Clothing Image Retrieval with Triplet Capsule Networks

Osman Furkan Kınlı

Advisor: Asst. Prof. M. Furkan Kıraç

Department of Computer Science
Özyeğin University

İstanbul, August 19th , 2019

Outline

- Introduction
- Clothing Image Retrieval
- Triplet-based Similarity Learning
- Capsule Networks
- Proposed Architectures
- Experimental Study
- Results
- Conclusion

Introduction

- Online shopping is a highly **growing** market.

- Global fashion e-commerce market has a volume of **480B $**[1].

- **Using visual information of the products** is one of the most sophisticated way to adapt developing technologies to the sales process.

[1]According to Fashion E-Commerce Report 2019 by Statista

Introduction

- With the help of novel techniques combining **CV** and **DNNs**, it has become easier to achieve.

- Mostly attacked to this problem by using **CNN-based architectures**.

- However, CNNs have **some intrinsic limitations** by their nature.

- Most recently proposed architecture, Capsule Networks, **claims to overcome** these limitations.

Introduction

- In this thesis, we investigate the performance of

Capsule Network architecture

on **clothing image retrieval** task.

Introduction

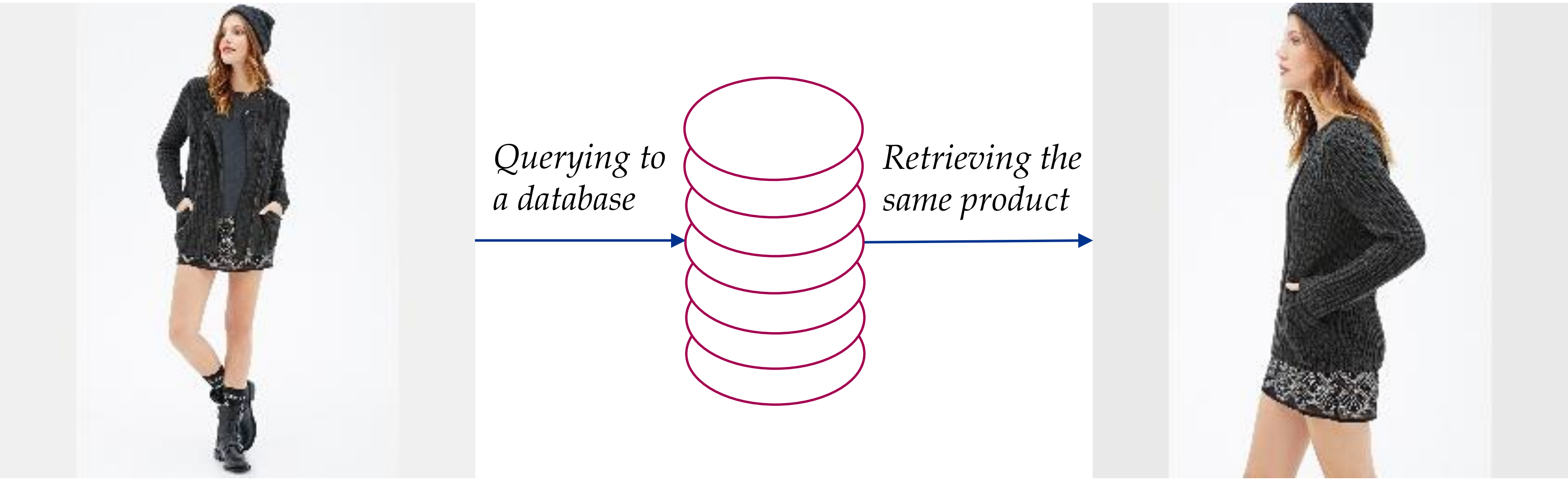
- Main goal:

- Investigating **the SOTA research** on clothing retrieval and Capsule Networks

  - The design of **Triplet-based** version of Capsule Networks
  - More **powerful feature extraction recipe** for Capsule inputs.

Clothing Image Retrieval

- Task of retrieving a clothing image in a gallery by querying an image of the same clothes.



Clothing Image Retrieval

- In fashion domain:
  - Kiapour *et al.* (2015): learning the similarity between the images is the best way to solve cross-domain image matching.
  - Huang *et al.* (2015): creating domain-specific representations by two sub-networks that are structurally similar, yet the weights are not shared is another solution for cross-domain image matching.
  - Liu *et al.* (2016): Employing the landmark information besides to the images helps to recover pose information in the images.

Clothing Image Retrieval

- In fashion domain:
  - Corbière *et al.* (2017): Integrating textual visual information (*i.e.* bag-of-words descriptors) into weakly-supervised learning process leads to get promising results.
  - Wang *et al.* (2017): Attention-based design focuses on important regions in clothing images and diminishes the effect of the background clutter.
  - Yuan *et al.* (2017): Ensembling a set of models with different complexities in cascaded manner and applying hard sampling strategies at the same time improves the performance by a wide margin.

Clothing Image Retrieval

- In fashion domain:
  - Opitz *et al.* (2018): Exploiting the independence within ensembles improves the robustness of the feature embeddings to the sampling strategy
  - Ge *et al.* (2018): Hierarchical Triplet Loss (HTL) addresses the random sampling issue during training Triplets
  - Kim *et al.* (2018): Representing different parts of the objects on the feature embeddings with different attention masks encourages the diversity in feature representation.

Clothing Image Retrieval

- Our approach:

Employing Capsule Networks to this problem

without utilizing any side information or extra module that

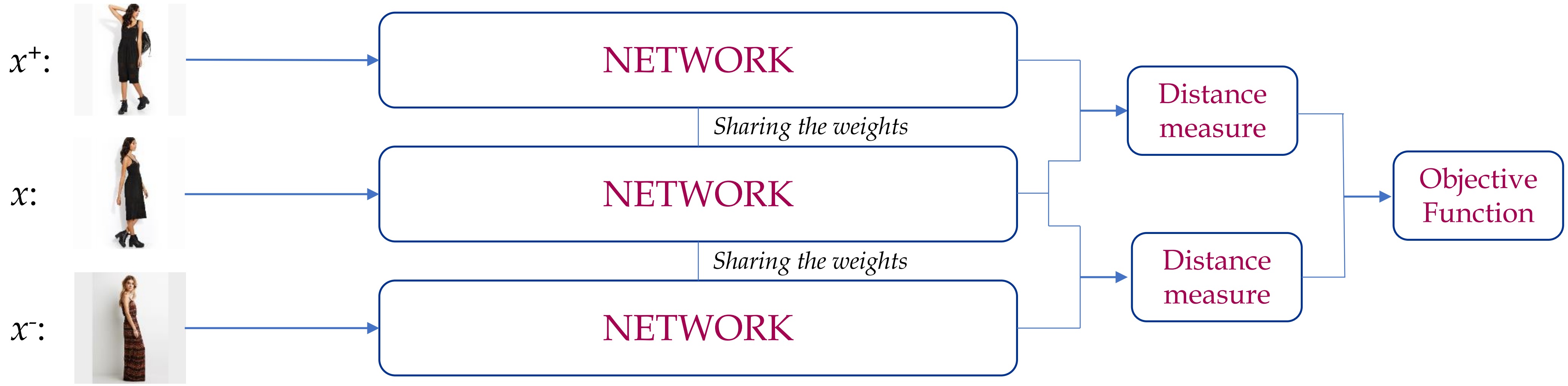
recovers the pose configuration in the images.

Triplet-based Similarity Learning

- Inspired by *Siamese Networks*.
- 3 instances of pairs for the same feed-forward Neural Network and denoted as:

x: Anchor instance; x^+: Positive instance; x^-: Negative instance

- Sharing the weights throughout the network.

