SIFED 2019

Cross-model Caricature Face Recognition via Dynamic Multi-task Learning

https://github.com/hengxyz/cari-visual-recognition-via-multitask-learning.git

Zuheng Ming

Jean-Christophe

Muzzamil Luqman

06, Juin 2019,

Plan

- Introduction
- Methode
- Evaluation
- Conclusion

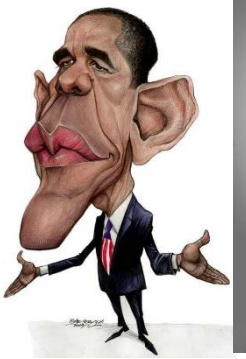
Introduction

- Caricature-Visual Face Recognition



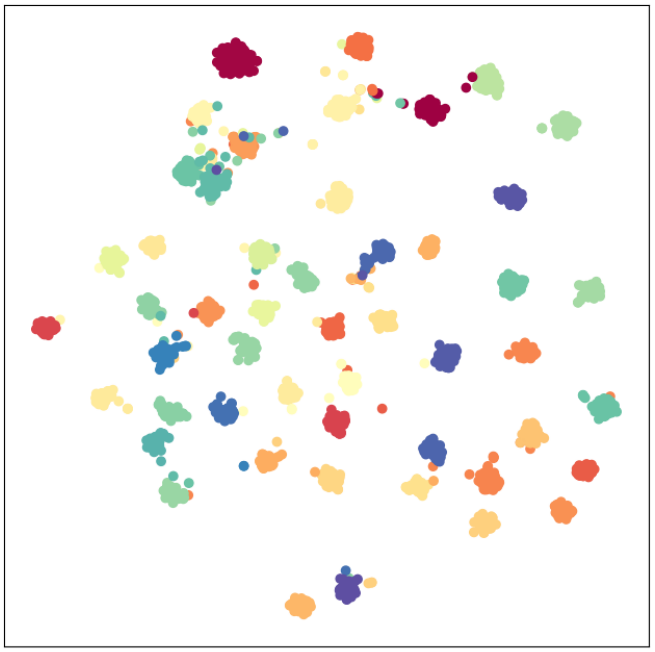





Introduction

- Caricature classification by visual model



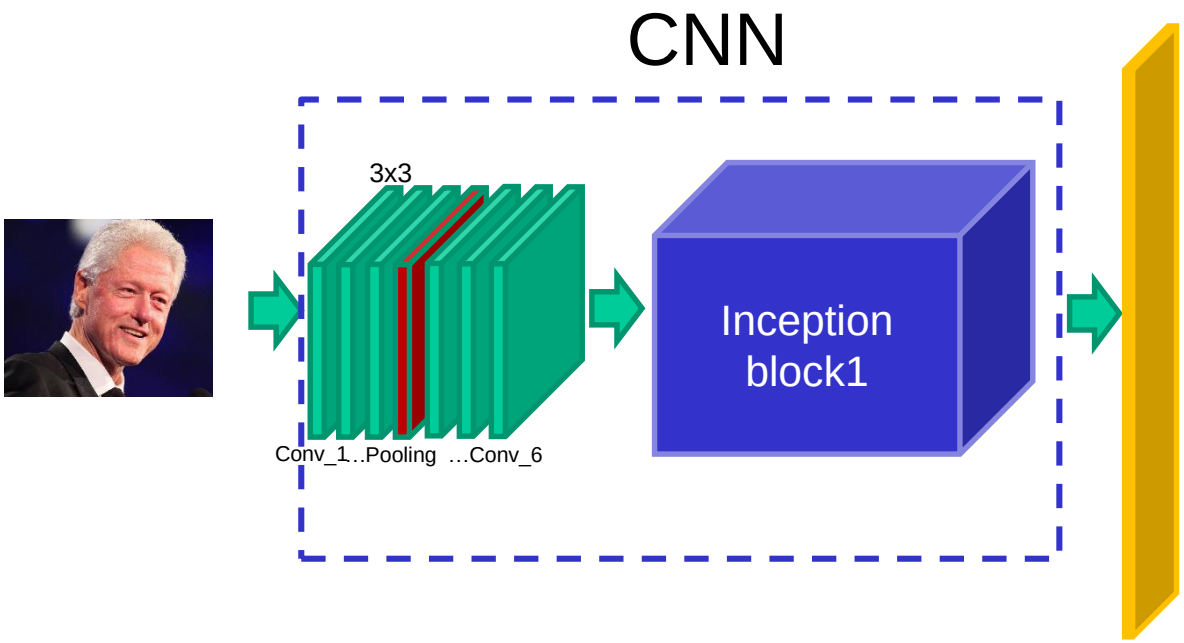
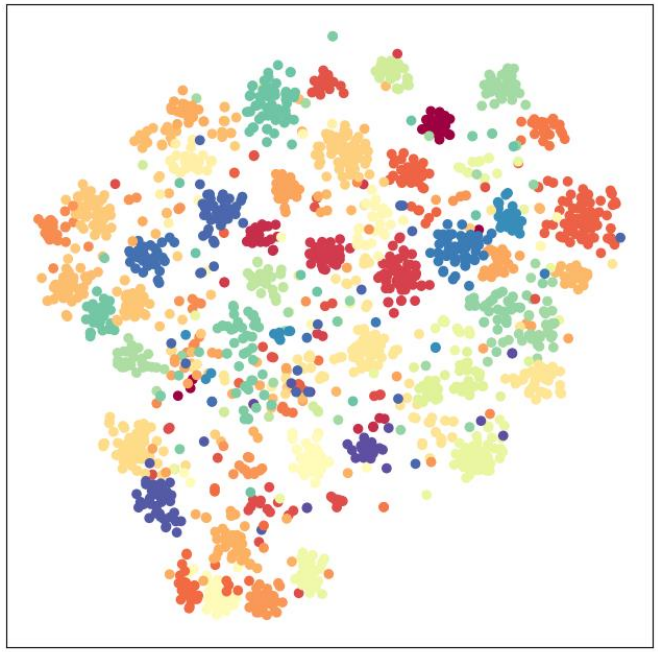


Fig 1. Caricature classification (t-SNE)

