Google Landmark Recognition Challenge

Albert Cheng

- About this competition
- Difficulties
- My solution
- Others solution
- Reference

About Competition

- Google Landmark Recognition Challenge (Label famous and not so famous landmarks in images)

About Competition

- Google Landmark Recognition Challenge (Label famous and not so famous landmarks in images)

About Competition

- Training data

	id	url	landmark_id
0	97c0a12e07ae8dd5	http://lh4.ggpht.com/-f8xYA5l4apw/RSziSQVaABI/...	6347
1	650c989dd3493748	https://lh5.googleusercontent.com/-PUnMrX7oOyA...	12519
2	05e63ca9b2cde1f4	http://mw2.google.com/mw-panoramio/photos/medi...	264
3	08672eddcb2b7c93	http://lh3.ggpht.com/-9fgSxDYwhHA/SMvGEoltKTI/...	13287
4	fc49cb32ef7f1e89	http://lh6.ggpht.com/-UGAXxvPbr98/S-jGZbyMIPI/...	4018

- Testing data

	id	url
0	cb9998b8cdaf6235	https://lh3.googleusercontent.com/-q8B91vDlQZY...
1	30728cf6e50a6bc6	https://lh3.googleusercontent.com/-91gJSKTgv5Q...
2	16afbc86b710337d	https://lh3.googleusercontent.com/-GHZdXuf2wMg...
3	d29b2166cf522450	https://lh3.googleusercontent.com/-cWDnYNQhyws...
4	dd5c03b20c21cfba	https://lh3.googleusercontent.com/-PSLN6BloM-k...

About Competition

- Google Landmark Recognition Challenge (Label famous and not so famous landmarks in images)
- Evaluation: Global Average Precision (GAP)

$$GAP = \frac{1}{M} \sum_{i=1}^{N} P(i) rel(i)$$

where:

- N is the total number of predictions returned by the system, across all queries
- M is the total number of queries with at least one landmark from the training set visible in it (note that some queries may not depict landmarks)
- $P(i)$ is the precision at rank i
- $rel(i)$ denotes the relevance of prediciton i: it's 1 if the i-th prediction is correct, and 0 otherwise

About Competition

- Evaluation: Global Average Precision (GAP)

$$GAP = \frac{1}{M} \sum_{i=1}^{N} P(i) rel(i)$$

id	landmark
a1f23b	A 0.33
a23d4b	A 0.67
12e22d	
4e76aa	C 0.11
923dd	B 0.91

→

id	landmark
923dd	B 0.91
a23d4b	A 0.67
a1f23b	A 0.33
4e76aa	C 0.11
12e22d	

→

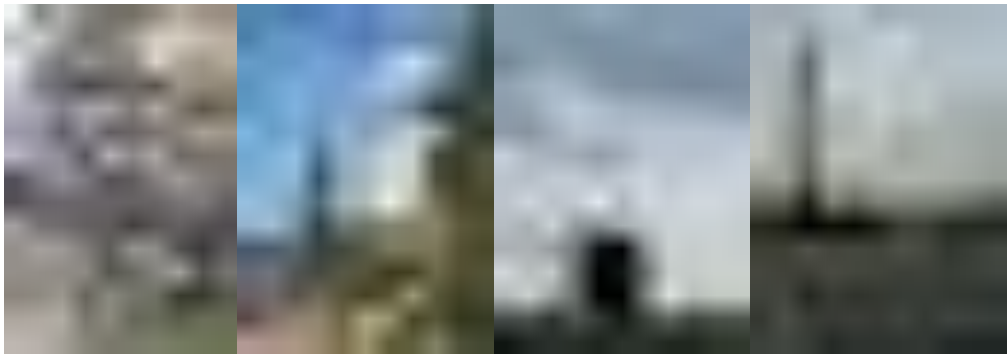
id	landmark	Precision
923dd	B 0.91	1
a23d4b	A 0.67	0
a1f23b	A 0.33	0.5
4e76aa	C 0.11	1
12e22d		

Difficulties

- Too many classes in training set
  - 14951 classes
- Very low resolution image in training set
- Same landmark_id with different angle of shooting
- Very imbalanced
  - Largest number of same landmark_id: 50010
  - 166 landmark_id have only 1 image
- Testing set contains new classes

Difficulties

- Too many classes in training set
  - 14951 classes
- Very low resolution image in training set (15*10)
- Same landmark_id with different angle of shooting
- Very imbalanced
  - Largest number of same landmark_id: 50010
  - 166 landmark_id have only 1 image
- Testing set contains new classes



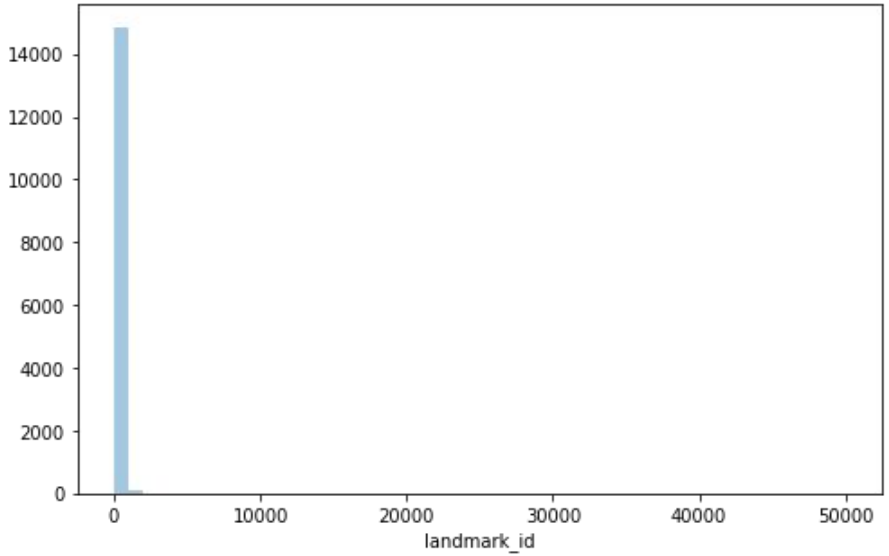
Difficulties

- Too many classes in training set
  - 14951 classes
- Very low resolution image in training set
- Same landmark_id with different angle of shooting
- Very imbalanced
  - Largest number of same landmark_id: 50010
  - 166 landmark_id have only 1 image
- Testing set contains new classes