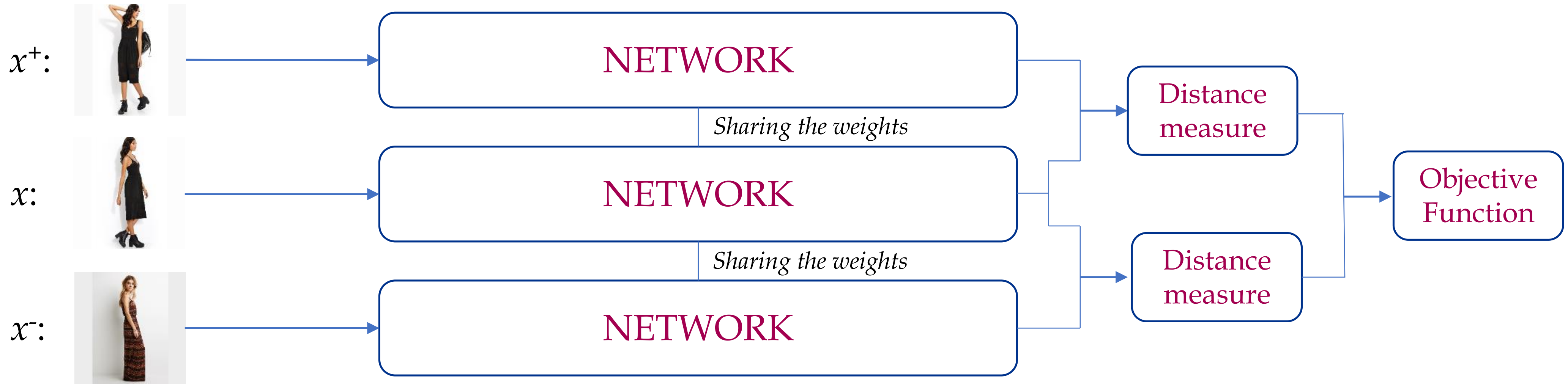


Triplet-based Similarity Learning

- Inspired by *Siamese Networks*.
- 3 instances of pairs for the same feed-forward Neural Network and denoted as:

x: Anchor instance; x^{+}: Positive instance; x^{-}: Negative instance

- <u>Sharing the weights throughout the network.</u>



Triplet-based Similarity Learning

where $f(x) \in \mathbf{R}^d$,

$d(l_1, l_2)$, $L(d_1, d_2) \in \mathbf{R}$

Feature embeddings:

$$f(x) = l$$

$$f(x^+) = l^+$$

$$f(x^-) = l^-$$

Triplet-based Similarity Learning

where $f(x) \in \mathbf{R}^d$,

$d(l_1, l_2)$, $L(d_1, d_2) \in \mathbf{R}$

Distance metric:

$$d(l, l^+) = \|f(x) - f(x^+)\|_2^2$$

$$d(l, l^-) = \|f(x) - f(x^-)\|_2^2$$

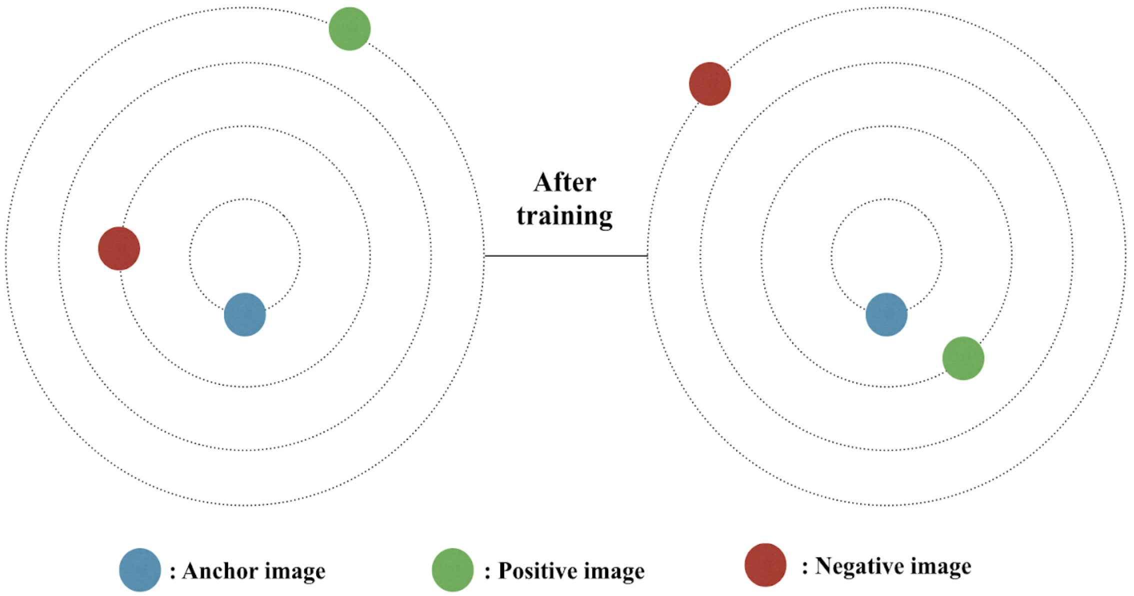
Triplet-based Similarity Learning

where $f(x) \in \mathbf{R}^d$,

$d(l_1, l_2)$, $L(d_1, d_2) \in \mathbf{R}$

Triplet relationship:

$$d(l, l^+) + \alpha < d(l, l^-)$$



Triplet-based Similarity Learning

where $f(x) \in \mathbf{R}^d$,

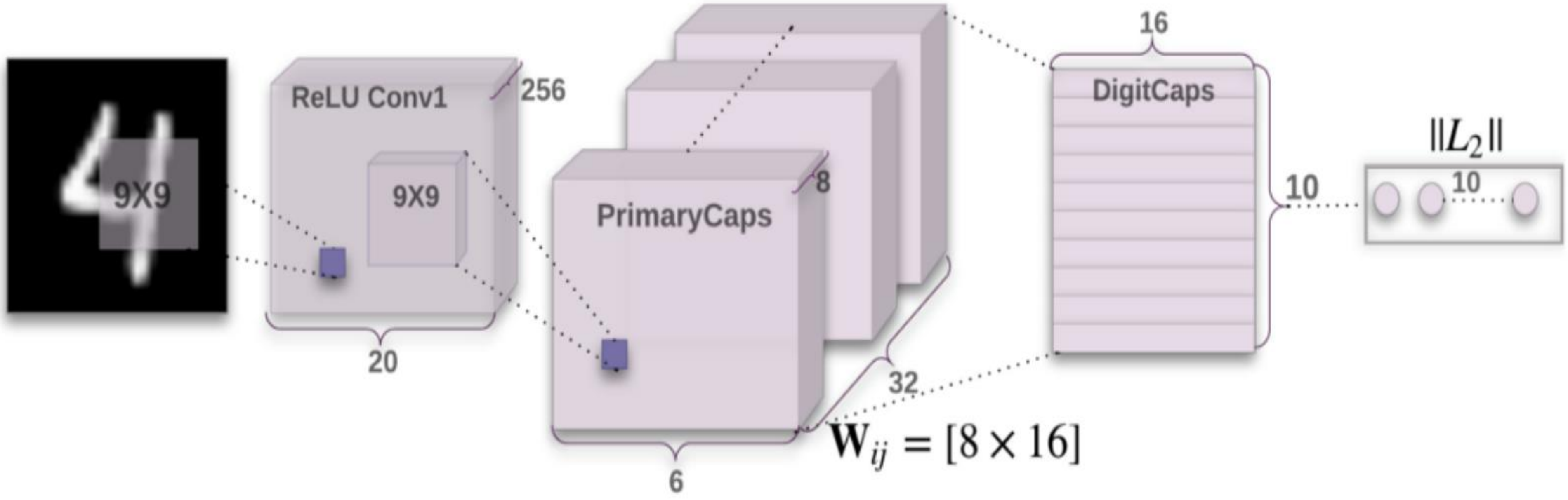
$d(l_1, l_2)$, $L(d_1, d_2) \in \mathbf{R}$

Triplet loss:

$$L(d_1, d_2) = \sum_i [d(l_i, l_i^+) - d(l_i, l_i^-) + \alpha]$$

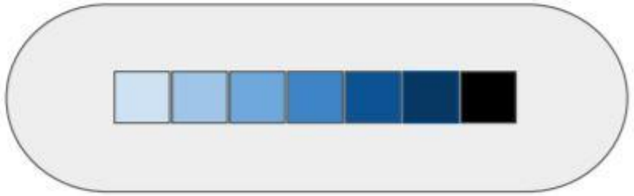
Capsule Networks

- Capsule Networks are recently proposed by Sabour and Hinton *et al.* (2017), with a novel routing algorithm between Capsules.



Capsule Networks

- Capsules are basically groups of neurons.



- High dimensional information:

the existence and pose configuration.

- The output of a Capsule is routed to the next Capsule layer by

a *dynamic routing algorithm*.