Introduction

- Caricature classification by visual model



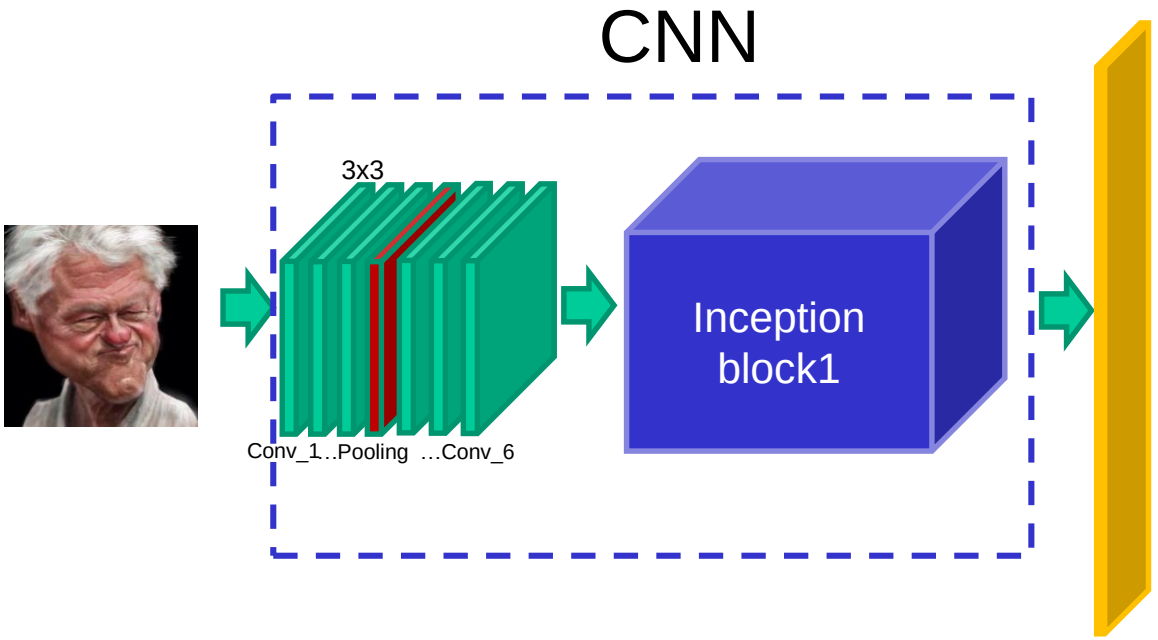


Fig 2. Caricature classification by visual model (t-SNE)

Introduction

- Visual face classification by caricature recogntion model

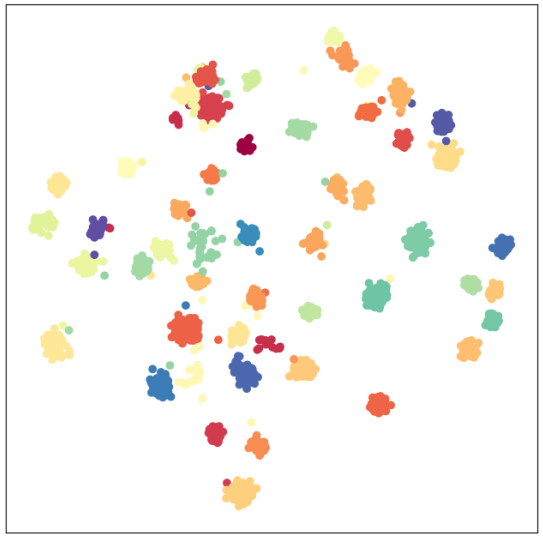


Fig 3. Visual face classification

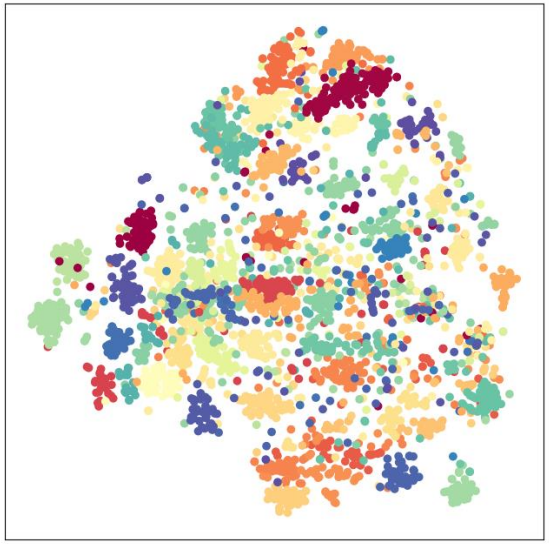
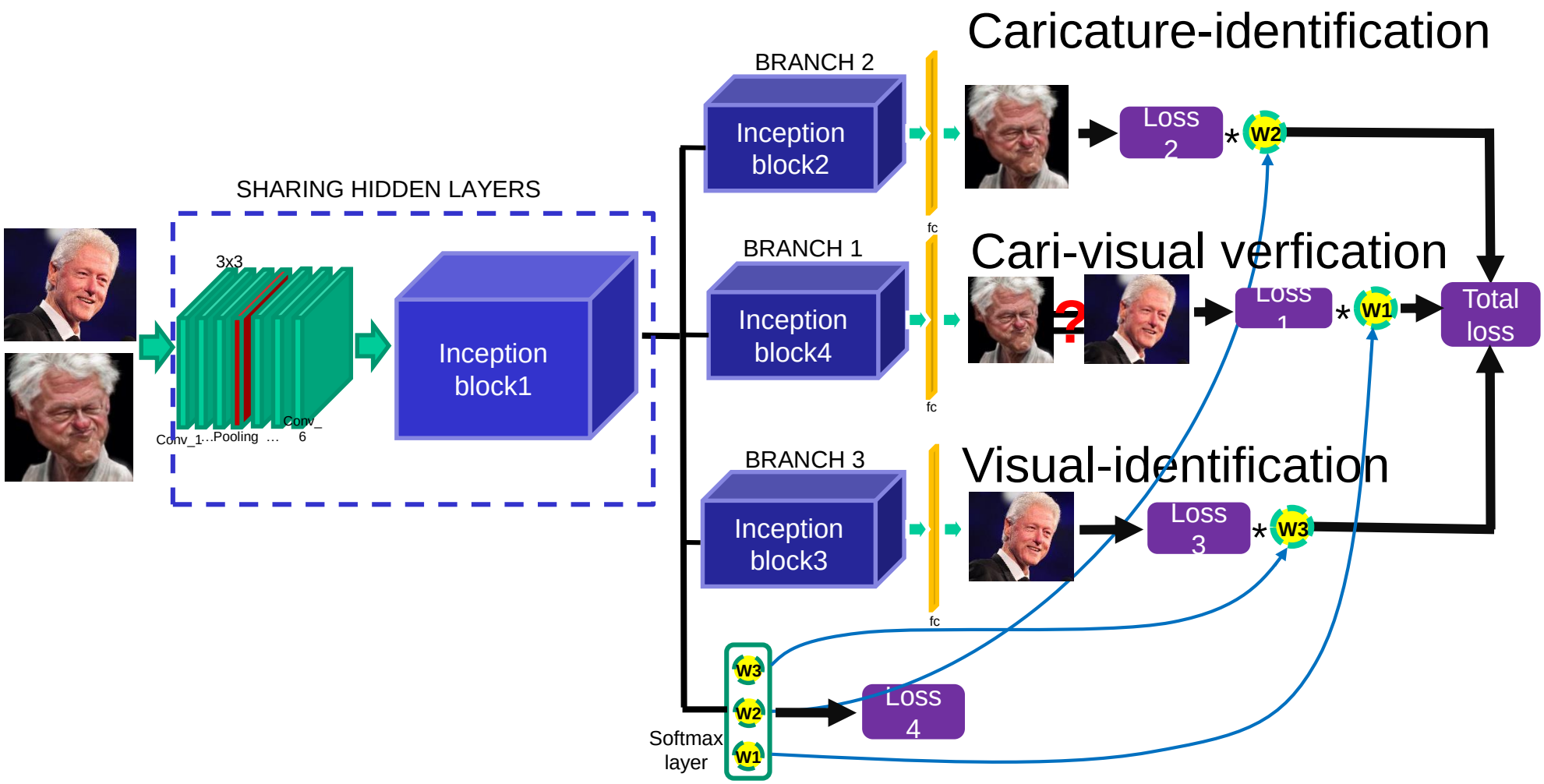


Fig 4. Visual face classification by caricature Recogntion model

Dynamic Multi-task learning for caricature-visual face recognition



Total loss function

- Total loss of the Multi-task learning is the sum of the weighted loss of each task:

$$\mathcal{L}(\mathbf{X};\Theta;\Psi) = \sum_{i=1}^{T} w_i(\Psi)\mathcal{L}_i(\mathbf{X}_i;\Theta_i) \qquad (1)$$

Where L is total loss, Wi is the weight of each task and Li is the loss of each task, T is the number of the task, here is three.

