root@i-8rx7rw1hjdk52lr6rjvf:/mnt/gpu1/dist-automl# streamlit run training_manager.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8502
Network URL: http://10.73.114.164:8502
```

← 替换成部署服务器的公网IP

Replace the part in the red bracket with the public IP of the server where you wish to deploy the service, and you will be able to access the training web service.

You may also offer your deployed service as a distributed Auto-ML service on the Mainnet.



Distributed AutoML Training Service

The application of AutoML DNN is widespread and benefits many industries.

Our goal is to estimate a kinematic model for the individual in each picture.

👉 Please select **Select a Demo Image** in the sidebar to start.

📌 Please make sure to checkout our favourite Kangaroo photo by pressing **Fur-riend**.

📷 Feel free to upload any image you want to get a pose estimation under **Upload an Image:** ⇨ SINGLE-person images ⇨ CENTER the human ⇨ HAVE FUN