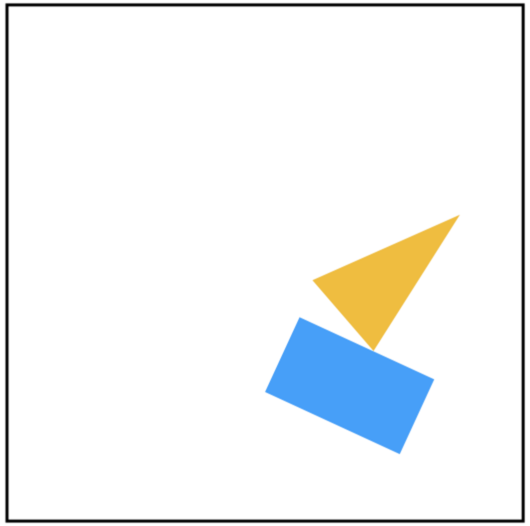
Capsule Networks

- In graphics:

Triangle	
Parameters	**Values**
x	50
y	80
height	200
width	120
color	yellow
angle	30

Rectangle	
Parameters	**Values**
x	60
y	120
height	350
width	220
color	blue
angle	30

rendering



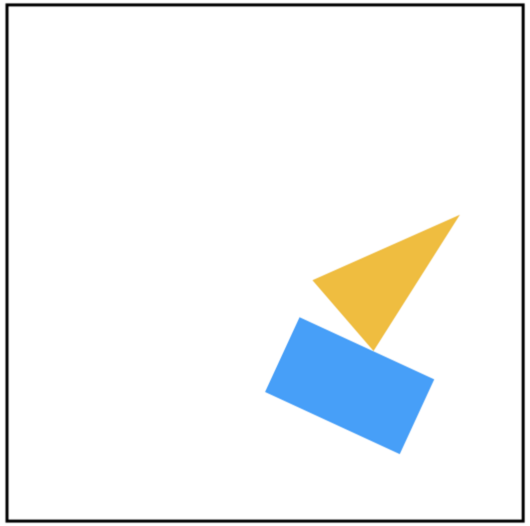
Capsule Networks

- In inverse graphics:

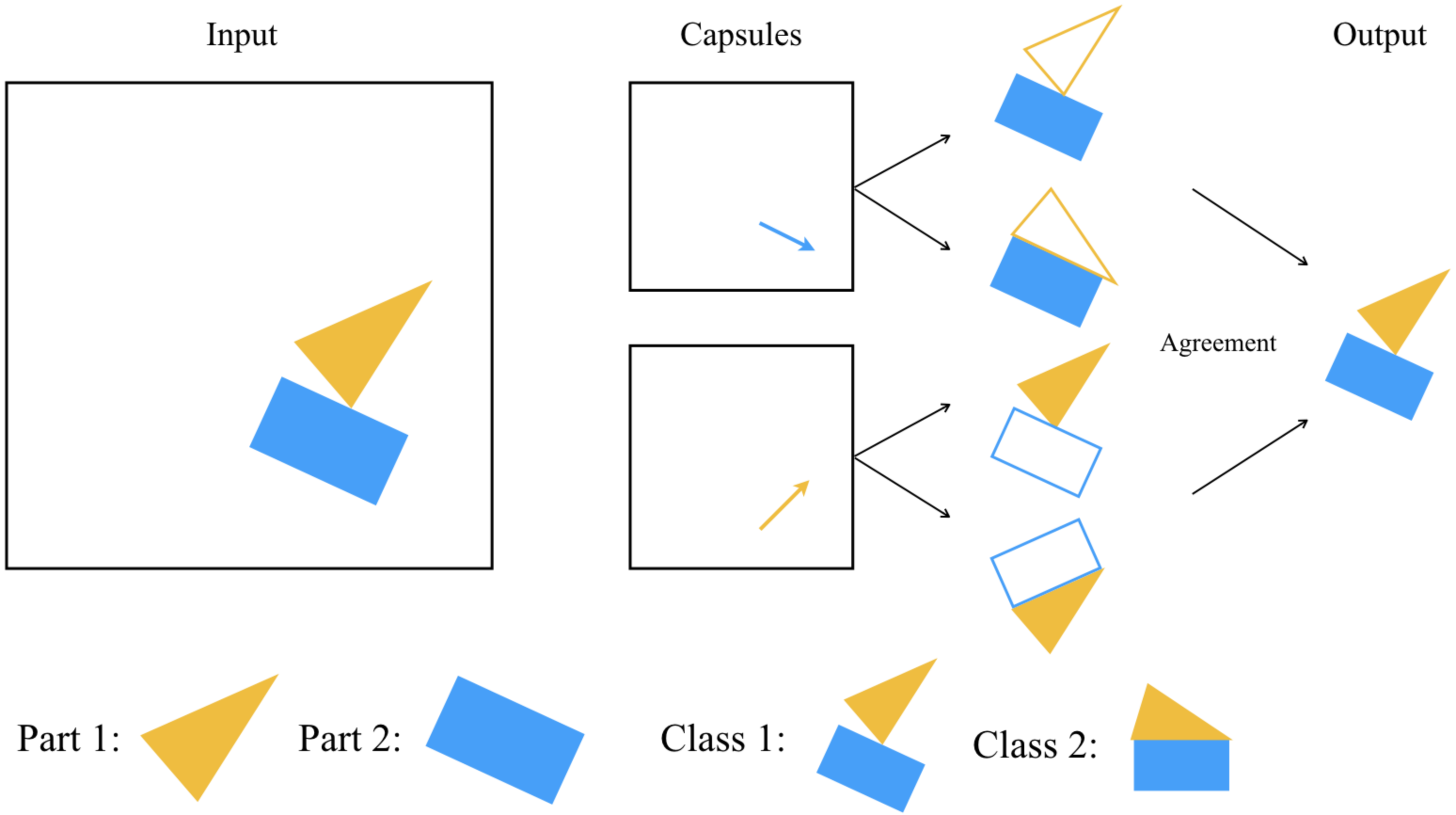
Triangle	
Parameters	**Values**
x	50
y	80
height	200
width	120
color	yellow
angle	30

Rectangle	
Parameters	**Values**
x	60
y	120
height	350
width	220
color	blue
angle	30

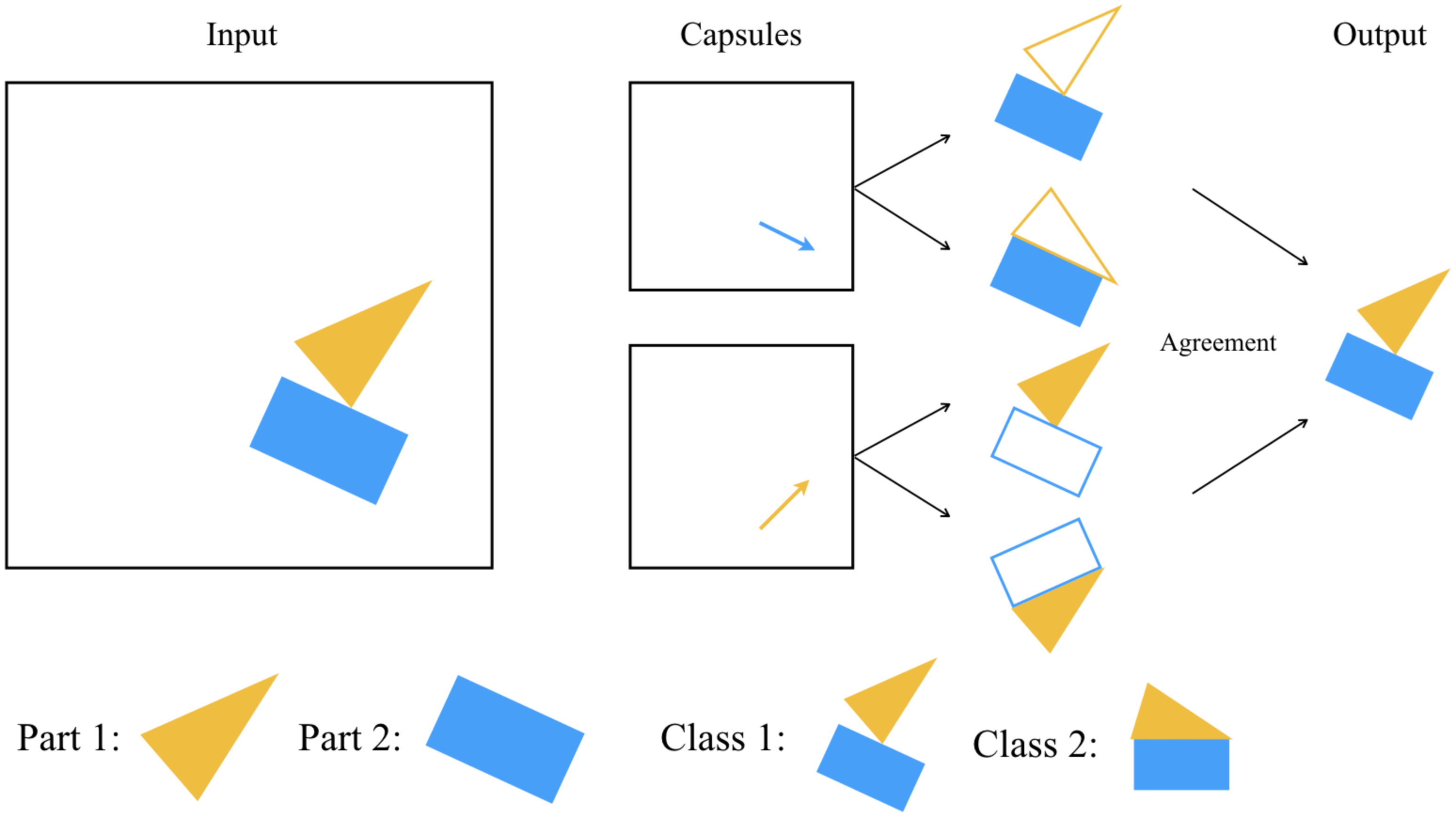
Inverse rendering



Capsule Networks



Capsule Networks



Capsule Networks

- In mathematical perspective:
  - The output of capsule *i* : u_i
  - Trainable transformation matrix : W_{ij}
  - Transformed output by coordinate frame relation

