

---

# **GluonFace Documentation**

*Release 1.0.1*

**haoxintong**

**Aug 05, 2019**



---

## Contents

---

<b>1</b>	<b>Losses in GluonFR:</b>	<b>3</b>
<b>2</b>	<b>Authors</b>	<b>5</b>
<b>3</b>	<b>Discussion</b>	<b>7</b>
<b>4</b>	<b>References</b>	<b>9</b>
4.1	Installation . . . . .	9
4.2	Datasets . . . . .	9
4.3	Model Zoo . . . . .	10
4.4	API Reference . . . . .	11
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



Gluon Face is a toolkit based on MXnet Gluon, provides SOTA deep learning algorithm and models in face recognition. If you are new to mxnet, please check out [dmlc 60-minute crash course](#).

---

**Hint:** For Chinese readers, here is the [zh-doc](#).

---

Gluon Face provides implement of losses in recent, including SoftmaxCrossEntropyLoss, ArcLoss, TripletLoss, RingLoss, CosLoss, L2Softmax, ASoftmax, CenterLoss, ContrastiveLoss, ... , and we will keep updating in future.

---

**Hint:** Github: see details in [gluon face](#).

---

If there is any method we overlooked, please open an [issue](#).





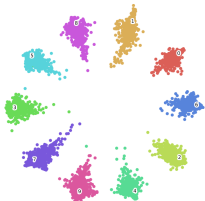


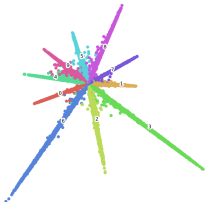
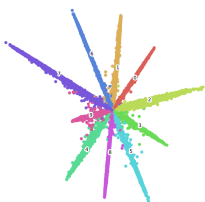
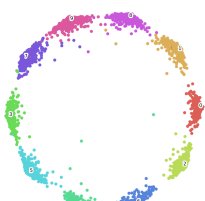
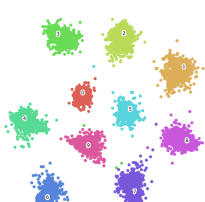
# CHAPTER 1

---

Losses in GluonFR:

---

The last column of this chart is the best LFW accuracy reported in paper, they are trained with different data and networks, later we will give our results of these method with same train data and network.

Method	Paper	Visualization of MNIST	LFW
Contrastive Loss	ContrastiveLoss		•
Triplet	1503.03832		99.63±0.09
Center Loss	CenterLoss		99.28
L2-Softmax	1703.09507		99.33
A-Softmax	1704.08063		99.42
CosLoss/AMSoftmax	1801.05599/1801.05599		99.17
Arcloss	1801.07698		99.82
Ring loss	1803.00130		99.52
LGM Loss	1803.02988		99.20±0.03



## CHAPTER 2

---

Authors

---

{ haoxintong Yangxv }



## CHAPTER 3

---

### Discussion

---

Gluon-Forum Feel free to use English here :D.



1. MXNet Documentation and Tutorials <https://zh.diveintodeeplearning.org/>
2. NVIDIA DALI documentation [NVIDIA DALI documentation](#)
3. Deepinsight [insightface](#)

## 4.1 Installation

Gluon Face supports Python 3.5 or later. To install this package you need install GluonCV and MXNet first:

```
pip install gluoncv --pre
pip install mxnet-mkl --pre --upgrade
# if cuda XX is installed
pip install mxnet-cuXXmkl --pre --upgrade
```

Then install gluonfr:

- From Socure(recommend)

```
pip install git+https://github.com/THUFutureLab/gluon-face.git@master
```

- Pip

```
pip install gluonfr
```

## 4.2 Datasets

`gluonfr.data` provides input pipeline for training and validation, all datasets is aligned by mtcnn and cropped to (112, 112) by DeepInsight, they converted images to `train.rec`, `train.idx` and `val_data.bin` files, please check out [\[insightface/Dataset-Zoo\]](#) for more information. In `examples/dali_utils.py`, there is a simple example of Nvidia-DALI. It is worth trying when data augmentation with cpu can not satisfy the speed of gpu training,

The files should be prepared like:

```

face/
  emore/
    train.rec
    train.idx
    property
  mslm/
    train.rec
    train.idx
    property
  lfw.bin
  agedb_30.bin
  ...
  vgg2_fp.bin
    
```

We use `~/ .mxnet/datasets` as default dataset root to match mxnet setting.