Difficulties

- Too many classes in training set
  - 14951 classes
- Very low resolution image in training set
- Same landmark_id with different angle of shooting
- Very imbalanced
  - Largest number in one landmark_id: 50010
  - 166 landmark_id have only 1 image
- Testing set contains new classes



Difficulties

- Too many classes in training set
  - 14951 classes
- Very low resolution image in training set
- Same landmark_id with different angle of shooting
- Very imbalanced
  - Largest number of same landmark_id: 50010
  - 166 landmark_id have only 1 image
- Testing set contains new classes

My solution

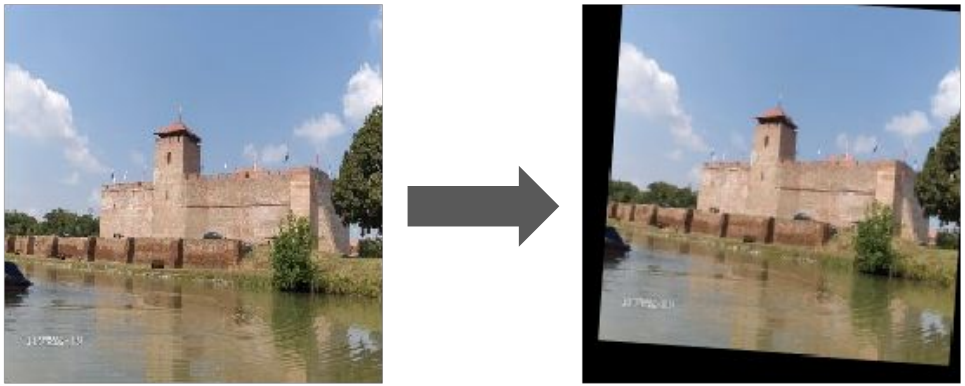
- Preprocessing
- ResNet
- FaceNet (triplet loss)
- ResNet + Hadamard Projection

My solution: Preprocessing

- Resize
- Image Augmentation
  - Shift
  - Rotate
- Imbalance Sampling
  - Under sample
  - Over sample

My solution: Preprocessing

- Resize
- Image Augmentation
  - Shift
  - Rotate
- Imbalance Sampling
  - Under sample
  - Over sample



My solution: Preprocessing

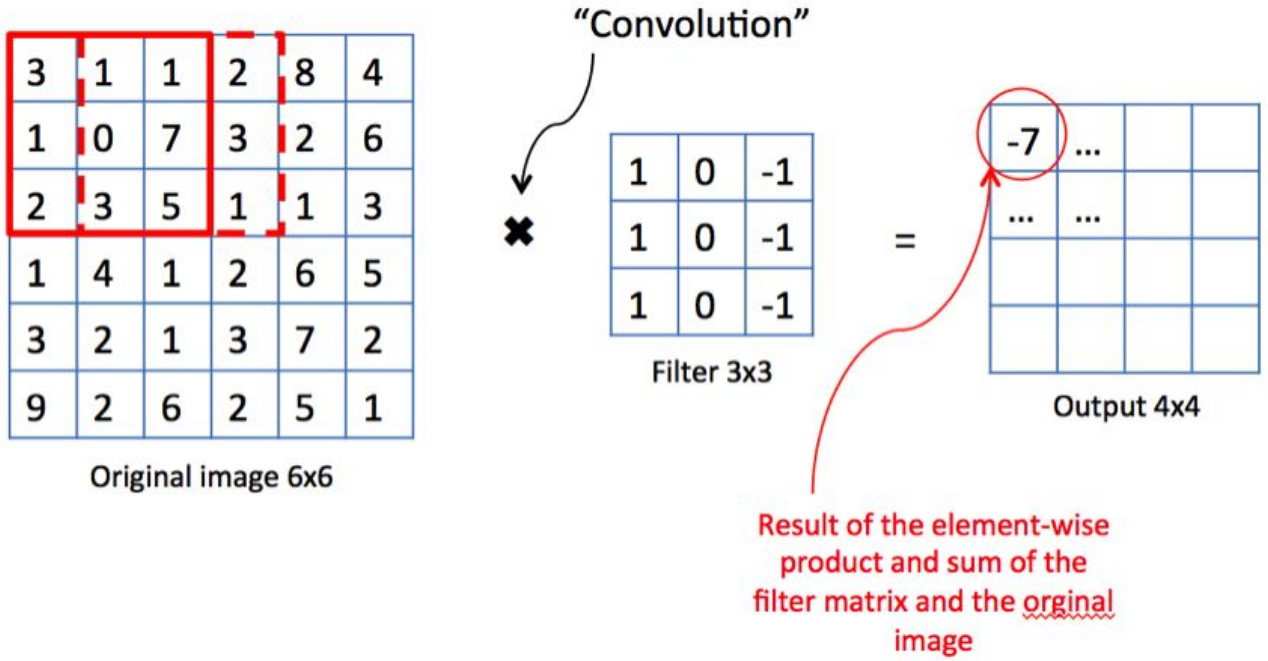
- Resize
- Image Augmentation
  - Shift
  - Rotate
- Imbalance Sampling
  - Under sample
  - Over sample