Total loss function

- Loss L1 for caricature recognition is the softmax loss, i.e. cross-entropy loss:

$$\mathcal{L}_{s1}(\mathbf{X}_1;\Theta_1) = \sum_{k=1}^{K} -y_k \log P(y_k = 1|\mathbf{X}_1,\theta_k)$$
$$= -\sum_{k=1}^{K} y_k \log \frac{e^{f^{\theta_k}(\mathbf{X}_1)}}{\sum_{k'}^{K} e^{f^{\theta_{k'}}(\mathbf{X}_1)}} \quad (3)$$

Where Ls1 is softmax loss of caricature recognition task, K is the number of the identities of caricatures.

- Loss L3 as L1 for visual face recognition is also the softmax loss

Total loss function

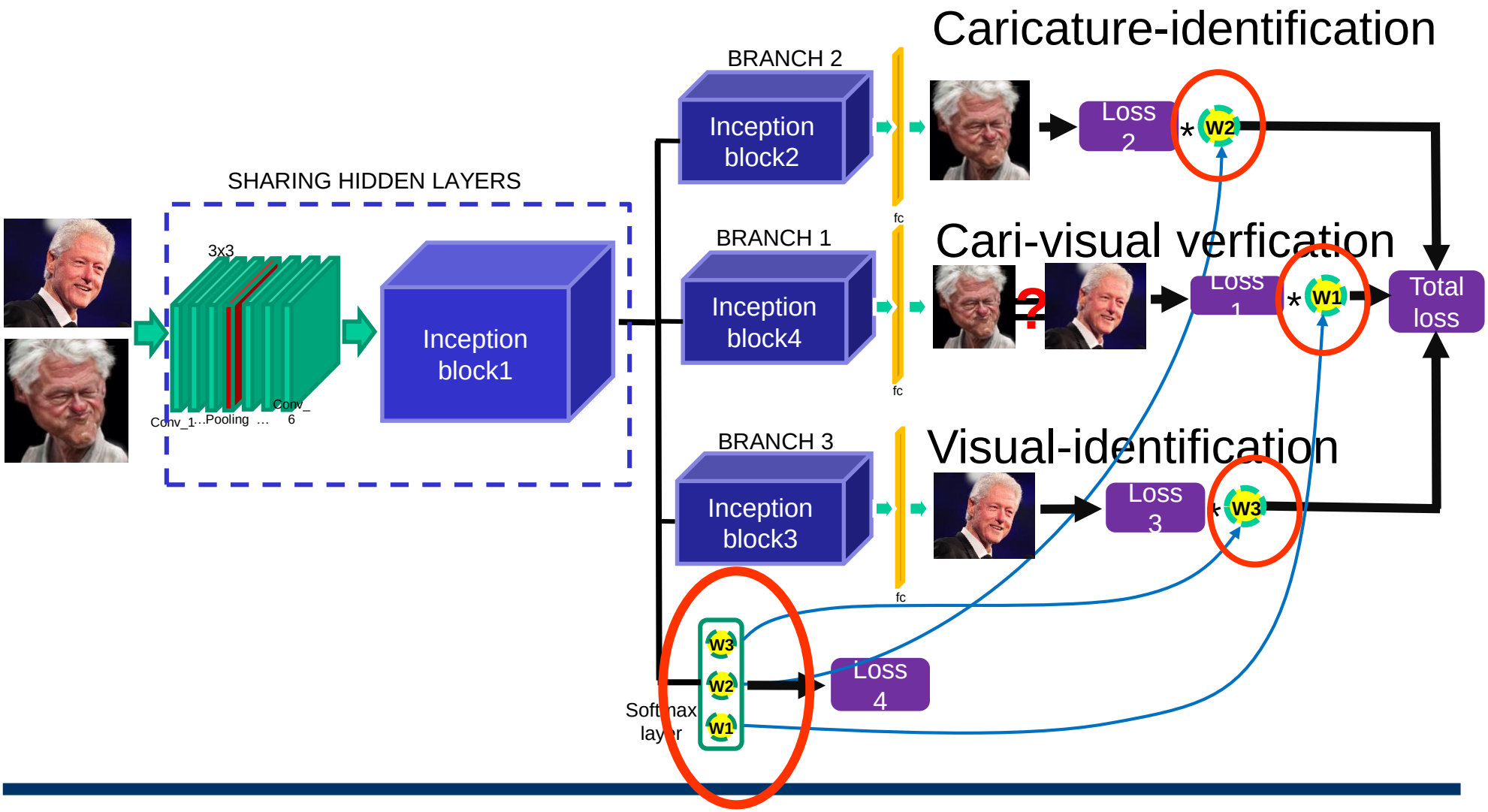
- Loss L2 for cari-visual face recognition is the center-loss [1]

$$\mathcal{L}_c(X_1;\Theta_1) = \|X_1 - C_{y_k}\| \qquad (4)$$

Where the Cyk is the center of the class which X1 belonging to.

[1] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In European Conference on Computer Vision. Springer, 499–515.

Dynamic weights of tasks



Dynamic weights

- How to set weights of tasks in Multi-task learning?
  - Static weights (the weights are fix during the training)