### 4.2.1 References

- **LFW** “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments”
- **CALFW** “A Database for Studying Cross-Age Face Recognition in Unconstrained Environments”
- **CPLFW** “Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments”
- **CFP\_fp, CFP\_ff** “Frontal to Profile Face Verification in the Wild”
- **AgeDB\_30** “AgeDB: the first manually collected, in-the-wild age database”
- **VGG2\_fp** “VGGFace2: A dataset for recognising faces across pose and age”

## 4.3 Model Zoo

### 4.3.1 Mobilefacenet Result.

TestSet	Ours	Insightface	Proposed
LFW:	<b>99.56</b>	99.50	99.55
CFP_FP:	<b>92.98</b>	88.94	•
AgeDB30:	95.86	95.91	<b>96.07</b>

Reference:

1. Our [code train script](#) and [log/model](#) in (Baidu:y5zh, Google Drive).
2. [Insightface result](#).
3. [Mobilefacenets papers](#).(No open project)

## Details

Flip	False	True
lfw:	0.995500+-0.003337	<b>0.995667+-0.003432</b>
calfw:	0.951000+-0.012069	<b>0.973083+-0.022889</b>
cplfw:	0.882000+-0.014295	<b>0.938556+-0.045234</b>
cfp_fp:	0.927714+-0.015309	<b>0.929880+-0.035907</b>
agedb_30:	<b>0.958667+-0.008492</b>	0.934903+-0.033667
cfp_ff:	<b>0.995571+-0.002744</b>	0.944868+-0.037657
vgg2_fp:	0.920600+-0.010920	0.940581+-0.032677

## Information

1. Github has some [projects](#) train to a high level, but with embedding\_size of 512, compare with them we use embedding\_size of 128 which origin paper proposed, model size is only 4.1M.
2. Welcome to use our train script to do more exploration, and if you get better results you could make a pr to us.
3. We Pre-trained model through L2-Regularization, output is cos(theta).

## 4.4 API Reference

### 4.4.1 gluonfr.data

---

**Hint:** Please refer to [Datasets](#) for the description of the datasets listed in this page, and how to download and extract them.

---

### API Reference

This module provides popular face recognition datasets.

**class** gluonfr.data.FRTrainRecordDataset

A dataset wrapping over a rec serialized file provided by InsightFace Repo.

#### Parameters

- **name** (*str.* Name of val dataset.) –
- **root** (*str.* Path to face folder. Default is '\$(HOME)/mxnet/datasets/face') –
- **transform** (*function*, default None) – A user defined callback that transforms each sample. For example:

```
transform=lambda data, label: (data.astype(np.float32)/255, label)
```

**class** gluonfr.data.FRValDataset

A dataset wrapping over a pickle serialized (.bin) file provided by InsightFace Repo.

#### Parameters

- **name** (*str.*) – Name of val dataset.

- **root** (*str.*) – Path to face folder. Default is ‘\$(HOME)/mxnet/datasets/face’
- **transform** (*callable, default None.*) – A function that takes data and transforms them:

```
transform = lambda data: data.astype(np.float32)/255
```

## 4.4.2 gluonfr.nn

Neural Network Components.

---

**Hint:** Not every component listed here is [HybridBlock](#), which means some of them are not hybridizable. However, we are trying our best to make sure components required during inference are hybridizable so the entire network can be exported and run in other languages.

For example, encoders are usually non-hybridizable but are only required during training. In contrast, decoders are mostly [HybridBlock](#) s.

---

### Basic Blocks

Blocks that usually used in face recognition.

### API Reference

Basic Blocks used in GluonFR.

```
class gluonfr.nn.basic_blocks.NormDense  
    Norm Dense
```

```
class gluonfr.nn.basic_blocks.SELayer  
    SE Layer
```

```
class gluonfr.nn.basic_blocks.FrBase
```

This is base class for all face recognition network. In this class, we defined the NormDense and control flow of the sub classes. In any sub classes, only need to implement features and embedding\_layer. Normally we add embedding\_layer to features.