My solution

- Preprocessing
- ResNet
- FaceNet (triplet loss)
- ResNet + Hadamard Projection

My solution: Resnet + FC

- Convolutional Neural Network



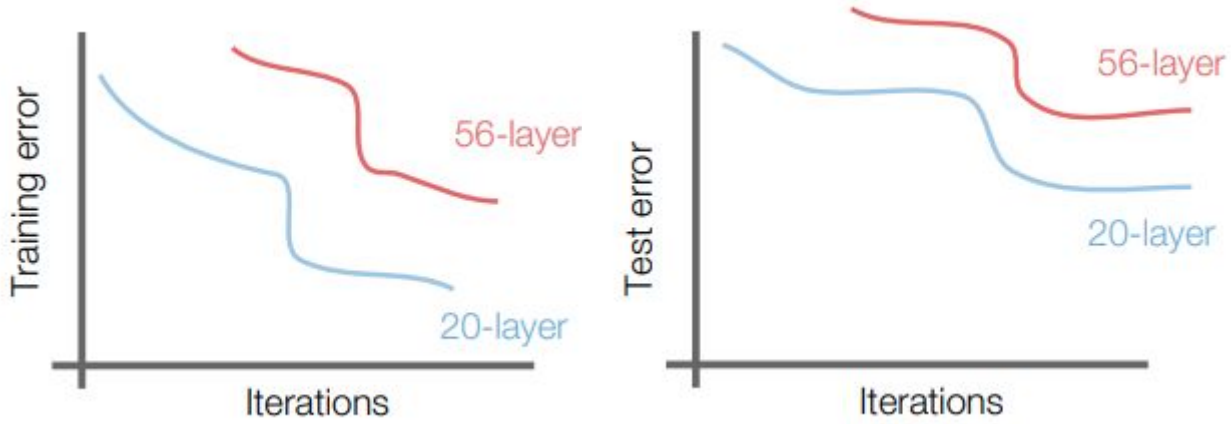
My solution: Resnet + FC

- Convolutional Neural Network

  - Accepts a volume of size $W_1 \times H_1 \times D_1$
  - Requires four hyperparameters:
    - Number of filters K,
    - their spatial extent F,
    - the stride S,
    - the amount of zero padding P.
  - Produces a volume of size $W_2 \times H_2 \times D_2$ where:
    - $W_2 = (W_1 - F + 2P)/S + 1$
    - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
    - $D_2 = K$
  - With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.

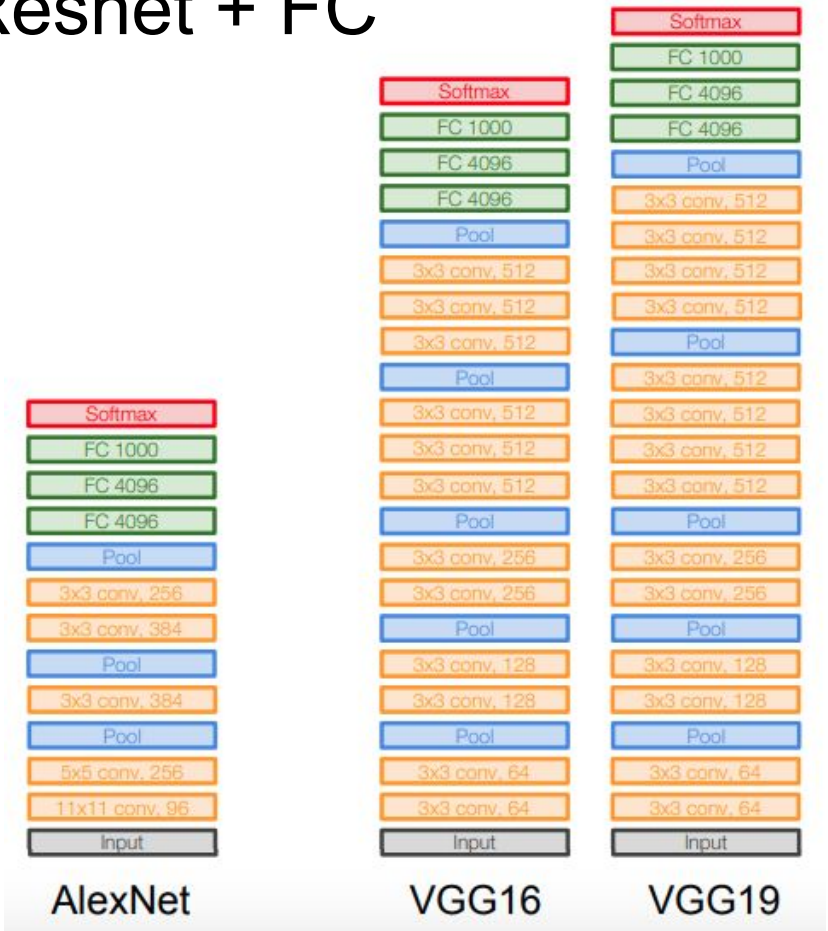
My solution: Resnet + FC

- Deeper is Better?