    a) manually setting

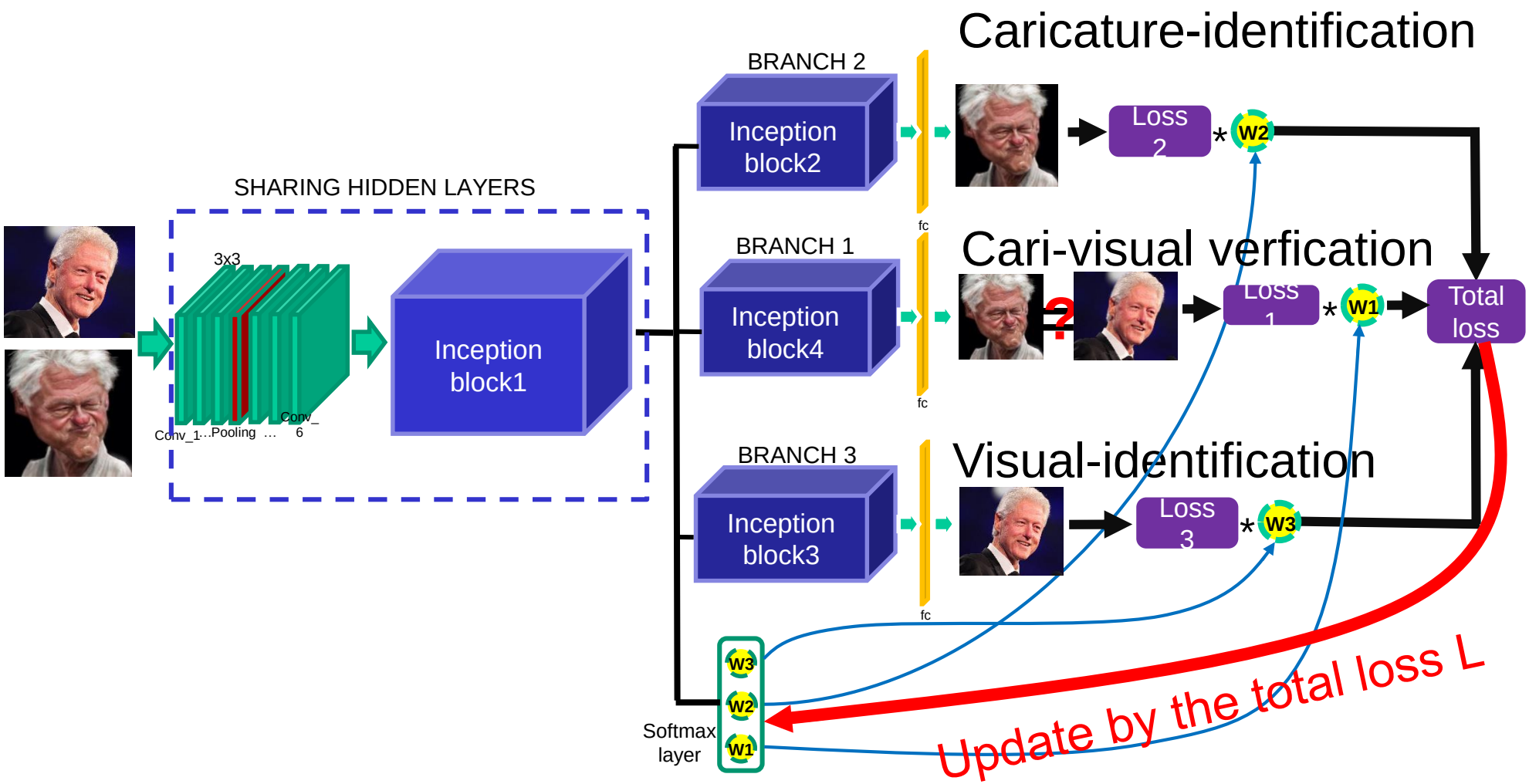
    b) greedy search
  - Dynamic weights (the weights can be update dynamically during the training)

    a) naive dynamic weight [2]

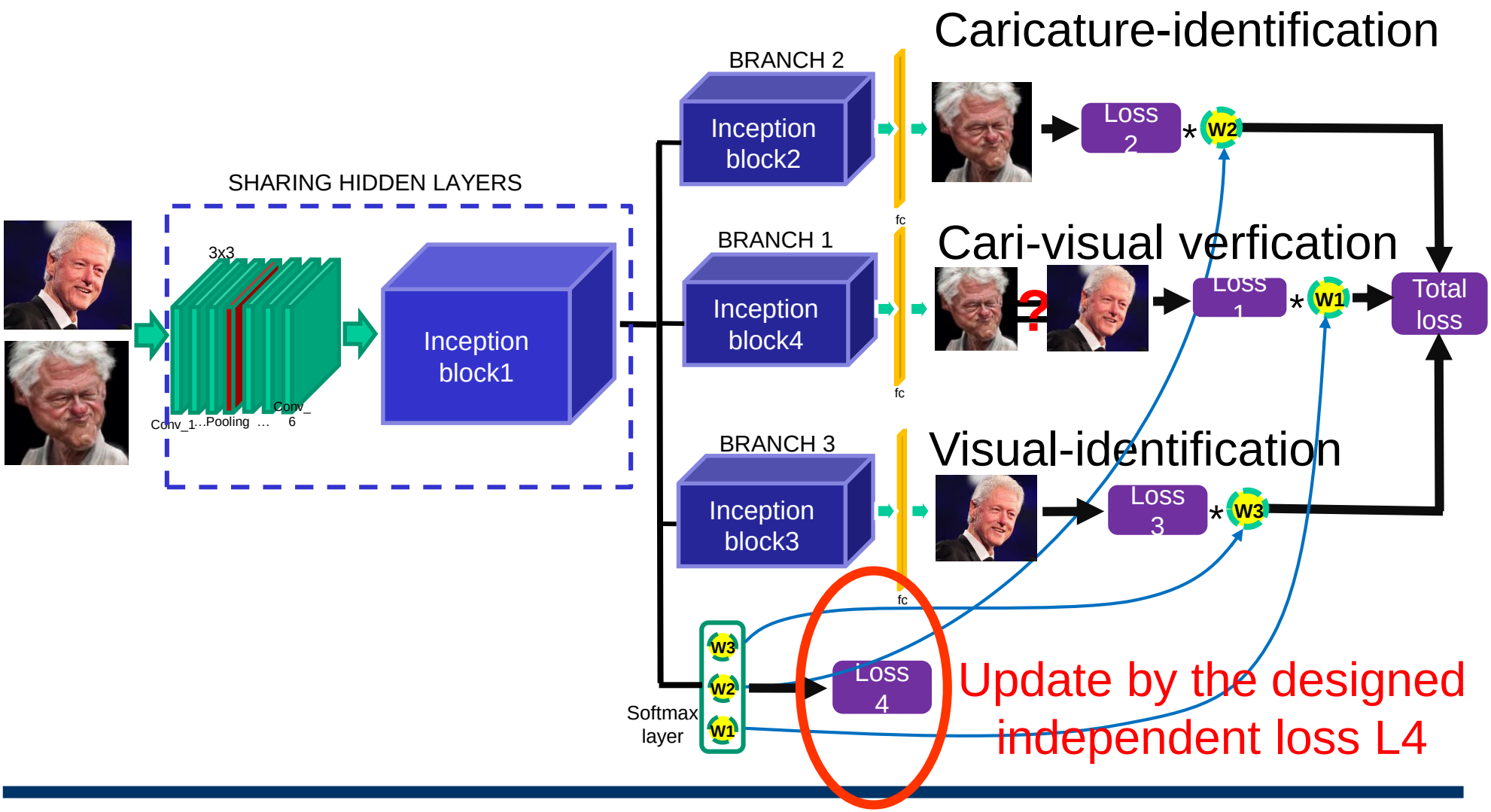
    b) our proposed dynamic weight

[2] Xi Yin and Xiaoming Liu. 2018. Multi-task convolutional neural network for pose-invariant face recognition. IEEE Transactions on Image Processing 27, 2 (2018), 964–975.