#### Parameters

- **classes** (*int*) – Number of classification classes.
- **embedding\_size** (*int*) – Units of embedding layer.
- **weight\_norm** (*bool, default False*) – Whether use weight norm in NormDense layer.
- **feature\_norm** (*bool, default False*) – Whether use features norm in NormDense layer.
- **need\_cls\_layer** (*bool, default True*) – Whether use NormDense layer. Normally it depends on your loss function. When you use Softmax, ArcLoss or based on Softmax loss, you need to set it to True. When you only need embedding output, like you are predicting or training with triplet loss, you need to set it to False.



### 4.4.3 gluonfr.loss

<code>gluonfr.loss.get_loss</code>	Return the loss by name.
<code>gluonfr.loss.get_loss_list</code>	Get the entire list of loss names in losses.

#### API Reference

Custom losses

`gluonfr.loss.get_loss(name, **kwargs)`

Return the loss by name.

##### Parameters

- **name** (*str.*) – Available losses name in gluon face
- **kwargs** (*str.*) – Check the docs for details.

**Returns** The loss.

**Return type** HybridBlock

`gluonfr.loss.get_loss_list()`

Get the entire list of loss names in losses.

**Returns** Entire list of loss names in losses.

**Return type** list of str.

**class** `gluonfr.loss.ArcLoss`

ArcLoss from “ArcFace: Additive Angular Margin Loss for Deep Face Recognition” paper.

##### Parameters

- **classes** (*int.*) – Number of classes.
- **m** (*float.*) – Margin parameter for loss.
- **s** (*int.*) – Scale parameter for loss.

##### Outputs:

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** `gluonfr.loss.TripletLoss`

Calculates triplet loss given three input tensors and a positive margin. Triplet loss measures the relative similarity between prediction, a positive example and a negative example:

$$L = \sum_i \max(\|pred_i - pos_i\|_2^2 - \|pred_i - neg_i\|_2^2 + margin, 0)$$

*pred*, *positive* and *negative* can have arbitrary shape as long as they have the same number of elements.

##### Parameters

- **margin** (*float*) – Margin of separation between correct and incorrect pair.
- **weight** (*float or None*) – Global scalar weight for loss.
- **batch\_axis** (*int, default 0*) – The axis that represents mini-batch.

##### Inputs:

- **pred**: prediction tensor with arbitrary shape
- **positive**: positive example tensor with arbitrary shape. Must have the same size as pred.
- **negative**: negative example tensor with arbitrary shape. Must have the same size as pred.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,).

**class** gluonfr.loss.RingLoss

Computes the Ring Loss from “Ring loss: Convex Feature Normalization for Face Recognition” paper.

$$L = - \sum_i \log(\text{pred})_{i, \text{label}_i} + \frac{\lambda}{2m} \sum_{i=1}^m (\|\mathcal{F}(x_i)\|_2 - R)^2$$

**Parameters**

- **lamda** (*float.*) – The loss weight enforcing a trade-off between the softmax loss and ring loss.
- **r\_init** (*float.*) – The initial value of Hyper Parameter R.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.CosLoss

**CosLoss** from “CosFace: Large Margin Cosine Loss for Deep Face Recognition” paper.

It is also AM-Softmax from “Additive Margin Softmax for Face Verification” paper.

**Parameters**

- **classes** (*int.*) – Number of classes.
- **m** (*float, default 0.4*) – Margin parameter for loss.
- **s** (*int, default 64*) – Scale parameter for loss.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.L2Softmax

L2Softmax from “L2-constrained Softmax Loss for Discriminative Face Verification” paper.

**Parameters**

- **classes** (*int.*) – Number of classes.
- **alpha** (*float.*) – The scaling parameter, a hypersphere with small alpha will limit surface area for embedding features.
- **p** (*float, default is 0.9.*) – The expected average softmax probability for correctly classifying a feature.
- **from\_normx** (*bool, default is False.*) – Whether input has already been normalized.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.ASoftmax

ASoftmax from “SphereFace: Deep Hypersphere Embedding for Face Recognition” paper. input(weight, x) has already been normalized

**Parameters**

- **classes** (*int.*) – Number of classes.
- **m** (*float.*) – Margin parameter for loss.
- **s** (*int.*) – Scale parameter for loss.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.CenterLoss

Computes the Center Loss from “A Discriminative Feature Learning Approach for Deep Face Recognition” paper.