$$\hat{u}_{j|i} = W_{ij} u_i$$

Capsule Networks

- In mathematical perspective:
  - Initial logits : b_{ij} (*i.e.* initialized to 0)
  - Represents the log prior probability of routing the output of capsule *i* to capsule *j* in the next layer.
  - Routing softmax

$$c_{ij} = \frac{e^{b_{ij}}}{\sum e^{b_{ij}}}$$

Capsule Networks

- In mathematical perspective:
  - Non-activated input for capsule *j*

$$s_j = \sum_i c_{ij} \hat{u}_{j|i}$$

  - Activation of the input for capsule *j* (*i.e.* squashing)

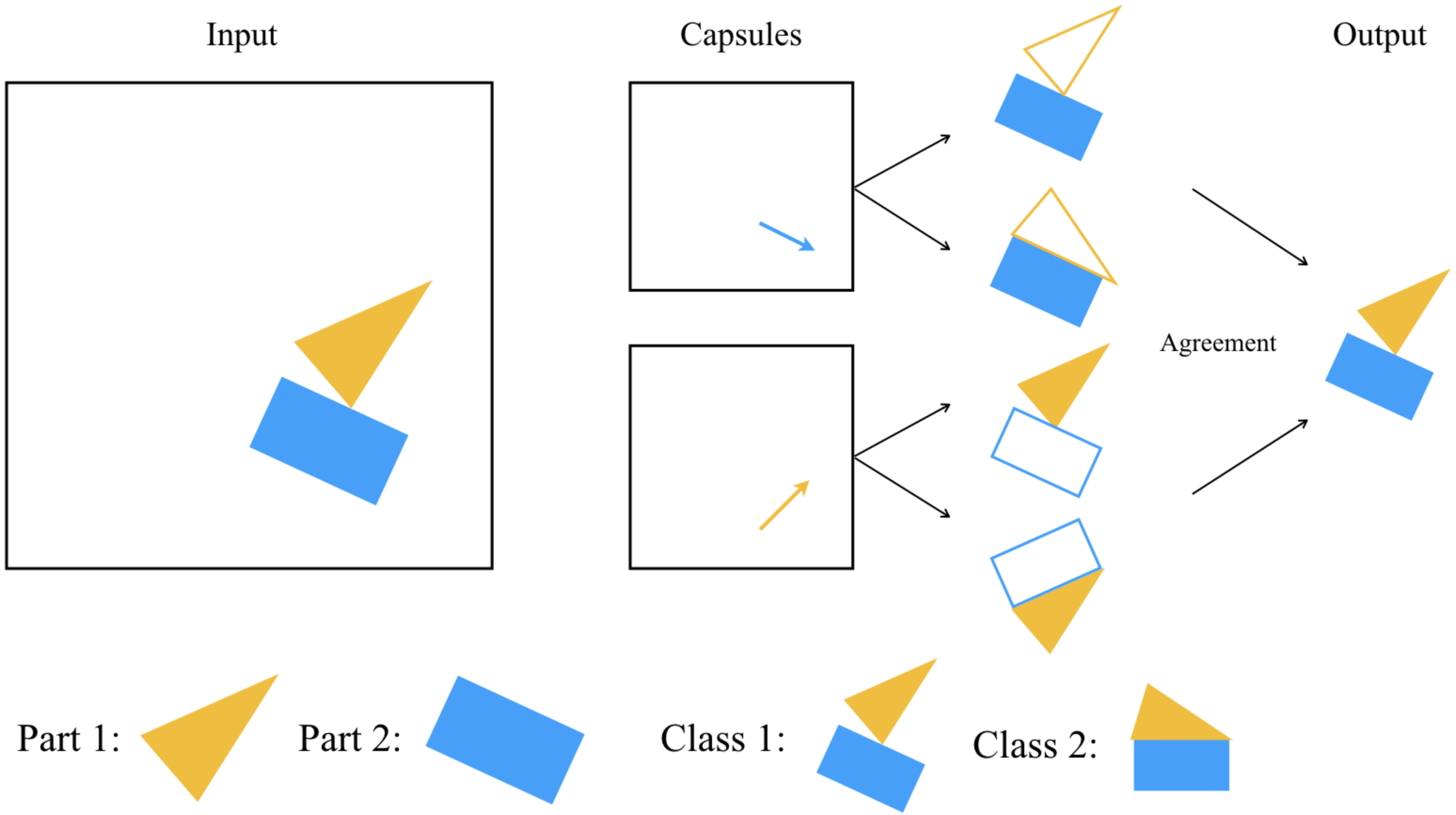
$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j|| + \epsilon}$$

Capsule Networks

- In mathematical perspective:
  - Agreement between coordinate frames (*i.e.* dot product of transformed output of capsule *i* and activated input of capsule *j*)

$$a_{ij} = v_j \hat{u}_{j|i}$$

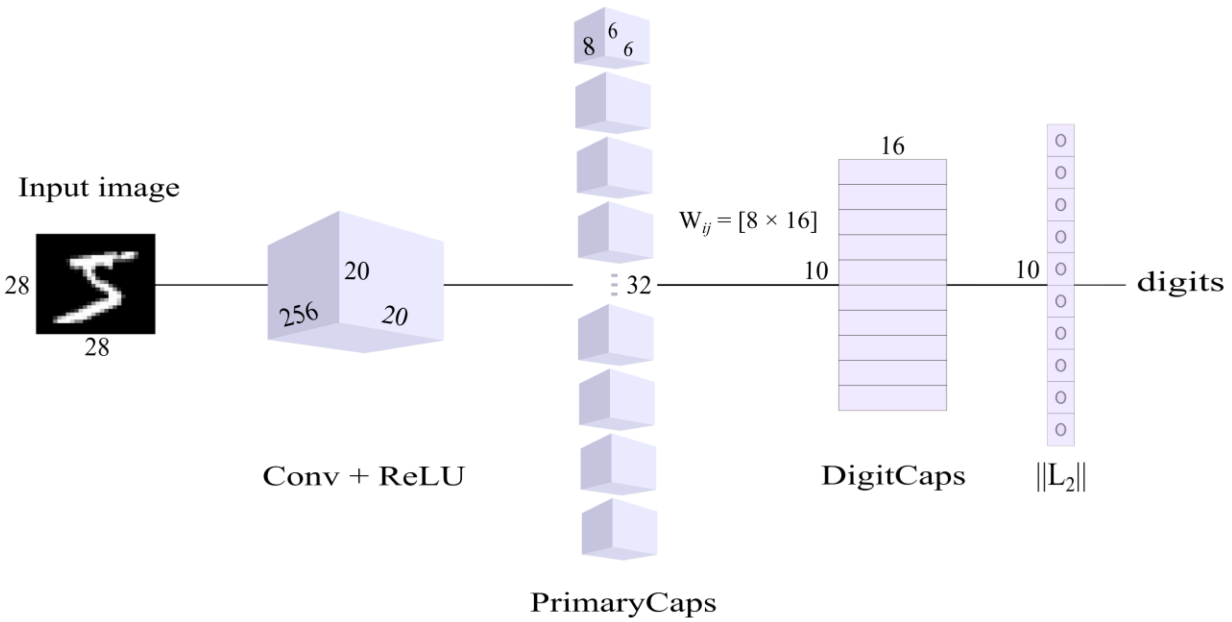
Capsule Networks



Capsule Networks

- Objective function:

$$L_k = T_k max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) max(0, \|v_k\| - m^-)^2$$



Capsule Networks

- Capsule Networks can **perform well** by
  - flowing **more descriptive** information between layers
  - preserving **the part-whole relationship** of the objects
- and regardless to
  - **the amount** of data
  - **the diversity** of data