Naive Dynamic weights of tasks

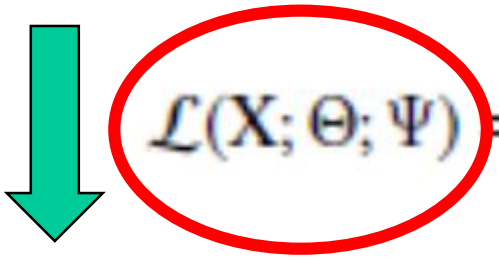


Proposed Dynamic weights of tasks



Naive Dynamic weights v.s. Ours

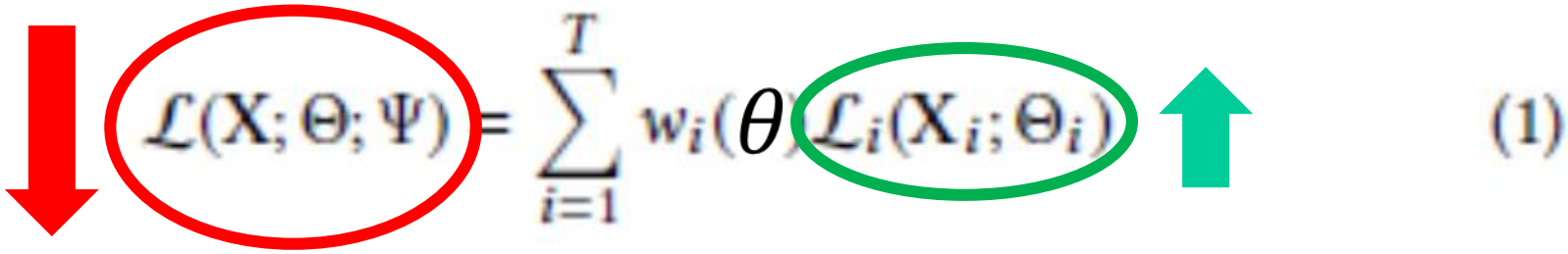
- Naive Dynamic weights [2] updating by the total loss L:



$$\mathcal{L}(\mathbf{X};\Theta;\Psi) = \sum_{i=1}^{T} w_i(\theta)\mathcal{L}_i(\mathbf{X}_i;\Theta_i) \quad (1)$$

Naive Dynamic weights v.s. Ours

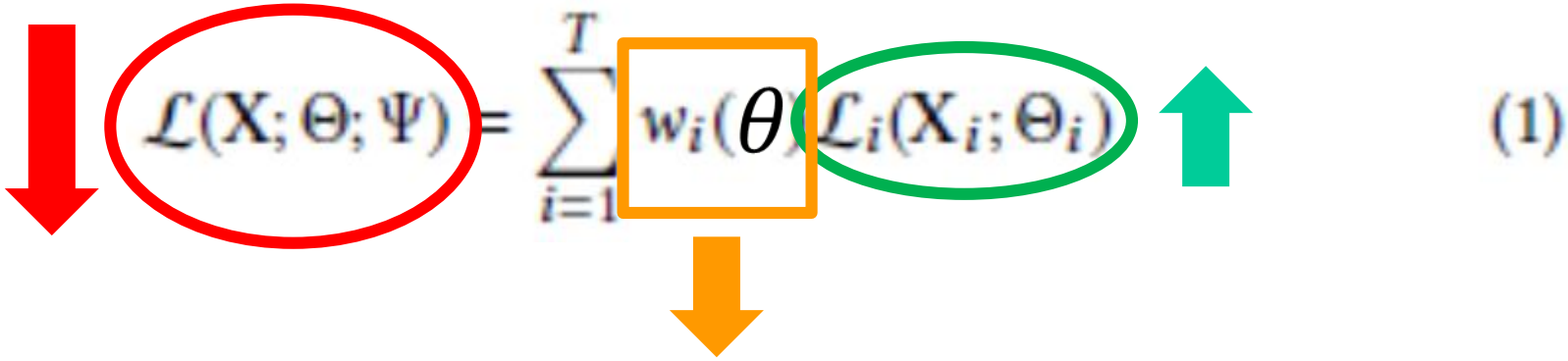
- Naive Dynamic weights [2] updating by the total loss L:



$$\mathcal{L}(\mathbf{X};\Theta;\Psi) = \sum_{i=1}^{T} w_i(\theta)\mathcal{L}_i(\mathbf{X}_i;\Theta_i) \qquad (1)$$

Naive Dynamic weights v.s. Ours

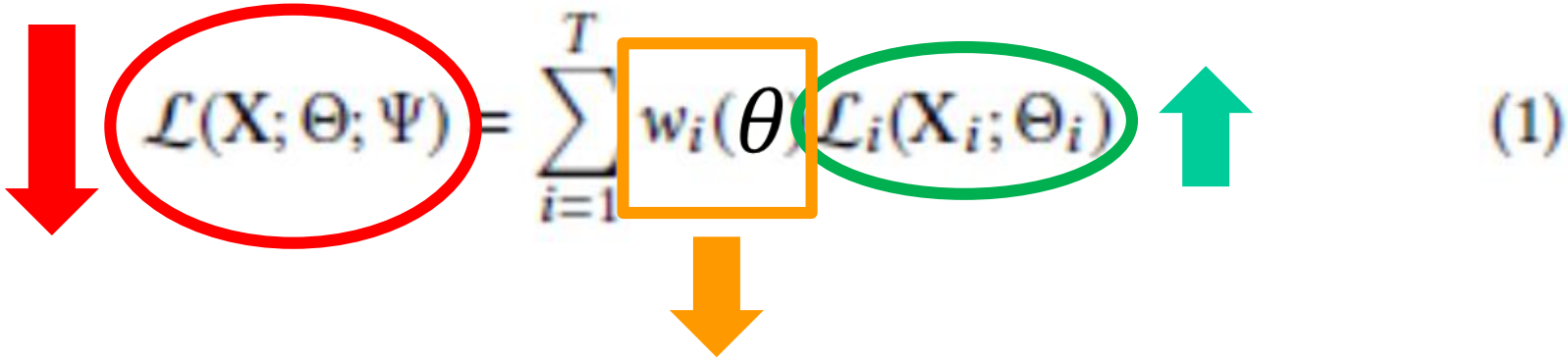
- Naive Dynamic weights [2] updating by the total loss L:



$$\mathcal{L}(\mathbf{X};\Theta;\Psi) = \sum_{i=1}^{T} w_i(\theta)\mathcal{L}_i(\mathbf{X}_i;\Theta_i) \quad (1)$$

Naive Dynamic weights v.s. Ours

- Naive Dynamic weights [2] updating by the total loss L:



$$\mathcal{L}(\mathrm{X};\Theta;\Psi) = \sum_{i=1}^{T} w_i(\theta)\mathcal{L}_i(\mathrm{X}_i;\Theta_i) \quad (1)$$

Weakness: The hard task with high loss always updated by the small weight of task, which lead the hard task cannot be well trained.