Implementation is refer to “[https://github.com/ShownX/mxnet-center-loss/blob/master/center\\_loss.py](https://github.com/ShownX/mxnet-center-loss/blob/master/center_loss.py)”

**Parameters**

- **classes** (*int.*) – Number of classes.
- **lamda** (*float*) – The loss weight enforcing a trade-off between the softmax loss and center loss.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.ContrastiveLoss

Computes the contrastive loss. See “Dimensionality Reduction by Learning an Invariant Mapping” paper. This loss encourages the embedding to be close to each other for the samples of the same label and the embedding to be far apart at least by the margin constant for the samples of different labels.

**Parameters** **margin** (*float, default is 1.*) – Margin term in the loss definition.

**Inputs:**

- **anchor**: prediction tensor. Embeddings should be l2 normalized.
- **positive**: positive example tensor with arbitrary shape. Must have the same size as anchor. Embeddings should be l2 normalized.
- **labels**: array with shape (batch\_size,) of binary labels indicating positive vs negative pair.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,).Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.LGMLoss

LGM Loss from “Rethinking Feature Distribution for Loss Functions in Image Classification” paper.

Implementation is refer to [https://github.com/LeeJuly30/L-GM-Loss-For-Gluon/blob/master/L\\_GM.py](https://github.com/LeeJuly30/L-GM-Loss-For-Gluon/blob/master/L_GM.py)

**Parameters**

- **num\_classes** (*int.*) – The num of classes.
- **embedding\_size** (*int.*) – The size of embedding feature.

- **alpha** (*float.*) – A non-negative parameter controlling the size of the expected margin between two classes on the training set.
- **lamda** (*float.*) – A non-negative weighting coefficient.
- **lr\_mult** (*float.*) – Var updating need a relatively low learning rate compared to the overall learning rate.

**class** gluonfr.loss.MPSLoss

Computes the MPS Loss from “DocFace: Matching ID Document Photos to Selfies” paper.

**Parameters** **m** (*float*) – Margin parameter for loss.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.GitLoss

Computes the Git Loss from “Git Loss for Deep Face Recognition” paper.

This implementation require the batch size not changing in training or validation. Commonly, it is ok, as when we train models last batch discard is applied, and no need for validation to compute the loss.

**Parameters**

- **classes** (*int.*) – Number of classes.
- **embedding\_size** (*int.*) – Size of feature.
- **lamda\_c** (*float.*) – The loss weight enforcing a trade-off between the softmax loss and center loss.
- **lamda\_g** (*float.*) – The loss weight enforcing a trade-off between the softmax loss and git loss.
- **batch\_size\_per\_gpu** (*int.*) – This size is sample numbers in each gpu or device, not total batch size

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** gluonfr.loss.COCOLoss

Computes the COCO Loss from “Rethinking Feature Discrimination and Polymerization for Large-scale Recognition” paper.

This loss can be replaced by NormDense with Softmax, it is not recommended to use this.

**Parameters**

- **classes** (*int.*) – Number of classes.
- **embedding\_size** (*int.*) – Size of feature.
- **alpha** (*float.*) – The scaling parameter, a hypersphere with small alpha will limit surface area for embedding features.

**Outputs:**

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

**class** `gluonfr.loss.SVXSoftmax`

SVXSoftmax from “Support Vector Guided Softmax Loss for Face Recognition” paper.

When use default parameter, the designed SV-X-Softmax loss becomes identical to the original softmax loss.

#### Parameters

- **classes** (*int.*) – Number of classes.
- **s** (*int.*) – Scale parameter for loss.
- **t** (*float.*) – Indicator parameter of SV.
- **m1** (*float.*) – Margin parameter for sphere softmax.
- **m2** (*float.*) – Margin parameter for cos/am softmax.
- **m3** (*float.*) – Margin parameter for arc softmax.

#### Outputs:

- **loss**: loss tensor with shape (batch\_size,). Dimensions other than batch\_axis are averaged out.

### 4.4.4 gluonfr.model\_zoo

<code>gluonfr.model_zoo.get_model</code>	Returns a model by name.
<code>gluonfr.model_zoo.get_model_list</code>	Get the entire list of model names in model_zoo.