My solution: Resnet + FC

- VGG net



My solution: Resnet + FC

- VGG family

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

My solution: Resnet + FC

- Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

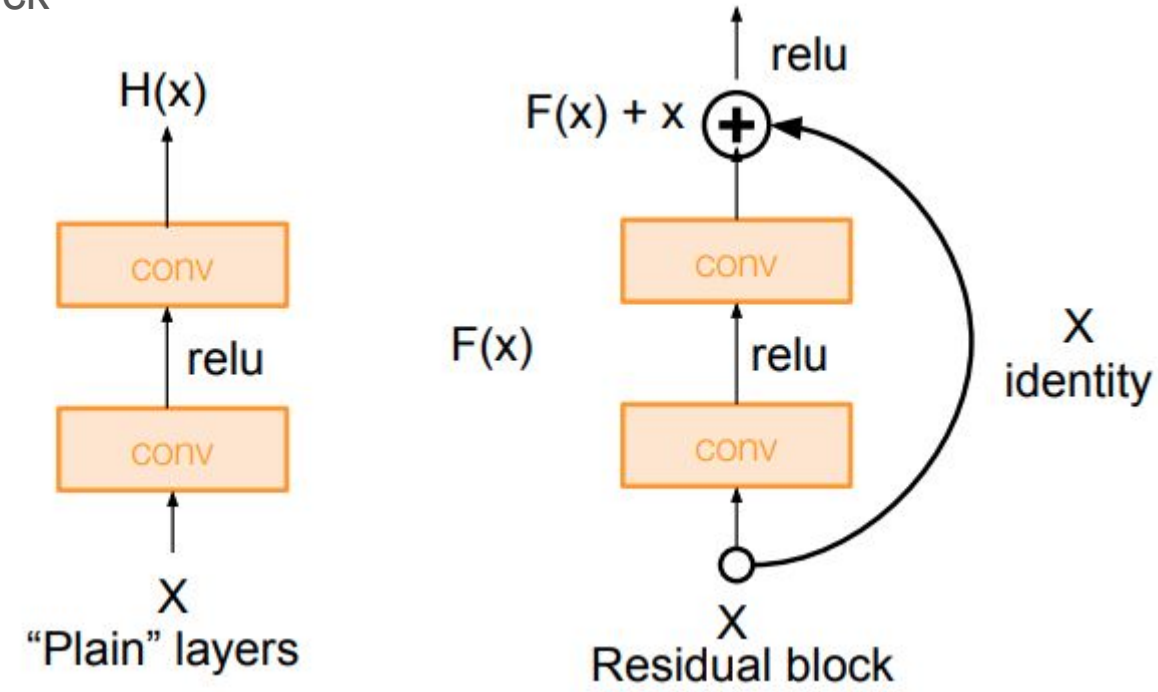
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