Proposed Architectures

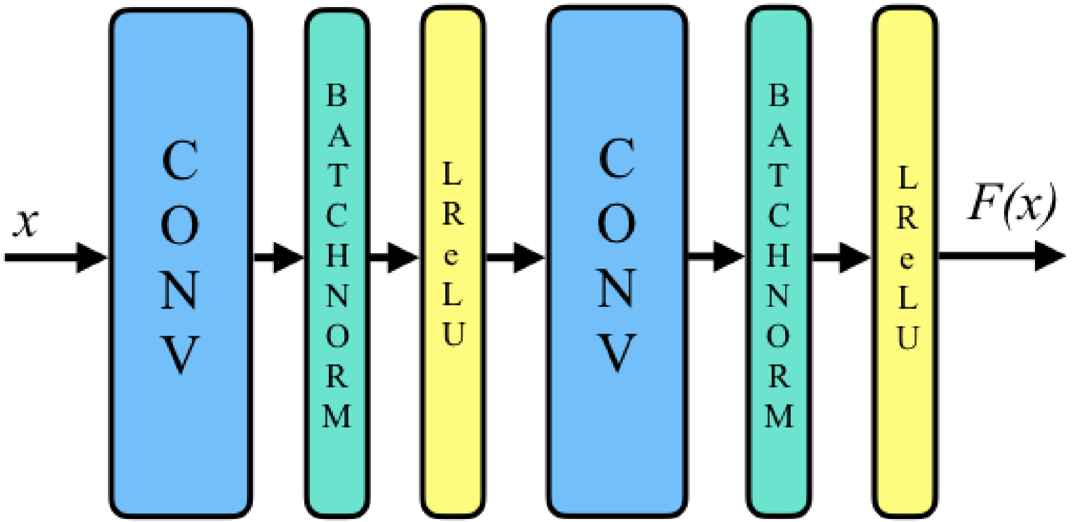
- We have **3** design steps:
  - **Powerful feature extraction blocks** for Capsule inputs
  - Adjusting the original architecture to **Triplet-based design**
  - Designing **Capsule layers**

Proposed Architectures

- Feature extraction blocks:
  - In default methodology, the feature extraction block has a **single** convolutional layer with **64** filters.
  - We design **two** different feature extraction blocks to generate Capsule inputs.

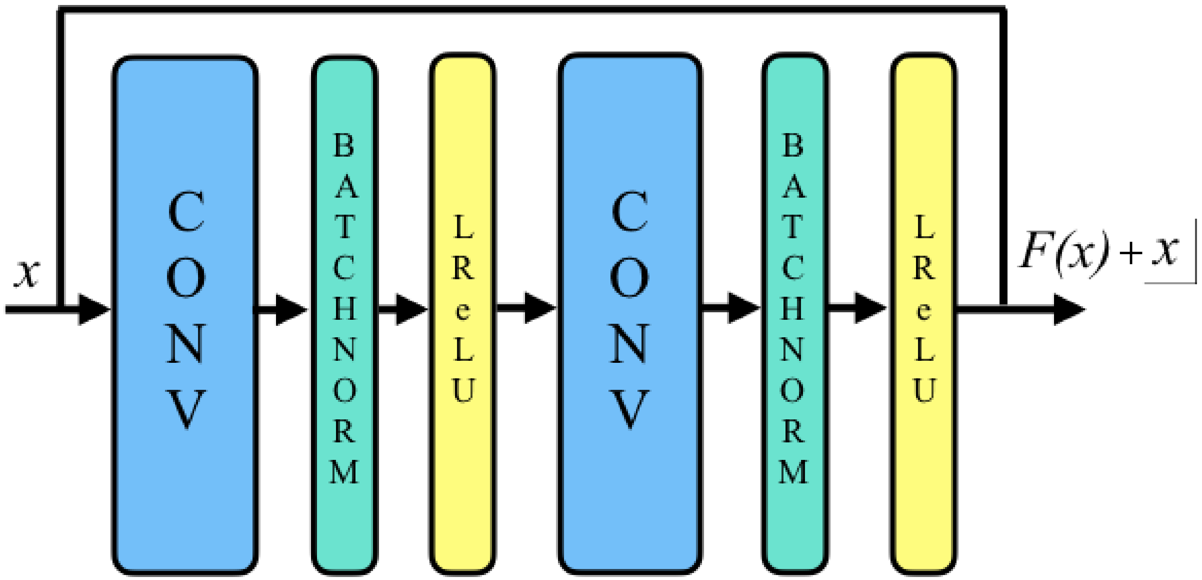
Proposed Architectures

1. **Stacking** several convolutional layers
   - with **different number of filters**
   - followed by **leaky-formed rectifiers** and **batch normalization**