**Hint:** This is the recommended method for getting a pre-defined model.

## API Reference

Models for face recognition

**class** `gluonfr.model_zoo.AttentionNet`

AttentionNet Model from “Residual Attention Network for Image Classification” paper.

#### Parameters

- **classes** (*int.*) – Number of classification classes.
- **modules** (*list.*) – The number of Attention Module in each stage.
- **p** (*int.*) – Number of pre-processing Residual Units before split into trunk branch and mask branch.
- **t** (*int.*) – Number of Residual Units in trunk branch.
- **r** (*int.*) – Number of Residual Units between adjacent pooling layer in the mask branch.
- **kwargs** –

**class** `gluonfr.model_zoo.AttentionNetFace`

AttentionNet Model for input 112x112.

#### Parameters

- **classes** (*int.*) – Number of classification classes.
- **modules** (*list.*) – The number of Attention Module in each stage.

- **p** (*int.*) – Number of pre-processing Residual Units before split into trunk branch and mask branch.
- **t** (*int.*) – Number of Residual Units in trunk branch.
- **r** (*int.*) – Number of Residual Units between adjacent pooling layer in the mask branch.
- **embedding\_size** (*int*) – Units of embedding layer.
- **weight\_norm** (*bool, default False*) – Whether use weight norm in NormDense layer.
- **feature\_norm** (*bool, default False*) – Whether use features norm in NormDense layer.
- **need\_cls\_layer** (*bool, default True*) – Whether use NormDense layer. Normally it depends on your loss function. When you use Softmax, ArcLoss or based on Softmax loss, you need to set it to True. When you only need embedding output, like you are predicting or training with triplet loss, you need to set it to False.

**class** gluonfr.model\_zoo.**MobileFaceNet**  
Mobile FaceNet

gluonfr.model\_zoo.**attention\_net128** (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 128 Model for face recognition.

**Parameters**

- **classes** (*int, -1*) – Number of classification classes.
- **need\_cls\_layer** (*bool, default True*) – Whether to use NormDense output layer.

gluonfr.model\_zoo.**attention\_net164** (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 164 Model for face recognition.

**Parameters**

- **classes** (*int, -1*) – Number of classification classes.
- **need\_cls\_layer** (*bool, default True*) – Whether to use NormDense output layer.

gluonfr.model\_zoo.**attention\_net236** (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 236 Model for face recognition.

**Parameters**

- **classes** (*int, -1*) – Number of classification classes.
- **need\_cls\_layer** (*bool, default True*) – Whether to use NormDense output layer.

gluonfr.model\_zoo.**attention\_net452** (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 452 Model for face recognition.

**Parameters**

- **classes** (*int, -1*) – Number of classification classes.
- **need\_cls\_layer** (*bool, default True*) – Whether to use NormDense output layer.

gluonfr.model\_zoo.**attention\_net56** (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 56 Model for face recognition.

**Parameters**

- **classes** (*int*, -1) – Number of classification classes.
- **need\_cls\_layer** (*bool*, default *True*) – Whether to use NormDense output layer.

`gluonfr.model_zoo.attention_net92` (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet 92 Model for face recognition.

**Parameters**

- **classes** (*int*, -1) – Number of classification classes.
- **need\_cls\_layer** (*bool*, default *True*) – Whether to use NormDense output layer.

`gluonfr.model_zoo.get_attention_face` (*classes=-1, num\_layers=128, embedding\_size=512, need\_cls\_layer=True, \*\*kwargs*)  
AttentionNet Model for 112x112 face images from “Residual Attention Network for Image Classification” paper.

**Parameters**

- **classes** (*int*, -1) – Number of classification classes.
- **num\_layers** (*int*, 128) – Numbers of layers. Options are 56, 92, 128, 164, 236, 452.
- **embedding\_size** (*int*, 256) – Feature dimensions of the embedding layers.
- **need\_cls\_layer** (*bool*, default *True*) – Whether to use NormDense output layer.

`gluonfr.model_zoo.get_attention_net` (*classes, num\_layers, \*\*kwargs*)  
AttentionNet Model from “Residual Attention Network for Image Classification” paper.

**Parameters**

- **classes** (*int*,) – Number of classification classes.
- **num\_layers** (*int*) – Numbers of layers. Options are 56, 92, 128, 164, 236, 452.