Naive Dynamic weights v.s. Ours

- Proposed Dynamic weights updating:

$$\mathcal{L}_4(Z;\Psi) = \sum_{i=1}^{T} \frac{w_i(\psi_i)}{\mathcal{L}_i} \quad s.t. \quad \sum w_i = 1 \qquad (7)$$

Naive Dynamic weights v.s. Ours

- Proposed Dynamic weights updating:

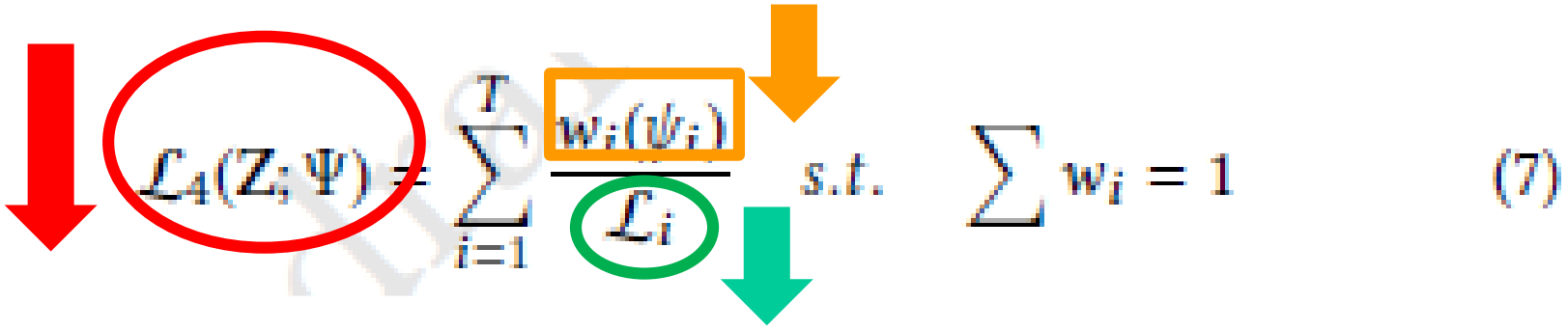
Not include in the θ as the Naive

$$\mathcal{L}_4(Z;\Psi) = \sum_{i=1}^{T} \frac{w_i(\psi_i)}{\mathcal{L}_i} \quad s.t. \quad \sum w_i = 1 \qquad (7)$$

  - Here the parameters φ for generating the weights of tasks is not a part of networks parameters θ,

Naive Dynamic weights v.s. Ours

- Proposed Dynamic weights updating:



$$\mathcal{L}_4(Z;\Psi) = \sum_{i=1}^{T} \frac{w_i(\psi_i)}{\mathcal{L}_i} \quad s.t. \quad \sum w_i = 1 \qquad (7)$$

- Advantage : the easy task with low loss updated/assigned by a small weight, hard task with large weights, which can drive the multi-task learning to train the hard task sufficienly.

Evaluations

- Dataset:
  - CaVI dataset[3]:

contains caricatures and visual images of 205 identities, which has 5091 caricatures ranging from 10-15 images per identity and 6427 visual images ranging from 10-15 images per identity.

  - WebCaricature dataset [4]:

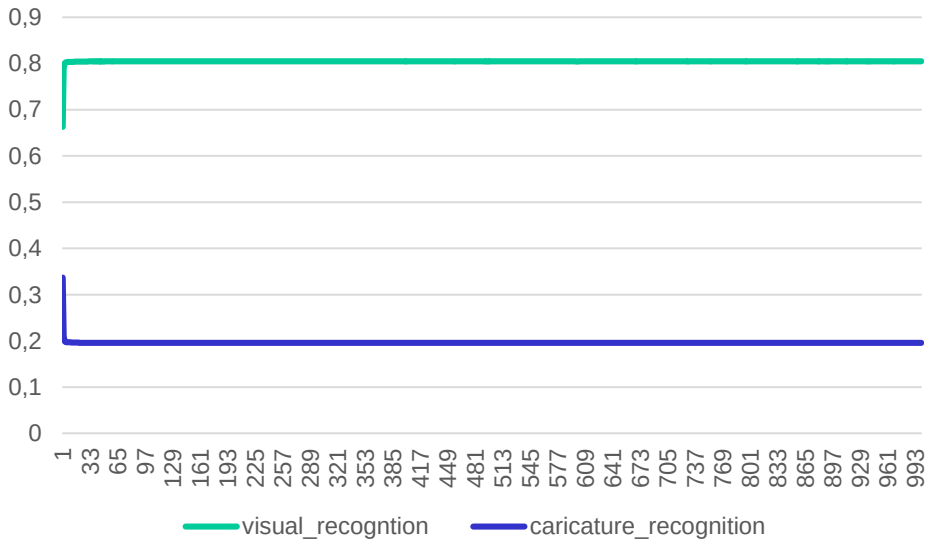
a large caricature dataset of 252 people with 6024 caricatures and 5974 photos

[3] Jatin Garg, Skand Vishwanath Peri, Himanshu Tolani, and Narayanan C Krishnan. 2018. Deep Cross Modal Learning for Caricature Verification and Identification (CaVINet). arXiv preprint arXiv:1807.11688 (2018).

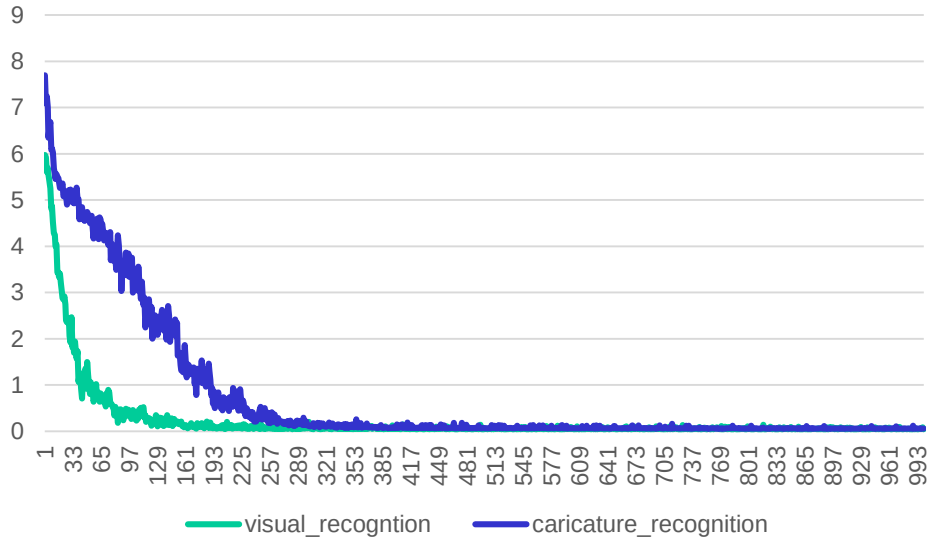
[4] Jing Huo, Wenbin Li, Yinghuan Shi, Yang Gao, and Hujun Yin. 2018. Web-Caricature: a benchmark for caricature recognition. In British Machine Vision Conference.