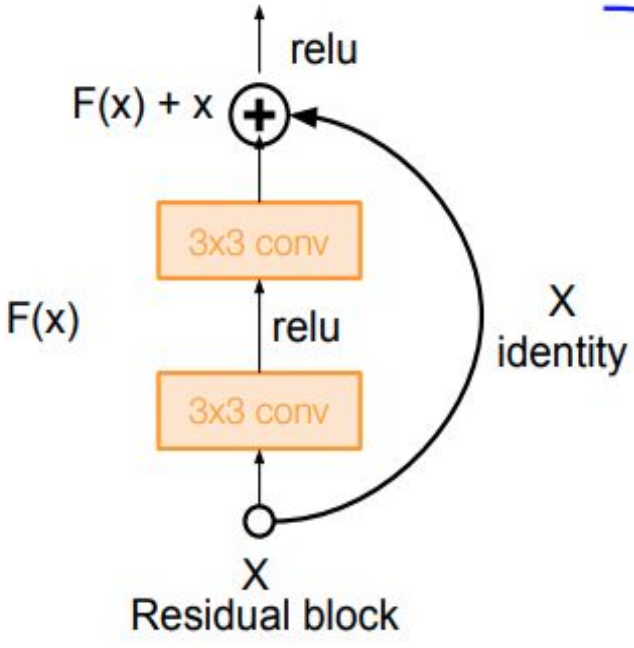
My solution: Resnet + FC

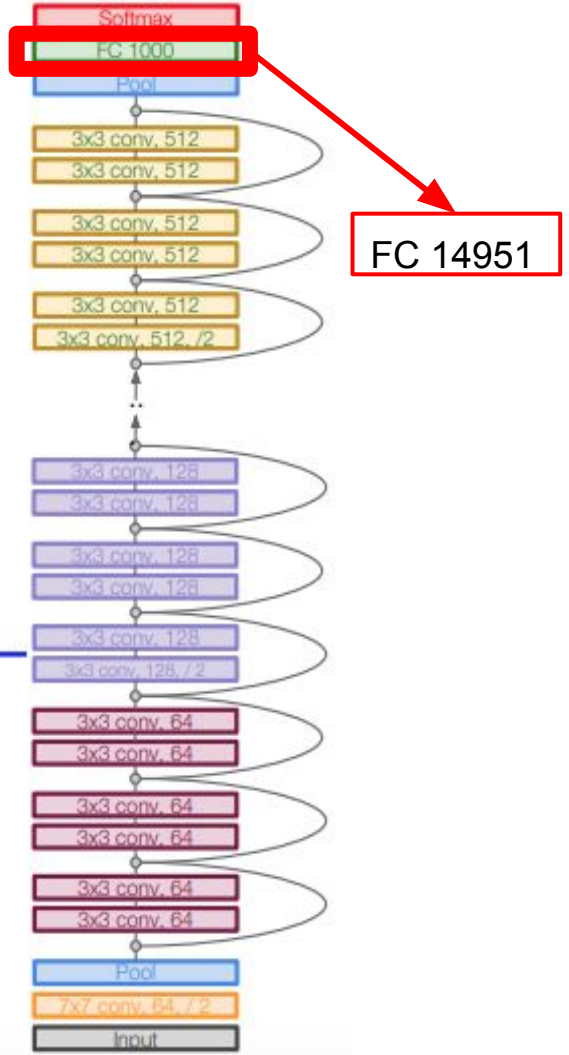
- Residual Block



My solution: Resnet + FC

- ResNet





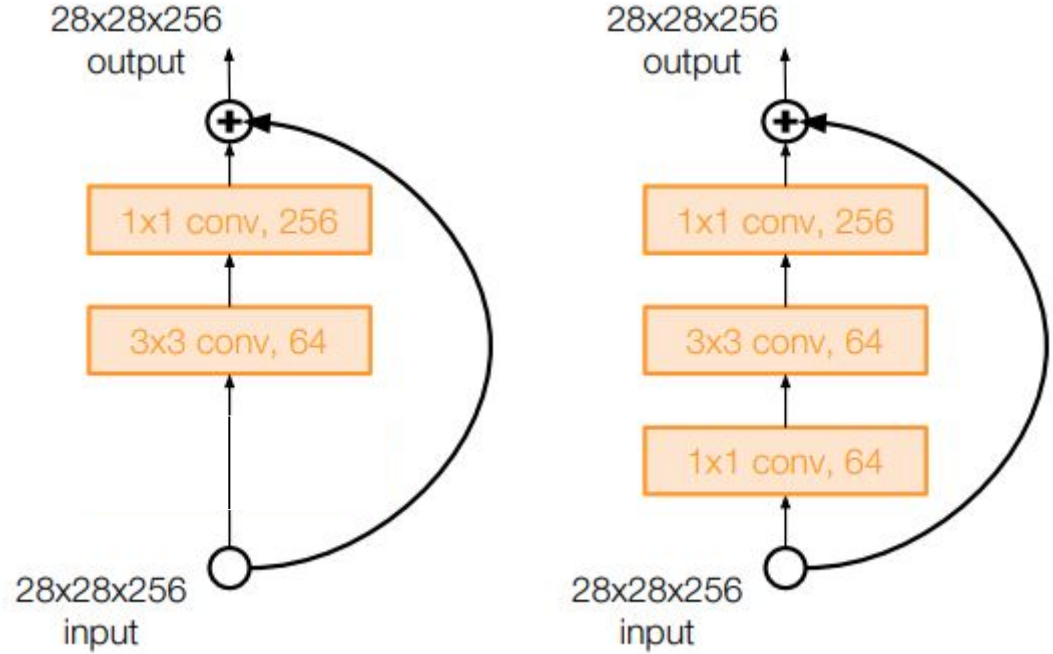
My solution: Resnet + FC

- Resnet family

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

My solution: Resnet + FC

- Bottleneck



Without Bottleneck		With Bottleneck	
64x3x3x256	147k	64x1x1x256	16k
256x1x1x64	16k	64x3x3x64	37k
		256x1x1x64	16k

My solution: Resnet + FC

- Resnet + Full Connection

Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 197, 197, 3)	12
resnet50 (Model)	(None, 1, 1, 2048)	23587712
dense_1 (Dense)	(None, 1, 1, 14951)	30634599

Total params: 54,222,323
Trainable params: 54,169,197
Non-trainable params: 53,126

My solution: Resnet + FC

- Resnet[Conv_5_2] + Average_pooling + FC

Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 197, 197, 3)	12
resnet50 (Model)	multiple	23587712
average_pooling2d_1 (Average	(None, 4, 4, 2048)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_1 (Dense)	(None, 14951)	489929319

Total params: 513,517,043
Trainable params: 513,463,917
Non-trainable params: 53,126

My solution: Resnet + FC

- Resample (drop rare events)

Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 197, 197, 3)	12
resnet50 (Model)	multiple	23587712
average_pooling2d_1 (Average	(None, 4, 4, 2048)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_1 (Dense)	(None, 6632)	217324008

Total params: 240,911,732
Trainable params: 240,858,606
Non-trainable params: 53,126

My solution

- Preprocessing
- ResNet and DenseNet + FC
- FaceNet (triplet loss)
- ResNet + Hadamard Projection