`gluonfr.model_zoo.get_mobile_facenet` (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)

**Parameters**

- **classes** (*int*, -1) – Number of classification classes.
- **need\_cls\_layer** (*bool*, default *True*) – Whether to use NormDense output layer.

`gluonfr.model_zoo.get_mobile_facenet_re` (*classes=-1, need\_cls\_layer=True, \*\*kwargs*)

**Parameters**

- **classes** (*int*, -1) – Number of classification classes.
- **need\_cls\_layer** (*bool*, default *True*) – Whether to use NormDense output layer.

`gluonfr.model_zoo.get_model` (*name, \*\*kwargs*)  
Returns a model by name.

**Parameters**

- **name** (*str*) – Name of the model.
- **classes** (*int*) – Number of classes for the output layer.

- **ctx** (*Context*, *default CPU*) – The context in which to load the pretrained weights.

**Returns** The model.

**Return type** HybridBlock

`gluonfr.model_zoo.get_model_list()`

Get the entire list of model names in `model_zoo`.

**Returns** Entire list of model names in `model_zoo`.

**Return type** list of str

`gluonfr.model_zoo.get_se_resnet(num_layers, **kwargs)`

SE\_ResNet V1 model from “Deep Residual Learning for Image Recognition” paper. SE\_ResNet V2 model from “Identity Mappings in Deep Residual Networks” paper.

**Parameters**

- **version** (*int*) – Version of ResNet. Options are 1, 2.
- **num\_layers** (*int*) – Numbers of layers. Options are 18, 34, 50, 101, 152.
- **pretrained** (*bool*, *default False*) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default CPU*) – The context in which to load the pretrained weights.
- **root** (*str*, *default '~/.mxnet/models'*) – Location for keeping the model parameters.

`gluonfr.model_zoo.se_resnet101_v2(**kwargs)`

SE\_ResNet-101 V2 model from “Identity Mappings in Deep Residual Networks” paper.

**Parameters**

- **pretrained** (*bool*, *default False*) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default CPU*) – The context in which to load the pretrained weights.
- **root** (*str*, *default '~/.mxnet/models'*) – Location for keeping the model parameters.

`gluonfr.model_zoo.se_resnet152_v2(**kwargs)`

SE\_ResNet-152 V2 model from “Identity Mappings in Deep Residual Networks” paper.

**Parameters**

- **pretrained** (*bool*, *default False*) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default CPU*) – The context in which to load the pretrained weights.
- **root** (*str*, *default '~/.mxnet/models'*) – Location for keeping the model parameters.

`gluonfr.model_zoo.se_resnet18_v2(**kwargs)`

SE\_ResNet-18 V2 model from “Identity Mappings in Deep Residual Networks” paper.

**Parameters**

- **pretrained** (*bool*, *default False*) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default CPU*) – The context in which to load the pretrained weights.



- **root** (*str*, *default* `'~/mxnet/models'`) – Location for keeping the model parameters.

`gluonfr.model_zoo.se_resnet34_v2(**kwargs)`

SE\_ResNet-34 V2 model from “Identity Mappings in Deep Residual Networks” paper.

#### Parameters

- **pretrained** (*bool*, *default* `False`) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default* `CPU`) – The context in which to load the pretrained weights.
- **root** (*str*, *default* `'~/mxnet/models'`) – Location for keeping the model parameters.

`gluonfr.model_zoo.se_resnet50_v2(**kwargs)`

SE\_ResNet-50 V2 model from “Identity Mappings in Deep Residual Networks” paper.

#### Parameters

- **pretrained** (*bool*, *default* `False`) – Whether to load the pretrained weights for model.
- **ctx** (*Context*, *default* `CPU`) – The context in which to load the pretrained weights.
- **root** (*str*, *default* `'~/mxnet/models'`) – Location for keeping the model parameters.

## 4.4.5 gluonfr.metrics

Metrics used in training a face recognition model.

### API Reference

This module provides metric used in face recognition

**class** `gluonfr.metrics.FaceVerification`

Compute confusion matrix of 1:1 problem in face verification or other fields. Use `update()` to collect the outputs and compute distance in each batch, then use `get()` to compute the confusion matrix and accuracy of the val dataset.