Evaluations

- Naive Dynamic (Toy experiments with 2 tasks) - CAVI:



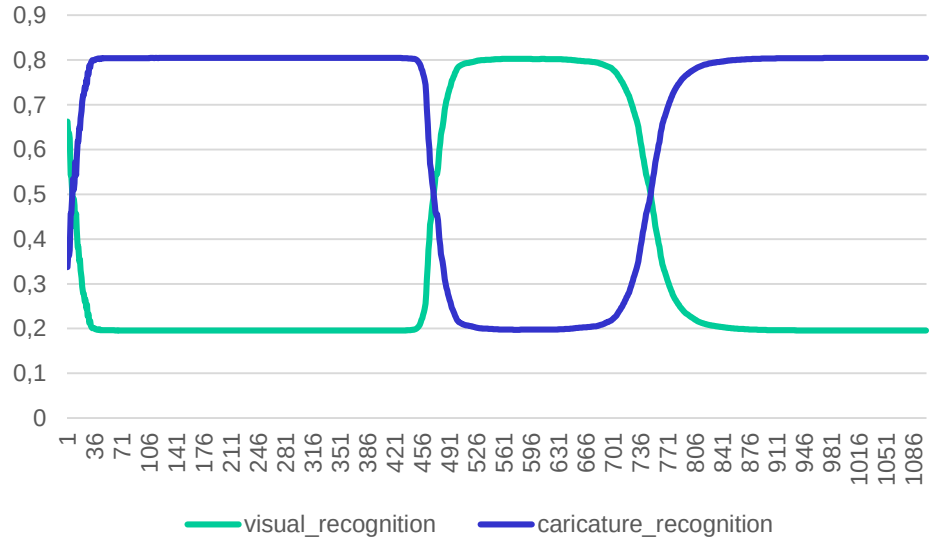
weight



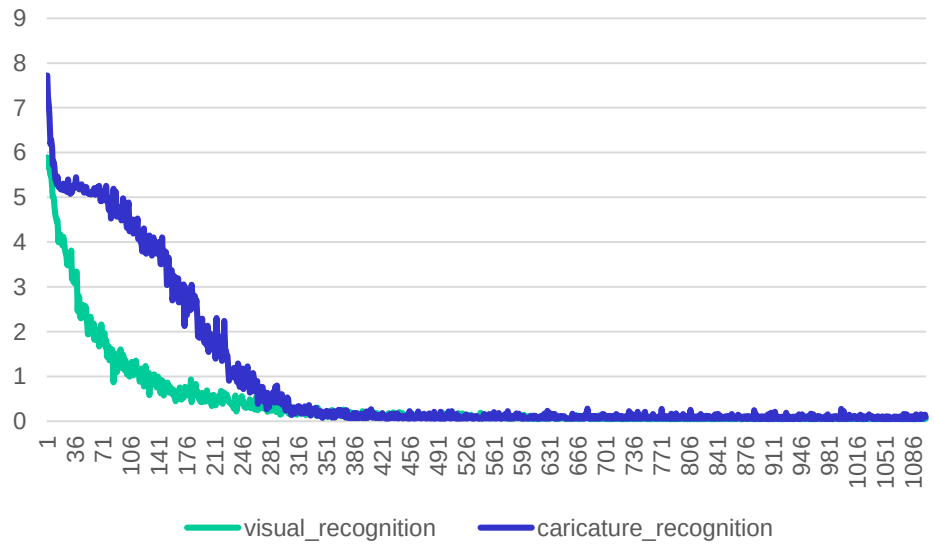
loss

Evaluations

- Proposed Dynamic (Toy experiments with 2 tasks):



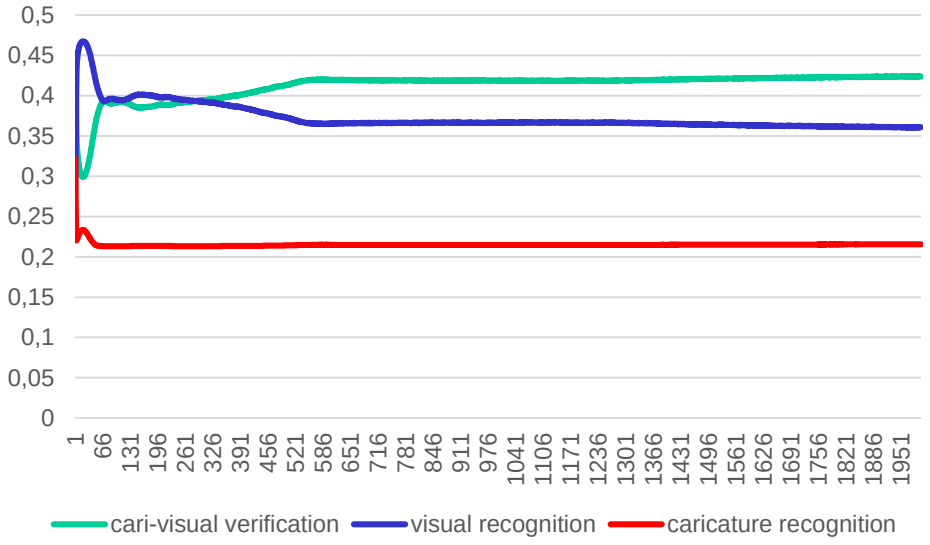
weight



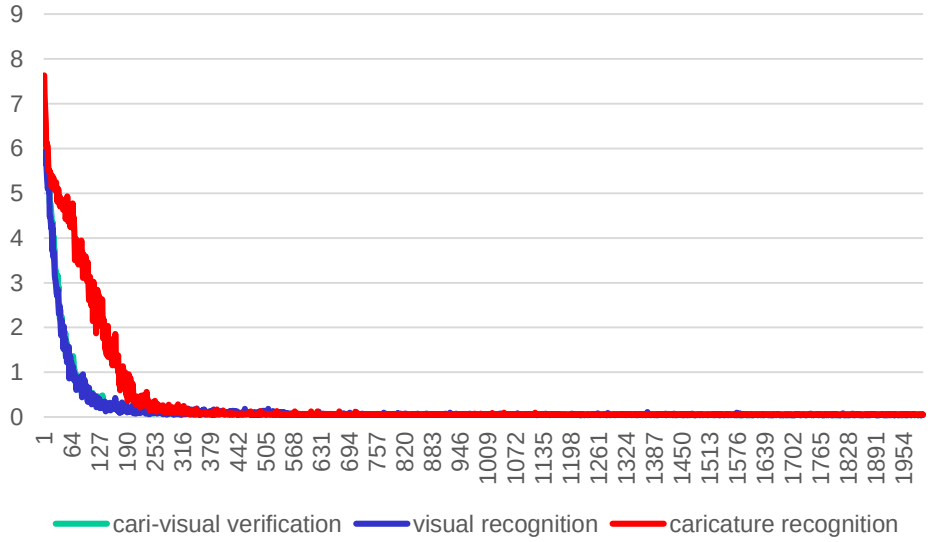
loss

Evaluations

- Naive Dynamic (3 tasks):