My Solution: FaceNet

- Triplet Loss

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T} . \quad (1)$$

The loss that is being minimized is then $L =$

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ . \quad (2)$$

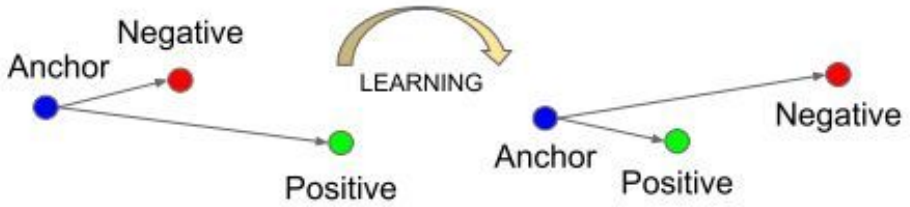


Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

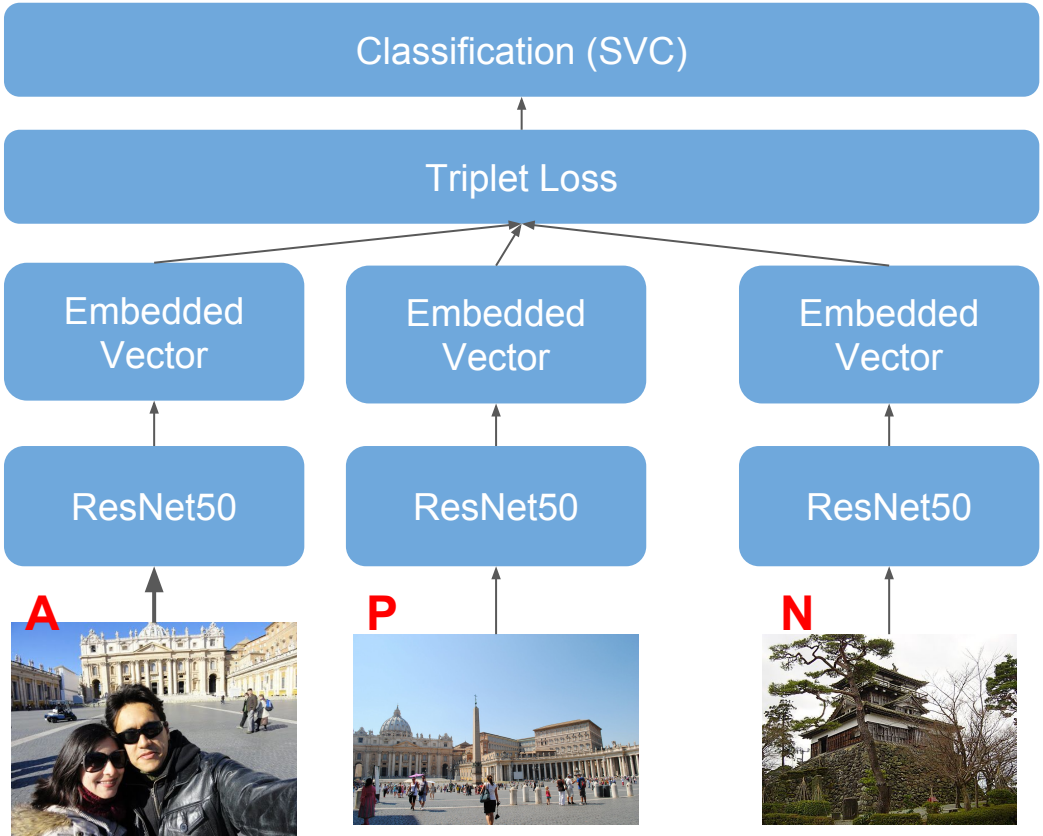
My Solution: FaceNet

- Structure

Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

My Solution: FaceNet

- Architecture



My Solution: FaceNet

- Classification

My solution

- Preprocessing
- ResNet and DenseNet + FC
- FaceNet (triplet loss)
- ResNet + Hadamard Projection

My solution: Hadamard Projection

- Hadamard Matrix

$$H_1 = [1],$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

and

$$H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} = H_2 \otimes H_{2^{k-1}},$$

for $2 \leq k \in N$, where $\otimes$ denotes the Kronecker product.

My solution: Hadamard Projection

- ResNet + Hadamard

Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 197, 197, 3)	12
resnet50 (Model)	multiple	23587712
average_pooling2d_1 (Average	(None, 4, 4, 2048)	0
flatten_1 (Flatten)	(None, 32768)	0
hadamard_classifier_1 (Hadam	(None, 14951)	14952
activation_50 (Activation)	(None, 14951)	0

Total params: 23,602,676
Trainable params: 23,549,550
Non-trainable params: 53,126

My solution

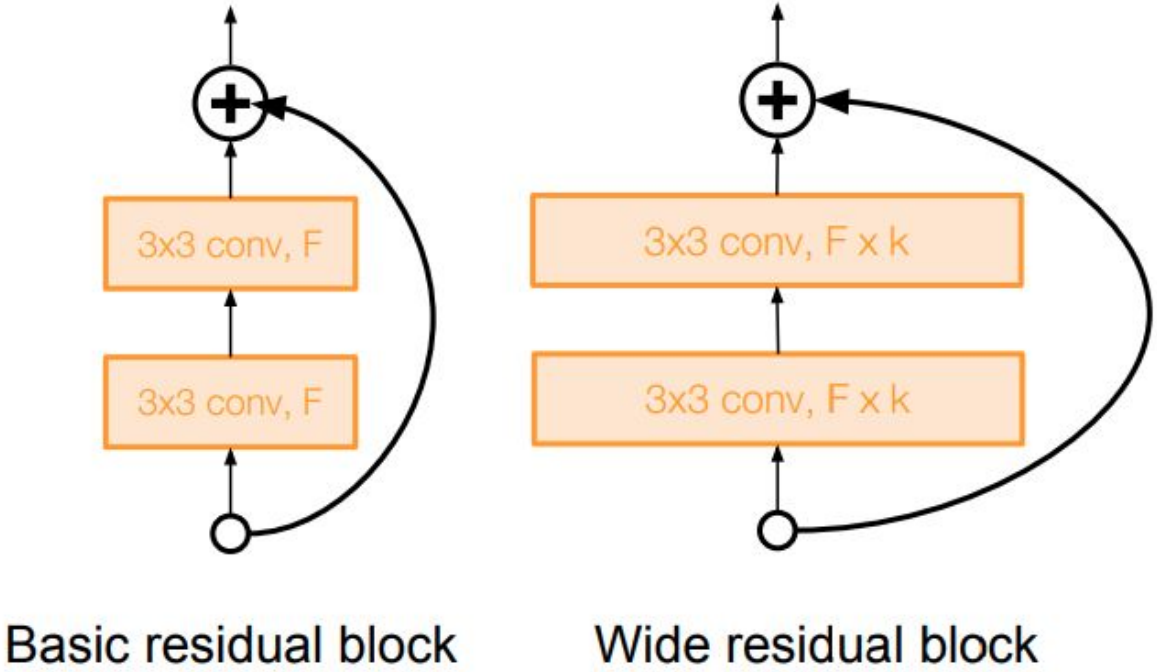

One year on this planet is 7 years on Earth
Great! I will train my model here

Others solution

- Train two CNN models, one for recognizing non-landmark image the other recognizing landmark_id
- Random crops
- Test Time Augmentation (TTA)
- Wide ResNet, DenseNet, Inception-v3, ...
- DEep Local Feature (DELF)
- Generalized Mean (GeM)
- Zero-shot learning, One-shot learning, Few-shot learning
- Ensemble
- ...