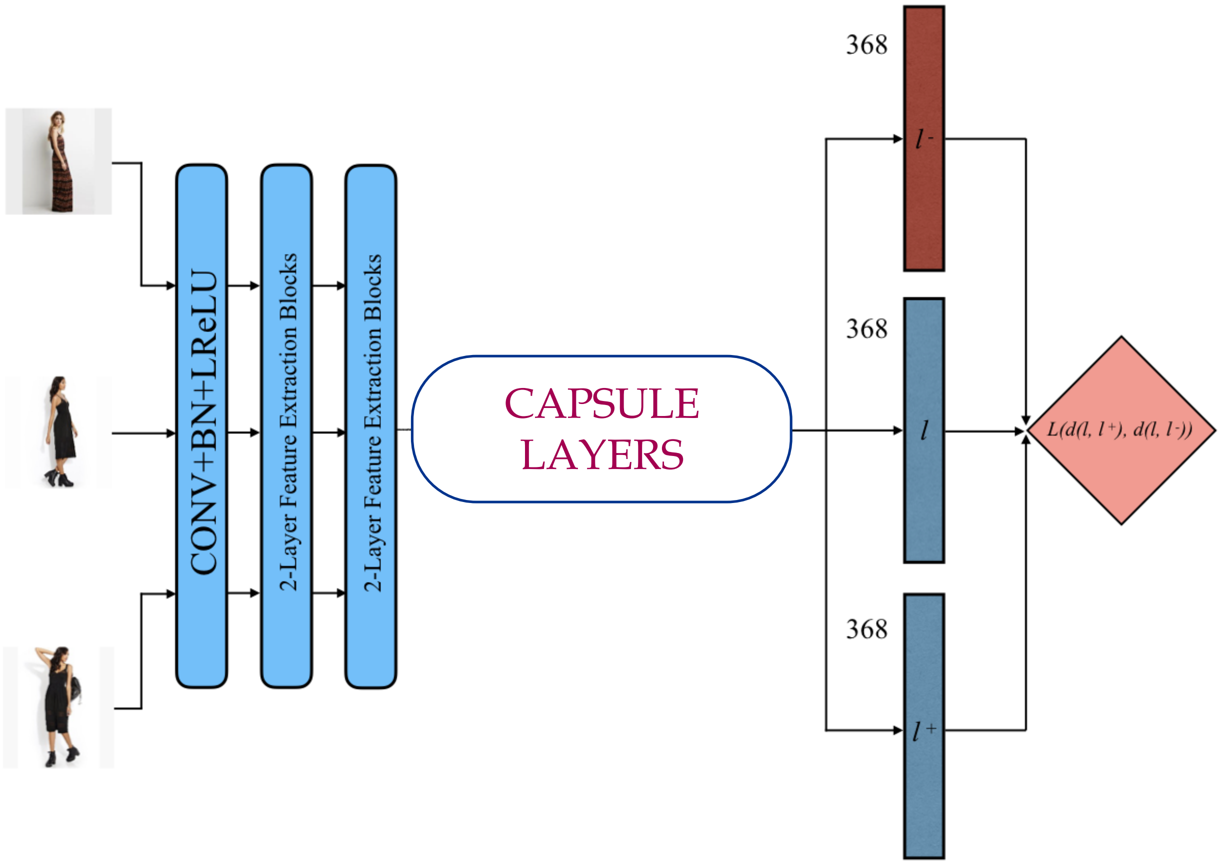
Proposed Architectures

2. Connecting stacked-convolutional layers as **residuals**



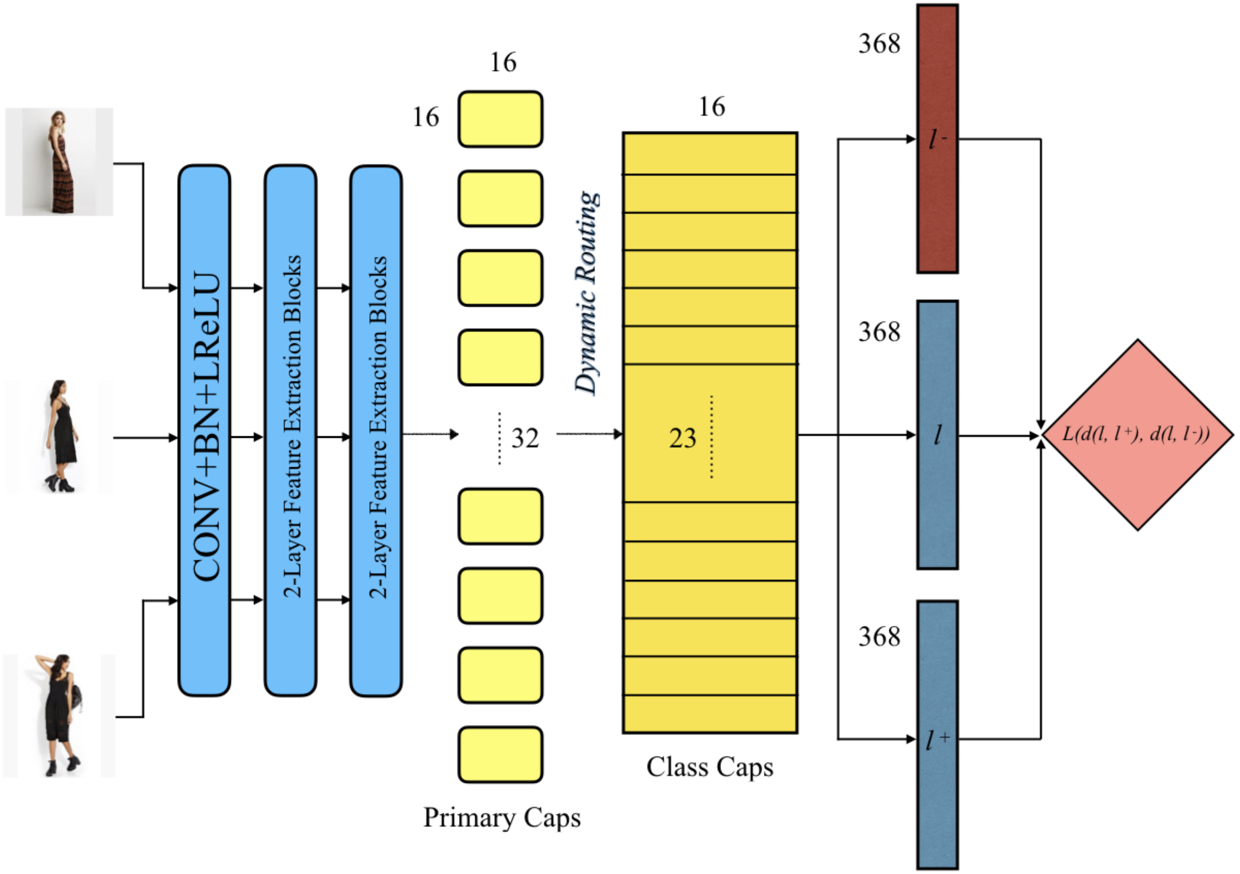
Proposed Architectures

- Triplet-based design:
  - Learning **the similarity** between images
  - Feeding the objective function with **the embedded sparse representations** extracted by **Capsules**



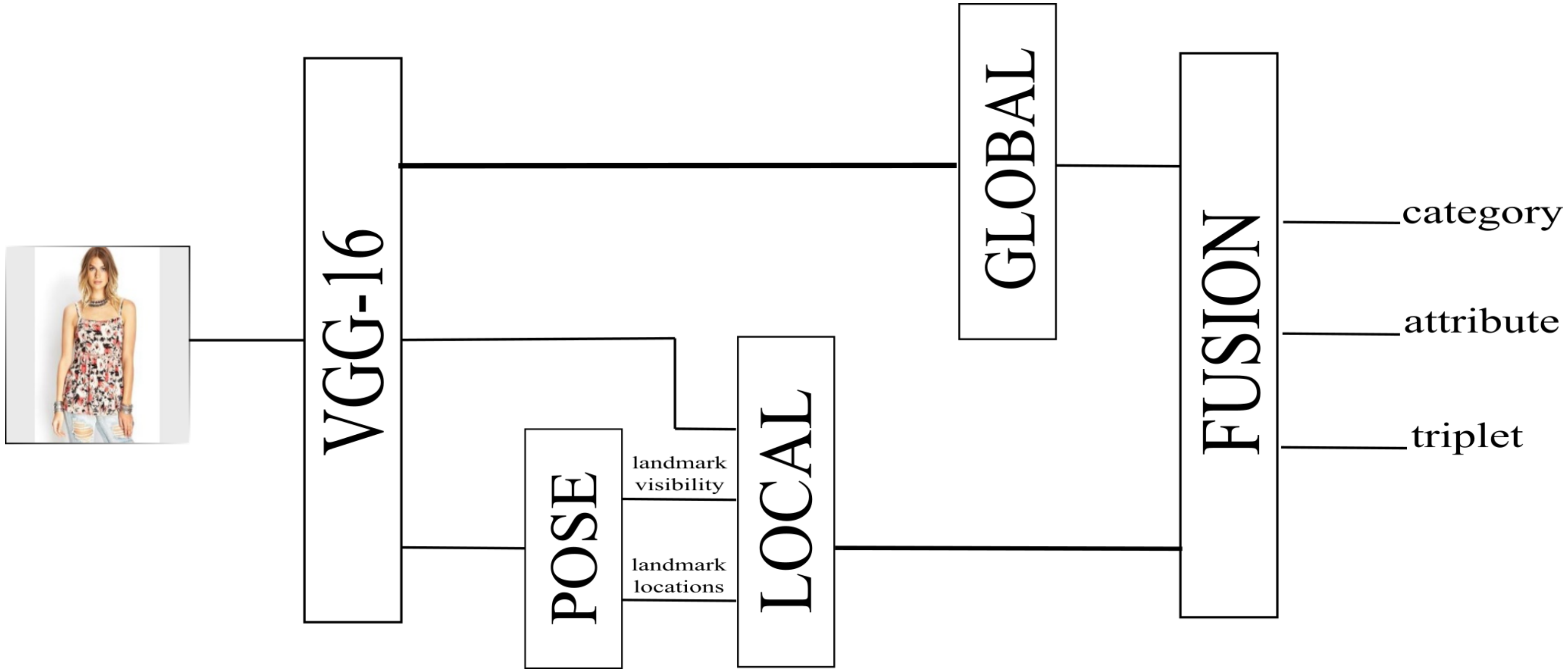
Proposed Architectures

- Capsule layers:
  - Two fully-connected Capsule layers which are called *Primary Capsule* and *Class Capsule*, respectively.



Experimental Study

- Baseline study:



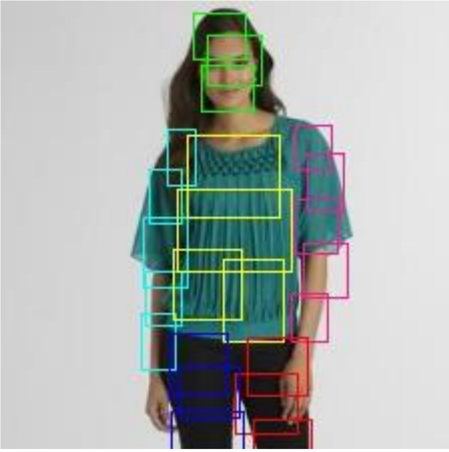
Experimental Study

- Data set:
  - **In-shop** partition of DeepFashion
  - 25k training, 14k query and 12k gallery images

Original

Landmarks

Human Joints



Poselets

Experimental Study

- Data set:
  - 1000 Attributes

Groups	Attributes
Texture	Floral, Stripe, Paisley, Distressed, Dot, Plaid, Panel, Raglan, ...
Fabric	*Lace, Denim, Chiffon, Pleated, Woven, Leather, Cotton, Linen, ...*
Shape	*Crop, Maxi, Fit, Longline, Boxy, Mini, Skinny, Midi, Pencil, ...*
Part	*Sleeveless, Pocket, V-Neck, Hooded, Racerback, Peplum, Strappy, ...*
Style	*Graphic, Muscle, Tribal, Peasant, Surplice, Polka, Retro, Yoga, ...*

  - **At most** 8 visible Landmarks

Experimental Study

- Implementation details:
  - 2 MSI GTX 1080 Ti Armor OC 11 GB
  - Framework: **Keras** with TF backend[1]

  - Hyper-parameter settings:

Hyper-parameter	Value
Optimizer	*Adam* [53]
Learning Rate	*0.001*
Decay Rate	5×10^{-4}
Batch Size	*32*
Routings	*3*
Normalization	*Pixel-wise*