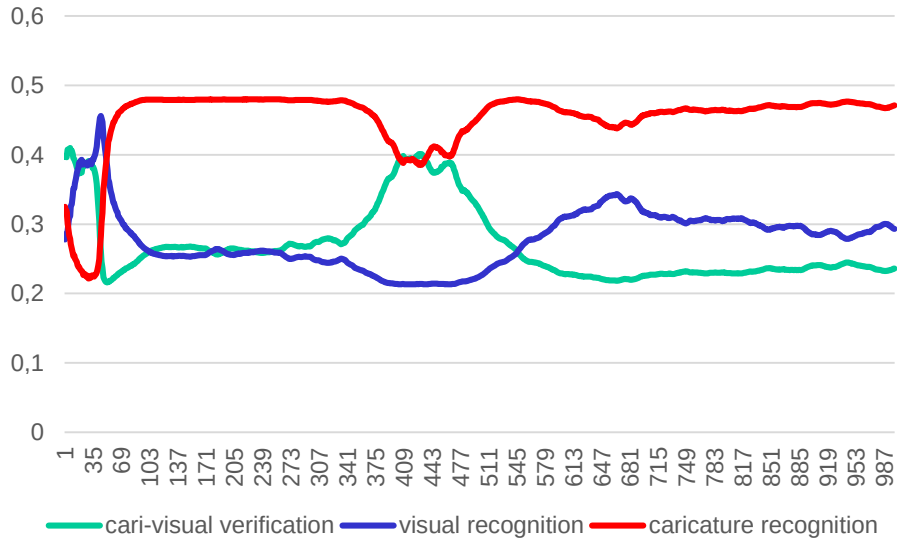
weight



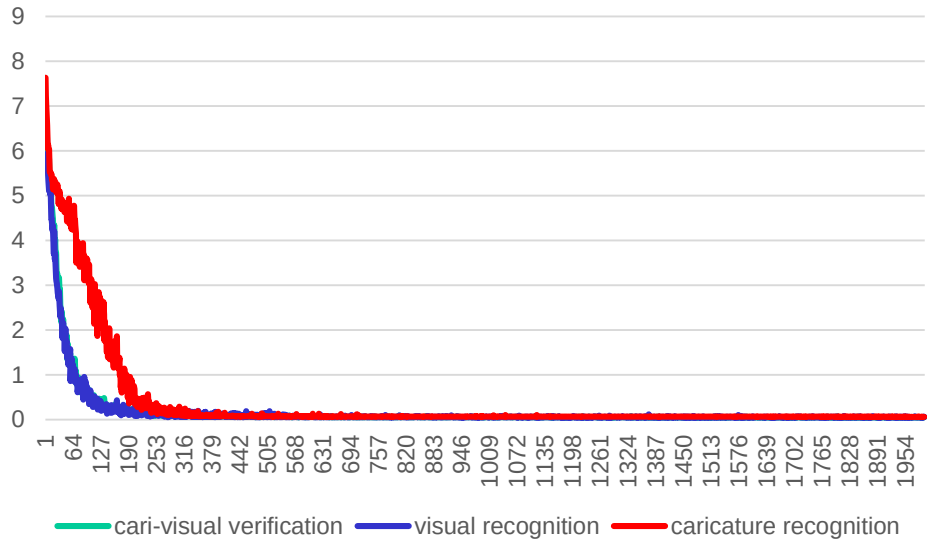
loss

Evaluations

- Proposed Dynamic (3 task):



weight



loss

Evaluations

- Performance on CAVI dataset:

Table 1: The evaluation of caricature-visual face verification, caricature identification and visual face identification (accuracy%) on dataset CaVI.

Method	Verification	Visual-id	Cari-id	C2V	V2C
CaVINet	91.06	94.50	85.09	-	-
CaVINet(TW)	84.32	85.16	86.02	-	-
CaVINet(w/o ortho)	86.01	93.46	80.43	-	-
CaVINet(shared)	88.59	90.56	81.23	-	-
CaVINet(visual)	88.58	92.16	83.56	-	-
Navie Dynamic	93.80	97.60	75.80	62.80	61.90
Ours (Single-verif)	92.46	-	-	-	-
Ours (Single-visual)	-	98.10	-	53.60	-
Ours (Single-cari)	-	-	78.20	-	41.8
Ours (Multi-task)	**94.92**	**98.35**	**85.61**	**64.39**	**80.04**

Evaluations

- Performance on WebCaricature dataset:

Table 2: The evaluation of caricature-visual face verification in terms of the validation rate (%) on dataset WebCaricature.

Method	VAL@FAR=0.1%	VAL@FAR=1%	AUC
SIFT-Land-ITML	5.08±1.82	18.07±4.72	0.841±0.018
VGG-Eye-PCA	21.42±2.02	40.28±2.91	0.896±0.013
VGG-Eye-ITML	18.97±3.90	41.72±5.83	0.911±0.014
VGG-Box-PCA	28.42±2.04	55.53±2.76	0.946±0.009
VGG-Box	34.94±5.06	57.22±6.50	0.954±0.010
Navie Dynamic	38.39±4.58	79.69±1.3	**0.987±0.002**
Ours (Single-verif)	42.10±3.05	**84.52±0.80**	0.948±0.002
Ours (Multi-task)	**45.82±1.65**	83.20±2.00	0.961±0.004

Evaluations

- Performance on WebCaricature dataset:

Table 3: The evaluation of Caricature to Photo identification (C2P) on dataset WebCaricature.

Method	Rank-1(%)	Rank-10(%)
SIFT-Land-KCSR	24.87 ± 1.50	61.57 ± 1.37
VGG-Eye-PCA	35.07 ± 1.84	71.64 ± 1.32
VGG-Eye-KCSR	39.76 ± 1.60	75.38 ± 1.34
VGG-Box-PCA	49.89 ± 1.97	84.21 ± 1.08
VGG-Box-KCSR	55.41 ± 1.41	87.00 ± 0.92
Navie Dynamic	86.00 ± 1.70	98.21 ± 1.08
Ours (Single-verif)	85.55 ± 1.30	96.31 ± 0.08
Ours (Multi-task)	**87.30 ± 1.20**	**99.21 ± 1.07**

Table 4: The evaluation of Photo to Caricature (P2C) identification (C2P) on dataset WebCaricature.

Method	Rank-1(%)	Rank-10(%)
SIFT-Land-KCSR	23.42 ± 1.57	69.95 ± 2.34
VGG-Eye-PCA	36.18 ± 3.24	68.95 ± 3.25
VGG-Eye-KCSR	40.67 ± 3.61	75.77 ± 2.63
VGG-Box-PCA	50.59 ± 2.37	82.15 ± 1.31
VGG-Box-KCSR	55.53 ± 2.17	86.86 ± 1.42
Navie Dynamic	82.80 ± 1.60	97.81 ± 0.88
Ours (Single-verif)	81.70 ± 2.60	95.25 ± 1.08
Ours (Multi-task)	**84.00 ± 1.60**	**99.01 ± 1.2**

Examples of failures

False positive (CAVI)

False negative (CAVI)

False positive (WebCari)

False negative (WebCari)

Conclusions

- The proposed dynamic weight module without introducing the additional hyperparameters can lead the multi-task learning to train the hard task primarily instead of being stuck in the overtraining of the easy task.
- No hyperparameters are introduced
- Easy to realise on other deep mult-task learning frameworks, such as the detection framework based on CNNs, e.g. Faster R-CNN, Mask RCNN

Thanks !