#### Parameters

- **nfolds** (*int*, *default* `is 10`) –
- **thresholds** (*ndarray*, *default* `is None.`) – Use `np.arange` to generate thresholds. If `thresholds=None`, `np.arange(0, 2, 0.01)` will be used for euclidean distance.
- **far\_target** (*float*, *default* `is 1e-3.`) – This is used to get the verification accuracy of expected far.
- **dist\_type** (*int*, *default* `is 0.`) – Option value is `{0, 1}`, 0 for euclidean distance, 1 for cosine similarity. Here for cosine distance, we use  $1 - \text{cosine}$  as the final distances.

## 4.4.6 gluonfr.utils

We implemented a broad range of utility functions which cover visualization, file handler, download and training helpers.

## **Visualization**

---

`plot_accuracy`

---

`plot_roc`

---

## **API Reference**

## g

`gluonfr.data`, 11

`gluonfr.loss`, 13

`gluonfr.metrics`, 21

`gluonfr.model_zoo`, 17

`gluonfr.nn.basic_blocks`, 12

`gluonfr.utils.viz`, 22



## A

ArcLoss (class in gluonfr.loss), 13  
 ASoftmax (class in gluonfr.loss), 15  
 attention\_net128 () (in module gluonfr.model\_zoo), 18  
 attention\_net164 () (in module gluonfr.model\_zoo), 18  
 attention\_net236 () (in module gluonfr.model\_zoo), 18  
 attention\_net452 () (in module gluonfr.model\_zoo), 18  
 attention\_net56 () (in module gluonfr.model\_zoo), 18  
 attention\_net92 () (in module gluonfr.model\_zoo), 19  
 AttentionNet (class in gluonfr.model\_zoo), 17  
 AttentionNetFace (class in gluonfr.model\_zoo), 17

## C

CenterLoss (class in gluonfr.loss), 15  
 COCOLoss (class in gluonfr.loss), 16  
 ContrastiveLoss (class in gluonfr.loss), 15  
 CosLoss (class in gluonfr.loss), 14

## F

FaceVerification (class in gluonfr.metrics), 21  
 FrBase (class in gluonfr.nn.basic\_blocks), 12  
 FRTrainRecordDataset (class in gluonfr.data), 11  
 FRValDataset (class in gluonfr.data), 11

## G

get\_attention\_face () (in module gluonfr.model\_zoo), 19  
 get\_attention\_net () (in module gluonfr.model\_zoo), 19  
 get\_loss () (in module gluonfr.loss), 13  
 get\_loss\_list () (in module gluonfr.loss), 13  
 get\_mobile\_facenet () (in module gluonfr.model\_zoo), 19

get\_mobile\_facenet\_re () (in module gluonfr.model\_zoo), 19  
 get\_model () (in module gluonfr.model\_zoo), 19  
 get\_model\_list () (in module gluonfr.model\_zoo), 20  
 get\_se\_resnet () (in module gluonfr.model\_zoo), 20  
 GitLoss (class in gluonfr.loss), 16  
 gluonfr.data (module), 11  
 gluonfr.loss (module), 13  
 gluonfr.metrics (module), 21  
 gluonfr.model\_zoo (module), 17  
 gluonfr.nn.basic\_blocks (module), 12  
 gluonfr.utils.viz (module), 22

## L

L2Softmax (class in gluonfr.loss), 14  
 LGMLoss (class in gluonfr.loss), 15

## M

MobileFaceNet (class in gluonfr.model\_zoo), 18  
 MPSPLoss (class in gluonfr.loss), 16

## N

NormDense (class in gluonfr.nn.basic\_blocks), 12

## R

RingLoss (class in gluonfr.loss), 14

## S

se\_resnet101\_v2 () (in module gluonfr.model\_zoo), 20  
 se\_resnet152\_v2 () (in module gluonfr.model\_zoo), 20  
 se\_resnet18\_v2 () (in module gluonfr.model\_zoo), 20  
 se\_resnet34\_v2 () (in module gluonfr.model\_zoo), 21  
 se\_resnet50\_v2 () (in module gluonfr.model\_zoo), 21

SELayer (*class in gluonfr.nn.basic\_blocks*), 12  
SVXSoftmax (*class in gluonfr.loss*), 16

## T

TripletLoss (*class in gluonfr.loss*), 13