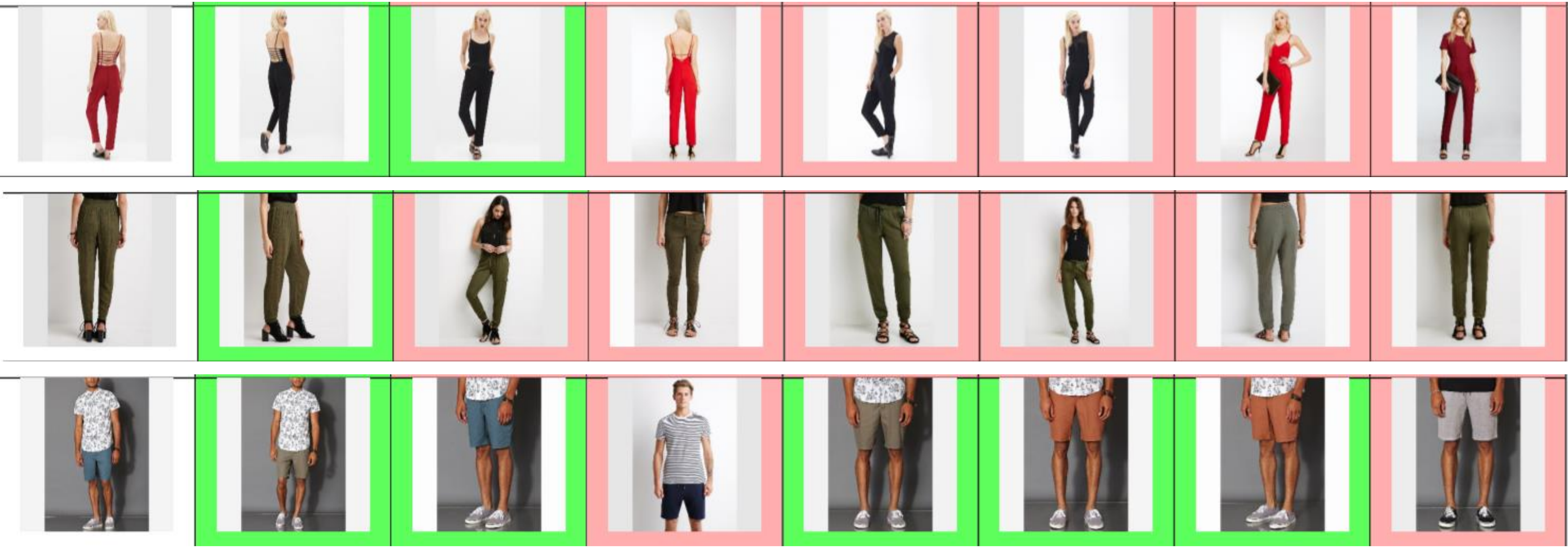
- Data augmentation:

Augmentation Methods	Applied	Range
Feature-wise Centering	✗	None
Sample-wise Centering	✗	None
Feature-wise STD Norm.	✗	None
Sample-wise STD Norm.	✗	None
ZCA Whitening	✗	None
Rotation	✓	[0°-30°]
Width Shifting	✓	[0-0.1]
Height Shifting	✓	[0-0.1]
Channel Shifting	✗	None
Brightness	✓	[0.5-1.5]
Shearing	✓	[0-0.1]
Zoom	✓	[0-0.1]
Horizontal Flipping	✓	None
Vertical Flipping	✗	None

[1]Source code: https://github.com/birdortyedi/image-retrieval-with-capsules

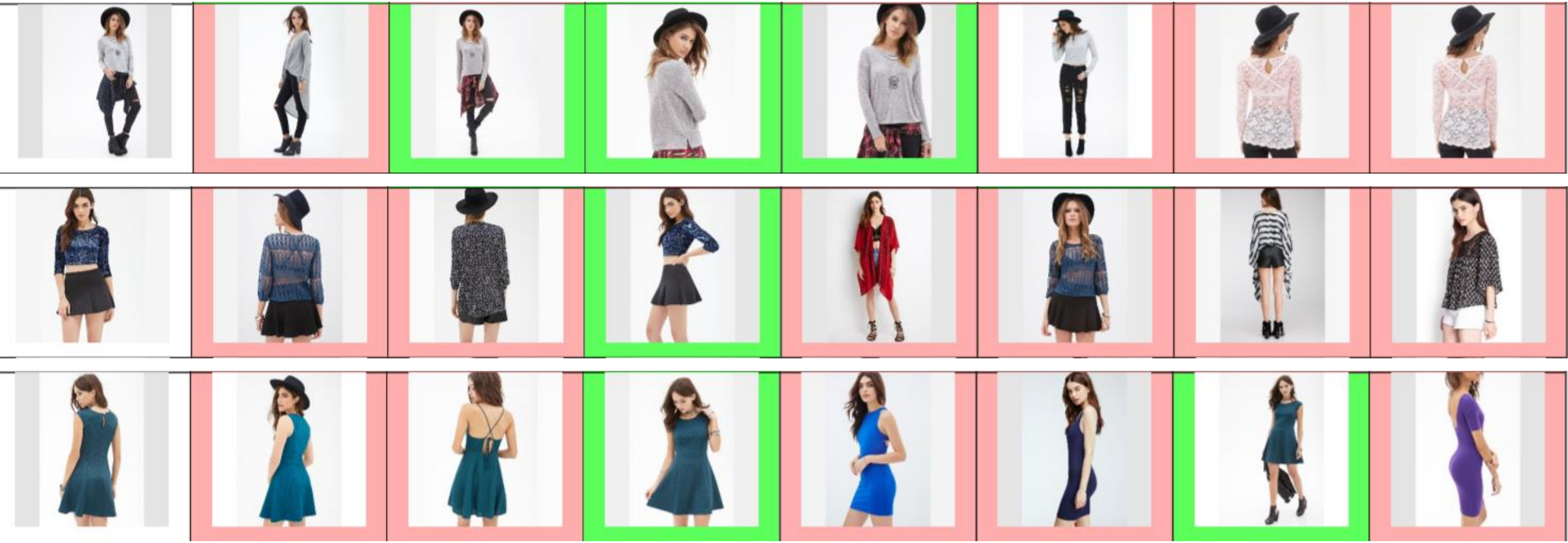
Results

- Qualitative Results:



Results

- Qualitative Results:



Results

- Quantitative Results:
  - **Details** of architectures in the comparison:

Model Name	**Backbone Architecture**	**Side Information (SI) Extra Module (EM)**	**# of (M) Params**
WTBI [25]	AlexNet [54]	Category-specific Similarity (SI)	60
DARN [26]	Custom NiN [55]	Visual Similarity (SI)	105
FashionNet [1]	VGG-16 [39]	Landmark Information (SI)	134
Corbiére *et al.* [27]	ResNet50 [6]	Bag-of-words Descriptors (EM)	25
SCCapsNet *(ours)*	CapsNet [4]	**No SI/EM Used**	2.5
RCCapsNet *(ours)*	CapsNet [4]	**No SI/EM Used**	4.5
HDC [29]	GoogLeNet [40]	Hard-Aware Cascaded Embedding (EM)	5
VAM [28]	GoogLeNet [40]	Attention with Impdrop Connection (EM)	6
BIER [20]	GoogLeNet [40]	Embedding Boosting (EM)	5
HTL [19]	GoogLeNet [40]	Hierarchical Triplet Loss (EM)	5
A-BIER [20]	GoogLeNet [40]	Embedding Boosting with Adversarial Loss (EM)	5
ABE [21]	GoogLeNet [40]	Attention-based Ensembling (EM)	10

Results

- Quantitative Results:
  - Inner comparison:

Models	Top-1 (%)	Top-10 (%)	Top-20 (%)	Top-30 (%)	Top-40 (%)	Top-50 (%)
SCCapsNet *(ours)*	32.1	72.4	81.8	86.3	89.2	90.9
RCCapsNet *(ours)*	33.9	**75.2**	**84.6**	**88.6**	**91.0**	**92.6**

Results

- Quantitative Results:
  - Comparison with **the Baseline study**:

Models	Top-1 (%)	Top-10 (%)	Top-20 (%)	Top-30 (%)	Top-40 (%)	Top-50 (%)
FashionNet+100A+L	36.0	53.0	57.3	60.0	62.0	62.5
FashionNet+500A+L	37.0	59.0	64.6	67.5	69.0	69.5
FashionNet+1000A+J	41.0	64.0	68.0	71.0	73.0	73.5
FashionNet+1000A+P	42.0	65.0	70.0	72.0	72.5	75.0
FashionNet+1000A+L	**53.2**	72.5	76.4	77.0	79.0	80.0
SCCapsNet *(ours)*	32.1	72.4	81.8	86.3	89.2	90.9
RCCapsNet *(ours)*	33.9	**75.2**	**84.6**	**88.6**	**91.0**	**92.6**

Results

- Quantitative Results:
  - Comparison with **the SOTA**:

Models	Top-1 (%)	Top-10 (%)	Top-20 (%)	Top-30 (%)	Top-40 (%)	Top-50 (%)
WTBI [25]	35.0	47.0	50.6	51.5	53.0	54.5
DARN [26]	38.0	56.0	67.5	70.0	72.0	72.5
FashionNet [1]	53.2	72.5	76.4	77.0	79.0	80.0
Corbiére *et al.* [27]	39.0	71.8	78.1	81.6	83.8	85.6
SCCapsNet *(ours)*	32.1	72.4	81.8	86.3	89.2	90.9
RCCapsNet *(ours)*	33.9	75.2	84.6	88.6	91.0	92.6
HDC [29]	62.1	84.9	89.0	91.2	92.3	93.1
VAM [28]	66.6	88.7	92.3	-	-	-
BIER [20]	76.9	92.8	95.2	96.2	96.7	97.1
HTL [19]	80.9	94.3	95.8	97.2	97.4	97.8
A-BIER [20]	83.1	95.1	96.9	97.5	97.8	98.0
ABE [21]	87.3	96.7	97.9	98.2	98.5	98.7

Results

- Quantitative Results:
  - Comparison with **the SOTA**:

Models	Top-1 (%)	Top-10 (%)	Top-20 (%)	Top-30 (%)	Top-40 (%)	Top-50 (%)
WTBI [25]	35.0	47.0	50.6	51.5	53.0	54.5
DARN [26]	38.0	56.0	67.5	70.0	72.0	72.5
FashionNet [1]	53.2	72.5	76.4	77.0	79.0	80.0
Corbiére *et al.* [27]	39.0	71.8	78.1	81.6	83.8	85.6
SCCapsNet *(ours)*	32.1	72.4	81.8	86.3	89.2	90.9
RCCapsNet *(ours)*	33.9	75.2	84.6	88.6	91.0	92.6
HDC [29]	62.1	84.9	89.0	91.2	92.3	93.1
VAM [28]	66.6	88.7	92.3	-	-	-
BIER [20]	76.9	92.8	95.2	96.2	96.7	97.1
HTL [19]	80.9	94.3	95.8	97.2	97.4	97.8
A-BIER [20]	83.1	95.1	96.9	97.5	97.8	98.0
ABE [21]	87.3	96.7	97.9	98.2	98.5	98.7

Results

Category Name	Number of Unique Items	Number of Total Items
Blouse/Shirts	697	2.094
Tees/Tanks	673	2.955
Dresses	624	1.091
Shorts	246	988
Sweaters	212	735
Jackets/Coats	195	545

- Ablation study 1:
  - **Category-specific** comparison:

Models	Category	Top-1 (%)	Top-10 (%)	Top-20 (%)	Top-30 (%)	Top-40 (%)	Top-50 (%)
SCCapsNet	Blouse/Shirts	36.3	74.8	82.5	86.4	88.6	90.6
	Tees/Tanks	20.0	64.1	75.9	82.7	86.3	88.5
	Dresses	24.8	65.4	75.4	81.8	85.9	88.0
	Shorts	25.4	66.1	78.5	83.8	88.3	90.5
	Sweaters	27.5	69.3	80.4	84.2	86.5	88.6
	Jackets/Coats	34.5	75.2	84.2	87.7	89.7	92.3
RCCapsNet	Blouse/Shirts	39.7	79.5	86.8	89.5	91.3	92.9
	Tees/Tanks	35.1	75.5	83.3	86.8	89.0	90.8
	Dresses	31.9	73.3	84.9	89.0	91.2	92.4
	Shorts	27.3	69.2	80.4	86.6	89.7	92.5
	Sweaters	27.6	69.8	80.8	85.0	88.3	89.8
	Jackets/Coats	36.5	75.2	84.8	90.5	92.8	94.5

Results

- Ablation study 2:
  - Category classification comparison:
  - with **the Baseline study**:

Models + the required side information (if any)	Top-3 (%)	Top-5 (%)
FashionNet + 100 A + L	47.38	70.57
FashionNet + 500 A + L	57.44	77.39
FashionNet + 1000 A + J	72.30	81.52
FashionNet + 1000 A + P	75.34	84.87
FashionNet + 1000 A + L	82.58	90.17
SCCapsNet-CLS *(ours)*	83.18	89.83
RCCapsNet-CLS *(ours)*	**85.12**	**91.41**

Results

- Ablation study 2:
  - Category classification comparison:

  - with the **SOTA**:

Architectures	Backbone	Side Information (SI) Extra Module (EM)	Top-3 (%)	Top-5 (%)
WTBI [25]	*AlexNet* [54]	*Category-specific Similarity (SI)*	43.73	66.26
DARN [26]	*Custom NiN* [55]	*Visual Similarity (SI)*	59.48	79.58
FashionNet [1]	*VGG-16* [39]	*Landmark Information (SI)*	82.58	90.17
SCCapsNet-CLS *(ours)*	*CapsNet* [4]	**No SI / EM Used**	83.18	89.83
RCCapsNet-CLS *(ours)*	*CapsNet* [4]	**No SI / EM Used**	85.12	91.41
Corbiére *et al.* [27]	*ResNet50* [6]	*Bag-of-words Descriptors (EM)*	86.30	92.80
Lu *et al.* [57]	*VGG-16* [39]	*Dynamic Branching (EM)*	86.72	92.51
Wang *et al.* [58]	*VGG-16* [39]	*Two Attention Modules (EM)*	90.99	95.78
Liu *et al.* [59]	*VGG-16* [39]	*Single Attention Module (EM)*	91.16	96.12

Conclusion

- To the best of our knowledge, nobody attacks to
    - Any information retrieval task
    - Any fashion-related task
    - Any task using ImageNet-sized data set
    - Any task using a data set with 6-digit number of samples

by using Capsule Networks so far.

Conclusion

- In this thesis, we show that
  - Capsule Networks can be designed as **Triplet-based** to learn the similarity between the images.
  - Employing **more powerful feature extraction methods** for Capsule inputs improves the performance of Capsules significantly.

Conclusion

- In this thesis, we also show that
  - Capsule Networks **can achieve even better results** than CNN-based architectures that use different side information or extra module **to recover pose configuration** of the objects.
  - Capsule Networks can get comparable results to the SOTA architectures by using **only images** and with **only half of the parameters** in the SOTA architectures.

References

[1] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “DeepFashion: Powering robust clothes recognition and retrieval with rich annotations,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[4] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in Advances in Neural Information Processing Systems 30, pp. 3856–3866, 2017.

[5] A. G´eron, “Capsule Networks (CapsNets): Tutorial,” 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2015.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” CoRR, vol. abs/1503.03832, 2015.

[19] W. Ge, “Deep metric learning with hierarchical triplet loss,” in The European Conference on Computer Vision (ECCV), September 2018.

[20] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, “BIER : Boosting independent embeddings robustly,” 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5199–5208, 2017.

[21] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon, “Attention-based ensemble for deep metric learning,” in The European Conference on Computer Vision (ECCV), September 2018.

[22] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in International Conference on Machine Learning (ICML) Deep Learning Workshop, vol. 2, 2015.

[25] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, “Where to Buy It: Matching street clothing photos in online shops,” pp. 3343–3351, 12 2015.

[26] J. Huang, R. Feris, Q. Chen, and S. Yan, “Cross-Domain Image Retrieval with a Dual Attribute-aware Ranking Network,” in Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1062–1070, 2015. 72

[27] C. Corbi`ere, H. Ben-younes, A. Ram´e, and C. Ollion, “Leveraging weakly annotated data for fashion image retrieval and label prediction,” 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 2268–2274, 2017.

[28] Z. Wang, Y. Gu, Y. Zhang, J. Zhou, and X. Gu, “Clothing retrieval with visual attention model,” 2017 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4, 2017.

[29] Y. Yuan, K. Yang, and C. Zhang, “Hard-aware deeply cascaded embedding,” in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 814–823, Oct 2017.

References

[30] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming Auto-encoders,” in Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I, ICANN’11, (Berlin, Heidelberg), pp. 44–51, Springer-Verlag, 2011.

[33] G. Hinton, S. Sabour, and N. Frosst, “Matrix capsules with EM routing,” in Proceedings of International Conference on Learning Representation (ICLR), 2018.

[36] D. Rawlinson, A. Ahmed, and G. Kowadlo, “Sparse unsupervised capsules generalize better,” in arXiv preprint arXiv:1804.04241, 04 2018.

[39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations (ICLR), 2015.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015

[53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[55] M. Lin, Q. Chen, and S. Yan, “Network in network,” in 2rd International Conference on Learning Representations (ICLR) 2014, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2014.

[56] A. R. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, “Stacked capsule autoencoders,” in arXiv preprint arXiv:1906.06818, 06 2019.

[57] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, “Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification,” 2016.

[58] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu, “Attentive fashion grammar network for fashion landmark detection and clothing category classification,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[59] J. Liu and H. lu, “Deep fashion analysis with feature map upsampling and landmark-driven attention,” 09 2018.

Thank you!