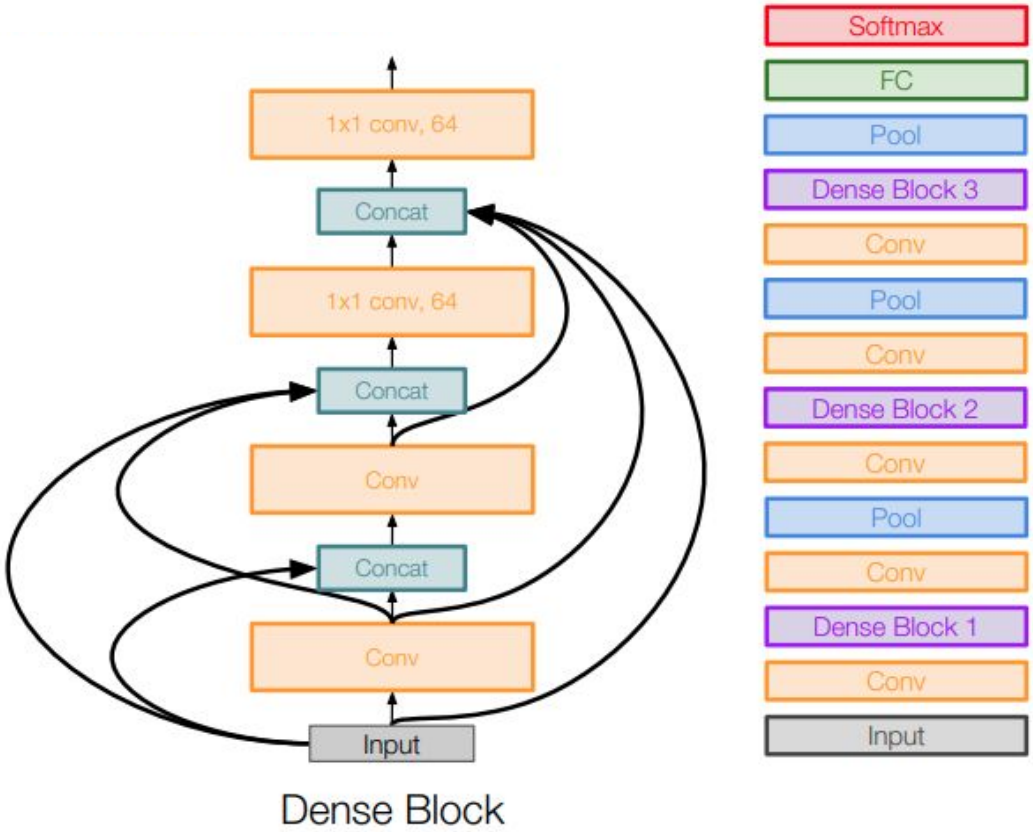
Others solution:

- Wide ResNet



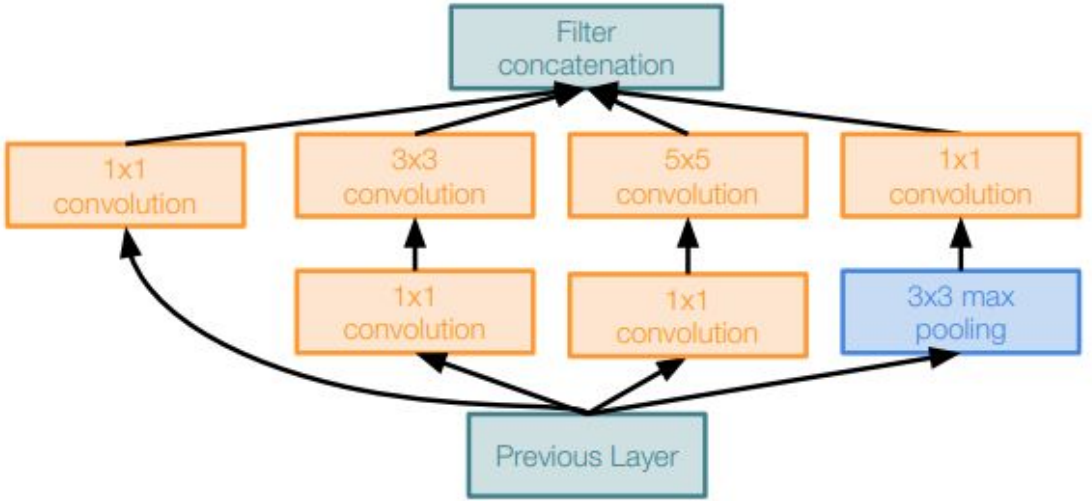
Others solution:

- DenseNet



Others solution:

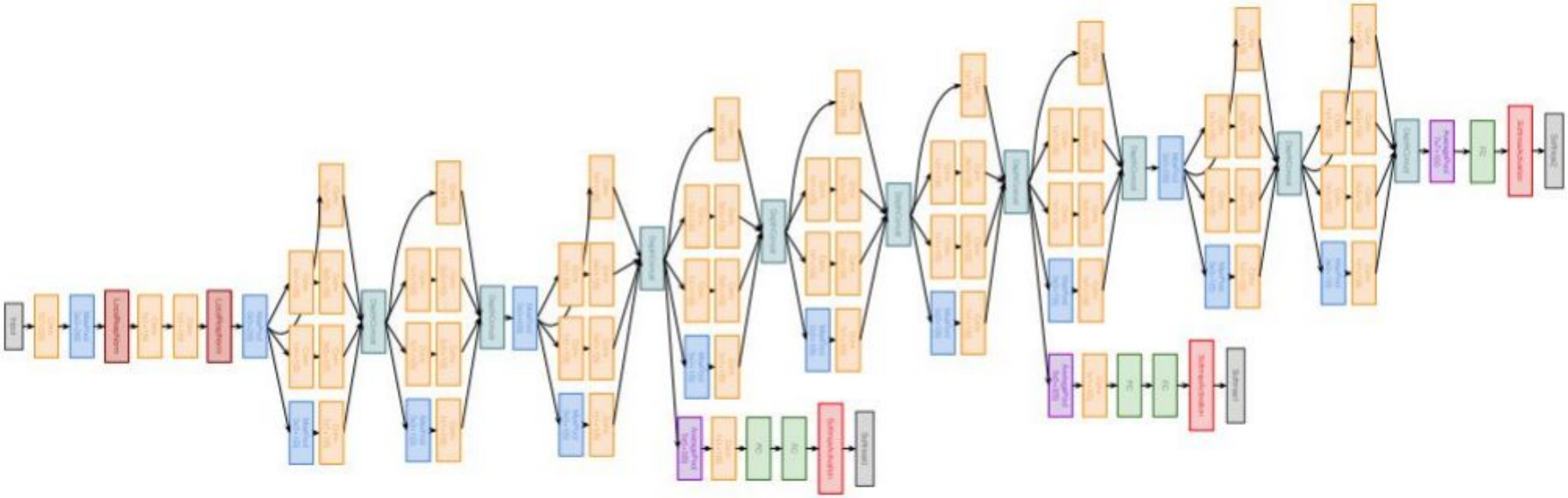
- Inception V3



Inception module with dimension reduction

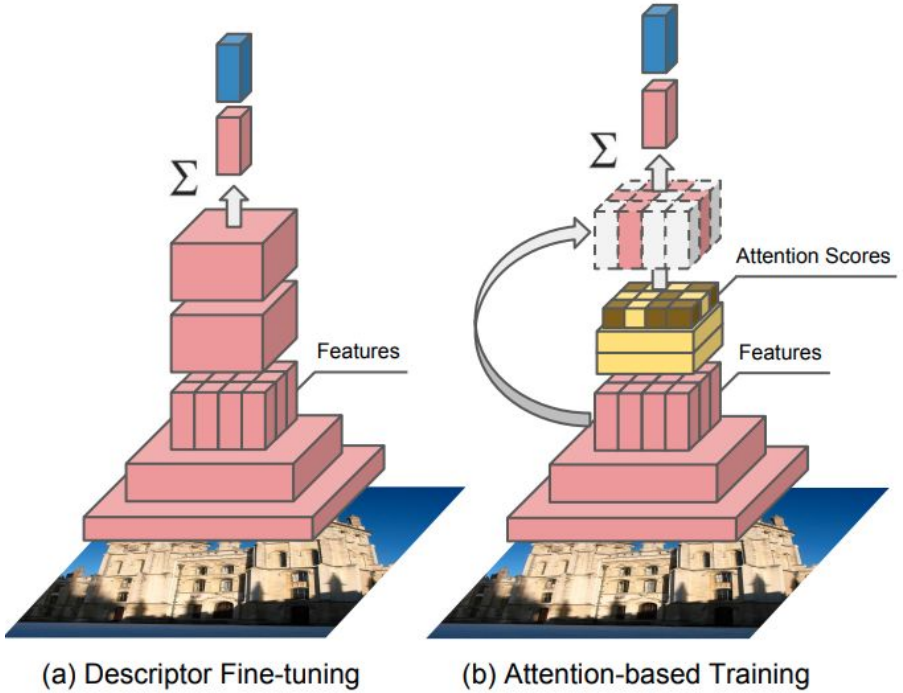
Others solution:

- Inception V3



Others solution

- DEep Local Feature (DELF)

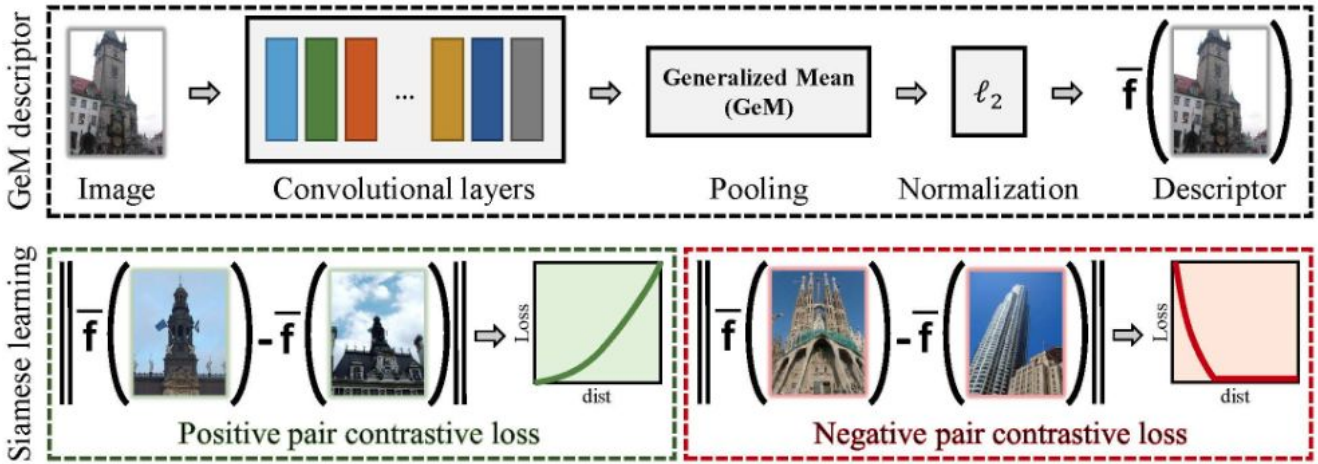


Based on ResNet50

$$\mathbf{y} = \mathbf{W}\left(\sum_n \alpha(\mathbf{f}_n;\theta)\cdot\mathbf{f}_n\right)$$

Others solution:

- Genralized Mean(GeM)



Generalized-mean pooling (GeM):

$$f_k = \left(\frac{1}{|X_k|} \sum_{x \in X_k} x^p \right)^{\frac{1}{p}}$$

$p \to \infty$ max pool MAC
$p = 1$ avg pool SPoC

Reference

https://www.kaggle.com/c/landmark-recognition-challenge

https://arxiv.org/pdf/1801.04540.pdf

https://github.com/davidsandberg/facenet

https://arxiv.org/pdf/1503.03832.pdf

http://cs231n.stanford.edu/

Fine-tuning CNN Image Retrieval with No Human Annotation